Trends in Mathematics

Klaus Meer
Alexander Rabinovich
Elena Ravve
Andrés Villaveces
Editors

Model Theory, Computer Science, and Graph Polynomials

Festschrift in Honor of Johann A. Makowsky

Birkhäuser

Trends in Mathematics

Trends in Mathematics is a series devoted to the publication of volumes arising from conferences and lecture series focusing on a particular topic from any area of mathematics. Its aim is to make current developments available to the community as rapidly as possible without compromise to quality and to archive these for reference.

Proposals for volumes can be submitted using the Online Book Project Submission Form at our website www.birkhauser-science.com.

Material submitted for publication must be screened and prepared as follows:

All contributions should undergo a reviewing process similar to that carried out by journals and be checked for correct use of language which, as a rule, is English. Articles without proofs, or which do not contain any significantly new results, should be rejected. High quality survey papers, however, are welcome.

We expect the organizers to deliver manuscripts in a form that is essentially ready for direct reproduction. Any version of TEX is acceptable, but the entire collection of files must be in one particular dialect of TEX and unified according to simple instructions available from Birkhäuser.

Furthermore, in order to guarantee the timely appearance of the proceedings it is essential that the final version of the entire material be submitted no later than one year after the conference.

Klaus Meer • Alexander Rabinovich •
Elena Ravve • Andrés Villaveces
Editors

Model Theory, Computer Science, and Graph Polynomials

Festschrift in Honor of Johann A. Makowsky

Birkhäuser

Editors
Klaus Meer
Department of Computer Science
Brandenburg Technical University
Cottbus-Senftenberg
Cottbus, Germany

Elena Ravve
Department of Software Engineering
ORT Braude College
Karmiel, Israel

Alexander Rabinovich
School of Computer Science
Tel Aviv University
Tel Aviv, Israel

Andrés Villaveces
Department of Mathematics
National University of Colombia
Bogota, Colombia

ISSN 2297-0215 ISSN 2297-024X (electronic)
Trends in Mathematics
ISBN 978-3-031-86318-9 ISBN 978-3-031-86319-6 (eBook)
https://doi.org/10.1007/978-3-031-86319-6

This book is published under the imprint Birkhäuser, www.birkhauser-science.com by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

This book is dedicated to our dear friend and colleague Johann Andreas Makowsky on occasion of his 75th birthday.

Preface

Johann (János) Andreas Makowsky was born on March 12, 1948, in Budapest, Hungary. In 1949 his family moved to Zurich in Switzerland, where he got his education, and in 1966 his citizenship. He majored in mathematics and physics at the Federal Institute of Technology (ETH) in Zurich, graduating with a master's degree in both fields in 1971. His MSc thesis was in Model Theory. He then spent part of 1972 and 1973 in Warsaw, first as an exchange student working under A. Mostowski, then as a participant in the Logic Year held at the Banach Center in 1973. Still enrolled at ETH, he completed his Ph.D. thesis under the supervision of H. Läuchli and E.P. Specker. Following the invitation by G. Kreisel, János worked as a visiting assistant professor at the philosophy department of Stanford University from fall 1973 to spring 1974. He returned to Zurich in spring 1974 and received the doctoral degree (Dr. math.sc.) for his thesis "Δ-Logics and Generalized Quantifiers." After a short post-doctoral research appointment at Simon Fraser University in Vancouver under A. Lachlan, in 1975 he was appointed a visiting professor at the mathematics department "Ulisse Dini" of Florence University. János held two regular positions: from 1976 to 1980 at FU Berlin as a junior faculty (H1), where he also got his Habilitation degree in 1980. Also in 1980 he joined the Faculty of Computer Science at the Technion in Haifa (Israel) as a senior research associate. He was given tenure in 1984 as associate professor, and became full professor in 2001 and professor emeritus in 2016. He still occasionally teaches advanced courses and supervises graduate students.

János held many visiting positions at various academic institutions: the Hebrew University and at Bar Ilan University in Israel; at the Simons Institute in Berkeley and at MIT in Cambridge, USA; at the Fields Institute in Toronto, Canada; at Lausanne University, Bern University, and ETH Zurich in Switzerland; at LaBRI of the University of Bordeaux and at Paris Diderot University in France; at Vienna Technical University in Austria; at Charles University in Prague, Czech Republic; the Stekhlov Institute in Moscow, Russian Federation; the Monash University, Melbourne, Australia; and at the Institute of Mathematical Sciences in Chennai, India.

János' research shows both an impressive broadness and scientific depth. The common thread of his research is Model Theory. Over the years János contributed significantly to areas ranging from Model Theory, Database Theory, Theory of Programming Languages and their Verification, to Data Modeling and Software Engineering, and more recently, to Algorithmic Graph Theory and Knot Theory, and a General Theory of Graph Polynomials.

János (co-)authored and edited 9 Books and Lecture Notes and published over 175 scientific papers. He gives a complete list and detailed comments in his contribution "My writing" in this volume. Without going more into detail here, the various contributions in the present volume convincingly testify to the diversity of his scientific interests and the breadth of the topics he has worked on.

Needless to say that his scientific career was flanked by numerous services to the community, such as long periods of being on the editorial board of various journals and a guest editor of special issues. He has also regularly served as a member of program and steering committees of scientific conferences worldwide. Among these multiple engagements let us highlight that he was one of the founding members of the European Association of Computer Science Logic (EACSL). During his presidency of the EACSL, in 2004 he initiated the now well-known "Ackermann Award" for outstanding dissertations in the field of Logic in Computer Science. As a teacher he developed numerous advanced courses and supervised 12 Ph.D. and 22 master's theses.

The present Festschrift gathers 24 research articles authored by scientific companions, friends, and colleagues. They cover a broad variety of areas to which János made significant contributions himself. This reflects impressively the significant impact of his work.

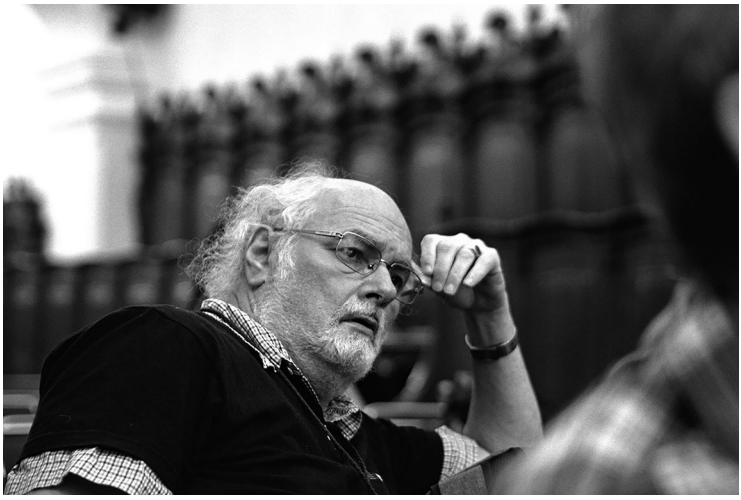
The volume is enriched by four essays shedding light on János as a personality, and with two contributions by the celebrant.

Last but not least, everyone who knows János as a private person or communicated with him during free time at conferences can affirm his formidable knowledge in so many areas besides science, whether it is literature, history, art, music, or other fields. For an impression of János' non-mathematical activities, especially creative and essay writing, one can access his homepage https://janos.cs.technion.ac.il/ and also read Claudia Gehrke's contribution in this volume. His own comments on his non-technical writing are also discussed in his contribution "My writing" in this volume. It is always a joy but also a challenge to discuss with János topics of his diverse interests.

We are happy to have you and your wife Masha as friends. We wish you many fruitful and healthy years full of activities!

Cottbus, Germany — Klaus Meer
Tel Aviv, Israel — Alexander Rabinovich
Haifa, Israel — Elena Ravve
Bogotá, Colombia — Andrés Villaveces
October 2024


UNIVERSIDAD
NACIONAL
DE COLOMBIA
facultad de Ingeniería
Sede Bogotá



Contents

My Writing

Johann A. Makowsky

Abstract I first describe how I became a mathematician, discussing other possibilities and paths not taken. Next I discuss my scientific publications. Finally, I describe my non-scientific publications, essays and literary miniatures. All this is arranged by topics rather than chronologically. Finally there is a complete list of my publications, scientific and otherwise.

1 Mathematics as a Literary Genre

I knew very early that I wanted to be an academic, creative but rooted in some kind of reality. Writing would be part of it, but what kind of writing was not clear yet.

1.1 Paths Not Taken

1.1.1 Path Not Taken: Chemistry

My maternal grandfather was an industrialist. In Hungary he was the CEO of a big chemical and mining company. After emigration to Switzerland he founded a small chemical firm and traded in commodities, mostly mining products and chemical components. Before the war he had talked his son, my uncle, into studying chemistry. He was groomed to be the "crown prince" of the chemical empire, but it did not work out as planned. Grandfather was sure I should follow suit, as a future heir of his enterprises. When I was eleven, he passed away and his enterprise was acquired by his silent partner who bought our shares cheaply due to our lack of funds for badly needed further investments. In a way I was lucky: I was now free to

J. A. Makowsky (✉)
Faculty of Computer Science, Israel Institute of Technology, Haifa, Israel
e-mail: jmakowsky@bluewin.ch

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_1

realize my own career choices. A path not taken not as a choice but by force majeur, hence path not taken number zero.

1.1.2 Path Not Taken: Atomic Physics

In 1960, when I was twelve, I wanted to become an atomic physicist. It was the time when passionate discussions about the effect of the atomic bomb on the future of mankind were reaching the peak. In 1962 the play "The Physicists" by Friedrich Dürrenmatt had its premiere, and my mother took me to the memorable performance of the Schauspielhaus in Zurich. In 1964, I saw a performance of "In the Matter of J. Robert Oppenheimer", a play by Heinar Kipphardt, based on the transcripts of the Oppenheimer security hearings. I became aware of the ambiguity of the role of science in technical progress. Eager to learn more about this ambiguity I got hold of Karl Jaspers' "The future of mankind" published in 1958.[1] It became clear to me that my wish to become an atomic physicist would confront me with questions of ethics and personal responsibilities which I was not willing to face at the time. This was the first time a path was not taken consciously.

1.1.3 Path Not Taken: Politics and Feuilleton

It was also the time of the Eichmann trial.[2] Mother was watching excerpts of the publicly broadcasted trial every evening. Due to my insistent questions, she started telling me about her own and her family's dramatic war-time experiences. I read Sartre's "Réflexions sur la question juive".[3] Jaspers and Sartre led to my early interest in existentialist philosophy and to my general interest in literature. During the six and a half years of gymnasium I developed a passion for writing and languages. I was reading the classics in the original Latin and Greek, and philosophy and literature in German and French. My school essays became ambitious writing projects and ambitious literary experiments. I wrote plays and excelled in translating Greek and Latin poetry. I was curious to understand the "behind-the-scenes"

[1] Karls Jaspers, "Die Atombombe und die Zukunft des Menschen: Politisches Bewusstsein in unserer Zeit", Piper, München, 1958.

[2] "The Eichmann trial was the 1961 trial in Israel of major Holocaust perpetrator Adolf Eichmann who was kidnapped in Argentina by Israeli agents and brought to Israel to stand trial. Eichmann was a senior Nazi party member and served at the rank of Obersturmbannführer (Lieutenant-Colonel) in the SS, and was one of the people primarily responsible for the implementation of the Final Solution. He was responsible for the Nazis' train shipments from across Europe to the concentration camps, even managing the shipment from Hungary directly, where 564,000 Jews died". (Cited from https://en.wikipedia.org/wiki/Eichmann_trial).

[3] "Reflections on the Jewish Question" is an essay about antisemitism written by Jean-Paul Sartre shortly after the Liberation of Paris from the German occupation in 1944. The first part of the essay, "The Portrait of the Antisemite", was published in December 1945 in Les Temps modernes. The full text was then published in 1946.

and wrote term papers based on interviews I conducted with the brewmaster of "Wädenswiler Bier", with the stage master of the Zurich Opera, and with the head of the typesetting department of *Die Tat*, a now defunct but then important evening paper, for which I also worked as a delivery boy to earn some pocket money. When I was 14 I started writing essays about theater and music which were published in the quarterly magazine *Die Mittelschulzeitung* edited by students of the Zurich's gymnasia, the *Zürcher Mittelschulzeitung* (ZMZ). Later I joined the magazine as an editor. Once at the university, I continued to write for the bimonthly *Zürcher Student* and in 1969 became one of its editors. I had learned newspaper making in all its aspects: writing, editing, designing page layout, and acquisition of advertisements. I could have started a journalistic career. I even had my moment of investigative journalism and political activism: One was when ZMZ defended the freedom of cinematic expression and exposed an obstinate enemy of the avant-guardish movie club. Another one was when in 1968 I suggested to the Swiss Student Union to initiate a referendum against a misconceived university reform, which took place in 1969, and we won. The *Zürcher Student* played an importan role in the campaign while I was one of its editors.

Many of my fellow students involved over the years with the *Zürcher Student* became journalists, writers, public servants in cultural institutions, or politicians. One became a professor of philosophy at Zurich University, and one even served as a Federal Council. I retained my wish to write essays and prose and I did publish about theater, music, politics and society during all my career, see Sect. 2.5, but not for a living. Second path not taken.

1.1.4 Path Not Taken: The World as a Stage

In 1966 my mother was diagnosed cancer just before starting a new job. She remained unemployed and I earned my money working in the newly founded *Theater am Neumarkt* as an all-purpose hand: selling programs, controlling tickets and managing the cloakroom. I also fulfilled tasks backstage for special effects involving sound and lightening. I attended rehearsals and occasionally offered my comments. I did have a rather solid background in drama, as a theatergoer and as a reader of plays, and I was familiar with some of the theoretical literature, see again Sect. 2.5. Soon the actors and the director began to pay attention to my sharp comments and constructive suggestions. I started flirting with the idea of making a career as a director. At the time there was only one way to learn the trade: One had to become a director's assistant and apprentice. I kept working in the theater part-time from 1966 to 1970 while Felix Rellstab[4] was its artistic director and Reinhardt

[4] Felix Rellstab 1966–1971 first head and director at the *Theaters am Neumarkt* in Zürich. 1960–1991 head and director of the *Bühnenstudio*, Switzerland's prestigeous acting school. Under his leadership the *Bühnenstudio* was renamed *Schauspiel-Akademie Zürich*. in 1972/1973.

Spörri[5] was its dramaturg who was also directing. Two years into my studies at ETH, Spörri was desperately looking for an assistant for his newest production planned at a very short notice. There it was, my dream had come true. I was offered the job. However, it came too late. I had already started to work on my diploma thesis, an equivalent to an MSc thesis. I did ponder over the offer seriously, but I finally rejected it. Mathematics and logic had captured me and did not let go. Third path not taken.

1.2 From Philosophy and Epistemology to Logic and Mathematics

Shortly before my matriculation exams in 1966 I decided to study mathematics and philosophy. Originally I wanted to go to Paris and study mathematics and philosophy. Paris for me was Sartre and Foucault on the one hand, and Bourbaki on the other. My father, who lived in Paris at the time, bought me the first 6 volumes of Bourbaki as a gift for my matriculation diploma. One could enroll at the École Normal Supérieure (ENS) as a foreign student without passing the concours. However, my Greek teacher introduced me to a friend of his who was a philosophy professor well versed in mathematics and exact sciences. He convinced me that for undergraduate studies I would fare better in Zurich, because in Paris I was unlikely to even get close to my admired intellectual superstars. In 1967, I enrolled at the Swiss Institute of Technology (ETH) in the Department of Mathematics and Physics.

I got interested in mathematics through the philosophy of Plato and Leibniz. In the last years of the gymnasium I discovered the early work of Ernst Cassirer.[6] His monograph on Leibniz and his two volumes on epistemology in modern times left a deep impact on me. The latter is a "magisterial and deeply original contribution to both the history of philosophy and the history of science. It is the first work, in fact, to develop a detailed reading of the scientific revolution as a whole in terms of the 'Platonic' idea that the thoroughgoing application of mathematics to nature (the so-called mathematization of nature) is the central and overarching achievement of this revolution."[7] Through Cassirer I was naturally led to the Vienna Circle and to Wittgenstein as well as to Russel and Frege. I was

[5] Reinhart Spörri was 1968–1971 dramaturg and first director at the *Theater am Neumarkt Zurich* under Felix Rellstab.

[6] My first readings of Cassirer's work: Leibniz' System in seinen wissenschaftlichen Grundlagen. Marburg, 1902; (1906) Das Erkenntnisproblem in der Philosophie und Wissenschaft der neueren Zeit. Band 1 und 2. Berlin: Bruno Cassirer Verlag 1906/7. Kant und die moderne Mathematik. Kant-Studien 12, 1–40. 1907.

[7] Quoted from: Michael Friedman, "Ernst Cassirer", The Stanford Encyclopedia of Philosophy (Fall 2023 Edition), Edward N. Zalta & Uri Nodelman (eds.), URL = https://plato.stanford.edu/entries/cassirer.

intrigued by epistemological questions of mathematics. Was mathematics a branch of logic? Or was it a language to describe abstract configurations? Are the basic elements of the mathematical language innate, hardwired in the human brain? I read Hadamard's booklet "Psychology of Invention in the Mathematical Field", Poincaré's philosophical works, especially "Science and Method", and Piaget's "The Child's Conception of Number".

In the first semester of my university studies I also enrolled in the philosophy seminar of ETH dedicated to Wittgenstein's early masterpiece, the"Tractatus logico-philosophicus" given by E. Specker and G. Huber. I had read the "Tractatus" while still in the gymnasium. In my precocious but innocent enthusiam I volunteered to give the first talk in this seminar, unaware that among the over eighty participants of the seminar all the philosophical elite of Zurich and beyond would be present. Among them Paul Bernays, the eminent logician and the author of an important critique of Wittgenstein's later work[8] Had I known more about the audience of the seminar, I might have shied away from volunteering to prepare a talk at such a short notice. Surprisingly, my talk, given freely and only based on some scribelled notes, was a stunning success. Later, both E. Specker and P. Bernays played decisive roles in my further mathematical development and career. From the first semester and until after my PhD I attended regularly the logic seminar, founded by Bernays and Gonseth in the 1930s. In my time it was run by Ernst Specker and my future MSc advisor Hans Läuchli. Other logic-related seminars I attended whenever I was in Zurich were organized by Ernst Specker and Volker Strassen on the complexity of computations, and by Erwin Engeler on logic and its role in computer science.

1.3 Epilogue

The paths not taken were many. I did not become an atomic phycisist, I did not choose to make literary criticism as my career, nor did I become a scholar of Latin and Greek literature and culture. I did realize that classical philology had passed its zenith and was on the way of becoming marginal. My dabbling in journalism and politics did not satisfy my intellectual ambitions. Besides my interest in mathematics in all its aspects, my passion for writing remained strong. I was 28 when my father passed away and I resolved to write a historical novel based on his life. But whenever I tried to dedicate time to this (or similar) projects I stumbled over a mathematical discovery which gently forced me to write mathematics. Even after my formal retirement, mathematics did not let go. Although I did spend more time on my literary projects and progressed well, mathematics kept intruding and interrupting.

For me mathematics is a literary genre.

[8] Paul Bernays, "Comments on Ludwig Wittgenstein's Remarks on the Foundations of Mathematics", Ratio 2: 1–22. 1959.

2 Comments on Selected Papers

My mathematical research interests still include Model Theory (Classical Model Theory, Abstract Model Theory, Finite Model Theory), Semantics of Programming Languages (Database Theory, Program Verification, Logic Programming), Logic and Complexity. However, since 1995 my work has concentrated more but not exclusively on Applications of Logic to Graph Theory, Knot Theory and Combinatorics. Since 2005 my work concentrates on a model theoretic view of Graph Polynomials and Combinatorial Counting Functions.

I have written two mathematical papers which have an autobiographical touch: [135] recounts how Model Theory is a recurrent theme in most of my work, and [75] recounts how graph polynomials became a dominant theme in my more recent research. My recollections of my encounters with A. Mostowski were published as [162]. For Yuri Gurevich's 80th birthday I wrote about our 40 years of friendship in [171]. I also co-edited and contributed to the memorial volume for my most influential teacher E. Specker (1920–2012) [55].

In the following subsections I comment on my publications grouped by topic. A complete list of my papers follows in Sects. 3 and 4.

2.1 Model Theory

2.1.1 Classical Model Theory

Already in my first semester I attended the Läuchli-Specker logic seminar and remained a faithful regular. Although the seminar was meant also for undergraduates, we also studied very advanced topics like Matyasevich's work on Hilbert's 10th problem, Morley's landmark paper on categoricity, and Tseitlin's work on the complexity of the resolution algorithm for satisfiability of propositional logic. In summer 1969, I spent 4 months as a recruit in the Swiss Army. During my service I had a copy of Morley's paper with me and used much of my free time in the army to prepare myself for the seminar, where I was to lecture about it. I also tried to attack some of the open problems listed in the paper. In summer 1970, I attended W. Hodges' logic conference at Bedford College, London. There I met not only W. Hodges, but also G. Müller, A. MacIntyre and A. Lachlan. All of them decisively influenced my future career as a logician. A. Lachlan gave a preprint of his and J. Baldwin's paper on categoricity after I told him what I was working on. The paper became the basis of my early research.

My work of 1971–1974 in classical Model Theory is based on my diploma thesis from 1971. The results were published as [1] and [3]. Two of the famous problems from M. Morley's papers[9] were (partially) solved in [3]: There is no

[9] Michael Morley, "Categoricity in power". Trans. Amer. Math. Soc. 114 (2): 514–538. 1965.

finitely axiomatizable complete almost strongly minimal ω-categorical (hence ω_1-categorical) theory, and there is a finitely axiomatizable complete superstable theory such that all of its types are non-principal.

Paper [3] is referenced in the two major monographs, one by C.C. Chang and J. Keisler[10] and one W. Hodges[11] and the more specialized monograph by B. Zil'ber[12] The book by M. G. Peretyat'kin[13] develops a rich theory of work done by its author between 1982 and 1993 which was influenced by my diploma thesis of 1971. In P. Rothmaler's Monograph[14] a whole chapter is devoted to my results of 1971.

Paper [3] also asks whether there is a finitely presented infinite group which has only finitely many conjugacy classes. This is still open and attributed to [3] in *Open problems in combinatorial group theory*.[15]

After my early success in classical Model Theory, I spent two periods of 6 months in Warsaw. I had realized that with the tools available to me at the time, I could not make any further progress in categoricity theory. Under the guidance of W. Marek and A. Mostowski, I turned to generalized quantifiers and Abstract Model Theory. I was fascinated by Lindström's characterization of First Order Logic using generalized quantifiers.[16]

2.1.2 Abstract Model Theory and Abstract Elementary Classes

My PhD thesis marked the beginning of my own work in Abstract Model Theory. It also marked the beginning of an intensive collaboration with S. Shelah.

The results of my PhD Thesis are published as [2, 4, 78]. When I lectured in S. Feferman's logic seminar at Stanford about my PhD, J. Stavi found a mistake in one of the theorems of PhD thesis. Jonathan had studied together with Saharon before joining Stanford's math department. He helped me to correct the mistake and we planned to publish the journal version together. However, J. Stavi insisted to add Saharon as a third author, although he was not personally involved in working on this paper at all. Stavi explained that all he needed to fix my mistake he had learned from Saharon while they both were in Jerusalem. Although I met Saharon already in 1971 at the European ASL Meeting in Cambridge, UK, personal collaboration with

[10] C.C. Chang and Jerome Keisler, "Model Theory", 2nd edition and later, North Holland, 1973.

[11] Wilfrid Hodges, "Model Theory", Cambridge University Press, 1993.

[12] iBoris Zil'ber. "Uncountably categorical theories". Mathematical Monographs Vol. 117. American Mathematical Soc., 1993.

[13] Mikhail G. Peretyat'kin, Finitely Axiomatizable Theories. 1997. English translation by V. Morley, J. Symbolic Logic 64.1 (1999): 1828–1830.

[14] Philipp Rothmaler. "Introduction to Model Theory". Vol. 15. CRC Press, 2000.

[15] Alexei G. Myasnikov, Gilbert Baumslag and Vladimir Shpilrain. "Open problems in combinatorial group theory." Combinatorial and Geometric Group Theory: AMS Special Session, Combinatorial Group Theory, November 4–5, 2000, New York: AMS Special Session, Computational Group Theory, April 28–29, 2001, Hoboken, New Jersey 296 (2002): 1.

[16] Lindström, P. (1969) "On Extensions of Elementary Logic." Theoria, 35: 1–11.

him started only when I met him again in 1974 in Vancouver. After that he offered me a 1 year Lady Davis post-doc position at the Hebrew University 1976. However, on his suggestion, this was cut short to a 2 month visit during a semester break due to my accepting a 5 year position in West-Berlin. During my Berlin appointment I visited Israel regularly and continued to work with S. Shelah and J. Stavi. This collaboration resulted in papers [9, 10, 12].

At the time I also worked in topological Model Theory [6, 11, 79, 80]. My approach to topological Model Theory was based on monotone quantifiers, which were also investigated in [5, 7]. I returned again to topological Model Theory most recently with the paper [77], where we count the number of finite topologies subject to various restrictions definable in monadic second order logic.

Some of my papers in abstract Model Theory (1973–1984) are widely referenced. The multi-author monograph 'Model Theoretic Logics' (J. Barwise and S. Feferman eds., 1985) gives best testimony for this: Chapters 18 and 20, [153], [155] were written by me alone summarizing my work with S. Shelah's and my own contribution to the field. Chapter 19 [154] is co-authored with D. Munidici. The three chapters comprise 146 pages and basically conclude my work in abstract Model Theory. Many other chapters quote my work.

In chapter 20, [155], I gave the first published presentation of S. Shelah's then unpolished results on "Abstract embedding relations". Both chapters also contain relevant original contributions to the field I had obtained before 1982. In 1978 in Berlin I supervised the MSc thesis by Gerhard Herrgott on the model theory of L^{pos}, but the results were never published. I also initiated and supervised (until I left for Israel) the MSc thesis of Sakae Fuchino.[17] When S. Shelah resumed work on this topic (published in 1987) he called the concept 'Abstract Elementary Classes", and misquoted my work with him. Due to this, Shelah's and my pioneering work from before 1983 was often overlooked.

I stopped working in abstract model theory due to my employment at the Computer Science Department of the Technion. However, in my later work in finite Model Theory my earlier work on generalized quantifiers plays a crucial role, see Sect. 2.2.7 below.

2.2 Foundations of Computer Science

2.2.1 My Early Exposure to Computer Science

Although ETH in Zurich was a pioneer in early computing there was no Computer Science Department until 1981. Programming was viewed as part of Numeric Analysis using ALGOL-60 as the programming language. In 1968, the Institute of

[17] Sakae Fuchino, "On the categoricity theorem in $L_{\omega_1,\omega}$", Tsukuba Journal of Mathematics, Vol.10, No.1 (1986), 117–120.

Applied Mathematics was split and the Group of Computer Science was established with H. Rutishauser (one of the fathers of ALGOL-60) and N. Wirth (the father of PASCAL). Later, they were joined by E. Engeler responsible for Theoretical Computer Science. Programming was still taught as part of Numeric Analysis, but switched to PASCAL. Also in 1968, V. Strassen joined the Math Department of Zurich University. He and E. Specker started a joint seminar on Complexity Theory. During my studies I attended various lectures delivered by Engeler and became a regular in the Specker-Strassen seminar. Quite a few of the early participants of this seminar became leading computer scientists, among them J. Heintz, J. von zur Gathen, E. Zachos and M. Fürer.

While I was employed at the Free University in West-Berlin I attended seminars at the Technical University: D. Siefkes' seminar on complexity theory and H. Ehrig's seminar on specification of abstract data types. There I started my collaboration with B. Mahr who would visit me later at the Technion in Haifa and with whom I wrote [88, 89, 13]. These papers became the basis of my [18] see also Sect. 2.2.5 below. I also supervised the MSc thesis of Michael Mötz on the complexity of resolution, but our results where superseded by Z. Galil's PhD thesis from 1977.[18]

Between 1978–1980, while visiting the Hebrew University on a German-Israeli Minerva Grant it became clear to me that for family reasons a permanent position in Israel would be beneficial for me. I was told by Eli Shamir that I would have good chances if I changed my research interest to Logic in Computer Science. This was 10 years before Logic in Computer Science became an established branch of theoretical computer science with its conferences LiCS (in the US) and CSL (in Europe).

Eli introduced me to Catriel Beeri. He knew that Catriel was working on databases, he was likely to be willing to discuss his work with a logician.

Consequently, I started working with Catriel on the foundations of database theory. Eli also suggested I should attend the 1979 Annual Symposium on Theory of Computing in Atlanta which I did. I made contact with V. Pratt, with whom I had a chance to discuss dynamic logic. He invited me for a 2 month stay at MIT starting in February 1980 in order to continue our emerging collaboration. I also learned about computable database queries from the talk by A. Chandra and D. Harel. Many other talks stressed the importance of logic for computer science.

Eli's suggestion opened my mind and helped me in choosing my first research topics in Computer Science. When I joined the Technion's Computer Science Department, I was well prepared to find ways to use my expertise in model theory for foundational problems in Theoretical Computer Science.

[18] Zvi Galil, On the complexity of regular resolution and the Davis-Putnam procedure, Theoretical Computer Science 4 (1), 1977, 23–46.

2.2.2 How to Use Model Theory in Computer Science

During 1980–82 I had various intensive discussions with Y. Gurevich and also with J. Stavi on the role model theoretic methods could play in Computer Science. In 1982 I was an invited speaker at the Logic Colloquium'82 in Florence, where I chose to speak about this, [87]. There were no technical results, but many suggestions and examples. At the time I missed what Y. Gurevich later pointed out:[19] Finite Model Theory is inherently different from model theory, in as much as most preservation theorems of classical Model Theory do not hold in the finite. My approach, however, was to use classical preservation theorems to explain how they can (still) be used when infinite models are allowed. I had two remarkable successes in this approach: I could explain why Horn formulas matter in Computer Science, [18], and find many new uses of the Feferman-Vaught Theorem and its variants for graph algorithms [43]. The first one is discussed further on pages in Sect. 2.2.5, the second in Sect. 2.2.8. I published two more papers similar in spirit to [87]. In 1992 [156] and in 1994 [99]. Finally, in 2011, I published a version of my retirement address as president of EACSL as [135] reflecting on how Model Theory remained a recurrent theme in my work over the years.

2.2.3 Dynamic Logic and Program Verification

My first published paper in computer science dealt with the expressive power of dynamic logic [82] and was presented at ICALP'80 in Noordweijkerhout, the Netherlands. There I also met S. Even who at the time was the head of the CS department of the Technion. Having heard that I was looking for an academic appointment in Israel, he recruited me to join his department. Already in my first semester at the Technion Michail Tiomkin, a very recent immigrant from the USSR became my PhD student. Michail was a graduate of Moscow State University. He had studied with Albert G. Dragalin who, together with A.N. Kolmogorov, had established a compulsory logic course for all mathematics majors at the university. He was also chosen to represent the USSR at the International Mathematics Olympiad, but, after he refused to join the KOMSOMOL (the Young Communist League), was dropped from the team. Michail's PhD thesis was published in two joint papers as [14, 23]. The first one dealt with propositional dynamic logic with local assignments, and the second with the decidability of finite probabilistic propositional dynamic logic. Michail left academia and joined the research laboratory of IBM, But he continued working with M. Kaminski, another immigrant from the same period, whom he knew from his Moscow times, and who joined our faculty after completing his PhD with M. Rabin in Jerusalem. They mostly worked together on non-monotonic

[19] Gurevich, Yuri. Logic and the challenge of computer science. 1985. (preprint), published in 1988 as: Gurevich, Yuri, Logic and the challenge of computer science, as: Chapter 1 of Trends in Theoretical Computer Science, E. Börger ed., Computer Science Press, 1988.

logic. I joined them once as a co-author with [110, 31]. All these papers had very limited impact. In the same year I also contributed to O. Grumberg's PhD thesis by providing a missing proof for a conjectured result, [84, 16]. It led to attempts at further collaboration with her supervisor N. Francez which did not come to fruition. I also lost interest in the topic, partially because I considered the approach in [84] inferior to the work published in the same year by D. Lehmann, A. Pnueli and J. Stavi[20]. N. Francez later wrote a book on the topic[21] which had a considerable impact. My last publication in program verification was accepted for the very first IEEE conference on Logic in Computer Science (LiCS) in 1986 [91], and its full version was published as [21]. It was based on Ildikó Sain's work after her first version received a negative referee report. I helped her revising and expanding it.

2.2.4 Database Theory

My work with C. Beeri never evolved into publishable papers. As a result, C. Beeri and M. Vardi picked up many themes I had discussed with C. Beeri before.

However, I was the first to define a framework for database dependencies, and to prove undecidability results in this framework. When I showed the undecidability result to M. Rabin, he dismissed it, as we now know wrongly, as previously known, while J. Ullman declared, the decidability question as "the fundamental open problem of database dependencies". After undecidability was established by myself and, independently, by A. Chandra and H. Lewis, J. Ullman declared, also wrongly, the study of database dependencies to be passé. The result was published jointly with A. Chandra and H. Lewis as [83, 85]. We did not publish the journal version because by the time C. Beeri and M. Vardi[22] had sharpened our results. My paper [86] and the resulting joint paper with M. Vardi [17] did deal with model theoretic aspects of database dependencies.

My first PhD student in databases was A. Zvieli, who wrote a thesis on the query language "Query by example" (QBE). His results were not published, because he dropped out of academia. His wife Dvora Tzvieli is a Teacher of Tibetan Buddhism and acts as a Lama of one of the Israeli Buddhist communities. A. Zvieli works as a teacher of mindfulness. At the time he worked on his PhD there were two ways of modelling databases: The relational model and the Entity-Relationsship (ER) model. Yoav Raz, then a faculty member, was a strict adherent to the ER school and convinced that ER was superior to the Relational model. I was convinced that they were intimately related. I recruited Udi Rotics as an MSc student to investigate

[20] Daniel Lehmann, Amir Pnueli, and Jonathan Stavi. "Impartiality, justice and fairness: The ethics of concurrent termination." Automata, Languages and Programming: Eighth Colloquium Acre (Akko), Israel July 13–17, 1981 8. Springer Berlin Heidelberg, 1981.

[21] Francez, Nissim. Fairness. Springer Science & Business Media, 2012.

[22] Beeri, Catriel, and Moshe Y. Vardi. "The implication problem for data dependencies." Automata, Languages and Programming: Eighth Colloquium Acre (Akko), Israel July 13–17, 1981 8. Springer Berlin Heidelberg, 1981.

this. Victor Markowitz, who had obtained his MSc under the supervision of Yoav Raz on designing ERROL, a query language for the ER model, joined us as a PhD student. Victor, Udi and I published our first findings as [95]. With Victor I published [97, 98] and finally established conditions for bi-interpretability of the ER and relational model in [22]. The latter is among my five most quoted papers with over 250 Google citations. Victor later became chief informatics officer and associate director at DOE Joint Genome Institute (JGI), and head of Lawrence Berkeley National Laboratory's Biological Data Management and Technology Center. Around 1985 Elias Dahlhaus joined me in Haifa for a year on a German-Israeli grant (Minerva). We mostly worked on extending relational query languages to hierarchic structures similar to file systems in UNIX. We published several papers on this, [92, 93,94,24] anticipating query languages for hierarchic databases. The paper [102] with my MSc student R. Hasson explored update languages versus query languages. Unfortunately, the student was not interested in continuing this line of research, so I abandoned it as well.

More recently, I worked with my former PhD student Elena Ravve, on normal forms of database design, especially on Boyce-Codd normal forms, [111, 113, 32, 166]. Elena's MSc and PhD theses were exploring applications of the Feferman-Vaught Theorem to decomposition problems in model checking, [109]. Ultimately, this led to my paper [43].

My contributions to Database Theory are referenced in many monographs.[23]

Four of my contributions in Database Theory had considerable impact. In my paper with A. Chandra and H. Lewis (1981), see [83, 85], we gave the first complexity analysis in the emerging field of database dependencies; in my papers with E. Dahlhaus starting in 1985 and published in conference proceedings [93, 92, 94], and finally in a journal publication [24]. We anticipated the emerging field of complex databases and clarified the notion of computable queries over hereditary finite structures; and in my papers with V. Markowitz and N. Rotics we contributed to the mathematical foundations of the Entity–Relationship Model. More recently, together with E. Ravve, I returned to the study of the foundations of database design. For this work I was invited to give a keynote address at the 15th International Conference on Conceptual Modeling in 1996. In 2012 I returned to work with E. Ravve and we laid the proper foundations for the role of Boyce-Codd Normalform in database design, see [166]. These findings finally showed that BCNF was prematurely dismissed as a useful design principle. This finally satisfied

[23] Most prominently in: Jeff Ullman, Principles of Database Systems, Computer Science Press, 1983,

Heikki Mannila, Karl-Jouko. Räihä, "The Design of Relational Databases", Addison-Wesley, 1992.

Serge. Abiteboul, Richard Hull and Victor Vianu, "Foundations of Databasea"s, Addison Wesley, 1995,

Mark Levene and George Loizou, "A Guided Tour of Relational Databases and Beyond", Springer 1999,

Bernhard Thalheim, "Fundamentals in Entity-Relationship Modelling", Springer, 2000.

my foundational interest in Database Theory. Subsequently, I lost interest in the field. But even earlier my research had gone in a different direction.

2.2.5 Satisfiability and Logic Programming

In 1982, the Japanese launched the 5th Generation Computer Systems Initiative: Logic Programming, especially the programming language PROLOG, played a key role in it. So did special cases of the Satisfiability Problem for propositional formulas and resolution-based algorithms underlying PROLOG. My paper *Why Horn formulas matter in computer science* [18] addressed the question why Horn formulas (a special class of first order formulas) play a prominent role both in Database Dependencies and in Logic Programming. The paper gives a semantic characterization of Horn formulas (and some generalizations thereof) which explained to a large extent why Negation by Failure has a simple semantics in Logic Programming when restricted to Horn formulas.

In paper [19] with A. Itai, we were first to show that propositional Horn formulas have a linear satisfiability problem. Often another paper[24] is credited to be the first with this result, due to its publication date of 1984. However, our result was published as a Faculty preprint already in 1982.[25]

In the paper [25] we study the expressive power of side effects in propositional PROLOG with assignments, and prove an undecidability result. Unfortunately, this paper has hardly been noticed in the literature.

My paper with my MSc student A. Sharell "On Average Case Complexity of **SAT** for Symmetric Distributions" [27] largely explains why on average **SAT** can be efficiently solved. It was prominently discussed in the historic survey on Satisfiability by S. Cook and D. Mitchell.[26]

The fundamental notions of average-case complexity of **SAT** were developed by Leonid Levin in 1986. He published a one-page paper defining average-case complexity and completeness while giving an example of a complete problem for distNP, the average-case analogue of NP.[27] The paper generated a wave of interesting papers[28] but interest soon died down. I still count paper [27] among my truly interesting results.

[24] William F. Dowling and Jean H. Gallier. "Linear-time algorithms for testing the satisfiability of propositional Horn formulae", Journal of Logic Programming, 1 (3), (1984) pp. 267–284.

[25] Alon Itai and Johann .A. Makowsky, "On the complexity of Herbrand's theorem", 12 pp., Technical Report No. 243, May 1982.

[26] Steven Cook and David Mitchell, "Finding Hard Instances of the Satisfiability Problem: A survey". DIMACS Series vol. 35, 1997.

[27] Leonid Levin, "Average case complete problems," SIAM Journal on Computing, vol. 15, no. 1, pp. 285–286, 1986.

[28] https://en.wikipedia.org/wiki/Average-case_complexity.

In the paper[47] with E. Ravve and E. Fischer we count the number of satisfying assignments of propositional formulas of size n. We show that for formulas in clausal normal form of fixed clique-width k this can be achieved in time $4^k n^3$

My papers [18], [19], [47] have over 125 Google citations each.

2.2.6 Notational Systems for Electronic Music

In the late 1970s I met the Israeli composer Josef Tal.[29] Learning about my publications about music, see Sect. 2.5, he invited me to collaborate in his project to design a notational systems for non-conventional music TALMARK. The project was even generously funded for 3 years by the Volkswagen foundation. J. Tal and U. Shimoni from the EE Department of the Technion had a PhD student, S. Markel, developing such a systems, and I had an MSc student, A. Ban, working on the project and published as [96]. A. Ban left academia and became one of the inventors of the USB flashdrive and one of the authors of JUNIOR, a multiple world champion chess program for small computers.[30] In 2003 DEEP JUNIOR played a six-game match against Garry Kasparov, which resulted in a 3–3 tie.

After talking to many composers[31] and lecturing at IRCAM[32] in Paris about Tal's ideas, I came to the conclusion that the system was ill-designed and not suitable for universal use by active composers. There were some very interesting ideas underlying Tal's reason for undertaking the project, but I never found the time and leisure to write those down.

The choice of notational systems, be it in mathematics, or in any other field where written records are used, defines the possibility of expressing and communicating one's thoughts. J. Tal's ambition was to provide composers with a notational system which would be both universally readable and playable. Its shortcomings taught me a lot which was also relevant to my understanding of programming and query languages. In some sense paper [158] was also influenced by this project.

2.2.7 Finite Model Theory

In 1994, I returned to the theory of generalized quantifiers in the context of *Finite Model Theory and its interaction with Complexity Theory*. This work was done together with my MSc students Y. Bargury, A. Calò. With Y. Bargury we investigated the expressive power of the transitive closure and 2–way multi-head

[29] Josef Tal 1910–2008, https://joseftal.org/, https://en.wikipedia.org/wiki/Josef_Tal.

[30] https://en.wikipedia.org/wiki/USB_flash_drive https://en.wikipedia.org/wiki/Junior_(chess_program).

[31] Y. Xenakis, H.W. Henze, P. Eötvös, W. Rihm, B. Ferneyhough, G. Grisey, M. Maros, R. Wernick, G. Sinopoli, G. Bertini. B Spoerri, G. Bennet, D. Fueter, D. Meier (of YELLO).

[32] https://en.wikipedia.org/wiki/IRCAM.

automata [100] and with A. Calò we investigated Ehrenfeucht–Fraïssé games for transitive closure [101]. Yachin Pnueli was not my PhD student. Nevertheless Yachin and I published five papers together while he was working on his PhD thesis in computer graphics. Papers [104], [108], [157], all deal with the use of generalized quantifiers in order to capture complexity classes. Papers [106, 28] investigates how restrictions on arity or quantifier alternation affect the expressive power of Monadic Second Order Logic over finite structures.

My two papers [112,115] deal with invariant definability, a topic which would have deserved more attention. In it I analyze a situation where an auxiliary definable predicate P is used in the definition of a property $\mathcal{P}$, but the choice of the interpretation of P is arbitrary as long as it satisfies its requirements.

My most cited paper in finite Model Theory is [43]. It shows how to use the classical Feferman-Vaught Theorem[33] in algorithmic applications. My favorite contribution to finite Model Theory is a new method for proving non-definability in First Order (or Monadic Second Order) Logic of properties of classes of finite structures using Connection Matrices (aka Hankel Matrices), [131, 136,63]. This method uses the results of [43] in an essential way. Both the new versions of the Feferman-Vaught Theorem and the method of connection matrices work also for numeric graph parameters and graph polynomials. I discuss this further in Sect. 2.3.1.

My MSc student Nadia Labai worked on various aspects of Hankel matrices.[34] With her I published six conference papers, [139], [141], [142], [145], [146], [170]. The first four papers deal with weighted automata, [146] deals with exact learnability of graph parameters, and [170] is an attempt to define a logic where all the definable classes of structures have finite Hankel rank. Unfortunately, the reviewer of the paper for MathSciNet found an irreparable mistake in the final step of our argument. It is still open whether such a logic exists.

2.2.8 Courcelle's Theorem and Clique-width

The famous theorem of B. Courcelle states that on classes of graphs of bounded tree-width, every monadic second-order property is decidable in polynomial time, actually it is even Fixed Parameter Tractable **FPT**. The converse is not true without further assumptions. With my postdoc Julian Mariño I investigated under what conditions the converse is true [41].

As mentioned earlier, Udi Rotics had been my MSc student 10 years before. Although he worked then on databases he was not very fond of logic. He left academia in order to work at IBM. Now, 10 years later he returned and wanted to work on a PhD. However, he insisted it should not involve mathematical

[33] Solomon Feferman Robert Vaught. "The first order properties of products of algebraic systems". Fundamenta Mathematicae. 47 (1) (1959) pp. 57–103.

[34] Nadia Labai, Definability and Hankel Matrices. M.Sc. Thesis, Technion-IIT, 2015.

logic. His first result treated the graph spanner problem. It turned out that V. Guruswami[35] (then a student of C. Pandu Rangan and of M.S. Mandanal at IIT Chennai) had obtained similar results. This resulted in our joint paper [30]. But Udi's main ambition was to find interesting generalizations of Courcelle's Theorem. His original idea proved promising, but only after I used Monadic Second Order Logic we finally found a satisfactory weaker condition than tree-width to prove a similar theorem. Logic always strikes back. When we showed our result to B. Courcelle he noticed that our condition was equivalent to clique-width, a notion he had introduced shortly before. He was aware of something like our theorem, but he had not yet published his version of it. We decided to merge our results and this led to the papers [114, 35, 37] which had a great impact reaching over 1000 Google citations. This was the beginning of my intensive collaboration with B. Courcelle and U. Rotics. Paper [117] won the best paper award at the 2000 conference on graph transformation and was further expanded into [38]. U. Rotics' PhD thesis contained material from [114, 35], 37] and also from [34]. A. Glikson wrote his MSc thesis under my guidance on NCE graph Grammars and Clique Width [122]. As already mentioned before, clique-width (and hence also tree-width) can be used to shed some light on the complexity of ♯**SAT**, the counting version of the satisfiability problem, see [47].

In [37] we noticed that Courcelle's classical theorem and its extension for clique-width could also be applied to numeric graph parameters and the permanent and the immanent of matrices[36] viewed as adjacency matrices of graphs. Later it turned out it also worked for graph polynomials in general, but at the time I did not know of other natural examples It was V. Turaev who suggested I should try to prove a generalization of Courcelle's Theorem for the Jones polynomial of knot theory. Turaev's research indeed deals with low-dimensional topology, quantum topology, and knot theory and their interconnections with quantum field theory. I spent a year learning knot theory. Later Boris Zil'ber showed me a model theoretic way for looking at graph polynomials. I wrote a personal account on how I got to like graph and knot polynomials, see [75]. It was a contribution to a special issue of the journal "Model theory" dedicated to Zil'ber's 75th birthday.

2.2.9 The Tutte Polynomial and Knot Polynomials

A knot diagram is a picture of a projection of a knot onto a plane which contains the information if the crossings are over-crossings or under-crossings so that the original knot can be reconstructed. In other words it is a planar graph of degree 4 where the edges are labeled. A signed graph is an edge-labeled graph. The Jones polynomial is a knot invariant which can be computed from a knot diagram. Turaev's challenge for

[35] Venkatesan Guruswami, as a student he wrote papers as G. Venkatesan. Once in the US he used V. Guruswami as his name. https://en.wikipedia.org/wiki/Venkatesan_Guruswami.

[36] Like the determinant, the permanent and the immanent are numeric invariants of matrices.

me was to analyze whether my method applies to the Jones polynomial. The Tutte polynomial is one of the most amazing graph inviariants. The Jones polynomial is related to the Tutte polynomial. It can be obtained from the Tutte polynomial directly for alternating knot diagrams. Otherwise it can be obtained from the Tutte polynomial for signed graphs.

My first result showed that for signed graphs of tree-width at most k the Tutte polynomial is fixed parameter tractable. Hence the same holds for the Jones polynomial and other related knot polynomials, [119, 45]. My method was based on showing that the signed Tutte polynomial was definable in a suitable extension of monadic second order logic. Hence the method was applicable to other cases as well. This included the matching polynomial and the chromatic polynomial, but at that time I was not yet aware of the abundance of graph polynomials hiding in the literature. Direct combinatorial proofs for the (unsigned) Tutte polynomial had been obtained by others at about the same time, but my result was truly new for the signed case. I continued to work on the parameterized complexity of knot polynomials with my postdoctoral student Julian Mariño, [40] and I was invited to present these result in 2003 at the 3 week mini-semester "Knots in Poland" at the Banach Center of the Polish Academy of Sciences.

While I was a visiting professor at ETH during a sabbatical I supervised Martin Lotz for his diploma thesis studying the complexity of the Tutte polynomial in an algebraic computational model, [42]. Much later I studied with my former PhD student, Tomer Kotek, the evaluation of the Tutte and the matching polynomial for a fixed graph at different evaluation points, see [72].

The Tutte polynomial is also related to the Potts model and the Ising model in statistical physics. The Ising polynomial is a graph polynomial derived from the Ising model. With Tomer Kotek we also studied efficient computations of generalized Ising polynomials on graphs with fixed clique-width, [144].

2.3 Back to Mathematics

When I joined the Technion in Haifa some of the senior professors were hesitant to give me tenure fearing that I was only a fake convert to Computer Science and that I would revert to Mathematics after they lost their grip on me. In a way they were not that wrong. Complexity of algorithms is still considered Computer Science, whereas Complexity Theory is more in the domain of Mathematics. Studying the complexity of evaluations of graph and knot polynomials led me naturally back into Pure Mathematics.

2.3.1 One, Two, Many Graph Polynomials

In 2005, while attending CSL, the European Conference in Computer Science Logic in Oxford, I paid a visit to Boris Zil'ber. While chatting about my recent work we discovered a model theoretic way of showing that many combinatorial

counting functions in graph theory turn out to be graph polynomials.[37] I was so overwhelmed by the abundance of graph polynomials that I decided to try to develop a general theory of graph polynomials. At one of the Dagstuhl seminars T. Zaslavsky suggested "From a zoo to a zoology" as the title of this research program. My first steps in this direction were published in [125, 49].

I was lucky to be able to build a team of graduate students and collaborators with whom I could study various aspects of graph polynomials. Between 2006 and 2016 I was running research seminars on graph polynomials. Regular participants were my own graduate students Ilya Averbouch, Tomer Kotek and Vsevolod Rakita. Other participants were Eldar Fischer, Elena Ravve, Benny Godlin, and later Orly Herscovici also attended the seminar regularly and contributed to the emerging results. When Peter Tittmann visited the Technion for the first time we discovered our joint interest in graph polynomials, and he became a remote collaborator.[38] Other collaborators were M. Bläser, A. Goodall, S. Noble, M. Hermann, R. Zhang.

One of the recurrent themes of our work was the use of model theoretic methods in dealing with graph polynomials and combinatorial counting functions. In 2007–2011 our efforts were funded by the Israeli Science Foundation under the heading "Model Theoretic Methods in Combinatorics",

In these seminars we were slowly building a comprehensive theory. I will not discuss each of the papers in this endeavour written between 2004 and 2024, but highlight only the most relevant ones. One of the amazing features of the Tutte polynomial is its universality with respect to deletion and contraction of edges in graphs.[39] Ilya Averbouch's PhD thesis introduces new graph polynomials which have similar universality properties.[40] One of them is the trivariate polynomial $\xi(x, y, z)$ which generalizes both the Tutte and the matching polynomial, [127, 52, 54]. Another is a polynomial based on the enumeration of vertex induced subgraphs with respect to the Number of Components, [132, 53]. Both of these polynomials are mentioned in contributions to this volume by G. Farr and K. Morgan, and by P. Tittmann.

Another graduate student of our seminar, Benny Godlin, contributed to our joint efforts, although his formal thesis work was with O. Strichmann on program verification. Benny obtained his MSc under Strichmann's supervision and they published four papers. With us he co-authored [124, 127, 128, 52, 56]. He could have easily gotten his PhD with me, but, although he enjoyed doing research, he

[37] Johann A. Makowsky and Boris Zil'ber. "Polynomial invariants of graphs and totally categorical theories". https://www.logique.jussieu.fr/modnet/Publications/Preprint%20server/papers/21/, 2006. MODNET Preprints, number 21.

[38] See his contribution to this volume.

[39] see: Martin Aigner, A Course in Enumeration, Graduate Texts in Mathematics 238, Springer, 2007, Chapter 9.

[40] Ilia Averbouch, Completeness and Universality Properties of Graph Invariants and Graph Polynomials. Ph.D. Thesis, Technion-IIT, January 2011.

disliked writing it up and taking additional required courses for credit. Neither did he like teaching, hence he decided to leave academia without his well-deserved PhD.

Tomer Kotek's PhD thesis studied definability and non-definability of combinatorial functions.[41] With Tomer we finally expanded and published my earlier work with B. Zil'ber, [129, 169]. Two other papers deal with holonomic functions and some generalizations thereof, [59], 57]. In the course of studying which graph polynomials are not definable in monadic second order logic we developed the new and very powerful method of *connection matrices* aka *Hankel matrices*, [136, 63].

Together with M. Grohe I organized a special session at the Annual AMS Meeting of 2009 in Washington, DC. Its theme was "Application of Logic to Finite Combinatorics". We then edited a book in the AMS Series "Contemporary Mathematics" under the same title.[42] Tomer co-authored with me and other collaborators two long survey papers. Besides [169], we also wrote about application of logic to combinatorial sequences and their recurrence relations, [168] putting our earlier work on recurrence relations into a more general perspective.

Two graphs are Tutte equivalent if they have the same Tutte polynomial. A graph is Tutte unique if no other graph is Tutte equivalent to it. There are graphs which are not Tutte unique. B. Bollobás, L. Pebody and O. Riordan conjectured that almost all graphs are Tutte unique.[43] The conjecture is wide open. The same can be asked for other graph polynomials P and specific graph classes C: Is (almost) every graph in C P-unique. In his MSc thesis[44] Vsevolod Rakita studied this question for large families of graph polynomials, [151, [70]. The same problem for hypergraphs was explored with R. Zhang in [67]. In his PhD thesis Vsevolod studied a special class of graph polynomials, the Harary polynomials,[45] see [71]. Given a graph G and a univariate graph polynomial P we say that $P(G : x)$ is unimodal if its coefficients are unimodal. P is (almost) unimodal if $P(G, x)$ is unimodal for all (almost) all graphs. The matching polynomial is unimodal, the independence polynomial is not.[46] In [73] Vsevolod and I study large classes of graph polynomials which are almost unimodal. An open question arising from Vsevolod's work on Harary polynomials, led to our joint work with Y. Filmus and E. Fischer on MC-finite integer sequences, [74], see further below.

One can view graph polynomials in two different ways. In one view graphs are used to index polynomials, in the other view polynomials are used to reflect

[41] T. Kotek, Definability of Combinatorial Functions, Ph.D. Technion-IIT, May 2012.

[42] Model Theoretic Methods in Finite Combinatorics, M. Grohe and J.A. Makowsky eds., Contemporary Mathematics, vol 558, American Mathematical Society (2011).

[43] Béla Bollobás, Luke Pebody, and Oliver Riordan. Contraction–deletion invariants for graphs. Journal of Combinatorial Theory, Series B, 80(2):320–345, 2000.

[44] Vsevolod Rakita, On Weakly Distinguishing Graph Polynomials. M.Sc. Thesis, Technion-IIT, 2020.

[45] Vsevolod Rakita, Harary polynomials and generating graph polynomials, Ph.D. Thesis, Technion-IIT, May 2023.

[46] Yousef Alavi, Paresh J Malde, Allen J Schwenk, and Paul Erdős. The vertex independence sequence of a graph is not constrained. Congressus Numerantium, 58(15–23):2, 1987.

combinatorial properties of graphs. In the latter view we say that two polynomials P and P' are semantically equivalent if for all graphs G, H such that $P(G) = P(H)$ implies that $P'(G) = P'(H)$. The coefficients or the location of the roots of semantically equivalent graph polynomials may behave very differently. Such issues are discussed in joint work with E. Ravve, [61], and from a general logical point of view with her and T. Kotek in [69].

With my collaborators and like-minded colleagues we have managed to create a new field in graph theory with two Dagstuhl Seminars, two workshops at the MATRIX Institute, two Special sessions at AMS meetings and one SIAM mini-symposium.

2009 AMS-ASL Special Session on *Model Theoretic Methods in Finite Combinatorics*, January 2009, Washington DC,
Organizers: M. Grohe and J.A. Makowsky

2014 SIAM Conference on Discrete Mathematics, Minneapolis, June 2014
Mini-symposium: *Graph Polynomials: Towards a General Theory*,
Organizers: Jo Ellis-Monaghan, Andrew Goodall and J.A. Makowsky

2016 Dagstuhl Seminar 16241:
Graph Polynomials: Towards a Comparative Theory,
Organizers: Jo Ellis-Monaghan, Andrew Goodall, Johann A. Makowsky, Iain Moffatt

2017 MATRIX Institute. Tutte Centenary Retreat.
Organizers: Graham Farr, Dillon Mayhew, Kerri Morgan, James Oxley and Gordon Royle.

2019 Dagstuhl Seminar 19401: *Comparative Theory for Graph Polynomials*
Organizers: Jo Ellis-Monaghan, Andrew Goodall, Iain Moffatt, Kerri Morgan

2022 Special Session on
Graph and Matroid Polynomials: Towards a Comparative Theory,
AMS-SMF-EMS Joint International Meeting, Grenoble, France, July 2022
Organizers: E.Gion, J.A.Makowsky and J.Oxley

2023 MATRIX Institute. Uniqueness and Discernement in Graph Polynomials.
Organizers: Jo Ellis-Monaghan, Iain Moffatt, Kerri Morgan and Graham Farr.

2.3.2 Recurrence Relations for Combinatorial Functions

In the joint work by C. Blatter and E. Specker from 1984,[47] they introduced a powerful method to show that certain combinatorial counting functions $f(n)$ are MC-finite, i.e., if computed modulo any fixed natural number m the sequence

[47] Blatter, Christian, and Ernst Specker. "Recurrence relations for the number of labeled structures on a finite set." Logic and Machines: Decision Problems and Complexity: Proceedings of the Symposium "Rekursive Kombinatorik" held from May 23–28, 1983 at the Institut für Mathematische Logik und Grundlagenforschung der Universität Münster/Westfalen. Springer Berlin Heidelberg, 1984.

$f(n) \pmod m$ is ultimately periodic. The counting function counted the number of binary relations $R \subseteq [n]$ on set $[n] = \{1, \ldots, n\}$ where $([n], R)$ satisfies a property expressible in Monadic Second Order Logic. I called it the Specker-Blatter Theorem, because the theorem was originally Specker's idea. There were many challenging open problems left in their work. In particular, it was left open, whether the same holds for relation of arity higher than two.

I got fascinated by their theorem and resolved to try to prove the theorem for arbitrary arities of R and dedicate the theorem to E. Specker for his 80th birthday in 2000. When I spoke to Specker about it, he said that he did not think this was possible. Indeed, I had no such proof for his 80th birthday. When I lectured about the Specker-Blatter Theorem in 2002, Eldar Fischer, then a newly hired junior faculty at the Technion, exhibited a counter-example with R of arity 4. Eldar and I have collaborated ever since. The papers [123, 168, 76] gradually solve all the open problems from the papers by E. Specker and C. Blatter. An obituary and homage to E. Specker can be found in [55].

As mentioned before, see Sect. 2.3.1, Tomer Kotek's PhD thesis dealt with definability questions of combinatorial counting functions, taking as a starting point the Specker-Blatter Theorem. In [57] and [59] we study variations on holonomic sequences based on lattice paths, and in [163] and [62] we study recurrence relations for graph polynomials. In [KMR-2018] we study sequences of polynomials arising from graph invariants.

2.4 Varia

My remaining papers document my various interests and involvement in questions of algorithmics, non-monotonic logics, program and hardware verification, and software engineering, as a result of collaboration with graduate students or colleagues, or as a reaction to my own reading, but they do not represent long-term involvement in their respective research areas. Among these there are:

Robotics: A paper on programming reactive systems with statecharts, [103];

Geometry: Papers on decidability questions in geometry [68, 150, 149, 174], including Origami geometry;

Teaching logic: Papers reflecting upon how to teach logic for Computer Science, [50, 143, 64];

Models of Computation: Papers reflecting upon computational models, [159, 158, 172]. In particular, in [167] we discuss alternative computational models for the study of graph polynomials.

Quantum computing: After D. Aharonov gave a colloquium talk in our department at the Technion, I suggested to her that the methods of Section in Sect. 2.2.8 could be used to classically simulate the quantum Fast Fourier Transform, [46].

The journal version of this paper was not published because our results were independently discovered.[48] However, our result was given due credit.

User interfaces: With my PhD student Jacob Ukelson I worked on user interfaces, [26]. I inherited Jacob from his supervisor Miki Rodeh after Miki left our department.

2.5 Essays and Prose

2.5.1 Theater and Music

I started to write about theater and music while still at the gymnasium. In [**E12**] I wrote about J. Girodoux's play "Electra" and in [**E13**] about S. Beckett's "Play". I also wrote a long essay (unpublished) about the genre of radio plays after listening regularly to the weakly radio play broadcasted late at night by the Swiss public radio. I also read all the radio plays published 1963 in an anthology.[49] Between 1965 and 1970 I was an intensive theatergoer not only in Zurich, but also in Paris and Berlin (East and West). In Paris it was the *TNP (Théatre National Populaire)* and the *Odéon*, where I saw productions by G. Wilson and J.L. Barrault of Brecht, Beckett, Kafka, Billetdoux, Audiberti, and Chekhov. In the *Théatre de la Huchette* I saw two short plays by Ionescu, "La cantatrice chauve" and "La leçon". In Berlin I saw Brecht's "Galileo" in the *Theater am Schiffbauerdamm* and his version of "Antigone" in the *Schaubühne am Halleschen Ufer*, I saw Büchner's "Leonce and Lena" and Valérie's "Mon Faust". I also attended the world premiere of P. Weiss' "The investigation" produced by E. Piscator with the music by L. Nono. I studied treatises on contemporary theater and the theater of the absurd.[50]

In [**E12**] I wrote about Shostakovitch and his 5th symphony on the occasion of its first Swiss performance in the middle of the Cold War. During the stormy events of 1967–1970 the students of the Zurich conservatory rebelled against its conservative cultivation of the classical repertoire. They published their own (mimeographed) periodical "Dissonanz" and organized concerts of and lectures about avant-guard music, which featured music and lectures by M. Kagel, J. Cage, Y. Xenakis and many others.

[48] Markov, Igor L., and Yaoyun Shi. "Simulating quantum computation by contracting tensor networks." SIAM Journal on Computing 38.3 (2008): 963–981.

[49] Spectaculum. Texte moderner Hörspiele. Brecht, MacLeish, Dürrenmatt, Herbert, Frisch, Camus, Andersch, Yacine, Eich, Harube, Bachmann, Dagerman, Weiss, Compton, Walser, Pinter, Konstatinovic, Hildesheimer, Pinget, Beckett. Michel, Karl Markus (Hg.): Verlag: Frankfurt a.M. : Suhrkamp, 1963.

[50] Peter Szondi, "Theory of the Modern Drama", Suhrkamp Verlag, Frankfurt a.M., 1956,

Martin Esslin, "The Theatre of the Absurd", Anchor Books, Doubleday and Company, Inc., Garden City, New York, 1961.

After attending Xenakis' lecture and a private talk with him in 1968 I published [**E1**]. A few months later I had a chance to interview H.W. Henze who conducted his "Undine" at the Zurich Opera in 1969. I published it as [**E2**]. Just before his visit to Zurich, the performance of Henze's "The raft of Medusa" in Hamburg ended in a political scandal and Henze refused to give any interview to the press of the establishment, including "Der Spiegel". My interview was quite spectacular, and excerpts of it are published in Henze's collected writings.[51] I remained friendly with Henze and also was invited to write an essays in his series "Die Zeichen" [**E4**]. At the time my former schoolmate Roger Cahn was chief editor of "Musik & Theater", an influential monthly periodical dedicated to the world of music and theater. In 1980 I wrote for him a review of Henze's book "Die englische Katze", [**E5**]. In 1981, on the occasion of G. Bertini's candidacy as chief conductor for the Zurich Tonhalle Orchester, I interviewed G. Bertini for "Musik & Theater", [**E3**]. It did not help. Ch. Eschenbach was chosen instead, but this choice was a failure. It was my last journalistic publication about music or theater.

2.5.2 Politics

In 1961, two high-school students, Dieter Neupert und Marcel Hoehn, founded in 1961 a movie club *Der Mittelschul-Filmklub Zürich*, MFK. The program was influenced by the "Cahiers du Cinéma". The screening of Jules Dassin's "Rififi" and Jean-Luc Godard's "Vivre sa vie" brought the club into trouble, when an ultra-conservative teacher, Dr. J. Egli, complained to the police that the club was violating the directives of the censorship board. When the police dismissed the case, Dr. Egli wrote letters to parents of the students of his school, warning them of the dangers of MFK which supposedly corrupted the morals of the tender souls of the students. The editorial board of the *Mittelschulzeitung* ZMZ[52] invited Dr. Egli for an interview, which he declined. We then decided to expose Dr. Egli's mean crusade against the MFK, [**E14**]. The joint conference of the rectors of the Gymansia decided to forbid selling ZMZ on the territory of the various cantonal schools. After the Social Democrat Party asked in an interpellation in Zurich's City Parliament whether the

[51] Hans Werner Henze, Schriften und Gespräche 1955–1979, expanded edition, Berlin 1981. original edition as: Hans Werner Henze, Schriften und Gespräche 1955–1975, Munich 1976, with Italian translation (Feltrinelli) and English translation (Faber & Faber).

[52] My co-editors were Ruth Gurny, Susanne Beyeler and Werner Sauber. Ruth Gurny later became a sociologist and social activist, member of the Cantonal Parliament of Zurich and professor of Sociology at the University of Applied Sciences, Zurich. Susanne Beyeler and Werner Sauber became cineasts and script writers. (https://de.wikipedia.org/wiki/Werner_Sauber). Werner Sauber was also involved with the "Bewegung 2. Juni", one of the anarchist organisations in Berlin (https://en.wikipedia.org/wiki/2_June_Movement). (Philip) Werner Sauber died in a fire exchange with the German police in 1975. Heis the main character in the novel by Ulrike Edschmid: "Das Verschwinden des Philip S.", Roman. Suhrkamp, Berlin 2013.

freedom of press was violated in the case of ZMZ, the scandal was perfect and the movie club was left in peace.

Later, in 1969, I became a co-founder of another movie club "Der Andere Film" (The Other Film) and wrote an essay about it, [**E22**]. The club's aim was to show movies not usually shown in commercial cinemas. I remember that we showed the Indian classic "Pather Panchali" by Satyajit Ray (1955), and the Cuban "Lucia" by Humberto Solás (1968). In 1972, the club was transformed into a movie cooperative called "Filmcoopi" featuring political and documentary movies. Today it still exists and it is one of the big Swiss distributors (and occasionally co-producer) of art-house movies.

The ZMZ was known for its subversive and avantguardish tendencies already before I joined its editorial board. In 1964 it published a list of doctors who were willing to prescribe the anti-baby-pill when it still was considered immoral and illegal. At that time the driving force behind ZMZ were G. Kohler, later professor of philosophy at the University of Zurich, and J.-P. Hoby, later a long-term director of the cultural program (culture minister) of the City of Zurich. In 1966 I, as a representative of the rebellious generation, I was interviewed together with some friends by the leading weekly magazine "Sie+ER" about difficulties of growing up, [**E16**]. The "establishment" started to take notice of the new generation and its specific problems.

I did have quite a bit of sympathy for various aspects of the anti-establishment activism of 1968. In 1968 I published two essays on the impact of language on equal chances in education, [**E18**, **E19**]. In the same year the international youth protest movement reached Switzerland. When in summer the Swiss parliament passed a constitutional amendment concerning the Swiss Federal Institute of Technology (ETH) uniting the Cantonal Institute of Technology of Lausanne (EPUL) with ETH, many thought of it as a missed chance for a more general university reform. I was among those who initiated a popular drive to bring this amendment to a popular vote (referendum). The establishment cried foul, the initiative is an abuse of democracy. In the end the students collected enough signatures to demand a Federal referendum. The vote was scheduled for the next year.

When the Zurich police overreacted in dispersing demonstrations of the rebellious youth in 1968 the mathematician and logician E. Specker and the sculptor H. Honegger[53] published a manifesto, the "Zürcher Manifest" taking sides with demonstrators. The manifesto turned into a popular movement, the *Zürcher Manifest*, where I was prominently active. The socialist newspaper "Volksrecht" started to give a voice to the protesters by printing a weekly supplement "Diskus", the "Voice of action for Zurich's youth". I published two pamphlets in it, [**E20**, **E21**].

In 1969 I co-authored a long manifesto for the "Progressive Student Union (FSZ)" [**E17**], outlining FSZ's view on necessary university reforms. Academic teaching and learning should be socially relevant. The power of the full professors should be curtailed and replaced by democratic procedures where students and

[53] https://fr.wikipedia.org/wiki/Gottfried_Honegger.

lower academic staff would be represented paritetically. FSZ was a Marxist student organisation where I served as an intellectual advisor without ever being a formal member.

As an editor of the *'Zürcher Student* I published three truly political items: One was related to political events in France. A. Krivine was the candidate of the Ligue Communiste revolutionaire (trotskyist) for the 1969 presidential elections in France. De Gaulle had resigned after loosing his constitutional referendum in May 1969. Krivine was one of the leaders of the May 1968 revolt in Paris. He was the last of the generation radicalised in the 1960s to serve on the political bureau of the LCR. I translated and published A. Krivine's TV address in the *Zürcher Student* [**E25**][54] The elections were finally won by G. Pompidou.

In the weeks before the referendum about ETH I published, again in the *Zürcher Student*, the lead article [**E24**] and a long article explaining why it was mandatory to reject the proposed constitutional amendment in its current form. I wrote that Switzerland deserved a more serious university reform, [**E23**]. We, the students, won a big victory, the amendment was rejected by a large majority of the people and also of the cantons. However, the true reform we had in mind never happened. E. Engeler referred in his contribution in this volume to the events described above.

In 1983 President Ronald Reagan launched his "Strategic Defense Initiative". He proposed to build a missile defense system intended to protect the United States from attacks by ballistic nuclear missiles. Over the course of 10 years, the government spent up to $30 billion on developing the concept. Universities and research institutes were given generously research money for this purpose. However, the futuristic program remained just that–futuristic. It was formally scrapped by President Bill Clinton in 1993. In 1985 I published an article in the then still liberal Jerusalem Post criticizing this program, [**E8**]. David Parnas,[55] a Canadian early pioneer of software engineering, was invited to Washington for a hearing about the feasibility of the Defense Initiative. He was very negative about it citing many publications in support of his view. He wrote in a letter to me in July 1985 that my article was among them. One of my conclusions I stated in [**E8**] is:

> Over-funded research is like heroin: It makes one addicted,
> weakens the mind and furthers prostitution.

2.5.3 Society

The "Kursbuch"[56] is a German cultural magazine founded by Hans Magnus Enzensberger and Karl Markus Michel in 1965. It became one of the important voices of the extra-parliamentarian opposition of the 1968 student movement in

[54] https://fr.wikipedia.org/wiki/Alain_Krivine.

[55] David Parnas, Software Aspects of Strategic Defense Systems, Preprint DCS-47-IR, Department of Computer Science, University of Victoria, B.C., Canada, July 1985.

[56] https://de.wikipedia.org/wiki/Kursbuch_(Zeitschrift).

German-speaking countries. Besides political reflections it also contained important cultural contributions. For me its volume 8 from 1967 was a revelation: It dealt with New Math, Foundational studies and the theory of automata. I remained subscribed to it until well over volume 100. In 1984 its editors planned a volume with essays inspired by G. Orwells "1984". I was invited to contribute, which resulted in [**E6**]. In it I sketched the possible future of a computerized society. In particular I said that it was not yet clear who would really benefit from the new technologies, the rulers or the ruled.

Another contribution of mine is [**E7**], an essay about "elites", distinguishing between the elites, meaning the privileged members of the ruling class, and the intellectual elite of outstanding people in science, humanities and the arts. I was, and I still am, worried about the populists declaring war on both without understanding the distinction.

After living in Israel for several years, I noticed the trend of secular Jews embracing extreme forms of orthodox Judaism. It seemed like a drug of a kind. In 1985, as a reaction to this, I published an essay in the then leading Jewish weakly of Switzerland "Die Jüdische Rundschau" my reflections about this phenomenon, [**E9**]. Its title "Postzionism and the return to religious life in today's Israel" may have been one of the first usages of the term "Postzionism". In the article I describe how the spirit of secular liberal Zionism was in decline. Postzionism then could manifest itself in two ways: As radical Messianism or as non-Zionist Israeliness.

2.5.4 konKursbuch Verlag Claudia Gehrke

Ruedi Lüscher was one of my closest friends from the last years of the gymnasium time until his premature death at age 35 in 1983. We shared many common interests in literature and philosophy. Through him I got acquainted with K. Kraus, R. Musil and L. Wittgenstein. We were both involved in the 1968 activism and even published together, [**E17**, **E20**]. He wrote political essays and comments in leftist media and enrolled at university, first in Zurich, then in Frankfurt in order to study philosophy. In Frankfurt he hoped to study under T. Adorno and J. Habermas, but Adorno died shortly after Ruedi arrived there. Later in Heidelberg he studied under E. Tugendhat. His big project was his study of Fordism and his analysis of fordistic society. Fordism, for him, was a special kind of consumerism, where the working class was given a new role in capitalism as consumers who keep the capitalist machine running. It was to be his PhD thesis, but his perfectionism did not allow him to finish it on time. When he died he left an almost finished manuscript.

Ruedi published in scientific and political periodicals. One of them was the newly established "KonKursbuch" published by Claudia Gehrke's small and newly founded publishing house. "KonKursbuch"is a pun consisting of "Konkurs" (bankruptcy) and "Kursbuch" (timetable of all the state-owned rail system). It was meant as a counterpart to the aforementioned "Kursbuch". After Ruedi's death I gathered a group of his friends in order to handle Ruedi's estate. We decided to edit and publish his manuscript on Fordism and finance its publication with

Claudia Gehrke's publishing house. Her editorial credo was to publish philosophy and literature and the series "KonKursbuch". Her other line consisted in publishing erotica from a feminist (and later LGBTIQ) point of view. This line published erotic prose and a series "Das heimliche Auge" (The secret eye), with verbal and visual contributions touching upon various interpretations of erotics.

After 4 years of collaboration with various of Ruedi's friends, Claudia published Ruedi's opus magnum as "Henry und die Krümelmonster" (Henry and the cookie monsters"[57]) [**E10**]. I served as the coordinating editor and contributed an editorial note and Ruedi's short biography.

Glaudia Gehrke and I became lifelong friends, see also her contribution in this volume. She also seduced me to contribute to both of her series "KonKursbuch" and "Das heimliche Auge". I gladly agreed and published irregularly, [**L1**, **L2**, **L3**, **L4**, **L5** ,**L6**, **L7**]. It kept my dream to write literary prose alive.

2.5.5 Mathematicians as Novelists

I was always fascinated by writers who were originally mathematicians or the like. Robert Musil and Hermann Broch are my prime examples, but also A. Solzhenitsyn comes to my mind. Others were attracted to mathematics and the exact sciences, P. Valéry and R. Queneau for example. Hermann Hesse was an intimate friend of the celebrated mathematician Hermann Weil, which may have influenced Hesse's "Magister Ludi". As I said in Sect. 1.3 I still hope to complete my many drafts and sketches of prose and turn them into one or more novels. A few years ago I was asked to write a review of a book entitled " Modernism, Fiction and Mathematics". I did write the review [**O13**] and posted a longer version on arxiv. The book under review was a study of three modernist writers, Musil, Broch and Pynchon. I was not too excited about the book, but instead of expressing my disappointment I took the opportunity to write my own essay about the subject.

For me Mathematics is a literary genre. I still plan to try my hand at other genres like writing one or more novels. Retirement from research mathematics, if at all possible, should give me a chance to do so. Who knows?

3 List of Scientific Publications

3.1 Edited Books and Lecture Notes

1. J.A. Makowsky, Logic for Computer Science (Technion Course 234292), 92 pp., reprinted and augmented annualy since 1988, last edition 1997

[57] Cookie Monster is a Muppet character on the PBS/HBO children's television show Sesame Street. He is best known for his voracious appetite. "Henry" is Henry Ford and the cookie monsters are the consumers.

2. J.A. Makowsky and E.V. Ravve, Translations, Interpretations and Reductions, Course given at ESSLLI'97, Aix-en-Provence, France, August 12–22, 1997. 280 slides.
3. J.A. Makowsky and E.V. Ravve (editors), Logic Colloquium '95, Proceedings of the 1995 Annual European Summer Meeting of the Association of Symbolic Logic, Haifa, August 1995, Lecture Notes in Logic, vol. 11, Springer Verlag, 1998 348 + xvi pp.
4. J.A. Makowsky, Introduction to Database Systems (Technion Course 236363), Lecture Notes in Hebrew by Gily Leshed and supplemented by Ofer Dubrovsky, Technion 1998. Second revised edition prepared by Z. Nevo and J.A. Makowsky, Technion 2001.
5. J.A. Makowsky, Logical Aspects of Combinatorial Algorithms, Course given at ESSLLI'99, Utrecht, The Netherlands, August 12–22, 1999 by J.A. Makowsky (assisted by U. Rotics), 180 slides.
6. J.A. Makowsky and E.V. Ravve, Logical Methods in Combinatorial Computations, Course given at ESSLLI'03, Vienna, Austria, August 18–28, 2003 by J.A. Makowsky (assisted by E. Ravve), ca. 180 slides.
7. M. Baaz and J.A. Makowsky (editors), Computer Science Logic, Proceedings of the 17th International Workshop CSL 2003, of the 12th Annual Conference of the EACSL and the 8th Kurt Gödel Colloquium KGC 2003, Vienna, August 2003, LNCS 2803.
8. M. Grohe and J.A. Makowsky (editors), Model Theoretic Methods in Finite Combinatorics, Contemporary Mathematics, vol 558, American Mathematical Society, 2011.
9. J.A. Makowsky, Classical graph properties and graph parameters and their definability in SOL. Lecture Notes for the DocCourse in Structural Graph Theory, Charles University, Prague, 2014
10. J.A. Makowsky and K. Meer, P=NP over arbitrary structures. Course given at ESSLLI'14, Tübingen 2014 and ESSLLI'19 Riga 2019.

3.2 Journal Publications 1972–2024

1. J.A.Makowsky, Note on almost strongly minimal theories, Bull. Acad. Pol. Sc. vol 20, No.7, 1972, pp. 529–534.
2. J.A.Makowsky, Langages engendres a partir des formules de Scott, C. R. hebd. Acad. Sc. Paris t. 276 , 1973, pp.1585–1587.
3. J.A.Makowsky, On some conjectures connected with complete sentences, Fund. Math., vol.81, 1974, pp. 193–202.
4. J.A.Makowsky, S.Shelah and J.Stavi, Δ-Logics and generalized quantifiers, Annals of Mathematical Logic 10, 1976, pp155–192.
5. J.A.Makowsky and A.Marcja, Completeness theorems for modal model theory with the Montague-Chang semantics,I., Zeitschrift fur mathematische Logik und Grundlagen der Mathematik, Bd. 23, 1977, pp 97–104.

6. J.A.Makowsky and A.Marcja, Problemi di decidibilita in logica topologica, Rend. Sem. Mat. Univ. Padova, vol.56, 1977, pp 67–78.
7. J.A.Makowsky and S.Tulipani, Some model theory for monotone quantifiers, Archiv fur Mathematische Logik, 18, 1977, pp 115–134.
8. J.A.Makowsky, Some observations on uniform reduction for properties invariant on the range of definable relations, Fundamenta Mathematicae 99, 1978, pp 199–203.
9. J.A.Makowsky and S.Shelah, The theorems of Beth and Craig in abstract model theory, I.The abstract setting, Transactions of the AMS 256, 1979, pp 215–239.
10. S.Shelah and J.A.Makowsky, The theorems of Beth and Craig in abstract model theory, II. Compact logics, Archiv für Mathematische Logik, 21, 1981, pp. 13–35.
11. J.A.Makowsky and M.Ziegler, Topological model theory with an interior operator, Archiv fur Mathematische Logik, 21, 1981, pp. 37–54.
12. J.A.Makowsky and S.Shelah, Positive results in abstract model theory: A theory of compact logics, Annals of Pure and Applied Logic, vol. 25.3 (1983) pp.263–300.
13. B.Mahr and J.A.Makowsky, Characterizing specification languages which admit initial semantics, (full version) Theoretical Computer Science, 31 (1984) pp.49–60. Extended abstract as Conference paper 88.
14. M.Tiomkin and J.A.Makowsky, Propositional dynamic logic with local assignments, Theoretical Computer Science, vol. 36.1 (March 1985) pp.71–87.
15. J.A.Makowsky, Vopenka's principle and compact logics, Journal of Symbolic Logic, vol. 50.1 (March 1985) pp.42–48.
16. O.Grumberg, N.Francez, J.A.Makowsky and W.de Roever, A proof rule for fair termination of guarded commands, Information and Control, vol. 66.1–2 (1986) pp.83–102. First version published as Conference paper 84.
17. J.A.Makowsky and M.Vardi, On the expressive power of data dependencies, Acta Informatica, vol 23.3 (1986) pp.231–244.
18. J.A. Makowsky, Why Horn formulas matter in computer science: Initial structures and generic examples, Journal of Computer and System Sciences, vol. 34.3/4 (1987), pp. 266–292. First version published as Conference paper 90
19. A. Itai and J.A. Makowsky, Unification as a complexity measure for logic programming, Journal of Logic Programming, vol. 4.2. (1987), pp. 105–117.
20. E. Dahlhaus, A. Israeli and J.A. Makowsky, On the existence of polynomial time algorithms for interpolation problems in propositional logic, Notre Dame Journal of Formal Logic, vol. 29.4 (1988), pp. 497–509.
21. J.A.Makowsky and I.Sain, Weak second order characterizations of various program verification systems, Theoretical Computer Science, vol. 66 (1989), pp. 299–321. First version published as 91
22. V.E. Markowitz and J.A. Makowsky, Identifying extended entity–relationship object structures in relational schemas, IEEE Transactions on Software Engineering, vol. 16.8 (1990), pp. 777–790.

23. M.Tiomkin and J.A.Makowsky, Decidability of finite probabilistic propositional dynamic logic, Information and Computation, vol. 94.2 (1991), pp. 180–203.
24. E. Dahlhaus and J.A. Makowsky, Query languages for hierarchic databases, Information and Computation vol 101.1, (1992), pp.1–32 Partial versions published as conference papers 92, 93 and 94.
25. J.A.Makowsky, J.C. Grégoire and M. Sagiv, On the expressive power of side effects in propositional PROLOG, Journal of Logic Programming, vol. 12 (1992), pp. 179–188.
26. J.A.Makowsky and J. Ukelson, A formalism for interactive menu design, Interacting with Computers, vol. 4(1), (1992), pp. 83–110.
27. J.A. Makowsky and A. Sharell, On the average case complexity of SAT for symmetric distributions, Logic and Computation, vol. 5.1 (1995), pp. 71–92.
28. J.A. Makowsky and Y.B. Pnueli, Arity vs. Alternation in Second Order Logic, Annals of Pure and Applied Logic, vol. 78 (1–3), 1996, pp. 189–202. Erratum in Annals of Pure and Applied Logic, vol. 92 (1998) p. 215. Conference version published as 106
29. J. Adamek, P.T. Johnstone, J.A. Makowsky and J. Rosicky, Finitary Sketches, Journal of Symbolic Logic, vol. 62.3 (1997) pp.699–707.
30. G.Venkatesan, U.Rotics, M.S.Madanlal, J.Makowsky and C.Pandu Rangan, Restrictions of Minimum Spanner Problems, *Information and Computation*, 136 (1997) pp. 143–164.
31. M. Kaminski, J. Makowsky and M. Tiomkin, Extensions for Open Default Theories via the Domain Closure Assumption, *Logic and Computation*, vol. 8.2 (1998), pp. 169–187. Conference version published as 110 M. Kaminski, J. Makowsky and M. Tiomkin, Extensions for Open Default Theories via the Domain Closure Assumption, in *Proceedings of the 5th European Workshop on Logics in Artificial Intelligence - JELIA'96*, J.J. Alfers, L.M. Pereira, and E. Orlowska eds., Springer, Berlin 1996, pp. 373–387 (*Lecture Notes in Artificial Intelligence* 1126).
32. J.A. Makowsky and E.V. Ravve, Dependency Preserving Refinements and the Fundamental Problem of Database Design, *Data & Knowledge Engineering*, vol. 24.3 (1998) pp. 277–312. Earlier version published as conference papers 111 and 113.
33. J.A. Makowsky and E.V. Ravve, Translation Schemes and the Fundamental Problem of Database Design (Invited lecture for ER'96), In *Conceptual Modeling–ER'96*, B. Thalheim ed., LNCS vol. 1157 (1996) pp. 5–26.
34. J.A. Makowsky and U. Rotics, On the Clique-width of Graphs with Few P_4's, International Journal of Foundations of Computer Science, 10 (1999) pp. 329–348.
35. B. Courcelle, J.A. Makowsky and U. Rotics, Linear Time Solvable Optimization Problems on Certain Structured Graph Families, Theory of Computing Systems, vol. 33.2 (2000) pp. 125–150. Conference version published as 114.

36. Courcelle, B., and J. A. Makowsky. “Operations on relational structures and their compatibility with monadic second order logic.” Math. Struct. Comput. Sci. xx (2000).
37. B. Courcelle, J.A. Makowsky and U. Rotics, On the Fixed Parameter Complexity of Graph Enumeration Problems Definable in Monadic Second Order Logic, Discrete Applied mathematics, vol. 108, No. 1–2 (2001), pp. 23–52.
38. B. Courcelle and J.A. Makowsky, Fusion in Relational Structures and the Verification of Monadic Second Order Properties, Mathematical Structures in Computer Science, vol. 12.2 (2002) pp. 203–235
39. J.A. Makowsky and J.P. Mariño, Farrell Polynomials on Graphs of Bounded Tree Width, Advances in Applied Mathematics, vol. 30, (2003), pp. 160–176
40. J.A. Makowsky and J.P. Mariño, On the Parametrized Complexity of Knot Polynomials, Journal of Computer and System Sciences, vol. 67.4, (2003) pp. 742–756
41. J.A. Makowsky and J.P. Mariño, Treewidth and the Monadic Quantifier Hierarchy, Theoretical Computer Science, 303 (2003) 157–170.
42. M. Lotz and J.A. Makowsky, On the Algebraic Complexity of Some Families of Coloured Tutte Polynomials, Advances in Applied Mathematics, 32.1–2 (2004) 327–349.
43. J.A. Makowsky, Algorithmic Uses of the Feferman-Vaught Theorem, Annals of Pure and Applied Logic, 126 (2004) pp. 159–213
44. E. Fischer and J.A. Makowsky, On Spectra of Sentences in Monadic Second Order Logic with Counting, Journal of Symbolic Logic, 69.3 (2004) pp. 617–640
45. J.A. Makowsky, Colored Tutte Polynomials and Kauffman Brackets for Graphs of Bounded Tree Width, Discrete Applied Mathematics, 145.2 (2005) pp. 276–290
46. D. Aharonov, Z. Landau and J.A. Makowsky, The quantum FFT can be classically simulated, arXiv: quant-ph/0611156vl (14. November 2006).
47. E. Fischer, J.A. Makowsky and E. Ravve, Counting Truth Assignments of Formulas of Bounded Tree Width or Clique Width, Discrete Applied Mathematics, 156 (2008), pp. 511–529. Available online since October 2007.
48. A. Cohen, M. Kaminski and J.A. Makowsky, Notions of sameness by default and their application to anaphora, vagueness and uncertain reasoning, Journal of Logic, Language and Information, 17.3 (2008), pp. 285–306.
49. J.A. Makowsky, From a Zoo to a Zoology: Towards a General Theory of Graph Polynomials, Theory of Computing Systems, 43 (2008), pp. 542–562. (available online since October 2007.)
50. J.A. Makowsky, From Hilbert’s Program to a Logic Toolbox. Annals of Mathematics and Artificial Intelligence, 53.1–4 (2008), pp. 225–250. Special issue to honor the 65th birthday of Victor Marek, edited by Michael Kaminski and Miroslaw Truszczynski.
51. M. Bläser, H. Dell and J.A. Makowsky, Complexity of the Bollobas-Riordan Polynomial: Exceptional points and uniform reductions. Theory of Computing

Systems, 46.4 (2010) pages 690–706. doi: https://doi.org/10.1007/s00224-009-9213-7
52. I. Averbouch, B. Godlin and J.A. Makowsky, An extension of the bivariate chromatic polynomial. European Journal of Combinatorics, Volume 31, Issue 1, January 2010, Pages 1–17.
53. P. Tittmann, I. Averbouch and J.A. Makowsky, The Enumeration of Vertex Induced Subgraphs with respect to the Number of Components, European Journal of Combinatorics 32.7 (2011), Pages 954–974.
54. I. Averbouch, T. Kotek, J. A. Makowsky, E. V. Ravve. The Universal Edge Elimination Polynomial and the Dichromatic Polynomial. Electronic Notes in Discrete Mathematics 38, (2011) pp 77–82
55. E. Engeler, N. Hungerbühler and J. A. Makowsky. Remembering Ernst Specker (1920–2011). Elemente der Mathematik 67.3 (2012): 89–115.
56. B. Godlin, E. Katz and J.A. Makowsky, Graph polynomials: From Recursive Definitions to Subset Expansion Formulas. Journal of Logic and Computation, Volume 22(2), (2012) Pages 237–265
57. T. Kotek and J.A. Makowsky, A Representation Theorem for Holonomic Sequences Based On Lattice Paths. Fundamenta Informaticae, 117.1–4 (2012), pp. 199–213.
58. A. Durand, N. Jones, J.A. Makowsky, M. Moore, Fifty Years of the Spectrum Problem: Survey and New Results, Bulletin of Symbolic Logic, 18.4 (2012) pp. 505–553.
59. T. Kotek and J.A. Makowsky, A Representation Theorem for (q)-Holonomic Sequences. Journal of Computer and System Sciences, 80.2 (2013), pp. 363–374
60. J. A. Makowsky, Elena V. Ravve, On the Location of Roots of Graph Polynomials. Electronic Notes in Discrete Mathematics 43, (2013), pp. 201–206
61. J.A. Makowsky, E.V. Ravve and N.K. Blanchard, On the location of of roots of graph polynomials, accpted for publication in the European Journal of Combinatorics, 2014 Eur. J. Comb. 41, (2014), pp. 1–19
62. T. Kotek and J.A. Makowsky, Recurrence Relations for Graph Polynomials on Bi-iterative Families of Graphs, Eur. J. Comb. 41, (2014), pp. 47–67
63. T. Kotek and J.A. Makowsky, Connection matrices and the definability of graph parameters, Logical Methods in Computer Science 10(4) (2014), pp. 1–33. Special issue of selected papers from CSL-2012.
64. J.A. Makowsky and A. Zamansky. Keeping logic in the trivium of computer science: a teaching perspective. Formal Methods in System Design 51.2 (2017): 419–430.
65. T. Kotek, J. A. Makowsky and E. V. Ravve. On sequences of polynomials arising from graph invariants. European Journal of Combinatorics 67 (2018): 181–198.
66. A. Goodall, M. Hermann, T. Kotek, J. A. Makowsky and Seven D. Noble. On the complexity of generalized chromatic polynomials. Advances in Applied Mathematics 94 (2018): 71–102.

67. J.A. Makowsky and R. X. Zhang. On P-unique hypergraphs. Australasian Journal of Combinatorics 73.3 (2019): 456–465.
68. J.A. Makowsky, Can one design a geometry engine? On the (un) decidability of certain affine Euclidean geometries. Annals of Mathematics and Artificial Intelligence 85 (2019): 259–291.
69. J.A. Makowsky, E. V. Ravve and T. Kotek. A logician's view of graph polynomials. Annals of pure and applied logic 170.9 (2019): 1030–1069.
70. J.A. Makowsky and V. Rakita. Weakly distinguishing graph polynomials on addable properties. Moscow Journal of Combinatorics and Number Theory 9.3 (2020): 333–349.
71. O. Herscovici, J. A. Makowsky and Vsevolod Rakita. Harary Polynomials. Enumerative Combinatorics and Applications ECA 1:2 (2021) Article #S2R13
72. T. Kotek and J. A. Makowsky. On the Tutte and matching polynomials for complete graphs. Fundamenta Informaticae 186.1–4 (2022): 155–173.
73. J.A. Makowsky and V. Rakita. Almost unimodal and real-rooted graph polynomials. European Journal of Combinatorics 108 (2023): 103637.
74. Y. Filmus, E. Fischer, J.A. Makowsky and V. Rakita. MC-finiteness of restricted set partition functions. Journal of Integer Sequences 26.2 (2023): 3.
75. J.A. Makowsky. How I got to like graph polynomials. Model Theory 3.2 (2024): 465–477.
76. E. Fischer and J. A. Makowsky. Extensions and Limits of the Specker-Blatter Theorem. The Journal of Symbolic Logic (2024): 1–29.
77. E. Fischer and J.A. Makowsky. Counting finite topologies. Enumerative Combinatorics and Applications ECA 4:4 (2024) Article S2R27.

3.3 Conference Publications 1973–2024

78. J.A.Makowsky, Securable quantifiers, κ-unions and admissible sets, Logic Colloquium '73, Rose et al.ed., Amsterdam 1975 pp 409–428.
79. J.A.Makowsky, Topological model theory: A survey, Model theory and applications, P. Mangani ed. Rome 1975, pp 121–150.
80. J.A.Makowsky, The reals cannot be characterized topologically with strictly local properties and countability axioms, Fourth Scandinavian Logic Symposium, Jyvaskyla 1976, Essays on Mathematical and Philosophical Logic, Dordrecht 1978 (Reidel), pp. 251–257.
81. J.A.Makowsky, Quantifying over countable sets: Stationary logic vs. positive logic, Logic Colloquium '77 , L.Pacholski et al.ed., Amsterdam 1978, pp183–194.
82. J.A.Makowsky, Measuring the expressive power of dynamic logics: An application of abstract model theory, ICALP 1980, Lecture Notes in Computer Science vol.85, pp.409–421 (MR 82c # 03022, ZB 465.68012). Erratum to "Measuring the expressive power of dynamic logic." ICALP 1981, Lecture Notes in Computer Science vol.115.

83. A.Chandra , H.Lewis and J.A.Makowsky, Embedded implicational dependencies and their inference problem (the untyped case), Proceedings XP1 Workshop on Relational Database Theory, Stony Brook, NY 1980 (SIGMOD 1981).
84. O.Grumberg, N.Francez, J.A.Makowsky and W.de Roever, A proof rule for fair termination of guarded commands, Algorithmic languages, de Bakker ed., Amsterdam 1981,pp 399–416. Journal version published as 16.
85. A.Chandra , H.Lewis and J.A.Makowsky, Embedded implicational dependencies and their inference problem (the typed case), ACM Symposium on the Theory of Computing 1981, pp.342–354.
86. J.A.Makowsky, Characterizing database dependencies, ICALP 1981 Lecture Notes in Computer Science, vol.115, 1981, pp. 86–97.
87. J.A.Makowsky, Model theoretic issues in theoretical computer science, part I: Relational Data Bases and Abstract Data Types, in: Logic Colloquium '82, Proceedings of the Colloquium held in Florence 23–28 August 1982, G. Lolli et al. eds., North Holland 1984, pp. 303–343.
88. B.Mahr and J.A.Makowsky, Characterizing specification languages which admit initial semantics, Proceedings of the 8th CAAP 1983, LNCS 159 (1983) pp.300–316. Journal version published as 13.
89. B.Mahr and J.A.Makowsky, An axiomatic approach to specification languages, extended abstract, Theoretical Computer Science, 6th GI-Conference, Dortmund 1983, Lecture Notes in Computer Science, vol. 145, 1983, pp. 211–220.
90. J.A. Makowsky, Why Horn formulas matter in computer science: Initial structures and generic examples, Mathematical Foundations of Software Development, CAAP 1985, LNCS 185 (1985) pp.374–385 Journal version published as 18
91. J.A.Makowsky and I.Sain, On the equivalence of weak second order and nonstandard time semantics for Floyd-Hoare logic, Proceedings of the First IEEE Symposium on Logic in Computer Science, Cambridge, Mass. 1986, pp.293–300 Journal version published as 21
92. E.Dahlhaus and J.A. Makowsky, The choice of programming primitives for SETL-like programming languages, ESOP '86 (European Symposium on Programming, Saarbrucken, March 17–19 1986), B. Robinet and R. Wilhelm eds., LNCS 213 (1986) pp. 160–172. Journal version published as 24
93. E. Dahlhaus and J.A. Makowsky, Computable directory queries, CAAP '86 (11th Colloquium on Trees and Algebra in Programming, Nice, March 24–26 1986), P. Franchi-Zannettacci ed., LNCS 214 (1986), pp. 254–265 Journal version published as 24
94. E. Dahlhaus and J.A. Makowsky, Computable directory queries, Proceedings of the workshop on "Logic and Computer Science, New trends and applications", Torino, October 1986, in: Rend. Sem. Mat. Univ. Pol. Torino, Fascicolo speciale 1987, pp. 165–197. Journal version published as 24

95. N. Rotitz (U. Rotics), J.A. Makowsky and V.E. Markowitz, Entity-Relationship Consistency for Relational Schemas, Proceedings of ICDT '86, Rome, September 1986, LNCS vol. 243 (1986), pp. 306–322
96. A. Ban and J.A. Makowsky, MUSICIAN - A Music Processing and Synthesis System, Proceedings of the 12th International Computer Music Conference, The Hague 1986, Addendum, pp. A22-A25
97. V.E. Markowitz and J.A. Makowsky, Incremental reorganization of relational databases, VLDB '87, Brighton England, September 1987, P. M. Stocker and W. Kent, eds., pp. 127–135.
98. V.E. Markowitz and J.A. Makowsky, Incremental restructuring of relational schemas, Proceedings of the 4th International Conference on Data Engineering, Los Angeles, February 1988, pp. 276–284.
99. J.A. Makowsky, The impact of model theory on computer science, Proceedings of the IX Congress of Logic, Methodology and Philosophy of Science, Uppsala 1991, D. Prawitz et al. eds., North Holland 1994, pp. 239–262.
100. Y. Bargury and J.A.Makowsky, The expressive power of transitive closure and 2–way multihead automata, Computer Science Logic, Refereed papers from CSL'91, LNCS vol. 626 (1992) pp. 1–14.
101. A. Calò and J.A. Makowsky, Ehrenfeucht–Fraïssé games for transitive closure, Logical Foundations of Computer Science–Tver '92, LNCS vol. 620, 1992, pp. 57–69.
102. R.A. Hasson and J.A. Makowsky, Update languages versus query languages, Proceedings of the XI International Conference of the Chilean Computer Science Society, Santiago de Chile, October 1991, pp. 17–30. Published also in book form by Plenum Publishing Corporation under the title: *Computer Science: Research and Applications*, R. Baeza-Yates and U. Manber eds., (1992), pp. 23–34.
103. J.C. Grégoire, J.A. Makowsky and S. Levin, Programming reactive systems with statecharts, The 5th Israeli Conference on Computer Systems and Software Engineering, Herzlia, May 1991, pp. 87–103.
104. J.A. Makowsky and Y.B. Pnueli, Oracles and Quantifiers. Computer Science Logic, Refereed papers from CSL'93, LNCS, vol. 832, 1994, pp. 189–222.
105. J.A. Makowsky, Capturing Complexity Classes with Lindström Quantifiers, Invited Lecture, Proceedings of the 19th International Symposium, MFCS'94 Kosice, Slovakia, LNCS vol. 841, 1994, pp. 68–71.
106. J.A. Makowsky and Y.B. Pnueli, Arity vs. Alternation in Second Order Logic, Logical Foundations of Computer Science–St. Petersburg 1994, LNCS, vol. 813, 1994, pp. 240–252. Journal version published as 28.
107. J.A. Makowsky, Capturing Relativized Complexity Classes with Lindström Quantifiers, Invited Lecture, *The Foundational Debate*, W. DePauli-Schimanovich ed., Kluwer Academic Publishers, 1995, pp. 133–140.
108. J.A. Makowsky and Y.B. Pnueli: Logics Capturing Relativized Complexity Classes Uniformly, Proceedings of LCC'94, Logic and Computational Complexity, D. Leivant ed., Lecture Notes in Computer Science vol. 960, 1995, pp. 463–479.

109. J.A. Makowsky and E. Ravve, Incremental model checking for decomposable structures, Mathematical Foundations of Computer Science 1995, J. Wiedermann and P. Hajek eds., Lecture Notes in Computer Science, vol. 969 (1995), pp. 540–551.
110. M. Kaminski, J. Makowsky and M. Tiomkin, Extensions for open default theories via the domain closure assumption, in *Proceedings of the 5th European Workshop on Logics in Artificial Intelligence - JELIA'96*, J.J. Alfers, L.M. Pereira, and E. Orlowska eds., Springer, Berlin 1996, pp. 373–387 (*Lecture Notes in Artificial Intelligence* 1126). Journal version published as 31.
111. J.A. Makowsky and E.V. Ravve, Translation Schemes and the Fundamental problem of Database Design (Invited lecture for ER'96), In *Conceptual Modeling–ER'96*, B. Thalheim ed., LNCS vol. 1157 (1996) pp. 5–26. Journal version published as 32
112. J.A. Makowsky, Invariant Definability, in *Computational Logic and Proof Theory*, Proceedings of the 5th Kurt Gödel Colloquium, KGC'97, Vienna, August 1997, LNCS vol. 1289 (1997), pp. 186–202.
113. J.A. Makowsky and E.V. Ravve, The Fundamental problem of Database Design (Invited lecture for SOFSEM'97), SOFSEM'97: Theory and Practice in Informatics, LNCS Volume 1338 (1997), pp. 53–69. Journal version published as 32
114. B. Courcelle, J.A. Makowsky and U. Rotics, Linear time solvable optimization problems on certain structured graph families, Proceedings of WG'98, Smolenice, Slovakia, June 1998, LNCS vol. 1517 (1998) pp. 1–16.
115. J.A. Makowsky, Invariant Definability and P/poly, *Computer Science Logic*, Proceedings of CSL'98, Brno, Czech Republic, August 1998, (G. Gottlob, E. Grandjean and K. Seyr, eds), LNCS, vol. 1584 (1999) pp. 142–158.
116. J.A. Makowsky and K. Meer, Polynomials of Bounded Tree Width, Extended Abstract, In: Formal Power Series and Algebraic Combinatorics, Porceedings of the 12th International Conference, FPSAC'00, Moscow, Russia, June 2000, D. Krob, A.A. Mikhalev and A.V. Mikhalev eds., Springer, 2000, pp. 692–703.
117. B. Courcelle and J.A. Makowsky, VR and HR Graph Grammars: A Common Algebraic Framework Compatible with Monadic Second Order Logic, "VR and HR graph grammars: A common algebraic framework compatible with monadic second order logic." Graph transformations (2000). Journal version published as 38
118. J.A. Makowsky and K. Meer, On the Complexity of Combinatorial and Metafinite Generating Functions of Graph Properties in the Computational Model of Blum, Shub and Smale, Proceedings of the 14th International Workshop, CSL'2000, Annual Conference of the EACSL, Computer Science Logic, LNCS vol. 1862 (P. Clote and H. Schwichtenberg eds.), pp. 399–410
119. J.A. Makowsky, Colored Tutte Polynomials and Kauffman Brackets for Graphs of Bounded Tree Width, Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms SODA'01, Washington DC, 2001, pp. 487–495 Journal version published as 45

120. J.A. Makowsky and J. Mariño, Graph Polynomials on Graphs of Bounded Tree Width, accepted to FPSAC'01, Phoenix, USA, May 2001. Journal version published as 39
121. J.A. Makowsky and J. Mariño, Treewidth and the Monadic Quantifier Hierarchy, accepted to LCCS'01, Paris, France, August 2001. Journal version published as 41
122. A. Glikson and J.A. Makowsky, NCE graph Grammars and Clique Width, in: Graph-Theoretic Concepts in Computer Science, Proceedings of WG'03, LNCS 2880 (2003) 237–248.
123. E. Fischer and J.A. Makowsky, The Specker-Blatter Theorem Revisited: Generating Functions for Definable Classes of Structures, in:Computing and Combinatorics, Proceedings of COCOON'03, LNCS 2697 (2003) 90–1001.
124. J.A. Makowsky, U. Rotics, I. Averbouch and B. Godlin, Computing graph polynomials on graphs of bounded clique-width, in: Graph-Theoretic Concepts in Computer Science, 32nd International Workshop, WG 2006, Bergen, Norway, June 22–23, 2006, Revised Papers, Lecture Notes in Computer Science vol. 4271 (2006) pp. 191–204.
125. J.A. Makowsky, From a Zoo to a Zoology: Descriptive Complexity for Graph Polynomials, in: Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, CiE 2006, Swansea, UK, July 2006, Lecture Notes in Computer Science vol. 3988 (2006) pp. 330–341
126. M. Bläser, H. Dell and J.A. Makowsky, Complexity of the Bollobas-Riordan Polynomial: Exceptional points and uniform reductions, 3rd International Computer Science Symposium in Russia (CSR 2008) June 7–12, 2008, Moscow, Russia, Lecture Notes in Computer Science vol. 5010 (2008) pp. 86–98.
127. I. Averbouch, B. Godlin and J.A. Makowsky, A Most General Edge Elimination Polynomial, WG 2008, 34th International Workshop on Graph-Theoretic Concepts in Computer Science. 30 June - 2 July 2008, Durham University, U.K. Lecture Notes in Computer Science vol. 5344 (2008) pp. 31–42.
128. B. Godlin, T. Kotek and J.A. Makowsky, Evaluations of Graph Polynomials, WG 2008, 34th International Workshop on Graph-Theoretic Concepts in Computer Science. 30 June - 2 July 2008, Durham University, U.K. Lecture Notes in Computer Science vol. 5344 (2008) pp. 183–194.
129. T. Kotek, J.A. Makowsky and B. Zilber On Counting Generalized Colorings, CSL 2008, 17th EACSL Annual Conference on Computer Science Logic, 15th-20th September 2008, Bertinoro, Italy Lecture Notes in Computer Science vol. 5213 (2008) pp. 339–353.
130. J. A. Makowsky, Uniform Algebraic Reducibilities between Parameterized Numeric Graph Invariants. Logic and Theory of Algorithms, Proceedings of CiE 2008, LNCS 5028 (2008), pp. 403–406.
131. J.A. Makowsky, Connection Matrices for MSOL-definable Structural Invariants, Proceedings of the Third Indian Conference on Logic and its Applications, January 7–11, 2009, Chennai, India ICLA 2009, LNCS (LNAI) 5378 (2009), pp. 51–64.

132. I. Averbouch, J.A. Makowsky and P. Tittmann, A Graph Polynomial Arising from Community Structure, Graph-Theoretic Concepts in Computer Science (WG'09), LNCS Volume 5911 (2010), pp. 33–43
133. I. Averbouch, T. Kotek, J.A. Makowsky and E.V. Ravve, The Universal Edge Elimination Polynomial and the Dichromatic Polynomial, Proceedings of EUROCOMB 2011, Electronic Notes in Discrete Mathematics 38 (2011): 77–82.
134. J.A. Makowsky, Application of Logic to Integer Sequences: A Survey. In: *Logic, Language, Information and Computation, 17th International Workshop, WoLLIC 2010, Brasilia, Brazil, July 6–9, 2010. Proceedings*, Anuj Dawar and Ruy J. G. B. de Queiroz eds., Lecture Notes in Computer Science, volume 6188, pages 34–41, Springer, 2010
135. J. A. Makowsky: Model Theory in Computer Science: My Own Recurrent Themes. CSL 2011, Marc Bezem, ed., Computer Science Logic, 25th International Workshop / 20th Annual Conference of the EACSL, CSL 2011, September 12–15, 2011, Bergen, Norway, Proceedings, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, LIPIcs, vol. 12 (2011) pp. 553–567.
136. T. Kotek and A. Makowsky. Connection Matrices and the Definability of Graph Parameters. CSL 2012, P. Cégielski and A. Durand eds., Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL, CSL 2012, September 3–6, 2012, Fontainebleau, France, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, LIPIcs, vol. 16 (2012) pp. 411–425.
137. J. A. Makowsky. Definability and Complexity of Graph Parameters (Invited Talk). CSL 2012, P. Cégielski and A. Durand eds., Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL, CSL 2012, September 3–6, 2012, Fontainebleau, France, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, LIPIcs, vol. 16 (2012) pp. 414–415.
138. T. Kotek, J. A. Makowsky, E. V. Ravve, A Computational Framework for the Study of Partition Functions and Graph Polynomials. IEEE Proceedings of SYNASC'2012, (2013), pp. 365–368
139. N. Labai and J.A. Makowsky, Weighted Automata and Monadic Second Order Logic. Proceedings of GandALF 2013, EPTCS 119 (2013), pp. 122–135
140. J.A. Makowsky and E.V. Ravve, On the Location of Roots of Graph Polynomials. Electronic Notes in Discrete Mathematics 43 (2013) pp. 201–206
141. N. Labai and J.A. Makowsky, Tropical Graph Parameters, Paper presented at FPSAC 2014, DMTCS Proceedings (2014), pp. 357–368
142. N. Labai and J.A. Makowsky, Hankel Matrices: From Words to Graphs. Proceedings of LATA 2015, LNCS 8977 (March 2015), pp. 47–55.
143. J.A. Makowsky, Teaching Logic for Computer Science: Are we teaching the wrong narrative? Porceedings of the 4th International Conference on Tools for Teaching Logic TTL, Rennes, France 2015, LIPICS, pp. 101–110. See also https://arxiv.org/abs/1507.03672v1.

144. T. Kotek, J. A. Makowsky, Efficient computation of generalized Ising polynomials on graphs with fixed clique-width, Proceedings of *Topics in Theoretical Computer Science (TTCS), Teheran, Augsut 2015*, M.T. Hajiaghayi and M.R. Mousavi (Eds.), LNCS 9541 (2016) pp. 135–146.
145. N. Labai and J.A. Makowsky, Hankel matrices for weighted visibly pushdown automata, accepted for LATA 2016, to appear in LNCS.
146. N. Labai and J. A. Makowsky. On the Exact Learnability of Graph Parameters: The Case of Partition Functions. 41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016). Vol. 58. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.
147. J.A. Makowsky and E. V. Ravve. Semantic Equivalence of Graph Polynomials Definable in Second Order Logic. Proceedings of the 23rd International Workshop on Logic, Language, Information, and Computation, LNCS Volume 9803. 2016.
148. J. A. Ellis-Monaghan, A. J. Goodall, J. A. Makowsky and I. Moffatt, Graph Polynomials: Towards a Comparative Theory (Dagstuhl Seminar 16241). Dagstuhl Reports, volume 6.6, (2016) pp. 26–48.
149. J.A. Makowsky The Undecidability of Orthogonal and Origami Geometries. Logic, Language, Information, and Computation: 25th International Workshop, WoLLIC 2018, Bogota, Colombia, July 24–27, 2018, Proceedings 25. Springer Berlin Heidelberg, 2018. pp. 250–270
150. J.A. Makowsky The undecidability of various affine Pappian geometries: wrong proofs and new true theorems. Logical Perspectives 2018 (2018): 21.
151. V. Rakita and J. A. Makowsky. On Weakly Distinguishing Graph Polynomials. Discrete Mathematics & Theoretical Computer Science 21 (2019). Proceedings of EUROCOM 2019.
152. E. Fischer and J. A. Makowsky. Extensions and Limits of the Specker-Blatter Theorem. 32nd EACSL Annual Conference on Computer Science Logic (CSL 2024). Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2024.

3.4 Papers in Collections

153. J.A.Makowsky, Compactness, Embeddings and Definability, Chapter 18 in "Model Theoretic Logics", J.Barwise and S.Feferman ed., Springer 1985, pp.645–716.
154. J.A.Makowsky and D. Mundici, Abstract Equivalence Relations, Chapter 19 in "Model Theoretic Logics", J.Barwise and S.Feferman ed., Springer 1985, pp.717–746.
155. J.A.Makowsky, Abstract Embedding Relations, Chapter 20 in "Model Theoretic Logics", J.Barwise and S.Feferman ed., Springer 1985, pp.747–791. Also: Technical Report #286, Computer Science Department, Technion. June 1983.

156. J.A.Makowsky, Model Theory and Computer Science: An Appetizer, Chapter I.6 in the "Handbook of Logic in Computer Science, vol. 1 (Background: Mathematical structures)", S. Abramsky, D.M. Gabbay, T.S.E. Maibaum eds., Oxford University Press, 1992, pp. 763–814.
157. u J.A. Makowsky and Y.B. Pnueli, Computable Quantifiers and Logics over Finite Structures, in "Quantifiers: Logics, Models and Computation, Volume I", M. Krynicki, M. Mostowski and L.W. Szczerba eds., Kluwer Academic Publishers, 1995, pp. 313–357.
158. J.A.Makowsky, Mental Images and the Architecture of Concepts, The Universal Turing Machine: A Half-Century Survey, R.Herken ed., Oxford University Press, London 1988, pp. 453–466. Second edition: Springer 1995.
159. E.Dahlhaus and J.A.Makowsky, Gandy Machines as a Model of Parallel Computation, in: The Universal Turing Machine: A Half-Century Survey, R.Herken ed., Oxford University Press, London 1988, pp. 309–314. Second edition: Springer 1995.
160. J.A. Makowsky and K. Meer, Polynomials of Bounded Tree Width, Refereed paper in: Foundations of Computational Mathematics, Proceedings of the Smalefest 2000, F. Cucker and M. Rojas (eds.), World Scientific, 211–250, 2002.
161. A. Cohen, M. Kaminski and J.A. Makowsky, Indistinguishibality be default, in: Sergei N. Artëmov, Howard Barringer, Artur S. d'Avila Garcez, Luís C. Lamb, John Woods (Eds.): We Will Show Them! Essays in Honour of Dov Gabbay, Volume One. College Publications 2005, pp. 415–428
162. J.A. Makowsky, Encounters with A. Mostowski, in: 70 Years of Foundational Studies, In memoriam of Andrzej Mostowski. W. Marek and M. Srebrny editors. IOS Press 2008. in press
163. E. Fischer and J.A. Makowsky Linear Recurrence Relations for Graph Polynomials, Special volume of LNCS in honour of Boris (Boaz) A. Trakhtenbrot on the occasion of his 85th birthday. LNCS 4800 (2008) pp. 266–279.
164. T. Kotek and J.A. Makowsky, Definability of Combinatorial Functions and Their Linear Recurrence Relations, In: *Fields of Logic and Computation, Essays Dedicated to Yuri Gurevich on the Occasion of His 70th Birthday*, Andreas Blass and Nachum Dershowitz and Wolfgang Reisig eds., Lecture Notes in Computer Science, vol. 6300, pages 444–462, Springer 2010.
165. J.A. Makowsky, From Parikh's Theorem to Many-Sorted Spectra, In: *Games, Norms and Reason. Logic at the Crossroads* J. van Benthem, A. Gupta and E. Pacuit eds. Synthese Library vol. 353 (2011), Springer, pp. 163–177.
166. J.A. Makowsky and E.V. Ravve, BCNF via attribute splitting, In: *Conceptual Modelling and Its Theoretical Foundations*, Essays Dedicated to Bernhard Thalheim on the Occasion of His 60th Birthday, A. Duesterhoeft, M. Klettke, K.-D. Schewe, eds. LNCS 7260 (2012), pages 73–84.
167. T. Kotek, J.A. Makowsky and E.V. Ravve, A computational framework for the study of partition functions and graph polynomials, In: Proceedings of the 12th Asian Logic Conference, R. Downey, J. Brendle, R. Goldblatt and B. Kim eds., World Scientific (2013), pp. 210–230.

168. E. Fischer, T. Kotek and J.A. Makowsky, Application of Logic to Combinatorial Sequences and Their Recurrence Relations, in: Contemporary Mathematics, vol 558, American Mathematical Society (2011). pp. 1.42.
169. T. Kotek, J.A. Makowsky and B. Zilber, On Counting Generalized Colorings, in: Contemporary Mathematics, vol 558, American Mathematical Society (2011). pp. 207–242.
170. J.A. Makowsky and N. Labai, Logics of Finite Hankel Rank. In: Fields of Logic and Computation II (pp. 237–252). Springer International Publishing, 2015.
171. J.A. Makowsky, To Yuri at 80 and More than 40 Years of Friendship. In: Fields of Logic and Computation III, (2020) Lecture Notes in Computer Science book series (LNPSE,volume 12180)
172. J.A. Makowsky, Scholar articles Some Thoughts on Computational Models: From Massive Human Computing to Abstract State Machines, and Beyond. In: Logic, Computation and Rigorous Methods: Essays in honour of E. Börger, pp 173–186 Lecture Notes in Computer Science book series (LNPSE,volume 12750) (2021)
173. T. Kotek and J. A. Makowsky. The exact complexity of the Tutte polynomial. In: Handbook of the Tutte Polynomial and Related Topics. (edt: J. Ellis-Monaghan and I. Moffatt), Chapman and Hall/CRC, (2022). pp. 175–193.
174. L. Beklemishev, A. Dmitrieva, and J. A. Makowsky. Axiomatizing Origami planes. In: Dick de Jongh on Intuitionistic and Provability Logics. Springer International Publishing, (2024) pp. 353–377.
175. J. A. Makowsky. Meta-theorems for Graph Polynomials. In: 2023 MATRIX Annals. Editors: David R. Wood, Alison Etheridge, Nalini Joshi, Jan de Gier. MATRIX Book Series, Vol. 6, Springer, in print.

3.5 Other Publications

This list contains a selection of various professionally relevant publications. It does neither contain the items of the computing column of Finanz & Wirtschaft nor my literary essays.

3.6 Reviews

O1 J.A.Makowsky, Review of *J.Ullman, Principles of Data Base Systems; D.Maier, The Theory of Relational Databases; A.Chandra and D.Harel, computable queries for Relational Data Bases.* Journal of Symbolic Logic, vol. 51.4 (1987) pp. 1079–1084.

O2 J.A. Makowsky, Review of *J. Paradeans et. al., The Structure of the Relational Database Model*, Journal of Symbolic Logic, Vol. 57.2 (1992) pp. 759–760.

O3 J.A. Makowsky, Review of *E.G. Manes, Predicate Transformer Semantics,* SIGACT News, Winter 1993.

O4 J.A. Makowsky, Review of *H. Mannila and K.-J. Räihä, The design of relational databases, S. Abiteboul, R. Hull and V. Vianu, Foundations of databases, P. Kanellakis, Elements of relational database theory*, Journal of Symbolic Logic, vol. 62.1 (1997) pp. 324–326.

O5 J.A. Makowsky, Zen oder die Kunst eine Turing Maschine zu warten, Review of *Oswald Wiener, Manuel Bonik und Robert Hoedicke, Eine Elementare Einfuerung in die Theorie der Turing–Maschinen*, Puplished in **Der Standard**, Vienna, 31. Juli 1998

3.7 Mathematics and Computing Related

O6 J.A.Makowsky, Das Problem als Triebfeder der Mathematik, Hochschulblätter für den Fachbereich Mathematik-FU, No. 52 (Mai 78), Berlin, Seiten 10–17.

O7 J.A.Makowsky, Logic Programming und die fünfte Computergeneration, Neue Zürcher Zeitung, Forschung und Technik Nr.250, 26.Oct.1983.

O8 J.A.Makowsky, Die elektronische Brieftasche, Computer World Schweiz, No.3 (2.December 1985) pp. 10–11. (This is an article about a patent in cryptography submitted by S. Even, O. Goldreich and Y. Yacobi)

O9 J.A.Makowsky, Computeranimation - High-Tech-Trickfilm oder Aufbruch in neue Visualisierungsmöglichkeiten, in: Simulation und Wirklichkeit, A.Schönberger ed., DuMont, Köln, 1988, pp. 149–155.

O10 J.A.Makowsky, Mental Images and the Architecture of Concepts, in: The Universal Turing Machine: A Half-Century Survey, R.Herken ed., Oxford University Press, London 1988, pp. 453–466. Second edition: Springer 1995.

O11 J.A. Makowsky, Contributions to "Finanz und Wirtschaft" 1988–1989. https://janos.cs.technion.ac.il/PUB/PUBLISHED/makowsky/FuW/FuW.pdf

O12 J.A. Makowsky, Computer Risks and Insurability of Software, 1990. reedited 2009 https://janos.cs.technion.ac.il/PUB/PUBLISHED/makowsky/ZURICH.pdf

O13 J.A. Makowsky, Modernisms, Fiction and Mathematics. Notices of the AMS, Vol. 67, Num. 10, November 2020, pp. 1589–1595. (extended version: arXiv preprint arXiv:1907.05787 (2019)).

4 List of Other Writings

4.1 Music

E1 J.A.Makowsky, Musik aus Raum und Zahl, (Bemerkungen zur Musik Iannis Xenakis), Zürcher Student, No.46.2 (Mai 1968), Seite 21.

E2 J.A.Makowsky, Musik ist nolens volens politisch (Interview mit H.W.Henze), Zürcher Student, No.47.3 (Juni 1969), Seiten 13–15. Abridged version reprinted in: Hans Werner Henze, Schriften und Gespräche, Deutscher Taschenbuchverlag, 1976.

E3 J.Makowsky, Autoritaet und ein klares Konzept, Interview mit Gary Bertini (Dirigent), Musik und Theater, Nr. 10 (1981), Seiten 12–15.

E4 J.A.Makowsky, Münchhausen's Theorem und seine Bedeutung fur die Musik und Musikwissenschaft, in "Die Zeichen, Neue Aspekte der musikalischen Aesthetik II", H.W.Henze ed. Frankfurt a.M. 1981, pp 278–307

E5 J.Makowsky, Arbeitsstile, Besprechung von Hans Werner Henze's "Die englische Katze", Musik und Theater, Nr10 (1983), Seiten 18–19.

4.2 Society and Politics After 1980

E6 J.A.Makowsky, 1984: Brave new work, in "Kursbuch 75: Computer Kultur", J.A.Makowsky, K.M.Michel et al. ed., Kursbuch Verlag Berlin 1984 (March), pp.119–143.

E7 J.A.Makowsky, Angst vor der Elite, in "Kursbuch 80: Begabung und Erziehung", K.M.Michel and T.Spengler ed., Kursbuch Verlag Berlin 1985 (May), pp.151–156. (Reprinted in Bulletin, Mitteilungen des Schweizerischen Bühnenverbandes, 8 (1985).)

E8 J.A.Makowsky, Blind rush for the new gold, The Jerusalem Post, Friday, April 19, 1985, page 9. (This is an article about software problems related to the Star War project of the Reagan administration)

E9 J.A.Makowsky, Postzionismus und religiöse Rückkehr im Israel von heute, Jüdische Rundschau (Maccabi), Nr.38 (19.Sept. 1985), pp.23–25.

E10 R.M. Lüscher, Henry und die Krümelmonster: Versuch über den fordistischen Sozialcharakter, edited by J.A.Makowsky, konKursbuch Verlag Claudia Gehrke, Tübingen,1988.

4.3 Gymnasium and University 1964–1969

E11 J.A.Makowsky (Janos Makowsky), Zu Dimitri Schostakovitsch und seiner 5. Sinfonie, Zürcher Mittelschulzeitung, No.15 (1963), ohne Seitenangabe.

E12 J.A.Makowsky (Janos Makowsky), Zur Auffüehrung von Girodoux' Elektra, Zürcher Mittelschulzeitung, No.20 (Dezember 1964), Seiten 14–16.

E13 J.A.Makowsky (jm), Höllisches Halblicht.... (Bemerkungen zu Samuel Beckett), Zürcher Mittelschulzeitung, No.21 (Maerz 1965) , Seiten 20–23.

E14 S.Beyeler, R.Gurny, J.A.Makowsky und W.Sauber (anonym), In der Sache Dr.J.Egli, Zürcher Mittelschulzeitung, No.23 (Dezember 1965) , Seiten 5–9.

E15 Janos Makowsky und Hanspeter Zürcher, Minimalismus als Folge eines Versagens der Lehrer und Erzieher, Zürcher Mittelschulzeitung, No.23 (Dezember 1965) , Seiten 24–25.
E16 SIE + ER Jugendforum, J.Ed.Morf (Chefredaktor), R.Bosshard (Soziologie), Ruth Gantner, Brigitte Stettbacher, Janos Makowsky, Willi Wottreng u.a., Das Abenteuer, erwachsen zu werden, SIE + ER 1966
E17 Autorenkollektiv der FSZ (Th.Held, R.M.Luescher, J.A.Makowsky, J.B. Neidhart, B. Niebuhr), FSZ Thesen zur Hochschulreform, Fortschrittliche Studentenschaft Zurich, Zurich 1969, 12p.
E18 J.A.Makowsky, Sprachstruktur und Unterricht, Zürcher Student, No.46.4 (Juli 1968), Seite 17.
E19 Johann Balthasar Neidhart und Johann-Andreas Makowsky, Intelligenz und Chancengleichheit, Volksrecht, No. 202, 29.August 1968, Beilage Kulturspiegel (auch im Winterthurer AZ).
E20 J.A.Makowsky (johann-andreas makowsky) und R.M.Luescher (jeanluc schwejger), Zürcher Manifest, Diskus (Aktionszeitung der Zürcher Jugend), No. 2, (12. September 1968), Seite 2.
E21 Joh.Andreas Makowsky, Scharf beobachtete Züge, Diskus (Aktionszeitung der Zürcher Jugend), No. 13, (12. Dezember 1968), Seite 1.
E22 J.A.Makowsky (JAM), Der andere Film, Zürcher Student, No.47.1 (Mai 1969), Seite 17.
E23 J.A.Makowsky (JAM), Links gehen, Gefahr sehen (Mitbestimmung und Transparenz, Zürcher Student, No.47.1 (Mai 1969), Seite 1–2.
E24 J.A.Makowsky (JAM), Lieber Leser, Züercher Student, No.47.2 (Mai 1969), Seite 1.
E25 J.A.Makowsky (jam), Alain Krivine, ein Student als Praesidentschaftskanditat, (Uebersetzung der Fernsehansprache Krivines) Zürcher Student, No.47.3 (Juni 1969), Seite 7.
E26 R.M.Lüscher und J.A.Makowsky, Archipel “Freie Welt”, Focus No.53 (1974), Seiten 8–12.

4.4 Literary Miniatures

L1 J. Makowsky. Leckerbissen. Mein heimliches Auge III, 1988, pp. 182–184
L2 J. Makowsky. Bambi’s Geheimnis. Mein heimliches Auge XXVII, 2012, pp. 202–203.
L3 J. Makowsky. Nasse Betten. Mein heimliches Auge XXXIII, 2018, pp. 76–79.
L4 J. Makowsky. Ums Haar sind wir noch davon gekommen. KonKursbuch 36, 1999, pp. 181–189.
L5 J. Makowsky. Ich packe meine Bibliothek aus. KonKursbuch 55, 2018, pp. 289–302. (Third expanded edition, 2019, pp. 315–318).

L6 J. Makowsky. Die Kunst in Würde und Gelassenheit zu sterben. KonKursbuch 56, 2020, pp. 329–335. (Second expanded edition, 2021. pp. 333–339.)
L7 J. Makowsky. Mut zur Musse. KonKursbuch 58, 2023, pp. 392–394.

Acknowledgments I would like to thank the editors of this volume who encouraged me to write and contribute this biographically teinted list of my publications.[58] I feel very honored by the contributions in this volume. Thanks a lot to all the contributors and editors of this volume.[59]

[58] Special thanks Maria N. Yelenevskaya for support and copy editing it.

[59] Special thanks to Klaus Meer for his efforts and patience in collecting all the contributions.

Some Personal Remarks About Johann A. Makowsky

Ilia Averbouch

Dedicated to my scientific supervisor, my mentor, my friend.

About half a year ago I was visiting my parents in Israel. It was a short and busy visit, and I decided not to go anywhere else, just stay with them. And yet, there were just a few people whom I was eager to see. János was among them, and I hoped he would be home as I wanted so much to see him. I believe there are people having the ability to change other people's lives. And I am convinced that when I met János, my entire life got the direction that I wanted it to have.

Actually, I first met him back in 1997 when as an undergraduate I attended his course of logic for computer science. I did not know then that I would continue my studies for higher degrees and that he would be my thesis supervisor.

I did enjoy the course though. It felt different from what I was used to at the the Technion. It was an astonishing discovery for me: the entire course was presented as one big theorem, from the beginning to the end. Maybe this is the reason that this course had such a powerful impression on me, so each time I need to teach somebody, I am hopelessly trying to make it Makowsky style . . .

Six years later, when I started to think about the Master's degree, one of my friends told me, "Go to János - he is fighting for his students like a lion". Then, I just followed this advice. Now, when I have learned a lot about the academic world and met many other graduate students and supervisors, I cannot overestimate the importance of this statement.

First and foremost, during all my studies János always made me feel unique, talented and valued. When a journal did not accept my paper, I really enjoyed the elegance of his communication with the editors, explaining to them why criticism was not justified, making them accept his arguments and even apologizing. After this the paper was successfully published in another journal.

I. Averbouch (✉)
Qualcomm Canada ULC, Markham, ON, Canada
e-mail: iaverbou@qti.qualcomm.com

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_2

When János had a slight impression that I was undervalued by my employer, he went directly to the head of IBM Haifa to express his disappointment.

When I finished my doctorate, I continued to feel my supervisor's interest in what was going on in my life and career; equally, does he follow events in the lives of his other students. And I must admit that like many years ago it makes me feel unique, talented and valued.

Last but not least, I was always overwhelmed by the amount and diversity of János' knowledge, be it maths or arts, sports or theater, economics or politics, or life in general. I am proud to tell my kids that I personally know a mathematician, a musician, an inspiring startup founder, a professional ski instructor, an art expert, a theater director, and many others–and all those are combined in the same person - my Teacher. I regard this as my personal achievement.

But the main thing for me is that János' house is always open for me, and I know that he is always behind us, his students, with all his spiritual power and wisdom.

I wish you, János, many healthy years, joy, and new achievements in the numerous areas of your interests!

Sincerely yours,

Ilia Averbouch

The Swiss Connection

Erwin Engeler

Johann (János) Makowsky was one of the last logic students at the ETH Zurich, who had the privilege of interacting with Paul Bernays, the mathematician who was the main collaborator in Hilbert's Program in the Foundations of Mathematics at Göttingen. After 1933 Bernays was fortunately able to move to Switzerland, and here he continued to be the untiring and conscientious correspondent of virtually all active logicians, including Gödel and Turing. In that he was comparable to Alonzo Church, who had assumed a similar role in Princeton (soon after he visited Hilbert's School in 1929). Bernays did not start a veritable School in Zurich, only the Monday Logic Seminar, which became a center of attraction to the surviving logic in Europe. Thus, János came to the right place, which by that time, in the 1960's, was importantly joined by Ernst Specker and Hans Läuchli who became his doctoral advisors. By 1972 the seminar enlarged to include theoretical computer science with Volker Strassen and the present author, (a former student of Bernays). János flourished in this environment.

I had met János before I came to Zurich; he was introduced to me by Beno Eckmann when I visited his Mathematical Research Institute at the ETH. He knew that young man as a member of the Zurich Jewish Community and as a student in his courses. He cautioned me against being misled by his political activism and his well-developed mathematical self-confidence. I found him sympathetic on first sight. By the time I was established in Zurich, János had finished his Dr. Sc.Math. He had also acquired a thorough political education through his active participation in the youth-upheavals of the 1968's.

The students at the ETH had their own target in the struggle: The Swiss parliament had quietly renewed the law governing this Technical University to include a new sister institution at Lausanne, without perceiving the signs of the time.

E. Engeler (✉)
Department of Mathematics, ETH Zurich, Zurich, Switzerland
e-mail: engeler@math.ethz.ch

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_3

This called forth so far unheard-of demonstrations, strikes and building occupations. János was one of the activists and participated in the very Swiss style of resolution of this political problem. He was one of the initiators suggesting that the students should collect citizens' signatures in order to force a referendum against the law. The Swiss people voted the law down by a satisfying and influential result of 65.5%. János told me later that that experience was formative for his future style when embarking on one of his activisms

Yes, all this was more than 50 years ago, and this account should now move to our relations and scientific exchanges of the ensuing decades. Much of this is more competently told by other contributors, the friends János made connection with throughout his academic career, his numerous academic visits around the globe. Zurich remained his base; his mother lived here, and he remained close to the circle of my students on many happy gatherings, in particular at my birthday symposia, marking my 60th, 70th, 80th and 90th.

From a Friend and Publisher

Claudia Gehrke

Janos Makowsky has accompanied my publishing house and me from afar for over 40 years.

It all began with his close friend, Rudolf M. Lüscher. Rudolf M. Lüscher (1948–1983), a Swiss author and philosopher who incorporated everyday life into his academic texts in a surprising way fitting in well with our recently founded periodical ***KonKursbuch***, which transcended genre boundaries. He had probably read its first issue and suggested a possible contribution for the second one shortly after the first issue was published. Back then, letters and contributions were still on paper, so these now historic documents can be found in folders. Unfortunately, I never printed out the subsequent e-mail exchanges with the authors and they gradually disappeared when hard disks broke or other computer mishaps occurred. We read Lüscher's contribution with enthusiasm, despite many annotations, because these were pointed extensions or digressions from the text. His readers were gently drawn into an extremely complex way of looking at reality. Walter Benjamin, Ludwig Wittgenstein, and Karl Kraus had inspired Lüscher as I learned from a biographical note that Janos Makowsky would later write for Lüscher's book.

Rudolf M. Lüscher contributed to the second volume of Konkursbuch titled "Faces of Violence" in 1978. His contribution "Sabotage and Surrealism" already contained hints of the theme of his book, which we published 10 years later, and which dealt with the modern social character ("Krümelmonster") in connection with work, or the organization of work in factories ("Fordism"). Rudi Lüscher also came to one of our first publishing house parties. Back then, in 1978 these events took place every 6 months, now about every 5 years, and Janos often attended them later. I still have a photo of breakfast with the guests in a garden in Tbingen. Rudi seemed to be on the sidelines and yet in the center of the group. I remember that he sat rather

C. Gehrke (✉)
KonKursbuchverlag, Tübingen, Germany
e-mail: gehrke@konkursbuch.com

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_4

quietly in the background, but when he said something, it brought the conversation forward in leaps and bounds. He died unexpectedly in 1983. Then a "Circle of Friends of Rudolf Lüscher" approached me to see if we would like to publish his bequeathed works. Among the friends, perhaps the driving force of the group was Janos. I got to know him personally during this period. I saw someone who went to extraordinary lengths for his friend to be read and appreciated; I saw someone whose friendship could be relied upon. He doesn't disappear from the lives of his friends after a short guest appearance. He stays, supports and encourages when he can.

Five years later, in 1988, after lengthy editorial work, Lüscher's posthumous book "Henri und die Krümelmonster" was published. In the biographical note, Janos also wrote about Lüscher's private library, which the circle of friends had bought from his mother in order to donate it to the Schweizerisches Sozialarchiv.[1] Lüscher and his friends shared a passion for books. Book collections tell the history of people, and not only intellectual history.

In the subsequent years Janos contributed on various topics to the Konkursbuch and also to the erotic yearbook "Mein heimliches Auge". In 2018 he published a wonderful autobiographical text about his own library (see Konkursbuch 55: über Bücher"', 2018). Through books, he recounted contemporary history and much of his own biography. There were only a few books from his mother's and grandparents' household, as they emigrated from Hungary to Switzerland in 1949, just a year after Janos' birth, and were unable to take many books with them. Other books came from the family of his mother's new husband, who was a former communist, for example letters by Rosa Luxemburg. As a ten-year-old, Janos enthusiastically read Stefan Zweig's "Schachnovelle" from the home library; as a high school student, he saved money from jobs such as delivering newspapers or working in construction during the fall vacations to buy books, such as seven volumes of Pierre Moland's twelve-volume Molière Complete Edition in Paris, and for his high school graduation, his biological father gave him a gift of six volumes of N. Bourbaki's Mathematics on his request. In 1990, the books by Masha, Maria N. Yelenevskaya (his life partner since this time) were added, "brought from the Soviet Union with her stories of preservation and hiding". In Konkursbuch 56: "Death" (2020), Janos' other biographical text followed, reflecting on stories of deaths: the dignity of his mother's death and death as defeat of his stepfather, the passing away of his grandmother Rose (Rozsi) and of grandma's friend Aunt Luise. But there is a lot of life in these stories too. Left alone with his mother, the eleven-year old was trying to restore the household that had vanished by learning to cook talking on the phone to aunt Luise and consulting grandma's cookbooks. There is also an intense text by Masha in the Konkursbuch edition on the subject of "death". And finally, the "Courage for leisure" in Konkursbuch 58: "Work" (2023). Biography, contemporary history and philosophy are wonderfully intertwined in the texts Janos wrote for us.

[1] The Sozialarchiv was founded in 1906 as a library, archive and documentation center of social movements, social change and social questions.

We have been seeing each other at irregular intervals for decades, sometimes it didn't work out for years, sometimes together with Masha he showed up in our garden in Tübingen-Hagelloch at short notice. He would also come to our publishing house parties, most recently to the 40th birthday event in 2018 and to my own seventieth in 2023. This beautiful connection, which began with Lüscher, has accompanied me my whole life, that's how I feel.

In 1989, my mother, my sister and I visited Janos and his two sons Yuval and Amichai in Tel Aviv with my niece Sarah, who was 5 years old at the time. He was already living in Haifa at the time. Janos kindly let us use his apartment in Tel Aviv as the starting point for our journey through Israel and beyond, to the Dead Sea and the Red Sea, a journey by car that took us through areas that have been unfortunately impossible to travel through for a long time. I can still see (and there are photos) Sarah and the two slightly older brothers romping around the apartment or having fun in the water on the beach in Tel Aviv. And I also remember this: Janos showed us the possibilities of computers—there was none in the publishing house at the time. Thanks to this "introduction to the digital world" I started thinking about the possibility of buying a PC for the publishing house. But it wasn't until around 1992 or 1993 that the time had come.

Janos studied mathematics, physics and logic, as well as philosophy, sociology and psychology. Until his retirement, he had been Professor of Computer Science at the Technion in Haifa. In addition to German studies and philosophy, I had also studied mathematics and taught as a hospital teacher in the early days of the publishing house. I was able to experience the wonderful "click experiences", i.e., the moment of understanding something anew together with the pupils, because I had to explain everything to myself again and again. Due to the intensity of the publishing work, which took up almost all my thinking, I gradually forgot what I had learned in my math studies. So my passion for mathematics was, if at all, only present as a "thought structure" and no longer connected to concrete knowledge. I think we rarely talked about mathematics with Janos; perhaps once about parallels that meet at infinity.

With that in mind, I wish myself many more encounters in infinity, whether short or long. I am sure that my friendship with this wonderful person will last:- Thank you for everything.

From Graph Polynomials to the Software Industry: Lessons from Janos

Tomer Kotek

Between 2007 and 2012 I was Janos' PhD student, and we continued to collaborate after that. Janos taught me how to do scientific research and how to do mathematics, how to ask the right questions and how to answer them. He exposed me to a rich mathematical theory investigating the intricate world of graph polynomials, which has been my most consistent line of research throughout my academic career. Later, I left academic research and embarked on a career in the computing industry, but I have taken the lessons I learned from Janos with me.

1 Logical Thinking

Janos' courses taught me how to reason about mathematics in a rigorous way. Until then I had surprisingly managed to be fairly successful in my mathematics courses without really understanding the concept of a mathematical proof. Writing a proof had been, for me, an amorphous process of writing my realizations about the problem as they popped into my head. In Janos' courses I learned how to write a proof as a sequence of inferences in which each conclusion follows logically from its premise. Over time, my way of thinking has become more similar to a mathematical proof - it is more structured, logical, and coherent. I have found this to be a significant advantage in industry, where it has helped me to explain and promote my ideas, convince stakeholders, direct discussions, influence priorities, and so on. It has also helped me to analyze, understand, and evaluate other people's suggestions and plans, and not be swayed by unconvincing arguments of charismatic speakers. It has helped me to construct high-level plans, make them detailed, and execute them.

T. Kotek (✉)
Berlin, Germany
e-mail: tomerkotek@gmail.com

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_5

2 Taking Feedback

During our joint work, we received many reviews of our papers. Obviously, the easiest reviews to handle were the positive ones. Sometimes we received negative reviews. It can also happen that reviews are written hastily, or that the reviewer didn't seem to understand our paper. Janos taught me to always appreciate the reviewer's effort, and never to dismiss a reviewer's comment, however unjustified or wrong it may seem. Ultimately, a reader may share our thoughts, so even in case the reviewer misunderstood the paper it only means that our explanation should be improved. I try to apply this approach to any criticism or feedback that I get now. I try to open my mind to the criticism, and even if I disagree, I try to find what I can do to improve the situation, if only to communicate my actions and intentions better.

3 Weaving a Narrative

In my opinion, one of the hallmarks of Janos' research is that it very naturally connects seemingly unrelated topics and transforms them into one coherent research program. Our work on graph polynomials derived objects of study from chemistry, physics, knot theory, graph theory, combinatorics, number theory, algebra, and computability. We studied them from different angles, in terms of their computational complexity, their algorithmic properties, their combinatorial properties, their algebraic properties, their descriptive properties, and their logical descriptions. We introduced our ideas to researchers in a wide variety of scientific communities, and published in their journals. I have learned from Janos how to take seemingly unconnected ideas, and weave them together into a wide, encompassing tapestry with one theme. I often find that working in industry can be a bit like that. One has to create and execute an overarching plan balancing writing code, developing IT infrastructure, and conducting research. It also requires doing project management, meeting customer demands, managing teams, adhering to the company's culture, overcoming financial constraints, observing market trends, and more. These numerous and multifaceted details, and these various groups of people with different expertise and often disparate perspectives must all fit together into a story that the organization tells, so that everyone can understand the goals of the organization, and their own roles in advancing them.

4 Giving Credit

I have learned from Janos to give credit generously. He taught me that ideas can take time to bake before they are ready to emerge from the oven, and by then we may

forget conversations we had, seminar talks we listened to, and papers we read. I have seen how Janos goes out of his way to give credit to the sources of his inspiration. Now I try to go out of my way to give credit for other people's ideas, suggestions, and work. It is particularly important for me to give credit to the people who I manage, so that they know I notice and appreciate their work, and so that other stakeholders in our organization are also aware of it.

There are many other lessons that I learned from Janos: trust and promote your people, collaborate to form relationships, keep promises, stay positive, and have the courage to take a stand. These lessons guided me when I was in academia, but they guide me even more today.

Acknowledgments I would like to thank Hadas Kotek for comments and editing, and to Maria Yelenevskaya for adding the finishing touches.

Emancipatory Aspects of Learning and Teaching Mathematics

Johann A. Makowsky

Abstract Mathematics can also be viewed as meditative activity. Like the martial arts, it provides us both with a technique to face the world and to face ourselves.

1 Preamble

The following is a text I wrote for my aspiring graduate students some 30 years ago. It also served as the bases of a talk I wanted to give at one of the conferences of the Palestinian Mathematical Society in 1997 or 1980. The Society was founded by Professor Marwan Awartani in 1992. In the late 1990ies Israeli mathematicians were invited to participate and my talk was accepted. However, by orders of Yassir Arafat personally, Israeli participation was thwarted. Following the Second Intifada in 2000, the Society's activity was halted due to the security situation. Since then the dialogue between Israeli and Palestinian Scientists became more and more difficult. The text I wrote, however, did not loose its actuality, on the contrary.

2 A Manifesto

Mathematics, like the martial arts, provides us both with a technique to face the world and to face ourselves. I will explore a view of mathematics which has its testified origins in Greek philosophy, Roman engineering and Arabic science. It is the basis of the Renaissance views of man, of the ethos of the autonomous individual and the underlying assumption of secularism. Here are some of its major points which should play a vital role when we ask ourselves why, among other reasons, we teach mathematics.

J. A. Makowsky (✉)
Faculty of Computer Science, Israel Institute of Technology, Haifa, Israel
e-mail: jmakowsky@bluewin.ch

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_6

- Mathematics deals with a *controlled virtual reality*, a landscape of mental images. *Virtual*, because we create it unless we are die–hard platonists, and *controlled*, because once we look at it, it does not change.
- The mathematical challenge is to explore this landscape, to map it out and to conquer it for exploitation (applications). We are allowed and encouraged to use landmarks of our predecessors, but we pride ourselves on being able to remap and reconquer this landscape using our own capacities and ingenuity if needed.
- The personal challenge of a mathematician is to control his/her hopes, feelings, weaknesses and fears in the face of the mathematical terra incognita. We want to know what it looks like there, and from there further on with the only goal to see.
- Each mathematical victory consists also in a victory over one's weaknesses. We reach a point on our own and the sole reward is to be there and to be able to generously and humbly share our view with our fellow travellers.
- Competitiveness among mathematicians tries to turn the other's weakness into our own strength. It makes us mean, petty and greedy. We should always remember to serve Mathematics humbly rather than use Her for our own purposes.
- *Even in teaching mathematics we can at least attempt to teach the students the flavour of freedom and critical thought, and to get them used to the idea of being treated as humans empowered with the ability of understanding*.

From: Roger Godement, Cours d'Algèbre, Hermann, Paris 1966.

Epsilon Calculus Provides Shorter Cut-Free Proofs

Matthias Baaz and Anela Lolić

Abstract In this paper we show that cut-free derivations in the epsilon format of sequent calculus provide for a non-elementary speed-up w.r.t. cut-free proofs in usual sequent calculi in first-order language. In addition, a non-elementary speed-up is shown w.r.t. cut-free proofs in calculi with relaxed eigenvariable conditions which proved a speed-up themselves w.r.t. **LK**.

1 Introduction

Epsilon calculus gives the impression to provide shorter proofs than other proof mechanisms. To make this claim precise, we compare in this paper an epsilon calculus variant of **LK** with **LK** and related calculi. The main property of epsilon calculus used is its ability to overbind bound variables. Furthermore, we show that cut-elimination for this epsilon calculus variant of **LK** along the lines of Gentzen or Schütte and Tait (i.e. induction after the complexity of the cut formulas) is not possible.

2 Epsilon Calculus

The ε-calculus uses ε-terms to represent $\exists x A(x)$ by $A(\varepsilon_x A(x))$. Consequently, $\forall x A(x)$ is represented by $A(\varepsilon_x \neg A(x))$. As the ε-calculus is only based on the

M. Baaz
Institute of Discrete Mathematics and Geometry, TU Wien, Vienna, Austria
e-mail: baaz@logic.at

A. Lolić (✉)
Kurt Gödel Society, Institute of Logic and Computation, TU Wien, Vienna, Austria
e-mail: anela@logic.at

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_7

representation by critical formulas

$$A(t) \to A(\varepsilon_x A(x))$$

for $A(t) \to \exists x A(x)$ and propositional axioms and rules, the unrestricted deduction theorem of propositional calculus transfers to this formalization of first-order logic: The ε-proof itself is a tautology

$$(\bigwedge_{i=1}^{n} A_i(t_i) \to A_i(\varepsilon_x A_i(x))) \to E,$$

where E is the original result translated into ε-calculus. Note that strong quantifier inferences are replaced by substitutions of $\varepsilon_x \neg A(x)$ for $\forall x A(x)$ positive and $\varepsilon_x A(x)$ for $\exists x A(x)$ negative. (Valid propositional formulas do not influence an ε-proof.) The extended first ε-theorem [8, 9] eliminates algorithmically the critical formulas obtaining a Herbrand disjunction $\bigvee_{i=1}^{m} E(\overline{t_i})$, where E is the ε-translation of $\exists \overline{x} E'(\overline{x})$, E' being quantifier-free. The argument can be easily extended to formulas E' which contain only weak quantifiers.

The language of epsilon calculus is based on the term language of epsilon expressions and other function symbols and on propositional language otherwise.

3 Lε, LK, and Related Sequent Calculi

To compare cut-free derivations we consider a sequent calculus format of the epsilon calculus.

Definition 1 (Lε) (In the language of epsilon calculus)
Axiom schema: $A \vdash A$, A atomic.
The inference rules are:

- for conjunction

$$\frac{A, B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} \wedge_l \qquad \frac{\Gamma_1 \vdash \Delta_1, A \qquad \Gamma_2 \vdash \Delta_2, B}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, A \wedge B} \wedge_r$$

- for disjunction

$$\frac{A, \Gamma_1 \vdash \Delta_1 \qquad B, \Gamma_2 \vdash \Delta_2}{A \vee B, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \vee_l \qquad \frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B} \vee_r$$

- for implication

$$\frac{\Gamma_1 \vdash \Delta_1, A \qquad B, \Gamma_2 \vdash \Delta_2}{A \to B, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \to_l \qquad \frac{A, \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \to B} \to_r$$

- for negation

$$\frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta} \neg_l \qquad \frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A} \neg_r$$

- weakening

$$\frac{\Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}\, w_l \qquad\qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A}\, w_r$$

- contraction

$$\frac{A, A, \Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}\, c_l \qquad\qquad \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A}\, c_r$$

- cut

$$\frac{\Gamma_1 \vdash \Delta_1, A \qquad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, cut$$

- quantifier inferences:
  the weak quantifier inferences $\exists_r$

$$\frac{\Pi \vdash \Delta, A(t)}{\Pi \vdash \Delta, A(\varepsilon_x A(x))}\, \exists_r$$

  and $\forall_l$

$$\frac{A(t), \Pi \vdash \Delta}{A(\varepsilon_x \neg A(x)), \Pi \vdash \Delta}\, \forall_l$$

  the strong quantifier inferences are redundant by substitution.

Example 1 Note that the $\forall_l$-rule can be defined symmetrically using an epsilon term $\tau_x A(x)$ to represent $\forall x A(x)$ by $A(\tau_x A(x))$. In the formulation above, $\forall_l$ introduces epsilon terms containing negation.

$$\frac{\begin{array}{c}\vdots \\ A(t) \vee B(t) \vdash A(t) \vee B(t)\end{array}}{A(\varepsilon_x \neg(A(x) \vee B(x))) \vee B(\varepsilon_x \neg(A(x) \vee B(x))) \vdash A(t) \vee B(t)}$$

We have to define first a translation of an expression in first-order language to an expression in epsilon calculus language.

Definition 2 Let A be a formula. Its epsilon translation is denoted as $[A]^\varepsilon$ and inductively defined as

- A is an atom. Then $[A]^\varepsilon = A$.
- $A = B \circ C$, where $\circ \in \{\wedge, \vee, \rightarrow\}$ and B and C formulas. Then $[A]^\varepsilon = [B]^\varepsilon \circ [C]^\varepsilon$.
- $A = \exists x A'(x)$. Then $[A]^\varepsilon = [A'(\varepsilon_x A'(x))]^\varepsilon$.
- $A = \forall x A'(x)$. Then $[A]^\varepsilon = [A'(\varepsilon_x \neg A'(x))]^\varepsilon$.

$[A]^{\forall\exists}$ is a translation from epsilon calculus language to first-order language when $A = [B]^\varepsilon$ for some expression B, and undefined otherwise.

Example 2 Note that $[A]^{\forall\exists}$ for an epsilon calculus expression A does not always exist: Consider for example the axiom $[\forall x.x = x]^\varepsilon = \varepsilon_x \neg(x = x) = \varepsilon_x \neg(x = x)$. With help of the critical formula

$$\varepsilon_x \neg(x = x) = \varepsilon_x \neg(x = x) \rightarrow \varepsilon_v(v = \varepsilon_x \neg(x = x)) = \varepsilon_x \neg x = x$$

we obtain

$$\varepsilon_v(v = \varepsilon_x \neg x = x) = \varepsilon_x \neg x = x$$

which has no meaning in first-order logic.

Definition 3 An epsilon term $\varepsilon_x A(x)$ overbinds if it is generated by the following inferences

$$\frac{\Pi \vdash \Delta, A(s(\varepsilon_x B), t(\varepsilon_x B))}{\Pi \vdash \Delta, A(\varepsilon_x A(x, t(\varepsilon_x B))}$$

$$\frac{A(s(\varepsilon_x B), t(\varepsilon_x B)), \Pi \vdash \Delta}{A(\varepsilon_x \neg A(x, t(\varepsilon_x B)), \Pi \vdash \Delta}$$

Proposition 1 *Every* **LK***-derivation possibly with cuts can be translated into an* **L**ε*-derivation of equal or smaller size. The translation proves the* ε*-translation of the end-sequent, it is structure-preserving, and cut-free proofs turn into cut-free proofs.*

Proof All inference steps are replaced by corresponding inference steps with exception of strong quantifier rules, which are replaced by substitution.

Remark 1 Note that the usual form of epsilon proofs can be obtained by deleting the quantifier inferences of **L**ε, and replacing them by

$$\frac{\begin{array}{c}(\psi') \\ \Pi' \vdash \Delta', A'(t) \qquad A'(\varepsilon_x A(x)) \vdash A'(\varepsilon_x A(x))\end{array}}{A'(t) \to A'(\varepsilon_x A(x)), \Pi' \vdash \Delta'} \exists_r$$

and

$$\frac{\dfrac{\begin{array}{c}(\psi') \\ A'(t), \Pi' \vdash \Delta' \qquad A'(\varepsilon_x \neg A(x)) \vdash A'(\varepsilon_x \neg A(x))\end{array}}{A'(\varepsilon_x \neg A(x)) \to A'(t), \Pi' \vdash \Delta'}}{\neg A'(t) \to \neg A'(\varepsilon_x \neg A(x)), \Pi' \vdash \Delta'} \forall_l$$

Recall that a function on the natural numbers is elementary if it can be defined by a quantifier-free formula from $+$, $\times$, and the function $x \to 2^x$. By independent results of Statman [11] and of Orevkov [10], the sizes of the smallest cut-free **LK**-proofs of sequents of size n are not bounded by any elementary function on n. (Note that in Statman's case equality can be axiomized. For a version of Statman's result in **LK** see [3].)

Definition 4 (Size) The size of a formula is the number of symbols occurring in it. The size of a sequent is the sum of the sizes of the formula occurrences in it. The size of a derivation is the sum of the sizes of the sequents occurring in it.

Example 3 A shortest cut-free **LK**-derivation of $\exists y(A(y) \to \forall x A(x))$ is

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A(a) \vdash A(a)}{A(a) \vdash A(a), \forall x A(x)}\, w_r}{\vdash A(a), A(a) \to \forall x A(x)}\, \to_r}{\vdash A(a), \exists y(A(y) \to \forall x A(x))}\, \exists_r}{\vdash \forall x A(x), \exists y(A(y) \to \forall x A(x))}\, \forall_r}{A(b) \vdash \forall x A(x), \exists y(A(y) \to \forall x A(x))}\, w_l}{\vdash A(b) \to \forall x A(x), \exists y(A(y) \to \forall x A(x))}\, \to_r}{\vdash \exists y(A(y) \to \forall x A(x)), \exists y(A(y) \to \forall x A(x))}\, \exists_r}{\vdash \exists y(A(y) \to \forall x A(x))}\, c_r$$

Its translation to **L**ε is

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A(e) \vdash A(e)}{A(e) \vdash A(e), A(e)}\, w_r}{\vdash A(e), A(e) \to A(e)}\, \to_r}{\vdash A(e), A(f) \to A(e)}\, \exists_r}{A(b) \vdash A(e), A(f) \to A(e)}\, (*) + w_l}{\vdash A(b) \to A(e), A(f) \to A(e)}\, \to_r}{\vdash A(f) \to A(e), A(f) \to A(e)}\, \exists_r}{\vdash A(f) \to A(e)\ (= \exists y(A(y) \to \forall x A(x))^{\varepsilon})}\, c_r$$

where $e \equiv \varepsilon_x \neg A(x)$ and $f \equiv \varepsilon_y(A(y) \to A(\varepsilon_x \neg A(x)))$.

$(*)$: $\forall_r$ has been replaced by the substitution of $\varepsilon_x \neg A(x)$ for a.

The shortest cut-free derivation of $\vdash A(f) \to A(e)$ in **L**ε is however

$$\dfrac{\dfrac{A(e) \vdash A(e)}{\vdash A(e) \to A(e)}\, \to_r}{\vdash A(f) \to A(e)}\, \exists_r$$

Theorem 1 ([10, 11]) *There is a specific family of sequents $\{S_i\}_{i<\omega}$ described in [3] and due to Statman [11], and specific **LK**-proofs thereof, such that they have the following properties:*

1. *the size of S_i is polynomial in i;*
2. *there is no bound on the size of their smallest cut-free **LK**-proofs that is elementary in i;*
3. *the size of these proofs (with cuts), however, is polynomially bounded in i.*

In the following we will consider the sequence of sequents $\{S_i\}_{i<\omega}$ from Theorem 1 above.

Corollary 1 *Each worst-case sequence as formulated in Theorem 1 generates a worst-case sequence, where the end-sequents contain weak quantifiers only.*

Proof Strong quantifiers in a cut-free **LK** proof can be replaced by Skolem functions without lengthening the proof or introducing cuts [5]. The same holds for a **LK**-proof with cuts, as long as the cuts are not Skolemized.

Definition 5 The matrix A^M of a first-order formula A is A, after deletion of all quantifiers and after replacement of bound variables by free variables.

Example 4 $[\exists x(\forall y A(x, y) \vee B(x))]^M = A(a, b) \vee B(a)$.

Lemma 1 *There is a specific family of sequents* $\{S_i\}_{i<\omega}$ *such that they have the following properties:*

1. *the size of* S_i *is polynomial in* i*;*
2. *there is no bound on the size of their smallest cut-free* ***LK****-proofs that is elementary in* i*;*
3. *the size of these proofs (with cuts), however, is polynomially bounded in* i*;*
4. *they contain only weak quantifiers;*
5. *on the left-side of the conclusion for every cut* A, $\forall \overline{x}(A^M \to A^M)$ *is added.*

Proof For the proofs with cut the addition of $\forall \overline{x}(A^M \to A^M)$ might lead to even shorter proofs, for the cut-free proofs the proofs may be double exponentially shorter if the newly added universal formulas are eliminated in the following way: In the moment where the corresponding implication left is inferred, replace this inference by a cut. In consequence, there is a proof with propositional cuts only, which can be eliminated in at most double exponential expense [12].

Theorem 2 *There is a sequence of cut-free* $\mathbf{L}\varepsilon$*-proofs such that*

1. *the size of* S_i *is polynomial in* i*;*
2. *the end-sequents* S_i *are translations of first-order sequents* S_i' *with weak quantifiers only;*
3. *the size of these proofs, however, is polynomially bounded in* i*;*
4. *there is no bound on the size of the smallest cut-free* ***LK****-proofs of the translation of* S_i *to first-order language that is elementary in* i*.*

Proof We choose a sequence of **LK**-proofs from the lemma above. We translate the proofs with cut into epsilon calculus (this does not lengthen the proof according to Proposition 1). In the $\mathbf{L}\varepsilon$-proof we replace all cuts on A by inferences of $A \to A$ on the left side. Derive immediately $[\forall \overline{x}(A^M \to A^M)]^{\epsilon}$. Contract it with $\forall \overline{x}(A^M \to A^M)$ which is already in the end-sequent.

Note that the extended first epsilon theorem [8] provides an upper bound for cut-free $\mathbf{L}\varepsilon$-derivations of translations of first-order sequents in the rough size of $2^{2^{\cdot^{\cdot^{\cdot}}}\}i}$ for the i-th cut-free $\mathbf{L}\varepsilon$-derivation.

4 $\mathbf{LK}^+$ and $\mathbf{LK}^{++}$

Another example of the speed-up of cut-free proofs as in Sect. 3 relates to the sequent calculi $\mathbf{LK}^+$ and $\mathbf{LK}^{++}$ introduced in [1]. They are obtained from **LK** by weakening the eigenvariable conditions. The resulting calculi are therefore globally

but possibly not locally sound. This means that all derived statements are true but that not every sub-derivation is meaningful.

Note that there is already a non-elementary speed-up of cut-free proofs of $\mathbf{LK}^{+}$, or $\mathbf{LK}^{++}$ w.r.t. cut-free **LK**-proofs [1]. In contrast, the transformation of cut-free $\mathbf{LK}^{++}$-proofs into cut-free $\mathbf{LK}^{+}$-proofs is elementary bounded [6].

Definition 6 (Side Variable Relation $<_{\varphi,\mathbf{LK}}$, cf. [1]) Let φ be an **LK**-derivation. We say b is a side variable of a in φ (written $a <_{\varphi,\mathbf{LK}} b$) if φ contains a strong quantifier inference of the form

$$\frac{\Gamma \vdash \Delta, A(a,b,\overline{c})}{\Gamma \vdash \Delta, \forall x A(x,b,\overline{c})}\,\forall_r$$

or of the form

$$\frac{A(a,b,\overline{c}), \Gamma \vdash \Delta}{\exists x A(x,b,\overline{c}), \Gamma \vdash \Delta}\,\exists_l$$

We may omit the subscript φ, **LK** in $<_{\varphi,\mathbf{LK}}$ if it is clear from the context.

In addition to strong and weak quantifier inferences we define $\mathbf{LK}^{+}$-suitable quantifier inferences.

Definition 7 ($\mathbf{LK}^{+}$-suitable Quantifier Inferences, cf. [1]) We say a quantifier inference is suitable for a proof φ if either it is a weak quantifier inference, or the following three conditions are satisfied:

- (substitutability) the eigenvariable does not appear in the conclusion of φ.
- (side variable condition) the relation $<_{\varphi,\mathbf{LK}}$ is acyclic.
- (weak regularity) the eigenvariable of an inference is not the eigenvariable of another strong quantifier inference in φ.

Definition 8 ($\mathbf{LK}^{+}$, cf. [1]) $\mathbf{LK}^{+}$ is obtained from **LK** by replacing the usual eigenvariable conditions by $\mathbf{LK}^{+}$-suitable ones.

Similarly to $\mathbf{LK}^{+}$, we define the calculus $\mathbf{LK}^{++}$ by further weakening the eigenvariable conditions

Definition 9 ($\mathbf{LK}^{++}$-suitable Quantifier Inferences, cf. [1]) We say a quantifier inference is suitable for a proof φ if either it is a weak quantifier inference, or it satisfies

- substitutability,
- the side variable condition, and
- (very weak regularity) the eigenvariable of an inference with main formula A is different to the eigenvariable of an inference with main formula A' whenever $A \neq A'$.

Definition 10 ($\mathbf{LK}^{++}$, cf. [1]) $\mathbf{LK}^{++}$ is obtained from **LK** by replacing the usual eigenvariable conditions by $\mathbf{LK}^{++}$-suitable ones.

Theorem 3

1. *If a sequent is* $\mathbf{LK}^{+}$*-derivable, then it is already* $\mathbf{LK}$*-derivable.*
2. *If a sequent is* $\mathbf{LK}^{++}$*-derivable, then it is already* $\mathbf{LK}$*-derivable.*

Proof ***(Proof Sketch)*** Consider an $\mathbf{LK}^{++}$-proof φ (an $\mathbf{LK}^{+}$ proof is also an $\mathbf{LK}^{++}$-proof). Replace every universal quantifier inference unsound w.r.t. $\mathbf{LK}$ by an $\rightarrow_l$ inference:

$$\frac{\Gamma \vdash \Delta, A(a) \qquad \forall x A(x) \vdash \forall x A(x)}{\Gamma, A(a) \rightarrow \forall x A(x) \vdash \Delta, \forall x A(x)} \rightarrow_l$$

Similarly, replace every existential quantifier inference unsound w.r.t. $\mathbf{LK}$ by an $\rightarrow_l$ inference:

$$\frac{\exists x A(x) \vdash \exists x A(x) \qquad A(a), \Gamma \vdash \Delta}{\Gamma, \exists x A(x), \exists x A(x) \rightarrow A(a) \vdash \Delta} \rightarrow_l$$

By doing this, we obtain a proof of the desired sequent, together with formulas of the form

$$A(a) \rightarrow \forall x A(x) \quad \text{or} \quad \exists x A(x) \rightarrow A(a)$$

on the left-hand side. Note that the resulting derivation does not contain any inference based on eigenvariable conditions. We can eliminate each of the additional formulas on the left-hand side by adding an existential quantifier inference and cutting with sequents of the form

$$\vdash \exists y (A(y) \rightarrow \forall x A(x))$$

or of the form

$$\vdash \exists y (\exists x A(x) \rightarrow A(y)),$$

both of which are easily derivable. For more details see [1].

Example 5 Consider the following locally unsound but globally sound $\mathbf{LK}^{+}$-derivation φ:

$$\dfrac{\dfrac{\dfrac{A(a) \vdash A(a)}{A(a) \vdash \forall y A(y)} \forall_r}{\vdash A(a) \rightarrow \forall y A(y)} \rightarrow_r}{\vdash \exists x (A(x) \rightarrow \forall y A(y))} \exists_r$$

As a is the only eigenvariable the side variable relation $<_{\varphi,\mathbf{LK}}$ is empty.

The focus in [1] has been on the strongly reduced complexity of cut-free $\mathbf{LK}^{+}$- and $\mathbf{LK}^{++}$-proofs (Theorem 2.6 and Corollary 2.7).

Note that all three conditions of Definitions 7 and 9 are necessary.

Example 6 If substitutability is violated, the following derivation is possible

$$\dfrac{A(a) \vdash A(a)}{A(a) \vdash \forall x A(x)}\,\forall_r$$

If the side variable relation is not acyclic, the following derivation φ is possible (with the side variable conditions $a <_{\varphi,\mathbf{LK}} b$ and $b <_{\varphi,\mathbf{LK}} a$, which loop)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A(a,b) \vdash A(a,b)}{A(a,b) \vdash \forall y A(a,y)}\,\forall_r}{A(a,b) \vdash \exists x \forall y A(x,y)}\,\exists_r}{\exists x A(x,b) \vdash \exists x \forall y A(x,y)}\,\exists_l}{\forall y \exists x A(x,y) \vdash \exists x \forall y A(x,y)}\,\forall_l$$

If weak regularity is violated, the following derivation is possible

$$\dfrac{\dfrac{\dfrac{A(a) \vdash A(a)}{A(a) \vdash \forall x A(x)}\,\forall_r}{\exists y A(y) \vdash \forall x A(x)}\,\exists_l$$

Lemma 2 *There is a specific family of sequents $\{S_i\}_{i<\omega}$ with the following properties:*

1. *the size of S_i is polynomial in i;*
2. *there is no bound on the size of their smallest cut-free $\mathbf{LK}^+$-proofs (or $\mathbf{LK}^{++}$-proofs) that is elementary in i;*
3. *the size of these proofs (with cuts), however, is polynomially bounded in i;*
4. *the end-sequents have only weak quantifiers;*
5. *on the left-side of the conclusion for every cut A, $\forall \overline{x}(A^M \to A^M)$ is added.*

Proof Note that Skolemization is not possible by direct substitution into strong quantifiers. However, Skolemization can be performed by adding additional cuts, which lengthen the proof linearly:

$$\dfrac{\dfrac{\Pi \vdash \Gamma, A(a,t)}{\Pi \vdash \Gamma, \forall x A(x,t)} \qquad \dfrac{A(f(t),t) \vdash A(f(t),t)}{\forall x A(x,t) \vdash A(f(t),t)}}{\Pi \vdash \Gamma, A(f(t),t)}$$

and

$$\dfrac{\dfrac{A(f(t),t) \vdash A(f(t),t)}{A(f(t),t) \vdash \exists x A(x,t)} \qquad \dfrac{A(a,t), \Pi \vdash \Gamma}{\exists x A(x,t), \Pi \vdash \Gamma}}{A(f(t),t), \Pi \vdash \Gamma}$$

Theorem 4 *There is a sequence of cut-free $\mathbf{L}\varepsilon$-proofs such that*

1. *the size of S_i is polynomial in i;*
2. *the end-sequents S_i are translations of first-order sequents S_i' with weak quantifiers only;*
3. *the size of these proofs, however, is polynomially bounded in i;*

4. *there is no bound on the size of the smallest cut-free* $\mathbf{LK}^{+}$*- or* $\mathbf{LK}^{++}$*-proofs of the translation of* S_i *to first-order language that is elementary in* i.

Proof We choose a sequence of $\mathbf{LK}^{+}$- or $\mathbf{LK}^{++}$- proofs according to Lemma 2. We translate the proofs with cut into epsilon calculus (this does not lengthen the proof according to Proposition 1). In the $\mathbf{L}\varepsilon$-proof we replace all cuts on A by inferences of $A \to A$ on the left side. Derive immediately $[\forall \overline{x}(A^M \to A^M)]^{\epsilon}$. Contract it with the $\forall \overline{x}(A^M \to A^M)$ which is already in the end-sequent. Note that cut-free $\mathbf{LK}^{+}$- or $\mathbf{LK}^{++}$-proofs with end-sequents with weak quantifiers only are $\mathbf{LK}$-proofs.

5 Problems with Syntactic Cut-Elimination for Lε

The problems of syntactic cut-elimination for $\mathbf{L}\varepsilon$ are widely known [13]. The results of this paper show that syntactic cut-elimination in the sense of Gentzen or Schütte-Tait is not possible as the induction after the complexity of the cut-formula fails.

Note that $\mathbf{L}\varepsilon$ is cut-free complete for end-sequents containing translations of first-order formulas only. This can be seen as follows: With additional cuts and without increasing the size much, the end-sequent can be transformed to an end-sequent which contains the translation of Skolemized formulas only. The extended first epsilon theorem could be understood as providing Herbrand expansions.[1] These Herbrand expansions are sequents in the language of $\mathbf{LK}$ and have a cut-free proof in $\mathbf{LK}$. Reintroduce quantifiers and eliminate the Skolem functions using [7]. Finally, translate this cut-free derivation to $\mathbf{L}\varepsilon$.

The constructions of this paper may be even refined by providing for every $\mathbf{LK}$-proof an $\mathbf{L}\varepsilon$-proof of about the same size with one universal cut, no terms in the cut, and one universal quantifier only. Every $\mathbf{LK}$-derivation can be transformed into an $\mathbf{LK}$-derivation with prenex cuts without increasing the size much [4]. The prenex cuts can be transformed into prenex atomic cuts as in [2] by adding universal formulas defining the matrices by atomic expressions. (These universal formulas can be eliminated elementarily from cut-free proofs.) Then replace every atom $P(\bar{t})$ by $T(p(\bar{t}))$ for a new T. Now translate the $\mathbf{LK}$-derivation to $\mathbf{L}\varepsilon$. Replace every cut on A by an inference $A \to A$ left. The $A \to A$ have necessarily the form $T(p(\bar{t})) \to T(p(\bar{t}))$. Use $\forall_l$ to obtain $T(\varepsilon_x \neg(T(x) \to T(x)) \to T(\varepsilon_x \neg(T(x) \to T(x)))$. All added formula $A \to A$ on the left size have now the same form and can be contracted. In the last step derive the sequent $\vdash T(\varepsilon_x \neg(T(x) \to T(x)) \to T(\varepsilon_x \neg(T(x) \to T(x)))$ and cut (this is the only cut).

[1] A Herbrand expansion of a formula (sequent) with weak quantifiers only replaces universal quantifiers by conjunctions, and existential quantifiers by disjunctions such that the resulting formula (sequent) is valid.

6 Conclusion

The effect that arbitrary cuts in $\mathbf{L}\varepsilon$ can be transferred into universal cuts with linear increase of size demonstrates that no cut-elimination for $\mathbf{L}\varepsilon$ for translation of first-order **LK**-sequents is possible by induction on the size of cut-formulas. This implies that e.g. Gentzen-style cut-elimination and Schütte-Tait-style cut-elimination are not feasible. Here the fundamental different nature of the (extended) first epsilon theorem becomes obvious [8]. The consequence is that cut-free $\mathbf{L}\varepsilon$-derivations of translations of first-order **LK**-sequents cannot be transformed elementarily to **LK**-derivations with cuts with bounded size. The question remains, whether cut-free $\mathbf{L}\varepsilon$-derivations of translations of first-order **LK**-sequents can be elementarily translated into **LK**-derivations with unbounded cut size.

References

1. Aguilera, J.P., Baaz, M.: Unsound inferences make proofs shorter. J. Symb. Log. **84**(1), 102–122 (2019)
2. Baaz, M., Fermüller, C.G.: Elementary elimination of prenex cuts in disjunction-free intuitionistic logic. In: CSL. LIPIcs, vol. 41, pp. 94–109. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2015)
3. Baaz, M., Leitsch, A.: On skolemization and proof complexity. Fundam. Informaticae **20**(4), 353–379 (1994)
4. Baaz, M., Leitsch, A.: Cut normal forms and proof complexity. Ann. Pure Appl. Log. **97**(1–3), 127–177 (1999)
5. Baaz, M., Leitsch, A.: Methods of Cut-elimination, vol. 34. Springer Science & Business Media, Berlin (2011)
6. Baaz, M., Lolic, A.: Effective skolemization. In: WoLLIC. Lecture Notes in Computer Science, vol. 13923, pp. 69–82. Springer, Berlin (2023)
7. Baaz, M., Hetzl, S., Weller, D.: On the complexity of proof deskolemization. J. Symb. Log. **77**(2), 669–686 (2012)
8. Hilbert, D., Bernays, P.: Grundlagen der Mathematik, vol. 2. Springer Berlin, Heidelberg (1939)
9. Moser, G., Zach, R.: The epsilon calculus and Herbrand complexity. Studia Logica **82**(1), 133–155 (2006)
10. Orevkov, V.P.: Lower bounds for increasing complexity of derivations after cut elimination. J. Soviet Math. **20**, 2337–2350 (1982)
11. Statman, R.: Lower bounds on herbrand's theorem. Proc. Am. Math. Soc. **75**(1), 104–107 (1979)
12. Weller, D.: On the elimination of quantifier-free cuts. Theor. Comput. Sci. **412**(49), 6843–6854 (2011)
13. Zach, R.: Semantics and proof theory of the epsilon calculus. In: ICLA. Lecture Notes in Computer Science, vol. 10119, pp. 27–47. Springer, Berlin (2017)

Variations on a Theme of Makowsky

John T. Baldwin

Abstract We distinguish the axiomatic study of proofs *in geometry* from the study *about geometry* from general axioms for mathematics. We briefly report on an abuse of that distinction and its unfortunate effect on US high school education. We review a number of twentieth century approaches to synthetic geometry. In doing so, we disambiguate (in the Wikipedia sense) the terms: metric, orthogonal, isotropic and hyperbolic. With some of these systems we are able to axiomatize 'affine geometry' over the complex field. The argument is trivial from Wu (Mechanical Theorem Proving in Geometry. Texts and Monographs in Symbolic Computation. Springer-Verlag, New York, 1994) or Szmielew (From Affine to Euclidean Geometry: An Axiomatic Approach. D. Reidel, Dordrecht, 1978), but not remarked by either of them.). We examine the general question of the connections between axioms for Affine geometries and the stability classification of associated complete first order theories of fields. We conclude with reminiscences of a half-century friendship with Janós.

1 Introduction

Our topic is inspired by Makowsky's article 'Can one design a geometry engine?' [25]. It introduced me to several ways to first order axiomatize 'Euclidean geometry' that were unusual because of very different choices of the fundamental notions. In Sect. 2 we contrast first order axiomatization of geometry (proofs in geometry) from arguments in ZFC or 2nd order logic (axioms about geometry, such as Birkhoff's [10, 11]).

Szmielew [32] carefully describes the *linear Cartesian plane* over a field F and the geometry on F^2 whose lines are the solutions of linear equations over F.

J. T. Baldwin (✉)
Department of Mathematics, Statistics, University of Illinois at Chicago and Computer Science M/C 249, Chicago, IL, USA
e-mail: jbaldwin@gmail.com

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_8

Offending all algebraic geometers,[1] we often abbreviate to 'plane over F' since the synthetic theory of such planes and straight lines is the target of this investigation. In particular, we will speak of the real and complex planes in this sense.

In the spirit of the clarification of the distinction in [25] between mutual and bi-interpretability, we explain several other terminological confusions. In fact a principal motive for Sect. 3 is to sort out for myself the rich diversity of first order approaches to coordinatizable plane geometry.

This analysis illuminates a deeper classification. Two of the prototypic structures in model theory are the (geometries over) the real and complex planes that lead to the 'twin' notions of strong minimality and o-minimality. The distinctions among the geometries discussed in Sect. 3 reflect this dichotomy. But we note in Sect. 4 that the geometry over the p-adic numbers fall into quite a different location in the map of stable theories at the webpage forkinganddividing.com. This raises the question, 'What distinguishes the geometries?'. How are the different choices of fundamental notions and approaches to coordinatization reflected in the stability classification?

2 Proofs in or About

We compare the synthetic proof of Euclid et al. with the twentieth century study of geometry by distinguishing three species of the proof of a proposition P in a geometry. What is a geometric proof? Any proof requires assumptions, rules of inference, and definitions. The three species are

- **Approach 1** *proof in* a formal language for geometry;[2]
- **Approach 2** *proof about* i.e., in a metatheory (e.g. ZFC), with geometry a defined notion.

  Whether such a proof in the second sense is 'geometric' is a purity issue.
- **Approach 3** We don't dwell here on a standard model theoretic technique: Use (2) to get (1). Using the completeness theorem [21] [6, p 257] outline the *method of semantic proof*. If a proposition is stated in first order logic and show to be true by a proof *about geometry* then in every model of a specific first order theory T of geometry, then it is provable in T.

Approach 1 Both Euclid and Hilbert (1899) wrote in natural language and had no explicit rules of inference. A formal proof in geometry requires:

1. Choosing a vocabulary (after conceptual analysis) of the fundamental notions (basic concepts). Euclid uses point, line, circle, incidence, congruence of segments of segments and of angles. Hilbert adds betweenness and order but

[1] Von Staudt published a 3 volume study of complex projective geometry including higher dimensional curves in 1856/1860.

[2] We restrict to geometry only for uniformity; the analysis applies to any formalized topic.

omits circle. In Sect. 3, we discuss such twentieth century basic concepts as orthogonality, parallelism, and perpendicularity.
2. Choosing a logic (first order, $L_{\omega_1,\omega}$, second order)
3. Choosing the axioms that reflect the conceptual analysis.

Approach 2 Through the late nineteenth and twentieth century as geometry metastasized from Euclid to hyperbolic, to differential, algebraic, etc., etc. the most published proofs were informal proofs (nominally reducible to ZFC for the last century) *about*, say, algebraic geometry. But they were not formalized in any specifically 'geometric' system. At best the appropriate geometry was defined in the (informal) metatheory.

For example, the *global method: analytic/metric* method of assigning area to a figure is described in [12]. Fix a unit; say, a square; tile the plane with congruent squares. Then to measure a figure, continually refine the measure by cutting the squares in quarters and count only those increasingly smaller squares which are contained in the figure. As one ponders this method, one realizes that it assumes a *real-valued* (to guarantee convergence) metric. This assumption is not mentioned but considered (correctly for most readers of that book) as a universally known assumption. Such is mathematics and I have no quarrel with it. But there is one hybrid which has had a disastrous impact on United States high school mathematics: Birkhoff's 'axioms'.

Our inspiration, Makowsky's article 'Can one design a geometry engine?', makes no mention of Birkhoff. Let us see why. Birkhoff [10, 11] works in a vocabulary of points, lines, distance ($d(A, B)$), and angle. Distance is a function from pairs of points to the real field (a topic assumed to be fully understood by students who survived 1 year of algebra.) (An angle is measured by a similar function from triples of points.) Postulate 1 (Ruler postulate) asserts that the points of any line can be put into 1-1 correspondence ($A \mapsto x_A$) with the reals so $d(A, B) = |x_A - x_B|$. The protractor postulate posits a similar measure for angles. In many texts [16], an early proof shows 'equals distances subtracted from equal distances are equal'. The proof is to apply the ruler postulate twice along with their deep understanding of the axioms of the algebra of the real numbers. This is in the first week of geometry for 14–15 year old students.

Raimi [30] presents Birkhoff's motivation for the high school text [11] as a reaction to shoddy treatment of limits in U.S. high schools during the first half of the twentieth century. Unfortunately, the cure is as bad as the disease. And the School Mathematics Study Group[3] adapted this system for high school geometry.

Contrary to Birkhoff, this is not a fully formalized axiom system. The properties of the reals are introduced as convenient oracles. Thus, as a proof *about but not in* geometry, it is not in the purview of [25].

[3] These are the architects of the 'new math'. Much of their work especially in Algebra I is aimed at understanding but the SMSG postulates [13, 31] remind one that a camel is a horse designed by a committee.

2.1 Proofs in Geometry: Choosing Basic Notions

In this subsection we survey several axiomatic approaches to the study of geometry. These systems are similar in that the initial axioms are first order and if/when Archimedes or Dedekind appears, it is explicitly mentioned. The distinction is in the choice of basic notions for geometry. We restrict to affine geometry as the translation (bi-interpretation) between projective and affine geometry is standard. In Sect. 3.1, we make a much finer distinction among six candidates for the title 'metric geometry'. The comparison between Hilbert style systems and the various orthogonal systems discussed there is the main concern of the paper.

2.2 Ordered Geometries

These are well-known; we just list them.

1. Hilbert/Euclid [18–20]: congruence is fundamental; two kinds of objects: point and lines
2. Tarski [32, 33]: congruence is fundamental; one kind (sort) of object: a line is a set of collinear points (given by a ternary betweenness relation).
3. Various authors [8, 14, 22, 26, 35]: Transformations are central but in most cases developed in axiomatized Euclidean geometry.[4]
4. Szmielew and Wu [32, 36] add the order notion at the end of their development; see Remark 4.3 and Fact 1.

3 Homonyms in Geometry: Is Order Essential?

This section relates more directly to [25]. We discuss three words which apply with apparently quite distinct meanings in developments of geometry from different choices of fundamental notions. We will then consider the relations of these developments with real and complex algebraic geometry. The three subsections address the three homonyms: metric, isotropic, and hyperbolic.

[4] While these systems are ostensibly second order by quantifying over transformations as arbitrary functions satisfying certain conditions, one can adopt the standard first order trick of adding a sort for transformations θ and requiring that each such θ indexes a set of ordered pairs, the graph of a rigid motion.

3.1 Metric

What is a metric geometry? We describe here four very different notions (congruence, distance, orthogonality (3 versions), parallelity) of a **metric** geometry with many specific axiomatizations in various vocabularies.

Definition 1 A *generalized* (pseudo) metric is a function f from $X \times X$ into an ordered field F, that is symmetric, $f(x, x) = 0$, other values are positive, and satisfies the triangle inequality. Normally, $F = \Re$.

Remark 1 (Equipped with Congruence (Line Segment/angle:)) This terminology is certainly inaccurate and likely only used when segment congruence is confused with the existence of a real valued distance metric. A congruence equivalence may not to be attached to a unit 'distance'. This is one of the crucial distinctions between Euclid and Hilbert. Euclid would not conceive of such a confusion because he viewed geometric and arithmetic magnitudes as incomparable (not merely incommensurable). Hilbert proves the existence of such a metric by finding the field and *adding* a constant to fix the unit.

Tarski, over many decades and fully laid out in the posthumous [34], produced a clearly first order system with a first order scheme of continuity axioms. These axioms are in a vocabulary with a ternary collinearity predicate as opposed to Hilbert's two-sorted approach. As described in [33] any model of these axioms is coordinatized by a real closed field.

Remark 2 (Equipped with a *Distance* Metric) Moise [27, p 137] carefully distinguishes between what he calls synthetic and metric approaches. Roughly speaking, his synthetic corresponds to Hilbert and metric to Birkhoff. Hilbert begins with congruence and, effectively but not explicitly[5], introduces a 'distance' measured on a field that varies with the model of the theory and with a unit distance in a model M as the congruence class of the segment 01. We analyze Birkhoff's ruler and protractor postulates in connection with school pedagogy in [7].

1. in some ordered field [19] or, more specifically,
2. equipped with a *real-valued* distance metric [10].

These are vastly different; the first is first-order axiomatized. As discussed in Section 2, the second is basically axiomatized in set theory and is really more describing a geometry from a global standpoint than giving axioms for geometry.

Remark 3 (Orthogonal Geometry 1 and 2:) 'Throughout this paper metric will always refer to a structure with an orthogonality relation or in which one such relation[6] can be defined. *It is in no way related to metrics defined as distances with*

[5] [19] does not use the word distance in this sense or 'metric' at all.

[6] Line reflections are a basic concept in this system.

real values.' [29, p 419]. We describe four variants on 'metric', one of which is the more extended discussion of orthogonal geometries in the categories.

1. [29] describes two approaches: group theoretic and geometric.
   (a) [29, p 423] axiomatize a group of rigid motions of a plane with a unary predicates for line reflections, an operation (composition), and a constant for the identity.[7] The geometry is recovered by first order definitions [29, §2.1] and one can distinguish the elliptic, euclidean and hyperbolic case.
   (b) Alternatively, 'geometric' axioms [29, §2.2] use the vocabulary of incidence, line orthogonality, and reflections in lines.
2. Artin [3, p 51] calls the problem of defining a field from a two-sorted axiomatic geometry 'much more fascinating' than the familar Cartesian reduction of geometric problems to analytic geometry. Thus, unlike Birkhoff, he is explicitly working in set theory and perhaps (not his word) doing metamathematics. However, because of this clarity, lack of a linear order, and the use of first order axiomatizations of some geometries, and the impact of this book on axiomatic geometry, I consider Artin here rather than as 'about' in Sect. 2.

   He writes [3, p 106] 'The study of bilinear forms is equivalent to the study of metric structures on V'. An orthogonality relation can be described as an 'inner product' possessing properties such as those imposed on real geometry by the inner product. But the field into which the form maps is not required to be ordered. The connection with 'metric' in the sense of Remark 2 arises from the fact that the real inner product of vector with itself is the square of the length. I classify this example as orthogonal because the inner product of two vectors determines the angle between them and thus perpendicularity. But this approach is far more general than a real inner product space (Remark 2) since it makes sense without any continuity hypothesis, for projective spaces, and for any vector space.
3. After some definitions we discuss Wu's orthogonal geometry (Remark 4) in more detail.

Definition 2

1. An incidence plane is a collection of points and lines such that two points determine a line (so ℓ may be denoted AB if both are on ℓ) and there are three non-collinear points.
2. An incidence plane, equipped with a binary (parallelism) predicate $\parallel$ on lines, is Pappian[8] if for A_1, A_2, A_3 on line ℓ_1 and B_1, B_2, B_3 on line ℓ_2 (distinct points on distinct lines)

$$(A_1B_2 \parallel A_2B_1 \wedge A_2B_3 \parallel A_3B_2) \rightarrow A_1B_3 \parallel A_3B_1.$$

[7] Pambuccian [28] axiomatizes the 'same' geometry using only the relation symbol $\perp$ (with $\perp(abc)$ to be read as 'a, b, c are the vertices of a right triangle with right angle at a').

[8] Each of Szmielew and Wu discuss various refinements of the Pappian notion and relations with various forms of Desargues; they agree on the statement here as the decisive condition for obtaining a commutative field.

Definition 3 (Wu's Orthogonality Axioms) The orthogonality of two lines is denoted by $\ell_1 \perp \ell_2$ or $\mathrm{Or}(\ell_1, \ell_2)$. This is a basic concept for Wu. A line ℓ is *isotropic* if it is self-perpendicular.

(O-1): $\ell_1 \perp \ell_2 \leftrightarrow \ell_2 \perp \ell_1$;
(O-2): For a point O and a line ℓ_1 there exists exactly one line ℓ_2 with $\ell_1 \perp \ell_2$ and $I(0, \ell_2)$;
(O-3): $(\ell_1 \perp \ell_1 \wedge \ell_3 \perp \ell_3) \rightarrow \ell_2 \parallel \ell_3$.
(O-4): For every O there is an ℓ with $I(O, \ell)$ (incidence) and $\ell \not\perp \ell$.
(O-5): The three heights of a triangle intersect in one point.

Remark 4 (Orthogonality 3) Wu [36, §2.2] axiomatizes in a vocabulary with points, lines, and perpendicular as basic concepts. He has four groups of axioms ordered by containment; the last two are metric.

1. A Wu-orthogonal plane satisfies the usual (Hilbert) incidence axioms, five orthogonality axioms, asserts lines are infinite, unique parallels, and two forms of Desargues.[9] He concludes that a Wu-orthogonal plane satisfies Pappus and has a definable commutative coordinatizing field.
2. An *unordered Wu-metric* plane arises by adding the symmetric axis axiom [36, p 91]: Any two non-isotropic (See Sect. 3.2. 2.) lines have a symmetric axis.[10] With these hypotheses, Wu [36, p 92] *defines* a notion of congruence (called equidistance and added to the vocabulary in [25]) and proves the Pythagorean (Kou-Ku) theorem.
3. Adding Hilbert's order axioms gives an *ordered Wu-metric plane* [36, §2.5].
   This system defines an ordered coordinatizing field. Thus it is bi-interpretable with Hilbert's system ([18, 20]). Hilbert relies directly on what he calls Pascal's theorem, a variant of Desargues and Pappus; Hartshorne [18, §19] uses the cyclic quadrilateral theorem.[11]
4. Adding Hilbert's (non-first order) continuity axiom Wu reaches his 'ordinary geometry' [36, §2.6].

Definition 4 (Affine and Parallelity Planes) A collinearity structure is a ternary relation (collinearity) such that two points determine a line. Such a structure is an *affine plane* if for any line ℓ and point A there exist a unique parallel to ℓ through A. Planarity is enforced by saying that if one line is parallel to two distinct lines then the two intersect. *By adding a constant to an affine plane we can fix a unit of distance.*

[9] [36, Section 2.1] shows that the 'linear Pascalian axiom' (a) allows the proof that the coordinatizing Skew field is commutative and (b) follows from axioms for Wu-orthogonality. Thus, unlike [32], there is not a separate Pappian field stage in his development.

[10] Let ℓ be the perpendicular bisector of (the segment between) two points A, B. Then ℓ is called the symmetric axis of (A, B).

[11] Thus, Hartshorne [18, p 173] differs from Hilbert in using circles, but does not use the intersection of circles postulate E.

Since naming constants has no effect on interpretability, we will be careless about whether a point is named.

We describe two variants on this approach.

1. Szmielew is discussed Remark 5.
2. Alperin [2, p 121] Alperin formulates first order 'paper folding' axioms 'for the origami constructible points of the complex numbers' using some basic notion as point, line, incidence, perpendicular bisector, and reflection. He writes, 'Our main contribution here is to show that with all six axioms we get precisely the field obtained from intersections of conics, the field obtained from the rationals by adjoining arbitrary square roots and cube roots and conjugates'. He provides six axioms for construction (which can be done by paper folding) and *working within the complex numbers* shows that his first three axioms allow the construction from $0, 1, \alpha$, where α is not real, a subfield of $\mathbb{C}$. His fourth and fifth axioms extend the result to Pythagorean and Euclidean fields (Definition 5); with the sixth axiom, solutions to cubics can be constructed.

Remark 5 Szmielew [32, §2] uses parallel ($\|$) as the only basic symbol and axiomatizes a two sorted system of points and lines, *parallelity planes* which are bi-interpretable with affine planes.[12]

Szmielew follows the 'projective geometry approach' of introducing ternary fields and gradually adding geometric conditions that strengthen the algebraic properties. This crucially distinguishes her approach from that of Hilbert, Hartshorne, and Wu. On the other hand, Wu and Szmielew differ from Hilbert/Hartshorne in applying Desargues/Pappus to find the field before introducing either order or congruence.

Szmielew [32, §] considers parallelity planes [32, §8] and introduces the notions of midpoint planes and midpoint-ordered planes.

Fact 1

1. Szmielew [32, 4.5.3.iii) and 4.5.7)] show commutative fields are binterpretable with Pappian parallelity planes.
2. Szmielew [32, 8.3.iii) and 8.5.iii)] and show ordered commutative fields are binterpretable with ordered midpoint Pappian planes.

The particular affine geometry on $\mathbb{C}$ with 'lines' defined by linear equations is an affine plane and $(\mathbb{C}, +, \cdot, 0, 1)$ is definable in $(S, L, \|)$. Of course this structure is very different from the 'complex plane' in the sense of algebraic geometry. With the field, we can define algebraic curves in the plane.

It seems to me that Remarks 3.1, 4, and 5, are very close together; each extends the orthogonality geometry to order to regain 'ordinary geometry' (although Wu equates 'ordinary' with $\mathfrak{R}$-geometry and so requires Dedekind's axiom for that description).

[12] Szmielew [32, p85]; a predicate for parallel is needed for AE-axiomatizability.

Definition 5 A Pythagorean field is a field in which every sum of two squares is a square. A Euclidean field is an *ordered* field in which all non-negative elements are squares.

A Euclidean field (axiom E: circle-circle intersection) is Pythagorean by the Pythagorean theorem and the use of Axiom E to construct a hypoteneuse for any pair of given lengths.

The crucial distinction between Remarks 1, 2, 4 and Definition 4 is that the systems in the latter pair, while called 'metric', **do *not* require a notion of length or ordering of segments**. They coordinatize with *unordered* fields. Item 4.iii defines congruence but the field remains unordered. Alperin's field do not admit a linear order.

Note that Pythagorean fields need not be ordered; [2, p 121] studies some as subfields of $\mathbb{C}$. However, the minimal Pythagorean field Ω is orderable and is the minimal field satisfying Hilbert's betweenness and congruence axioms [18, 16.3.1].

A key feature of (axiomatic) orthogonal geometries is that the existence of a field is either assumed (Artin) or arises directly from assumed Pappian configurations rather than Desarguesian/Pappus being derived from the parallel postulate using segment congruence as in Hilbert.

3.2 Isotropic

1. Artin says a subspace of an orthogonal space in the sense of Remark 3.2 is isotropic if it is annihilated by the form f. In particular two lines ℓ_1, ℓ_2 are orthogonal (i.e. perpendicular) if $f(x, y) = 0$ for $x \in \ell_1$, $y \in \ell_2$.
2. Wu says a line is isotropic if it is self-perpendicular, $f(x, y) = 0$ for distinct x, y on ℓ. An example of an isotropic line through the origin in the complex plane is $x_2 = -ix_1$. Use the bilinear form $f(x_1, x_2) = x_1^2 + x_2^2$. The distance between any points on the line is 0. (See [36, p 141].)
3. Schwartz (https://www.math.brown.edu/reschwar/INF/handout10.pdf) says a geometry is isotropic if for any point and any angle can find a symmetry (distance preserving bijection) which fixes that point and rotates by that angle around the point.

The first two notions are closely related; the third distinct.

3.3 Hyperbolic Space

1. The standard notion in non-euclidean geometry:
2. Artin [3, Def III.3.8] A non-singular plane which contains an isotropic vector is called hyperbolic.

It seems pretty clear that these notions of hyperbolic and isotropic are really distinct. The question is whether, as in my comment in item Remark 3. 2, there is some etymological explanation for the overlap in terminology.

4 Classifying Geometries Model Theoretically

By Fact 1 we know the (linear cartesian) plane π over any commutative field (constructed as in e.g. [18, §14] satisfies the parallelity axioms. So π is bi-interpretable with its coordinatizing field. The bi-interpretability, indeed interdefinability, is particulary easy to see for the orthogonality case.

Remark 6 (Bi-interpretability) Given the plane. Fix two orthogonal lines and interpret the field on one line ℓ_1 using Pappus. By fixing a family of lines of the same slope define a bijection f (and field isomorphism) between the lines. Formally define over that field the plane on $\ell_1 \times \ell_2$. Now it is definably isomorphic to the original plane) by mapping $\langle a_1, a_2 \rangle$ to the intersection in the plane of the line parallel to ℓ_1 meeting ℓ_2 in a_2 and the line parallel to ℓ_2 meeting ℓ_1 in a_1.

So if the coordinatizing field has a recursively axiomatizable complete first order theory, the first order theory of a particular plane is a complete decidable theory; for example, the real and complex planes.

Fact 2 [Bi-interpretations] The following classes of geometries and fields are quantifier-free bi-interpretable.[13]

1. Pappian geometries (Wu—unordered metric planes and Szmielew–Paffian affine planes) and fields;
2. Infinite Pappian geometries with linearly ordered lines (Hilbert planes, Wu-ordered metric geometries, ordered affine planes [32, §8]) and ordered fields;
3. Hyperbolic geometries with limiting parallels and ordered Euclidean fields.

The following is immediate from the existence of a suitable bi-interpretation as in Fact 2.

Theorem 3 *The complete theory of the complex affine plane is axiomatized by adding the axioms of ACF_0 to the incomplete theory of fields given by the bi-interpretation with either (1) theory of Pappian parallelity planes [32, Theorem 4.5.3 iii) p 82] or (2) the theory of Wu-orthogonal planes.*

The following remarkable theorem (Fact 4) of Ziegler is essential to understand undecidability of fields and geometries.

[13] The first two are proved with an argument emphasizing the quantifier eliminability are summarised in [25, Theorems 5–7] and the third in [18, §43].

Fact 4 ([9, 37]) If T is a finitely axiomatized subtheory of RCF or ACF_0 then T is undecidable.

Fact 5 (**Makowsky** [25, Thm 17 pg 26; Prop 6 pg 10]) The universal first order consequences of (a) any extension of the orthogonal geometries in Remark 4, Remark 4.2 or Remark 5, or (b) HP5 whose interpretation is consistent with either (1) ACF_0 or (2) RCF_0 is decidable.

The proof uses heavily the quantifier-free interpretations laid out in [25]. Recalling Ziegler, Fact 4 and noticing that the axioms of the various geometries described in Remark 1 are $\forall\exists$-axiomatizable[14] Thus, decidability of universal sentences is the most that can be hoped for in any general geometry; Fact 5 is optimal.

We have described a family of different axiomatizations in different vocabularies that have some claim to 'axiomatizing geometry'. Many are bi-interpretable. Such theories are often regarded as 'the same'. But 'same' is far from true here. The orthogonal geometries are not ordered; Hilbert's are. Tarski's first order completion is the first order theory of the reals—real closed fields while the orthogonal geometries are exemplified by the Complex affine plane. Note these interpretations are 2-dimensional. Is 1-dimensional any better?

What do we know about the fields? A Hilbert field is ordered using betweenness ([32, 7.1.9]). But orthogonality geometries don't have betweenness. Alperin's origami give subfields of the complexes.

What are axioms for linear Cartesian planes over p-adic fields? Fix p and consider the affine plane over Q_p (or perhaps a countable elementary submodel?). We include in the vocabulary of Q_p a predicate for the valuation since the topological information is central to the notion. Let T' be the theory of Q_p. By [17], T' can be formalized in a one-sorted language as a theory that is NIP but neither distal nor o-minimal but is dp-minimal, look at forkinganddividing.com. It is easy to see Q_p is not linearly ordered as for various p, there are negative integers that are perfect squares.[15]

But (linear cartesian) geometry over Q_p is bi-interpretable with the field (without the valuation) Q_p (since the geometry is Pappian). What (if anything) needs to be added to the geometric vocabulary to define the valuation? It is not clear that dp-minimality is preserved by a 2-dimensional interpretation. If it were, we would know from [17] that its complete first order theory has the same place in the stability geography. Which formalism is most useful for axiomatizing the geometry?

[14] As described (e.g. [1, 707]) the propositions of Euclid fall into (1) *theorems* which are universal quantification of an implication of two diagrams (conjunction of atomic and neg-atomic formulas) and (2) constructions: π_2 sentences: For any instance of a diagram there are witness to an extended diagram.

[15] By an intriguing application of elementary descriptive set theory, https://math.stackexchange.com/questions/49990/the-p-adic-numbers-as-an-ordered-group shows there is no linear order compatible with the addition is definable in the field Q_p (since it would then have the Baire property).

The referee asked, 'what about the model theory of less 'well-behaved structures than fields?'. Work of the 1970's shows that if there is a coordinatizing division ring that is superstable, it must be an algebraically closed field [15, 23]. In the 90's I constructed an $\aleph_1$-categorical projective plane at the lowest level of the Lenz-Barlotti hierarchy[16]—the ternary field operation cannot be split into addition and multiplication [4, 5]. The Lenz-Barlotti classification charactizes ternary rings in terms of 16 properties such properties as associativity, commutativity, etc. I don't know of any work connecting this hierarchy with the stability classification. Another project! In particular, find complete theories of the coordinatizing rings in the work Wu and Szmielew and their place in the stability classification.

5 Reminiscences

I met Janos in the summer of 1974 during the International Congress of Mathematics in Vancouver. A group of us traveled to Banff and Calgary. I recall two small episodes: his insisting on swimming in his underwear in Shuswap Lake and refusing a bottle of wine in a fancy restaurant in Calgary. Much more memorable was his spelling me in carrying my daughter in a back carrier up a mountain near Banff. (My wife thinks this happened not in Banff but closer to Vancouver. But an ancient CV shows I gave a talk in Calgary that summer.) Sometime in the late 70's, my wife Sharon, daughter Katie, and I joined Janos and Irit in a tour of Switzerland. The highlight was pre-school Katie directing us, 'Follow the D-car'. (Janos was working in Berlin so had a German license plate.) I returned the child-on-back favor in 1980, carrying Amichai during our excursion from the Patras Conference to Delphi. We have no joint papers yet; our closest 'collaboration' was extended discussions about his contribution [24] to the Model-theoretic logics book. A later adventure whose date escapes me was following up a swank dinner in Colmar (Strasbourg?), by smuggling (details may vary) a computer into West Germany. Maybe it was that the computer was smuggled out and then reimported to establish 'legality'. The fine dining stories continued with a visit to Perroquet in Chicago where Janos won an argument with the maître'd by insisting that any reasonable high class restaurant would recognize his cardigan as a 'jacket' or provide jackets to traveling guests. We have exchanged visits over the years. Perhaps our long and highly-valued friendship can continue with another visit to Chicago by Janos and Masha.

[16] https://www.math.uni-kiel.de/geometrie/klein/math/geometry/barlotti.html.

References

1. Avigad, J., Dean, E., Mumma, J.: A formal system for Euclid's elements. Rev. Symb. Logic **2**, 700–768 (2009)
2. Alperin, R.C.: A mathematical theory of origami constructions and numbers. N.Y. J. Math. **6**, 119–133 (2000)
3. Artin, E.: Geometric Algebra. Interscience, New York (1957)
4. Baldwin, J.T.: An almost strongly minimal non-Desarguesian projective plane. Trans. Am. Math. Soc. **342**, 695–711 (1994)
5. Baldwin, J.T.: Some projective planes of Lenz Barlotti class I. Proc. A.M.S. **123**, 251–256 (1995)
6. Baldwin, J.T.: Model Theory and the Philosophy of Mathematical Practice: Formalization Without Foundationalism. Cambridge University Press, Cambridge (2018)
7. Baldwin, J.T., Mueller, A.: Focusing a GeT course on axiomatic systems for geometry. In: The GeT Course: Resources and Objectives for the Geometry Courses for Teachers (2025). Accepted 30 page chapter imbedded in 50 page 'supplement' online: http://homepages.math.uic.edu/~jbaldwin/CTTIgeometry/ctti
8. Barker, W., Howe, R.: Continuous Symmetry: From Euclid to Klein. American Mathematical Society, Providence (2007)
9. Beeson, M.: Some undecidable field constructions. translation of [37] http://www.michaelbeeson.com/research/papers/Ziegler.pdf
10. Birkhoff, G.: A set of postulates for plane geometry. Ann. Math. **33**, 329–343 (1932)
11. Birkhoff, G.D., Beatley, R.: Basic Geometry, 3rd edn. Chelsea Publishing Co., New York (1959). 1st edition 1941: Reprint: American Mathematical Society, 2000. ISBN 978-0-8218-2101-5; online at https://kupdf.net/download/birkhoff-amp-beatley-basic-geometry_58b4448e6454a79179b1e939_pdf
12. Boltyanskii, V.: Hilbert's Third Problem. V.H. Winston and Sons, Washington (1978)
13. Cederberg, J.: A Course in Modern Geometries. Springer, Berlin (2001)
14. Clark, D.M.: Euclidean Geometry: A Guided Inquiry Approach. Mathematical Circles Library. Mathematical Sciences Research Institute, Berkeley (2012)
15. Cherlin, G.L., Shelah, S.: Superstable groups and rings. Ann. Pure Appl. Logic **18**, 227–270 (1980)
16. Cummins, B., et al.: Geometry. Glencoe Mathematics. Glencoe, Chicago (2005)
17. Dolich, A., Goodrick, J., Lippel, D.: Dp-minimality: basic facts and examples. Notre Dame J. Formal Logic **52**, 267–288 (2011)
18. Hartshorne, R.: Geometry: Euclid and Beyond. Springer-Verlag, Berlin (2000)
19. Hilbert, D.: Foundations of Geometry. Open Court Publishers, LaSalle (1962). Original German publication 1899: reprinted with additions in E.J. Townsend translation (with additions) 1902: Gutenberg e-book #17384 http://www.gutenberg.org/ebooks/17384
20. Hilbert, D.: Euclid's Elements. Dover, New York (1956). In 3 volumes, translated by T.L. Heath; first edition 1908. Online at http://aleph0.clarku.edu/~djoyce/java/elements/
21. Hilbert, D., Ackermann, W.: Grundzüge der Theoretischen Logik, 2nd edn. Springer, Berlin (1938). First edition, 1928
22. Libeskind, S.: Euclidean and Transformation Geometry: A Deductive Inquiry. Jones and Bartlett, Burlington (2008)
23. Macintyre, A.J.: On ω_1-categorical theories of fields. Fundamenata Mathematicae **71**, 168–175 (1971)
24. Makowsky, J.A.: Abstract embedding relations. In: Barwise, J., Feferman, S. (eds.) Model-Theoretic Logics, pp. 747–792. Springer-Verlag, Berlin (1985)
25. Makowsky, J.A.: Can one design a geometry engine? On the (un)decidability of affine Euclidean geometries. Ann. Math. Artif. Intell. **85**, 259–291 (2019). Online: https://doi.org/10.1007/s10472-018-9610-1

26. Martin, G.E.: Transformation Geometry, an Introduction to Symmetry. Springer-Verlag, Berlin (1982)
27. Moise, E.: Elementary Geometry from an Advanced Standpoint, 3rd edn. Addison-Wesley, Boston (1990)
28. Pambuccian, V.: Orthogonality as single primitive notion for metric planes. Contributions Algebra Geom. **48**, 399–409 (2017)
29. Pambuccian, V., Struve, H., Struve, R.: Metric geometries in an axiomatic perspective. In: Ji, L., et al. (ed.) From Riemann to Differential Geometry and Relativity, vol. 48, pp. 399–409. Springer International Publishing AG, Berlin (2007)
30. Raimi, R.: Ignorance and innocence in the teaching of mathematics (2005). https://homepages.math.uic.edu/~jbaldwin/pub/Raimi.pdf
31. SMSG: The SMSG Postulates for Euclidean Geometry (1995). Search for: Geometry/The SMSG Postulates for Euclidean Geometry, e.g. https://faculty.winthrop.edu/pullanof/MATH%20393/The%20SMSG%20Postulates.pdf
32. Szmielew, W.: From Affine to Euclidean Geometry: An Axiomatic Approach. D. Reidel, Dordrecht (1978). Edited by Moszyńska, M.
33. Tarski, A.: What is elementary geometry? In: Henkin, L., Suppes, P., Tarski, A. (eds.) Symposium on the Axiomatic method, pp. 16–29. North Holland Publishing Co., Amsterdam (1959)
34. Tarski, A., Givant, S.: Tarski's system of geometry. Bull. Symb. Logic **5**, 175–214 (1999)
35. Weinzweig, A.I.: Geometry Through Transformations. Bent Tree Press, Reno Nevada (2007).
36. Wu, W.-T.: Mechanical Theorem Proving in Geometry. Texts and Monographs in Symbolic Computation. Springer-Verlag, New York (1994). Chinese original 1984
37. Ziegler, M.: Einige unentscheidbare Körpertheorien. L'Enseignement Math. **28**, 269–280 (1982). Michael Beeson has an English translation

Automatic Structures and the Problem of Natural Well-orderings

Lev D. Beklemishev and Fedor N. Pakhomov

Abstract We explore the idea of using automatic and similar kind of presentations of structures to deal with the conceptual problem of natural proof-theoretic ordinal notations. We conclude that this approach still does not meet the goals.

1 Introduction

This paper is written for the Festschrift volume dedicated to the 75th anniversary of Johann Makowsky. While thinking on the topic that would be appropriate for this volume, we decided to go for one that would link the interests of the authors in proof theory and some topics that play a role in Johann's own work: classes of structures with decidable (MSO) theories and interpretations. We decided to record our attempts to explore one particular approach to the (in)famous problem in proof theory—the problem of canonicity of proof-theoretic ordinal notation systems. Even though this problem is well-known, few written accounts and discussions of it exist in the literature.

Arguably, historically the first encounter with this problem occurs in the famous work of Alan Turing "System of logics based on ordinals" [47] presenting the content of his PhD dissertation. Substantial discussions are found in the papers by Georg Kreisel, in particular [27], and especially Solomon Feferman [18] who describes it as one of the three problems that 'bug him'.

The work of Fedor Pakhomov was funded by the FWO grant G0F8421N.

L. D. Beklemishev (✉)
Steklov Mathematical Institute, Moscow, Russia
e-mail: bekl@mi-ras.ru

F. N. Pakhomov
Steklov Mathematical Institute, Moscow, Russia

Ghent University, Ghent, Belgium
e-mail: fedor.pakhomov@ugent.be

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_9

The problem requires delineating canonical, well-behaved ordinal notation systems from pathological ordinal notation systems, such as a computable well-ordering of order type ω such that the associated induction scheme implies the consistency of Peano arithmetic (Kreisel's example). All examples of such pathological well-orderings have some external notions (such as consistency statements) encoded into them; however, it is unclear exactly what is required from an ordinal notation system to forbid such counterexamples and at the same time to be sufficiently general.

The absence of a good mathematical solution of this problem is not only annoying, but it makes the basic question what constitutes an ordinal analysis of a formal theory intuitive rather than fully rigorous. The commonly used expression 'to calculate the proof-theoretic ordinal of a theory' does not really have a definite meaning: it is not clear in which terms the result of this 'calculation' needs to be specified. Thus, ordinal analysis, the characterization of proof-theoretic ordinals of theories, is sometimes described as an art [37, 38]. We usually recognize individual ordinal notation systems arising in concrete situations as natural, the simplest one is the system of ordinal notation for ε_0 based on Cantor normal forms. Many other systems are surveyed, e.g., in [18, 34, 37]. However, a general mathematical definition of a 'natural ordinal notation system' is lacking.

At this point we would like to remark that the term 'natural ordinal notation system' may be somewhat misleading. Our understanding of this problem is purely mathematical: the goal is not to explicate the vague notion of 'naturality' in a philosophical sense of the word, but rather to find a suitably general definition or a framework that would delineate a wide class of ordinal notation systems (well-ordering representations) suitable for proof-theoretic analysis.

Our main goal in this note is to consider this problem from the point of view of the theory of automatic structures. We will motivate this approach and see where it will lead us to. Our attempts do not really solve the problem, however we believe that there is value in studying failures. In the process we learn new insights, and ultimately such a study may help to point us in the right direction.

2 Turing

Arguably the first discussion of the *problem of natural ordinal notations*, as we would call it today, is found in the work of Alan Turing [47]. There, Turing studied transfinite progressions of theories based on iteration of the process of extending a theory by consistency assertions and by some more general reflection principles. The main goal of this study was to obtain a classification of arithmetical sentences[1] according to the stages of this process. To quote:

[1] Turing considered sentences of logical complexity at most Π^0_2 that he called 'number-theoretic theorems'.

> We might also expect to obtain an interesting classification of number-theoretic theorems according to "depth". A theorem which required an ordinal α to prove it would be deeper than one which could be proved by the use of an ordinal β less than α. However, this presupposes more than is justified.

The last sentence here apparently indicates that Turing realized that his results fall short of the stated goal. To explain this, we first remark that to construct a Turing progression

$$T_0 := T, \quad T_{\alpha+1} = T_\alpha + \mathrm{Con}(T_\alpha), \quad T_\lambda = \bigcup_{\alpha<\lambda} T_\alpha, \text{ if } \lambda \in \mathrm{Lim},$$

one really needs to associate theories T_α with ordinal notations (or some kind of constructive representations) rather than with ordinals α in the set-theoretic sense. In order to formulate consistency assertions $\mathrm{Con}(S)$ according to the recipe of Gödel, the axiom set of a theory S must be r.e. and represented in the language of arithmetic by a Σ_1-formula. Thus, the arithmetical formula $\mathrm{Con}(T_\alpha)$ must somehow refer to the ordinal α, and one has to deal with computable and arithmetized ordinal representations rather than with the ordinals themselves. With this understanding, Turing showed how to accurately define such a progression for a given ordinal notation system. This was further elaborated by Feferman in 1962 [17]. Both Turing and Feferman used Kleene's universal ordinal notation system O for this purpose.

One of the main results of Turing's paper can be stated (in modern terms) as follows, where $|a|$ denotes the order type of $a \in O$.

Theorem 1 (Turing) *For each true Π^0_1-sentence π there is an ordinal notation $a \in O$ such that $|a| = \omega + 1$ and T_a proves π.*

Turing's completeness result is a mixed blessing. The negative side of it is that any true Π^0_1-sentence can be proved already at stage $\omega + 1$ of a suitable Turing progression. These include all sentences of the form $\mathrm{Con}(T_b)$ with $|b|$ much larger than $\omega + 1$, which entails that the theories T_a heavily depend on particular ordinal representations rather than on their order types. Thus, the idea of a meaningful classification of sentences according to the progression stages breaks down. Turing put it in a remarkably pessimistic form:

> *This completeness theorem as usual is of no value. Although it shows, for instance, that it is possible to prove Fermat's last theorem with Λ_P (if it is true) yet the truth of the theorem would really be assumed by taking a certain formula as an ordinal formula.*[2]

Further in the paper Turing suggests a partial way out, a careful selection of specific ordinal notations.

> *We can still give a certain meaning to the classification into depths with highly restricted kinds of ordinals. Suppose that we take a particular ordinal logic Λ and a particular ordinal formula Ψ representing the ordinal α say (preferably a large one), and that we restrict*

[2] Λ_P is progression based on iteration of consistency, and 'ordinal formulas' are his ordinal notations.

ourselves to ordinal formulae of the form $\mathrm{Inf}(\Psi, a)$.[3] *We then have a classification into depths, but the extents of all the logics which we so obtain are contained in the extent of a single logic.*

Thus, Turing essentially suggests to consider linearly ordered subsets of O defined by specific 'highly restricted' ordinal notations. It seems likely that he means here particular notations such as, for example, the one for ε_0 based on Cantor normal forms, in other words, ordinal notation systems we would describe as 'natural'. However, he admits that by doing this we have to give up the idea of classification of *all* true Π^0_1 sentences. We would refer to this proposal below as *"limited Turing's program."*

Turing did not pursue this suggestion any further than that. This kind of approach has much later reappeared in the works of Schmerl [40, 41], it was taken up in [3] and other papers with many positive results. Turing's idea has been used to classify, for example, Π^0_n-consequences of specific theories, such as PA and its predicative extensions. These results were indeed based on highly specific 'natural' ordinal notations.

We know many examples of natural ordinal notation systems for fairly large constructive ordinals. Several types of such notation systems are reviewed in Feferman's paper [18]. However, we still lack a general understanding of what constitutes a 'natural' ordinal notation system. Essentially the same problem appears in the study of proof-theoretic ordinals based on Gentzen's approach.

3 Proof-theoretic Ordinals

Since Gentzen gave his consistency proof of Peano arithmetic by transfinite induction for the ordinal ε_0, much of the work in proof theory has been the exploration of the relationships between formal theories and well-orderings [26, 34, 35, 37, 46]. A proof-theoretic study of a formal theory usually culminates in the calculation of its *proof-theoretic ordinal*, that is, a well-ordering representing a bound on the strength of the system. With this ordinal, the other important characteristics of a formal theory, such as its class of provably total computable functions, are often connected.

Although it is not always duly emphasised in proof-theoretic literature, there are different ways of associating ordinals to theories. They are sensitive to different levels of logical complexity and lead, in general, to inequivalent notions of proof-theoretic ordinals. The most common notion, known since Gentzen and prevalent in the work on proof theory, is the so-called Π^1_1*-ordinal* which is defined as the supremum of order types of recursive well-orderings that are provably well-founded in a given formal theory. Of course, for this definition to be applicable, the language of the theory must be able to define recursive relations and to express the well-

[3] These formulas define initial segments of α.

foundedness property of such relations (which is Π^1_1-complete). Thus, we usually assume the language to contain at least that of first-order arithmetic with free second-order variables (denoting arbitrary sets of natural numbers). A fundamental result of Gentzen stated in these terms is that the Π^1_1-ordinal of $\mathsf{PA}(X)$, a version of Peano arithmetic in the language expanded by free set variables, is ε_0.

The Π^1_1-ordinal is a well-defined and robust measure of proof-theoretic strength, however it is not very sensitive: Extension of a theory by true Σ^1_1-axioms does not change its Π^1_1-ordinal [27, 37]. In particular, theories with the same Π^1_1-ordinal may have vastly different consistency strength and the classes of provably total computable functions. For example, PA and $\mathsf{PA} + \mathrm{Con}(\mathsf{ZF})$ have the same Π^1_1-ordinal. Moreover, Gentzen's consistency proof for Peano arithmetic is not exactly captured by the characterization of its Π^1_1-ordinal: Whereas a proof of $\mathrm{Con}(\mathsf{PA})$ by transfinite induction on ε_0 entails that the well-foundedness of ε_0 is unprovable in $\mathsf{PA}(X)$ (by Gödel's second incompleteness theorem), the converse cannot in general be concluded.

Attempts to define the notions of proof-theoretic ordinals relevant for arithmetical complexity classes such as Π^0_2 and Π^0_1 only succeed provided some natural system of ordinal notation is given (as in Turing's limited program), see [3] for a discussion of various proposals of this kind. Without such an assumption, these attempts invariably fail. This issue was treated by Georg Kreisel who constructed a number of pathological counterexamples in order to demonstrate such failures.

4 Pathological Well-orderings

Given an arithmetically definable binary relation $\prec$ and a class of formulas $\mathcal{F}$ we denote by $\mathrm{TI}(\prec; \mathcal{F})$ the schema of transfinite induction for $\varphi \in \mathcal{F}$:

$$\forall x\, (\forall y \prec x\, \varphi(y) \to \varphi(x)) \to \forall x\, \varphi(x).$$

$\mathrm{TI}(\prec; \mathcal{F})$ is true in $\mathbb{N}$ if $\prec$ is well-founded w.r.t. $\mathcal{F}$-definable sets. As remarked by Kreisel, Gentzen's consistency proof for PA was naturally formalizable in Primitive Recursive Arithmetic PRA extended by $\mathrm{TI}(\prec; \Delta_0)$, where $\prec$ is the canonical primitive recursive well-ordering of order type ε_0, and Δ_0 is the class of primitive recursive arithmetical formulas.

Now we turn to pathological well-orderings. As our starting point we consider the simplest example from [26].

Proposition 1 *For any true Π^0_1 sentence π, there is a primitive recursive well-ordering $\prec_\pi$ of order type ω such that $\mathsf{PRA} + \mathrm{TI}(\prec_\pi; \Delta_0) \vdash \pi$.*

Proof If π has the form $\forall x\, \pi_0(x)$ with $\pi_0 \in \Delta_0$, then one can define:

$$x \prec_\pi y \iff^{\mathrm{def}} (x < y \wedge \forall z < x\, \pi_0(z)) \vee (y < x \wedge \exists z < y\, \neg\pi_0(z)).$$

It is easy to see that, once π is false and n is the minimal natural number such that $\neg\pi_0(n)$, the ordering $\prec_\pi$ has a Δ_0-definable subset $\{y : y > n\}$ without the least element. Hence $\mathrm{TI}(\prec_\pi; \Delta_0)$ is false. Formalizing this in PRA shows $\mathsf{PRA} + \mathrm{TI}(\prec_\pi; \Delta_0) \vdash \pi$. On the other hand, since π is true in the standard model of arithmetic, the witness of $\neg\pi$ does not exist and $\prec_\pi$ is, in fact, isomorphic to ω (but not provably so unless one can prove π). □

This effect is essentially the same as the one observed in the above quotation from Turing: If π is Fermat's last theorem (or, better now, some Π^0_1-equivalent of Riemann's Hypothesis), the truth of π is encoded in the fact that $\prec_\pi$ is a well-ordering. This rather cheap trick is to be compared with Gentzen's theorem stating that $\mathrm{Con}(\mathsf{PA})$ is provable in $\mathsf{PRA} + \mathrm{TI}(\prec; \Delta_0)$ where $\prec$ is Cantor's canonical notation system for the ordinal ε_0, but not any proper initial segment of it.

Kreisel's example immediately shows the inadequacy of the naive definition of a proof-theoretic ordinal of a theory T as the order type of the shortest primitive recursive well-ordering $\prec$ such that $\mathrm{TI}(\prec, \Delta_0)$ over PRA proves $\mathrm{Con}(T)$: Take $\mathrm{Con}(T)$ for π and observe that the ordinal of T then equals ω irrespectively of T.

Remark 1 Kreisel's example shows that $\mathrm{TI}(\prec, \Delta_0)$ can be pathologically strong. However, there are other examples showing that this schema can be pathologically weak [2, 26]. For example, Kreisel demonstrated that there are non-wellfounded primitive recursive linear orderings $\prec$ such that PA proves the schema of transfinite induction on $\prec$ for arbitrary arithmetical formulas $\mathrm{TI}(\prec, \Pi^0_\infty)$. Accordingly, for all $\alpha < \omega_1^{CK}$, one can construct a primitive recursive well-ordering $\prec$ of order type α, such that $\mathrm{TI}(\prec, \Pi^0_\infty)$ is provable in PA. This phenomenon is closely related to the construction of linear orderings well-founded w.r.t. hyperarithmetical sets, but not actually well-founded, see [19].

Other important uses of ordinals in proof theory occur in the constructions of subrecursive hierarchies [39]. In turn, the latter are used to characterize provably total computable functions of theories [8] and as technical tools to prove the independence of certain combinatorial theorems [21].

An important hierarchy of functions is the so-called *fast-growing hierarchy* $(F_\alpha)_{\alpha<\Lambda}$ (also known as the extended Grzegorczyk hierarchy) [29]:

1. $F_0(x) = x + 1$,
2. $F_{\alpha+1}(x) = \underbrace{F_\alpha(\ldots(F_\alpha}_{x\text{–times}}, (x))\ldots)$
3. $F_\lambda(x) = F_{\lambda[x]}(x)$, for limit ordinals λ.

Here Λ is supposed to be a countable ordinal, and $\lambda[\cdot]$, for each limit ordinal $\lambda < \Lambda$, denotes a *fundamental sequence*, that is, a monotonically increasing sequence of ordinals $\lambda[n]$ such that $\mathsf{sup}_{n<\omega}\lambda[n] = \lambda$. Once a system of fundamental sequences is fixed, the functions F_α are uniquely defined for each $\alpha < \Lambda$.

If we want the functions F_α to be computable, then Λ should be represented as a computable well-ordering; the set $\Lambda \cap \mathsf{Lim}$ and the functions $x + 1$ and $x[y]$ should be computable as well.

The typical expectation for non-pathological computable systems of ordinal notations and fundamental sequences is that the growth rate of F_α reflects the size of α, e.g., F_ω is expected to be of the growth rate of the Ackermann function. The fast-growing hierarchy is often used to characterize provably total computable functions of theories. For natural ordinal notation systems, the functions F_α are provably total in PA, if $\alpha < \varepsilon_0$, while F_{ε_0} grows faster than any PA-provably total computable function.

Examples of computable ordinal notation systems leading to pathological sub-recursive hierarchies are well-known and going back at least to [16] (see also the references therein). We give an easy example of this kind here.

Proposition 2 *For any total computable function g there is a (polytime) computable well-ordering of order type $\omega + 1$ and a fundamental sequence for ω such that the growth rate of F_ω is faster than g.*

Proof Without loss of generality we switch from g to a perhaps faster growing monotone function f such that $f(x)$ is computable in time polynomial in $f(x)$. The domain of the ordering $\prec$ is the union of $\{\omega\}$ and all pairs of natural numbers (n, m) such that $m \leq f(n)$. We put, for each n, n', m, m', $(n, m) \prec \omega$ and

$$(n, m) \prec (n', m') \text{ if either } n < n' \text{ or } n = n' \text{ and } m > m'.$$

The fundamental sequence is $\omega[n] := (n, 0)$. Notice that the comparison relation, the domain of the ordering, and the functions $x + 1$ and $\omega[x]$ are computable in polynomial time. The only non-trivial algorithm here is the one for the domain.

Since $f(n)$ is computable in time polynomial in $f(n)$, there is a polynomial bound $s(m)$ such that whenever the computation of $f(n)$ does not terminate in $s(m)$ steps, $f(n) \geqslant m$. To check if (n, m) belongs to the domain, on input (n, m) run the computation for $f(n)$ for $s(m)$ steps. If $f(n)$ has not yet terminated tell that (n, m) is in the domain, otherwise check the inequality $m \leq f(n)$ directly.

We have $F_\omega(x) \geqslant f(x)$, since $F_\omega(x) = F_{(x,0)}(x)$ and the position of $(x, 0)$ in the ordering is $x + \sum_{y \leq x} f(y) \geqslant f(x)$. □

5 Complexity Considerations

How can one define a general class of ordinal representations excluding pathological examples like Kreisel's? We first remark that the idea of restricting the well-orderings to low complexity classes does not really work. The complexity of the formula $\prec_\pi$ is already low (Δ_0, which corresponds to the linear time hierarchy). In fact, one can lower the complexity of $\prec_\pi$ even further.

Let us call a class of binary relations a *basis of r.e. sets*, if it is closed under boolean operations and explicit transformations, and any r.e. set can be obtained as a projection from a relation in that class. (The term is due to R. Smullyan,

see [28, 45].) The class of Δ_0-definable relations, for example, is a basis of r.e. sets, as is the class of linear time computable relations (on multitape TMs).

Given a basis of r.e. sets $\mathcal{S}$, any Π^0_1-sentence π can be represented in the form $\forall x\, \pi_0(x)$ with $\pi_0 \in \mathcal{S}$ in the standard model of arithmetic. We say that $\mathcal{S}$ is a basis of r.e. sets *provably in* T if this holds in all models of T. If T is sufficiently strong (as strong as PRA would certainly suffice) we have that linear time computable relations, for example, are a provable basis of r.e. sets in T.

Let f be a monotone, provably total computable function in T whose graph is linear time computable, and let $g(x) := \min\{y : f(y) > x\}$. Notice that f can be fast growing (e.g. as fast as any primitive recursive function if $T = \mathsf{PRA}$), and g is therefore very slow growing. We can modify Kreisel's example as follows:

$$x \prec_\pi y \iff^{\text{def}} (x < y \wedge \forall z < g(x)\, \pi_0(z)) \vee (y < x \wedge \exists z < g(y)\, \neg\pi_0(z)).$$

We think of x and y now as of binary strings, and $<$ corresponds to the lexicographic ordering. This defines a well-ordering of order type ω whose computational complexity is only slightly above the linear function of $\max(|x|, |y|)$ assuming that $\pi_0(z)$ is computed in $O(|z|)$ steps. However, the argument for

$$\mathsf{PRA} + \mathrm{TI}(\prec_\pi; \Delta_0) \vdash \pi$$

goes through. We remark that one needs the provable totality of f in order to show the existence of a decreasing chain in $\prec_\pi$: If m is the first z such that $\pi_0(z)$ then the decreasing chain starts from $f(m)$.

We conclude that well-ordering representations that do not allow for a version of Kreisel's trick (of incorporating the consistency into the very definition of the ordering) must be more restrictive than most natural complexity classes.

The first class that comes to mind which is not a basis of r.e. sets is the class of regular languages. As is well-known, regular languages are closed under boolean operations and projection. So, projection of a regular language cannot be Σ_1-complete. This leads us to the more general idea of using automatic presentations of well-orderings as candidates for canonical ones.

6 Automatic Structures

Automatic presentations of first-order structures emerged in the fundamental works of Büchi, Rabin and others, and are still an ongoing topic of active research. An important milestone in this development was the work by Khoussainov and Nerode [24] where a general program to study such structures was initiated (the notion of automatic structure already appeared in an earlier paper by Hodgson [20]). A (relational) structure is called *automatic* if it has an isomorphic copy where the universe is a regular set of words and all relations can be recognized by synchronous

multi-tape automata. We recommend the surveys [5, 23] for an introduction to the topic of automatic structures and a historical overview.

The study of automata presentable structures was mainly motivated by the problems in computable model theory. The main advantage of automatic structures compared to (polynomially) computable structures is that they enjoy, in general, nicer computational properties. For example, automatic structures have decidable first-order theories. The class of automatic structures is closed under first-order interpretations (hence under factorization by definable congruences and definable substructures) and finite products. They also have nice alternative characterizations in terms of logical languages. The following theorem summarizes the results by Büchi, Bruyère, Blumensath and Grädel (see [5]), where $\mathcal{P}^{<\omega}(\mathbb{N})$ denotes the set of all finite subsets of $\mathbb{N}$:

Theorem 2 *Let $\mathcal{M}$ be a first-order structure. $\mathcal{M}$ is automatic iff any of the following conditions hold:*

(i) *$\mathcal{M}$ is first-order interpretable in $(\mathbb{N}, \mathcal{P}^{<\omega}(\mathbb{N}); \in, S)$, where S is the successor function.*[4]
(ii) *$\mathcal{M}$ is first-order interpretable in Büchi arithmetic $(\mathbb{N}; +, V_2)$, where $V_2(x)$ is the function that returns the maximal power of 2 dividing x.*
(iii) *$\mathcal{M}$ is first-order interpretable in $(\{0,1\}^*; \sqsubset, S_0, S_1, E)$, where $\sqsubset$ is the prefix relation, S_i are successor relations, and $E(x, y)$ is true iff the words x, y have equal length.*

Remark 2 In (ii) above the function V_2 can be replaced by any other V_k, with $k > 2$, as all structures of the form $(\mathbb{N}; +, V_k)$ are mutually interpretable. This is possible because first order interpretations allow for relativization of quantifiers. Similarly, every automatic structure is automatic over the binary alphabet. In contrast, it is well-known that V_n is definable in $(\mathbb{N}; +, V_k)$ iff n and k are multiplicatively dependent, that is, if $n^m = k^l$, for some k, l [48].

Automatic well-orderings have been studied early on, and one of the basic results is the following theorem by Delhommé [15]:

Theorem 3 *An ordinal α has an automatic presentation iff $\alpha < \omega^\omega$. Moreover, from an automaton recognizing the binary relation one can effectively construct a Cantor normal form presentation of its order type α:*

$$\alpha = \omega^{n_1} m_1 + \omega^{n_2} m_2 + \cdots + \omega^{n_k} m_k$$

with $m_i > 0$ and $n_1 > n_2 > \cdots > n_k \geqslant 0$.

Corollary 1 *The isomorphism problem for automatic well-orderings is decidable.*

[4] We consider here the two sorted (weak second-order) structure as a first-order structure in the usual way. This allows us to use the standard notion of first-order interpretation here (see e.g. [5, 30]).

This result shows that automatic presentations of well-orderings are equivalent to Cantor normal form presentations (of ordinals below ω^ω). The theorem itself is provable by elementary methods, which means that automatic well-orderings are natural in the sense of proof theory. However, a major drawback is that they do not work beyond the very small ordinal ω^ω, which is at the lowermost end of ordinal notations of interest in proof theory (it corresponds to the proof-theoretic ordinal of primitive recursive arithmetic).

Can one do better? A way out is to look for more general types of presentations than the automatic ones. In the recent literature several other kinds of automatic-like structures were considered, especially: *tree-automatic* (which accept as inputs finite labelled trees), *Büchi automatic* (which accept as inputs infinite words), and *Rabin automatic* ones (which accept as inputs infinite binary labelled trees). A structure is called tree (respectively, Büchi, Rabin) automatic, if its domain and basic relations are recognizable by tree (respectively, Büchi, Rabin) automata. Classical results of Büchi [9] and Rabin [36] relate this to definability in (weak) monadic second order structures of, respectively, natural numbers with successor and the infinite binary tree. See also [5].

Theorem 4 (Büchi, Rabin)

(i) A structure is Büchi automatic iff it is definable in $(\mathbb{N}, \mathcal{P}(\mathbb{N}); \in, S)$.
(ii) A structure is tree-automatic iff it is definable in $(\{0,1\}^*, \mathcal{P}^{<\omega}(\{0,1\}^*); \in, S_0, S_1)$.
(iii) A structure is Rabin automatic iff it is definable in $(\{0,1\}^*, \mathcal{P}(\{0,1\}^*); \in, S_0, S_1)$.

The class of tree-automatic structures is genuinely larger than the class of automatic ones. For example, the structure of natural numbers with multiplication (Skolem arithmetic) is tree-automatic but not automatic. In the case of well-orderings, however, the gains are relatively minor. For tree-automatic ordinals, Delhommé [15] showed that Theorem 3 can be improved by one exponent: An ordinal α is tree-automatic iff $\alpha < \omega^{(\omega^\omega)}$.

The classes of Büchi and Rabin automatic structures are advantageous when one wants to represent uncountable structures. Their first-order theories are decidable and in general the classes have nice closure properties. For example, the structure of reals with addition is Büchi automatic [5].

Proof-theoretic ordinal notation systems are supposed to be interpreted in first-order arithmetic (otherwise, transfinite induction schema can hardly be stated), and in this case we want ordinal representations to be finite objects. However, e.g. in the case of the binary tree, if one only considers MSO definable structures whose elements correspond to *finite* sets of words, then these will be the same as tree-automatic (or WMSO definable) structures. Thus, prima facie there is no advantage in considering Büchi and Rabin representations of well-orderings. However, this discussion brings us to a few useful general observations.

Firstly, the class of relations definable in decidable theories, such as MSO theories of the binary tree, can never be a basis of r.e. sets in the sense of

Sect. 5. Hence, they are good candidates for avoiding pathological well-ordering counterexamples. The more expressive is such a decidable theory, the larger a class of interpretable well-orderings we potentially obtain. This draws our attention towards expressively strong decidable theories.

Secondly, suppose a structure $\mathcal{A}$ has a decidable monadic second-order theory (as opposed to its first-order theory). If a relation $\prec$ is interpreted in $\mathcal{A}$ in such a way that its domain consists of elements of $\mathcal{A}$ (is MSO definable), then the well-foundedness of $\prec$ is expressible by the MSO formula

$$\forall X\ \forall x\ (\forall y \prec x\ y \in X \rightarrow x \in X) \rightarrow \forall x\ x \in X.$$

Hence, given a formula defining $\prec$, one can effectively check whether $\prec$ is a well-ordering. Thus, the class of interpretable well-orderings will have an additional nice property of being effectively recognizable.

Thirdly, because of the previous property, if a model has a decidable MSO theory in the language expanded by constants for each element of the domain, then one can enumerate all the well-orderings definable in it. Hence, their order types will be uniformly bound by some constructive ordinal $\alpha < \omega_1^{\mathrm{CK}}$.

The only example we know of where this idea works beyond the tree-automatic ordinals is the so-called Caucal hierarchy.

7 Caucal Hierarchy

Muchnik's theorem [43] generalizes the decidability result of Rabin by establishing that the decidability of the MSO theory of a structure is preserved under the iteration operation that maps relational structures $\mathcal{M} = (M; R_0, \ldots, R_{n-1})$ to certain natural structures $\mathcal{M}^*$ on finite words over its domain M. In particular, a well studied corollary of Muchnik's results is the decidability of MSO theories of graphs from the Caucal hierarchy [13, 14].

Here we consider directed graphs with edges colored in finitely many colors such that, for all vertices x, y and color i, there is at most one edge from x to y of that color. Formally, we say that a structure $\mathcal{G}$ is a *directed graph with colored edges* if its signature consists of finitely many binary relations $\{R_i \mid i < n\}$. Later in this section we will simply call them graphs.

We have two natural operations on these graphs preserving the decidability of MSO theories:

1. MSO interpretations,[5] i.e., the interpretations where the first-order domain is interpreted by a one-dimensional MSO definable set of first-order elements and the relations are interpreted as MSO definable relations.

[5] Transductions in the terminology of [30].

2. The unfolding operation Unf, where for a graph $\mathcal{G}$, the vertices of $\mathsf{Unf}(\mathcal{G})$ are all possible paths $\mathsf{Path}(\mathcal{G})$ through $\mathcal{G}$, and we have an i-colored edge from a path $\alpha = (v_0, \ldots, v_m)$ to $\beta = (u_0, \ldots, u_k)$ if $k = m + 1$, $v_0 = u_0, \ldots, v_m = u_m$, and in $\mathcal{G}$ there is an i-colored edge from v_m to u_{m+1}.

Level 0 of the Caucal hierarchy C_0 consists of all finite graphs, and level $n + 1$ of the hierarchy C_{n+1} consists of all the graphs that are MSO interpretable in the unfoldings of graphs from C_n.

From the automata-theoretic perspective, the Caucal hierarchy naturally corresponds to *higher-order pushdown automata* first introduced by Maslov [31]. We call a 0-*pds* over a finite alphabet A (pds stands for *higher order pushdown store*) just a letter from A. A $(n + 1)$-*pds* over A is a finite sequence of n-pds. We think about $(n + 1)$-pds as a stack of n-pds. Further, we have a natural pop^k operation, $0 < k \leq n$, removing the topmost $(k - 1)$-pds (if $k < n$ consider the topmost $(n - 1)$-pds and apply pop^k to it). The operation $\mathsf{push}^k(a)$ modifies the topmost k-pds by creating a copy of the topmost $(k - 1)$-pds and putting it on the top of the k-pds, followed by replacing the topmost 0-pds with the letter a.

Now one can naturally define the notion of a *n-pushdown automaton* as a finite state transition device that uses a single n-pds as its memory. We do not go into the details here, but it is important to allow ε-transitions, i.e., the transitions where the automaton can perform an operation on n-pds and transit to a new state, but is not reading any characters from the input. The *configuration graph* of a (possibly non-deterministic) n-pushdown automaton is the graph whose vertices are the states reachable from the starting state (with an empty n-pds). Its edges are labeled with ε and characters of the input alphabet and correspond to the one-step transitions. As proved by Carayol and Wöhrle [12], level n of the Caucal hierarchy consists precisely of (the graphs isomorphic to) ε-contractions of the configuration graphs of n-pushdown automata.

In the previous section we connected ordinals with structures with decidable MSO theories by means of first-order interpretations. Although this question is also natural to ask for the case of the Caucal hierarchy, to the best of our knowledge it is open.

In this section we will be looking at a stronger notion of representability of ordinals in the sense of being MSO interpretable. Structures in the Caucal hierarchy have decidable MSO theories and all their elements are definable. Hence, the remarks at the end of the previous section apply. In particular, one can effectively recognize well-orderings in the Caucal hierarchy, and there is a uniform bound on their order types. The following theorem by Braud and Carayol [6, 7] explicitly describes this bound.

Define $\omega_0 := 1$ and, for $k < \omega$, let $\omega_{k+1} := \omega^{\omega_k}$. Then ε_0 is the supremum of $\{\omega_k : k < \omega\}$.

Theorem 5 *An ordinal α is MSO interpretable in a graph from the Caucal hierarchy iff $\alpha < \varepsilon_0$. An ordinal α is interpretable in the n-th level of the Caucal hierarchy iff $\alpha < \omega_{n+1}$.*

Remark 3 We do not really know if the Cantor normal form of an ordinal can be computed from its Caucal hierarchy representation. Nor do we know if the isomorphism problem for well-orderings in the Caucal hierarchy is decidable.

Remark 4 We know that MSO theories of the structures in the Caucal hierarchy are decidable. A fortiori, this holds for the well-orderings in the Caucal hierarchy.

By the results of Kołodziejczyk and Michalewski [25], Rabin's theorem on the decidability of the MSO theory of the binary tree is surprisingly strong: it is unprovable in the fragment of the second-order arithmetic with Δ^1_3-comprehension axioms. This theory is very much stronger than PA and its proof-theoretic ordinal is larger than all currently known ones. We do not really know the corresponding lower bound for the Caucal hierarchy, but in any case it can only be worse.

Therefore, we are in a curious situation that Caucal hierarchy presentations, in general, may not be recognizably decidable within a given formal system. It puts some doubts on the hope that any Caucal presentation of a well-ordering would be provably isomorphic to a computable one (such as the canonical one based on Cantor normal forms). However, there is a caveat: By a well-known theorem of Büchi [10, 11] any countable well-order has a decidable MSO theory. This theorem seems to be proof-theoretically weaker than Rabin's. So, Caucal presentations of *well-orderings* may not actually require as strong axioms as Δ^1_3-comprehension.

In any case, there is a strange discrepancy between the order types of Caucal representable well-orderings and the very strong axioms needed to show that they are decidable.

8 Fundamental Sequences in the Caucal Hierarchy

As we discussed in Sect. 4, the assignment of fundamental sequences to ordinals within an ordinal notation system is important for the construction of subrecursive hierarchies of functions such as the fast-growing hierarchy. Given that such hierarchies play a significant role in proof theory, in this section we will discuss a result from [32] stating that fundamental sequences can also be represented with the Caucal hierarchy and yield, under some natural conditions, the hierarchies of functions equivalent to the one for the standard fundamental sequences assignment. We are very sketchy and refer the reader to [32] for more details.

Firstly, we remark that an ordinal represented in the Caucal hierarchy can always be expanded by a system of fundamental sequences in the same hierarchy. We represent a system of fundamental sequences by the predicate $\mathsf{FS}(\beta, \gamma)$ expressing that β is an element of the fundamental sequence for γ. Here we naturally assume that only the limit ordinals have non-empty fundamental sequences and, for any limit ordinal λ, the set of ordinals $\{\beta \mid \mathsf{FS}(\beta, \lambda)\}$ has order type ω and λ as its limit. We then let $\lambda[n]_{\mathsf{FS}}$ denote the n-th element of the set $\{\beta \mid \mathsf{FS}(\beta, \lambda)\}$.

To see that, given a well-ordering in the Caucal hierarchy, some system FS exists that can be defined in the Caucal hierarchy, we consider deterministic trees. A

deterministic tree is a tree where from each vertex there is at most one outgoing edge of any given color. It is known [12] that every graph from C_n is MSO interpretable in a deterministic tree from C_n. It is then fairly easy to expand an MSO interpretation of an ordinal in a deterministic tree by an MSO definable system of fundamental sequences FS.

A commonly considered nice property of systems of fundamental sequences is the so-called *Bachmann property* [1, 39, 42]. It demands that each $\lambda[\cdot]\colon \omega \to \lambda$ is strictly increasing and that for any limit α from $(\lambda[n], \lambda[n+1]]$ the value $\alpha[0] \geqslant \lambda[n]$. As explained in [32], if a well-ordering equipped with a system of fundamental sequences FS is MSO interpreted in a deterministic tree, then one can construct another MSO definable system $\mathsf{FS}' \subseteq \mathsf{FS}$ with the Bachmann property.

As before, when we have a computable ordinal equipped with system of computable fundamental sequences $\cdot[\cdot]_{\mathsf{FS}}$, we can form the associated fast-growing hierarchy. If FS satisfies the Bachmann property, then the asymptotic growth rate of the functions F_α from the hierarchy grows monotonically in α [39]. One of the criteria of an ordinal notation system of being natural is that it leads to the levels of fast-growing hierarchy having the expected growth rate reflecting the value of ordinal. The following theorem [32] confirms that this is the case for the Caucal hierarchy representations.

Theorem 6 *Suppose a structure* $(\Lambda, <, \mathsf{FS}_1, \mathsf{FS}_2)$ *in the Caucal hierarchy is an ordinal with two systems of fundamental sequences, both satisfying the Bachmann property. For each* $i \in \{1, 2\}$, *let* $(F_\alpha^{\mathsf{FS}_i}\colon \mathbb{N} \to \mathbb{N})_{\alpha<\Lambda}$ *be the fast-growing hierarchy according to* FS_i. *Then for each* $\alpha < \beta < \Lambda$ *there is an* N *so large that*

$$F_\alpha^{\mathsf{FS}_1}(x) < F_\beta^{\mathsf{FS}_2}(x), \text{ for } x \geqslant N.$$

Hence, all systems of fundamental sequences in the Caucal hierarchy yield fast-growing hierarchies of similar growth rates. An important tool to get bounds on the hierarchy functions is the pumping lemma for higher-order pushdown automata proved by Parys [33].

The standard system of fundamental sequences for ordinals $\lambda < \varepsilon_0$ given in Cantor normal form is defined by:

$$\lambda[n] := \begin{cases} \alpha + \omega^\beta \cdot (n+1), & \text{if } \lambda = \alpha + \omega^{\beta+1}; \\ \alpha + \omega^{\beta[n]}, & \text{if } \lambda = \alpha + \omega^\beta \text{ and } \beta \in \mathrm{Lim}. \end{cases}$$

The standard system satisfies the Bachmann property and is MSO definable in $(\Lambda, <)$ for ordinals $\Lambda < \omega^\omega$ (but not above). Hence, we obtain the following corollary.

Corollary 2 *Suppose* $(\Lambda, <, \mathsf{FS})$ *in the Caucal hierarchy is an ordinal* $< \omega^\omega$ *together with a system of fundamental sequences satisfying the Bachmann property. Let* $(F_\alpha^{\mathsf{FS}}\colon \mathbb{N} \to \mathbb{N})_{\alpha<\Lambda}$ *be the fast-growing hierarchy according to* $\cdot[\cdot]_{\mathsf{FS}}$ *and*

let $(F_\alpha : \mathbb{N} \to \mathbb{N})_{\alpha<\omega^\omega}$ *be the fast-growing hierarchy (defined using the standard fundamental sequences). Then for each* $\alpha < \beta < \Lambda$ *there is an* N *so large that*

$$F_\alpha(x) < F_\beta^{\mathsf{FS}}(x) \quad \textit{and} \quad F_\alpha^{\mathsf{FS}}(x) < F_\beta(x), \textit{for } x \geqslant N.$$

We do not know if the counterpart of Corollary 2 holds without the restriction on Λ. One can also ask similar questions about the other classes of automatically represented well-orders considered in Sect. 6.

9 Automatic Well-founded Relations

We have seen that the notion of automatic well-order provides natural ordinal representations, however is too restrictive for proof-theoretic applications. The situation is the same for well-founded partial orders [22]. In proof theory, transfinite induction can be stated more generally for well-founded relations rather than just for well-founded orders (or partial orders).

Are automatic presentations of well-founded relations always natural? We answer this question negatively by employing a construction by Khoussainov and Minnes from [22] who showed that there exist automatic well-founded relations of arbitrary large ordinal rank $< \omega_1^{CK}$.

Theorem 7 *For each true* Π_1^0*-sentence* π *there exists an automatic well-founded binary relation* R_π *such that* $\mathsf{PRA} + \mathrm{TI}(R_\pi, \Delta_0) \vdash \pi$.

Proof The proof is essentially an application of a construction from [22, Theorem 1.2] to Kreisel's example. We only sketch it here and refer to [22] for additional details.

The proof relies on several general facts. Firstly, the configuration graph of a Turing machine is automatic. Vertices of the graph are tuples of words (representing the content and the position of the head on each tape). Edges represent one-step transitions of the machine between the configurations.

Secondly, for each Turing machine there is a reversible three tape Turing machine accepting the same language. For such a machine both the in-degree and the out-degree of the configuration graph are at most one, in fact, the graph is a union of chains that are either finite or of the type of natural numbers. Therefore, the graph is a well-founded relation (of rank $\leqslant \omega$).

Given a Π_1-sentence π, let $\prec_\pi$ denote Kreisel's ordering on $\{0, 1\}^*$ (as defined in Sect. 4), and let $\mathcal{M}$ be the Turing machine computing $\prec_\pi$ in the sense that on input (x, y) it outputs 'yes' or 'no' depending on whether $x \prec_\pi y$ holds. Kreisel's definition of $\prec_\pi$ can be read as an algorithm for computing $\prec_\pi$. Moreover, the associated Turing machine $\mathcal{M}$ can be represented in arithmetic in such a way that the result of its computation provably in PRA meets its specification: $\mathcal{M}(x, y) =$ 'yes' iff $x \prec_\pi y$. Moreover, we can assume $\mathcal{M}$ to be (PRA provably) reversible.

Let (D, E) denote the configuration graph of $\mathcal{M}$. We define the domain of the automatic structure A as $\{0,1\}^* \cup D$ (the union is disjoint). The relation R_π is defined as the union of E (on D) and:

- All pairs (x, z) such that $x \in \{0,1\}^*$ and z is the initial configuration of $\mathcal{M}$ on input (x, y), for some $y \in \{0,1\}^*$;
- All pairs (z, y) where $y \in \{0,1\}^*$ and z is the final configuration of an accepting computation of $\mathcal{M}$ on input (x, y), for some $x \in \{0,1\}^*$.

Since (D, E) is automatic, it is easy to see that so is R_π.

Let R^* denote the transitive closure of R.

Lemma 1 *For all $x, y \in \{0,1\}^*$, if $x \prec_\pi y$ then $x R_\pi^* y$.*

Proof Suppose $x \prec_\pi y$, then $\mathcal{M}(x, y) =$ 'yes'. Let $I(x, y)$ denote the initial configuration of $\mathcal{M}$ on input (x, y), and $F(x, y)$ its final configuration. Then we have a path in (A, R_π) from x to y: $x R_\pi I(x, y) R_\pi^* F(x, y) R_\pi y$. □

Using Kreisel's argument we see that, if π is false, then there is a descending sequence w.r.t. $\prec_\pi$. By Lemma 1 it generates a descending sequence w.r.t. R_π, hence R_π is not well-founded. Formalizing this in PRA yields that PRA + $\mathrm{TI}(R_\pi, \Delta_0) \vdash \pi$.

Now we show that R_π is actually well-founded. Since π is true in the standard model, $\prec_\pi$ is isomorphic to ω. Consider a nonempty set $X \subseteq A$. If $X \cap \{0,1\}^* \neq \varnothing$ we first consider the $\prec_\pi$ minimal element $a \in X \cap \{0,1\}^*$. By the construction of R_π (second item), the elements u such that $u R_\pi a$ must be final configurations of accepting computations of $\mathcal{M}(x, a)$, for some x. Select any such x. Then $x \prec_\pi a$ and, by the minimality of a, $x \notin X$. The computation chain of $\mathcal{M}(x, a)$ is finite. We claim that its minimal element in X (if exists) will be R_π-minimal, otherwise a will be R_π-minimal. This is clear for all but the first element of the computation chain. However, if v is the initial configuration then its only R_π-predecessor is x, but x is not in X. So, v, if in X, must be R_π-minimal.

If $X \cap \{0,1\}^* = \varnothing$, then X is a non-empty subset of the computation graph D, which is well-founded by the reversibility condition. The minimal element of X in D will also be R_π minimal in A. □

This example shows that automatic presentation of a structure is not always nice. Whether or not crucially depends on the choice of the relations of the structure one assumes to be automatic. So, this brings us back to the question, what kind of structures are proof-theoretic ordinals?

10 Conclusion and Open Questions

We would like to mention the following questions resulting from our analysis. We think they are interesting irrespectively of the problem of naturality of ordinal notation systems.

– Are there mathematically interesting classes of structures expressively stronger than the Caucal hierarchy for which the MSO theory (FO theory) is decidable? We would like to find such structures (or to show that they do not exist) for important proof-theoretic ordinals such as the Feferman–Schütte ordinal, the Howard ordinal, etc.

– Can one relate in an intrinsic way the Caucal hierarchy representations of ordinals below ε_0 and Peano arithmetic, that is, to use them directly for a proof-theoretic analysis of PA? For example, it seems natural to represent in the Caucal hierarchy the infinitary derivation trees arising from PA-proofs and possibly the non-well-founded proofs in a cyclic version of PA [44].

– As we have discussed above, automatic well-founded partial orders have very small ranks, whereas automatic well-founded binary relations do not, but neither do they exclude pathological counterexamples. Are there natural types of well-founded structures, whose automatic presentations are tame, yet much larger proof-theoretic ordinals are presentable? This is related to the more traditional view of proof-theoretic ordinals as orders equipped with an additional structure. In the proof-theoretic literature there are various proposals as to possible structures (see [4, 18, 27, 35, 37, 46]).

Acknowledgments The first author would like to thank Andrei Muchnik (1958–2007) with whom he has had several memorable conversations about the problem of natural proof-theoretic ordinal notations. Curiously, they were not related to MSO definability and Muchnik's own remarkable work playing a role in the present paper. We also thank Iskander Kalimullin and Alexander Shen for helpful remarks and references.

References

1. Bachmann, H.: Transfinite Zahlen. J. Symb. Logic **24**(3) (1959)
2. Beklemishev, L.D.: Another pathological well-ordering. In: Buss, S.R., Hájek, P., Pudlák, P. (eds.) Logic Colloquium '98. Lecture Notes in Logic, pp. 105–108. Cambridge University Press, Cambridge (2000)
3. Beklemishev, L.D.: Proof-theoretic analysis by iterated reflection. Archive Math. Logic **42**, 515–552 (2003)
4. Beklemishev, L.D.: Provability algebras and proof-theoretic ordinals, I. Annals Pure Appl. Logic **128**, 103–123 (2004)
5. Blumensath, A., Grädel, E.: Finite presentations of infinite structures: Automata and interpretations. Theory Comput. Syst. **37**, 641–674 (2004)
6. Braud, L.: Covering of ordinals. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, vol. 4, pp. 97–108 (2009)
7. Braud, L., Carayol, A.: Linear orders in the pushdown hierarchy. In: International Colloquium on Automata, Languages, and Programming, pp. 88–99. Springer, Berlin (2010)
8. Buchholz, W., Wainer, S.: Provably computable functions and the fast growing hierarchy. In: Contemporary Mathematics, vol. 65, pp. 179–198 (1987)
9. Büchi, J.R.: On a decision method in restricted second order arithmetic. In: Logic, Methodology and Philosophy of Science; Proceedings of the 1960 International Congress. Stanford (California), pp. 1–11. Stanford University Press (1962)

10. Büchi, J.R.: Decision methods in the theory of ordinals. Bull. Am. Math. Soc. **71**, 767–770 (1965)
11. Büchi, J.R.: The monadic second order theory of ω_1. In: The Monadic Second Order Theory of All Countable Ordinals (Decidable Theories, II). Lecture Notes in Mathematics, vol. 328, pp. 1–127. Springer, Berlin (1973)
12. Carayol, A., Wöhrle, S.: The Caucal hierarchy of infinite graphs in terms of logic and higher-order pushdown automata. In: FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science: 23rd Conference, Mumbai, India, December 15-17, 2003. Proceedings 23, pp. 112–123. Springer (2003)
13. Caucal, D.: On infinite transition graphs having a decidable monadic theory. In: International Colloquium on Automata, Languages, and Programming, pp. 194–205. Springer (1996)
14. Caucal, D.: On infinite terms having a decidable monadic theory. In: International Symposium on Mathematical Foundations of Computer Science, pp. 165–176. Springer (2002)
15. Delhommé, C.: Automaticité des ordinaux et des graph homogénes. C.R. Acad. Sci. Paris, Ser. I **339**, 5–10 (2004)
16. Feferman, S.: Classifications of recursive functions by means of hierarchies. Trans. Am. Math. Soc. **104**(1), 101–122 (1962)
17. Feferman, S.: Transfinite recursive progressions of axiomatic theories. J. Symb. Logic **27**, 259–316 (1962)
18. Feferman, S.: Three conceptual problems that bug me. Lecture text for 7-th Scandinavian Logic Symposium (1996). ftp://math.stanford.edu/pub/papers/feferman/conceptualprobs.ps.gz
19. Harrison, J.: Recursive pseudo-well-orderings. Trans. Am. Math. Soc. **131**, 526–543 (1968)
20. Hodgson, B.R.: On direct products of automata decidable theories. Theor. Comput. Sci. **19**, 331–335 (1982)
21. Ketonen, J., Solovay, R.: Rapidly growing Ramsey functions. Ann. Math. **113**(2), 267–314 (1981)
22. Khoussainov, B., Minnes, M.: Model-theoretic complexity of automatic structures. Ann. Pure Appl. Logic **161**, 416–423 (2009)
23. Khoussainov, B., Minnes, M.: Three lectures on automatic structures. In: Delon, F., Kohlenbach, U., Maddy, P., Stephan, F. (eds.) Logic Colloquium 2007, pp. 132–176. Cambridge University Press, Cambridge (2010)
24. Khoussainov, B., Nerode, A.: Automatic presentations of structures. In: Lecture Notes in Computer Science, vol. 960, pp. 367–392. Springer, Berlin (1995)
25. Kołodziejczyk, L., Michalewski, H.: How unprovable is Rabin's decidability theorem? In: Grohe, M., Koskinen, E., Shankar, N. (eds.) Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016, pp. 788–797. ACM (2016)
26. Kreisel, G.: A survey of proof theory. J. Symb. Logic **33**, 321–388 (1968)
27. Kreisel, G.: Wie die Beweistheorie zu ihren Ordinalzahlen kam und kommt. Jahresbericht der Deutschen Mathematiker-Vereinigung **78**(4), 177–223 (1977)
28. Lewis, F.D.: On unsolvability in subrecursive classes of predicates. Notre Dame J. Formal Logic **20**(1), 55–67 (1979)
29. Löb, M.H., Wainer, S.S.; Hierarchies of number-theoretic functions. Archive Math. Logic **13**, 39–51, 97–113 (1970)
30. Makowsky, J.A.: Algorithmic uses of the Feferman–Vaught theorem. Ann. Pure Appl. Logic **126**(1), 159–213 (2004)
31. Maslov, A.N.: The hierarchy of indexed languages of an arbitrary level. Doklady Akademii Nauk **217**(5), 1013–1016 (1974)
32. Pakhomov, F.: Ordinal notations in Caucal hierarchy. Preprint (2015). arXiv:1512.05036
33. Parys, P.: A pumping lemma for pushdown graphs of any level. In: STACS'12 (29th Symposium on Theoretical Aspects of Computer Science), vol. 14, pp. 54–65. LIPIcs (2012)
34. Pohlers. W.: A short course in ordinal analysis. In: Axcel, A., Wainer, S. (eds.) Proof Theory, Complexity, Logic, pp. 867–896. Oxford University Press, Oxford (1993)
35. Pohlers, W.: Proof Theory: The First Step into Impredicativity. Springer, Berlin (2009)

36. Rabin, M.O.: Decidability of second-order theories and automata on infinite trees. Trans AMS **141**, 1–35 (1969)
37. Rathjen, M.: The realm of ordinal analysis. In: Cooper, S.B., Truss, J.K. (eds.) Sets and proofs. London Math. Soc. Lect. Note Series 258, pp. 219–279. Cambridge University Press, Cambridge (1999)
38. Rathjen, M.: The art of ordinal analysis. In: Proceedings of the International Congress of Mathematicians, vol. 2, pp. 45–69 (2006)
39. Rose, H.E.: Subrecursion: Functions and Hierarchies. Clarendon Press, Oxford (1984)
40. Schmerl, U.R.: A fine structure generated by reflection formulas over Primitive Recursive Arithmetic. In: Boffa, M., van Dalen, D., McAloon, K. (eds.) Logic Colloquium'78, pp. 335–350. North Holland, Amsterdam (1979)
41. Schmerl, U.R.: A proof-theoretical fine structure in systems of ramified analysis. Archive Math. Logic **22**, 167–186 (1982)
42. Schmidt, D.: Built-up systems of fundamental sequences and hierarchies of number-theoretic functions. Arch. Math. Log. **18**(1), 47–53 (1977)
43. Semenov, A.L.: Decidability of monadic theories. In: International Symposium on Mathematical Foundations of Computer Science, pp. 162–175. Springer (1984)
44. Simpson, A.: Cyclic arithmetic is equivalent to Peano Arithmetic. In: Esparza, J., Murawski, A.S. (eds.) Foundations of Software Science and Computation Structures – 20th International Conference, FOSSACS 2017, Uppsala, Sweden, April 22–29, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10203, pp. 283–300 (2017)
45. Smullyan, R.: Theory of Formal Systems. Princeton University Press, Princeton (1961)
46. Takeuti, G.: Proof Theory, 2nd edn. Dover Publications, New York (2013)
47. Turing, A.M.: System of logics based on ordinals. Proc. Lond. Math. Soc. Series 2 **45**, 161–228 (1939)
48. Villemaire, R.: The theory of $\langle \mathbb{N}, +, V_k, V_l \rangle$ is undecidable. Theor. Comput. Sci. **106**, 337–349 (1992)

On the Counting Complexity of the Cover Polynomial for Simple Graphs

Markus Bläser and Nico Mansion

Abstract Graph polynomials are graph invariants that map graphs to polynomials. As an analogue of the famous Tutte polynomial for directed graphs, Chung and Graham (J. Comb. Theory Series B 65(2):273–290, 1995) define the cover polynomial $C_G(x, y)$. Bläser and Dell (Automata, Languages and Programming, pp. 801–812. Springer, Berlin, 2007) prove that evaluating the cover polynomial is #P-hard, except for the points $(0, 0)$, $(0, -1)$, $(1, -1)$. There the evaluation is easy. However, the graphs used in this reduction are not simple. Motivated by a connection between the cover polynomial and the drop polynomial for simple graphs, Chung and Graham (J. Comb. Theory Series B 126:62–82, 2017) conjecture that the cover polynomial is also hard when restricted to simple graphs (which do not allow parallel edges or loops). As our main result, we confirm this conjecture.

1 Introduction

Graph polynomials are functions that map directed or undirected graphs to polynomials such that isomorphic graphs are always mapped to the same polynomial. There are many graph polynomials, however one of the most interesting ones is the Tutte polynomial T, see e.g. [10], as many of its points encode interesting properties of the given graph. For example, $T_G(1, 1)$ is the number of spanning trees of a graph G, $T_G(1, 2)$ the number of spanning subgraphs and $T_G(2, 0)$ the number of acyclic orientations of G.

The Tutte polynomial is defined for undirected graphs. Chung and Graham [7] define an analogue for directed graphs, the (factorial) cover polynomial $C_G(x, y)$. When we talk about the complexity of the evaluation of a graph polynomial at a given point, as for example $T_G(x, y)$, we regard x and y as constant and only consider the complexity in dependence of the input graph G. We write $T(x, y)$ for

M. Bläser (✉) · N. Mansion
Saarland University, Saarbrücken, Germany
e-mail: mblaeser@cs.uni-saarland.de; s8nimans@stud.uni-saarland.de

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_10

this problem, the problem of evaluating the Tutte polynomial at the point (x, y) given some input graph.

We can view a graph polynomial as a family of computational problems, parameterized by the evaluation point. Given a fixed point (x, y), we can now study the complexity of the map $G \mapsto T_G(x, y)$. For the Tutte polynomial, Jaeger, Vertigan, and Welsh [13] completely classified these complexities. For almost all points $(x, y) \in \mathbb{Q}^2$, the problem is #P-hard, except for one hyperbola and four points. There the problem can be solved in polynomial time. Many other graph polynomials show a similar behaviour, like the cover polynomial [2, 3, 6], the interlace polynomial [4], the Bollobás-Riordan polynomial [5], the most general edge elimination polynomial [1, 12], and other variants of coloring polynomials [11]. This resulted in Makowsky's difficult point conjecture [15].

1.1 Previous Results

For the cover polynomial, the difficult point property was shown by Bläser and Dell [3]. They prove that the evaluation of the factorial cover polynomial is #P-hard everywhere, except for three polynomial time computable points and that the geometric cover polynomial is #P-hard everywhere except for two easy points. Later, Bläser and Curticapean [2] showed that both the factorial as well as the geometric cover polynomial are #P-hard for planar graphs except for a few points. However, during the constructions, multiple lines and self-loops are introduced, even if the original graph was simple. Thus, strictly speaking, the above results only hold for multigraphs with self-loops. Chung and Graham [8] conjectured that the cover polynomial is also hard when restricted to simple graphs, which do not allow parallel edges or self-loops. The proof of this conjecture will be our main result.

1.2 Our Results

In this paper, we consider the complexity of computing the factorial cover polynomial for simple graphs as well as for planar, simple graphs. We prove that for simple graph, the same hardness classification as obtained by Bläser and Dell [3] holds. For simple, planar graphs, we also confirm Makowsky's difficult point conjecture, however for some lower dimensional set of points, the complexity remains open (as in [2]). In the end, we translate our results to the geometric cover polynomial and the drop polynomial.

2 Preliminaries

Let $\mathbb{N} = \{0, 1, ...\}$ denote the natural numbers. The graphs considered in this work are directed multigraphs $G = (V, E)$, that is, parallel edges and self-loops are allowed. We will usually call the number of vertices n and the number of edges m, counted with multiplicities. Simple graphs are graphs without any loops or parallel edges. Furthermore, we call a graph planar, if one is able to draw it without any intersecting edges in the plane.

2.1 Counting Complexity Basics

The class #P consists of all functions $f : \{0, 1\}^* \to \mathbb{N}$, such that a polynomial time verifier has exactly $f(x)$ accepting certificates of polynomial length, for all inputs x. For two functions $f, g : \{0, 1\}^* \to \mathbb{N}$, we say f Turing-reduces to g in polynomial time (denoted by $f \leq_P^T g$) if there is a deterministic oracle Turing machine that, given an oracle to g, computes f in polynomial time. If the oracle is only used once, we will call this a polynomial time many-one reduction ($f \leq_P^m g$). If the output of the oracle in a many-one reduction is $f(x)$, we call this a parsimonious reduction ($f \leq_P g$). The notions of hardness and completeness are defined as usual: A function f is #P-hard (under Turing reductions) if $\forall g \in \#\mathsf{P} : g \leq_P^T f$ and #P-complete if it is #P-hard and in #P. Hardness under the other two types of reductions is defined similarly.

2.2 Polynomials

Polynomials $p(x_1, ..., x_m) \in \mathbb{Q}[x_1, ..., x_m]$ are elements of a polynomial ring, in this case over $\mathbb{Q}$. Given $d+1$ input-value pairs, one can compute a univariate polynomial p with $\deg(p) \leq d$ using Lagrangian interpolation that maps all inputs to their respective values. Furthermore, p is unique, that is, a polynomial of degree d can be exactly reconstructed just from $d + 1$ distinct input-value pairs.

2.3 Isomorphic Graphs

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called isomorphic if there is a bijection $f : V_1 \to V_2$, such that $(u, v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2$. This means that G_1 can be transformed into G_2 (and vice versa) by renaming the vertices.

2.4 Graph Invariants and Graph Polynomials

A graph invariant is a function f that maps graphs to some values such that isomorphic graphs get the same value. If that set is a polynomial ring, we call f a graph polynomial.

2.5 The Cover Polynomial and Its Relatives

Definition 1 (Path-cycle Cover) A path-cycle cover of a directed graph $G = (V, E)$ is a set of disjoint simple paths and cycles, such that all vertices are covered by exactly one path or one cycle. We will denote the number of path-cycle covers of G with i paths and j cycles by $c_G(i, j)$.

Definition 2 (Factorial Cover Polynomial) The factorial cover polynomial, defined by Chung and Graham [7], is a twovariate polynomial and is defined by

$$C_G(x, y) := \sum_{i=0}^{n} \sum_{j=0}^{n} c_G(i, j) x^{\underline{i}} y^j,$$

where $x^{\underline{i}} := x(x-1)\cdots(x-i+1)$ denotes the falling factorial and n is the number of vertices of G.

Definition 3 (Geometric Cover Polynomial) The geometric cover polynomial of D'Antona and Munarini [9] is defined in a similar fashion as follows:

$$C_G^{\mathrm{geo}}(x, y) := \sum_{i=0}^{n} \sum_{j=0}^{n} c_G(i, j) x^i y^j,$$

where n is the number of vertices of G.

Remark 1 When we consider the evaluation of graph polynomials like C or C^{geo} at a particular point $(\xi, \eta) \in \mathbb{Q}^2$, we will write $C(\xi, \eta)$ and $C^{\mathrm{geo}}(\xi, \eta)$ for the function that maps a given directed graph G to the value $C_G(\xi, \eta)$ and $C_G^{\mathrm{geo}}(\xi, \eta)$, respectively.

Definition 4 (Drop) For a graph $G = (V, E)$ and a permutation $\pi : V \to V$, we say that π has a drop at u if $(u, \pi(u)) \in E$ and we denote the number of permutations on G which have exactly k drops by $\left\langle {G \atop k} \right\rangle$.

Definition 5 (Drop Polynomial) The drop polynomial is defined by

$$B_G(x) = \sum_{k=0}^{n} \left\langle {G \atop k} \right\rangle \binom{x+k}{n},$$

where n is the number of vertices of G and $\binom{x+k}{n} := \frac{(x+k)^{\underline{n}}}{n!}$ is the extension of the binomial coefficient to arbitrary values x.

We will introduce two useful operations on graphs that will be used to define the so-called contraction–deletion identities:

- *Deletion:* For a graph $G = (V, E)$ and an edge $e \in E$, we define $G \setminus e := (V, E \setminus e)$.
- *Contraction:* For a graph G with an edge $e = (u, v)$, we define G/e as the graph that results from "glueing" u and v together: We replace the vertices u and v by a vertex uv and all edges of the form (w, u) now become edges of the form (w, uv) and all edges of the form (v, w') now become edges of the form (uv, w') for all vertices w and w'. Also note that edges of the form (u, w) or (w, v) are removed. (This is different to undirected graphs.) If e is a loop and thus of the form (u, u) we just delete u and all adjacent edges from the graph.

The factorial cover polynomial satisfies the following *contraction–deletion identities*:

- If e is an edge of G that is a loop, then we have

$$C_G(x, y) = C_{G\setminus e}(x, y) + yC_{G/e}(x, y),$$

- and if e is not a loop, then

$$C_G(x, y) = C_{G\setminus e}(x, y) + C_{G/e}(x, y).$$

Similar identities are known for the drop polynomial.

3 The Cover Polynomial is #P-hard for Simple Graphs

In the following, let G be a graph. Our first main theorem is the following:

Theorem 1 *Let $x, y \in \mathbb{Q} \setminus \{(0, 0), (0, -1), (1, -1)\}$. Then $C(x, y)$ is #P-hard for simple graphs.*

We will prove this by reducing the cover polynomial for (multi)graphs to the cover polynomial for simple graphs. This reduction will consist of two steps: First,

we remove loops from the graph using a function f and second, we remove parallel edges using a function g. They will be defined in their respective section.

3.1 Loops

3.1.1 The Gadget

The reduction function f maps a directed graph to a new directed graph, eliminating all loops by replacing any vertex u with a loop by three vertices $u_{\text{in}}, u', u_{\text{out}}$ and edges (u_{in}, u'), (u', u_{out}). We then add as many edges $(u_{\text{out}}, u_{\text{in}})$ as there are loops at u. Furthermore, every edge of the form (v, u) for $v \neq u$ becomes (v, u_{in}) and every edge (u, v) becomes (u_{out}, v), see Fig. 1. Intuitively, this is the opposite of contracting edges of a triangle twice: If u is a vertex in a graph G, then contracting the triangle $(f(G)/(u_{\text{in}}, u'))/(u', u_{\text{out}})$ reverses this operation locally on this single vertex. Note that we may still have parallel edges in the resulting graph. They will be removed in the next step.

3.1.2 The Properties of the Gadget

First, we consider the line $(0, y)$. On this line, path-cycle covers containing any path do not contribute to the value $C_G(0, y)$. Thus, we only need to consider cycle covers. We will show that

$$\forall y : C_G(0, y) = C_{f(G)}(0, y).$$

We will prove this by induction on the number of loops. Assume that G has at least one self-loop. Otherwise, we are done. Let G' be the graph obtained from G

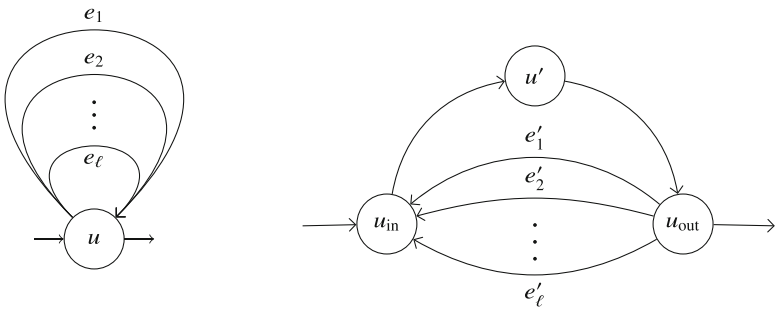


Fig. 1 A subgraph consisting of a vertex with ℓ loops and an incoming and an outgoing edge before applying f (left-hand side) and after applying f (right-hand side)

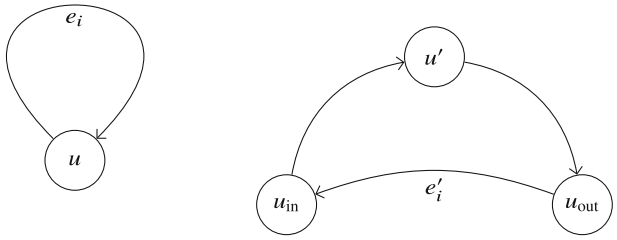


Fig. 2 A cycle cover locally covering u taking a self-loop (left-hand side) and the corresponding local covering in $f(G)$ (right-hand side)

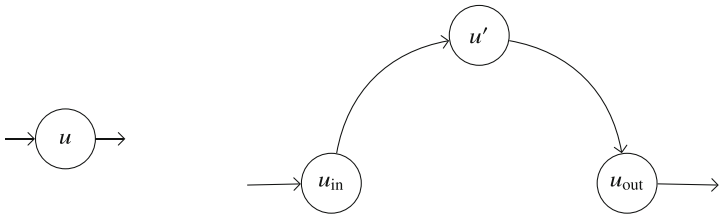


Fig. 3 A cycle cover covering u not taking a self-loop (left-hand side) and the corresponding covering for the triangle resulting from u in $f(G)$ (right-hand side)

by replacing just the self-loops of one node u by the gadget construction. Note that $f(G') = f(G)$.

Lemma 1 *There is a bijection b between cycle covers of G and cycle covers of G', such that if C is a cycle cover of G with j cycles, then $b(C)$ is a cycle cover of G' with j cycles.*

Proof Let C be a cycle cover of G with j cycles. Let u be the vertex of G that was replaced by a gadget as in Fig. 1. If u is covered by a self loop e_i in C, then we map the cover C to a cover C' of G' that contains the triangle (u_{in}, u'), (u', u_{out}), e'_i instead. See Fig. 2 for an illustration. If u is part of a larger cycle, then this larger cycle contains two edges (v, u), (u, w). ($v = w$ is possible.) To obtain C', we replace them with (v, u_{in}), (u_{in}, u'), (u', u_{out}), (u_{out}, w), as depicted in Fig. 3. This mapping is obviously injective. Moreover the mapping does not change the number of cycles.

Conversely, we have $(G'/(u_{\text{in}}, u'))/(u', u_{\text{out}}) = G$. Note that the node u' in G' can only be covered by the unique edges entering and leaving it. To close these two edges to a cycle, we either have to use one of the edges e'_i or they are part of a larger cycle that leaves through u_{out} and comes back through u_{in}. Let C' be a cycle cover with j cycles in G' and let C be the image of C' in G after contracting. In the first case u is now covered by a self-loop. In the second case, u is part of a larger cycle obtained by contracting the two edges $(u_{\text{in}}, u'))$, (u', u_{out}). In both cases the number of cycles is not changed and on all other nodes C' and C coincide. Thus the mapping has an inverse and hence is a bijection. □

Lemma 2 *There is a bijection b between cycle covers of G and cycle covers of $f(G)$, such that if C is a cycle cover of G with i cycles, then $b(C)$ is a cycle cover of $f(G)$ with i cycles.*

Proof The proof is by induction on the number k of nodes with self-loops of G.

If $k = 0$, then there is nothing to prove. If $k = 1$, then $f(G) = G'$, where G' is the graph from the previous lemma and the result follows from the previous lemma.

For the induction step, let $k \geq 2$. G' has $k - 1$ nodes with self-loops. By the induction hypothesis, there is a bijection between the cycle covers of G' and of $f(G')$ preserving the number of cycles. By the previous lemma, there is also a bijection between the cycle covers of G and of G' preserving the number of cycles. The claim now follows since $f(G) = f(G')$. □

Lemma 3 *We have $C_G(0, y) = C_{f(G)}(0, y)$ for all graphs G and $y \in \mathbb{Q}$.*

Proof Follows directly from the last lemma, as the number of cycles contained in the cycle covers is invariant under b and since b is a bijection. □

3.2 Parallel Edges

Next, we are going to define our function g which eliminates parallel edges from a graph. Let G be a directed graph with n vertices and no loops. Let $u, w \in V$ be vertices and $\ell_{u,w}$ be the number of edges from u to w. Now, we say that there are parallel edges from u to w, if $\ell_{u,w} > 1$.

3.2.1 The Gadget

Our function g maps directed graphs to directed graphs by subdividing all parallel edges e_i with a vertex v_i, adding an additional vertex b_i for each of them, the cycle $(v_i, b_i), (b_i, v_i)$ and the loop (b_i, b_i). This is done for every such instance of a parallel edge. The construction is illustrated in Fig. 4.

3.3 The Properties of the Gadget

Let G be a graph with at least one pair of nodes u, w with parallel edges from u to w and let G' be the graph obtained from G by replacing it with our gadget. Note that $g(G) = g(G')$. Let $\ell > 1$ be the number of parallel edges from u to w.

Lemma 4 *There is a bijection b between cycle covers of G and cycle covers of G', such that if C is a cycle cover of G with i cycles, then its image C' is a cycle cover of G' with $\ell + i$ cycles.*

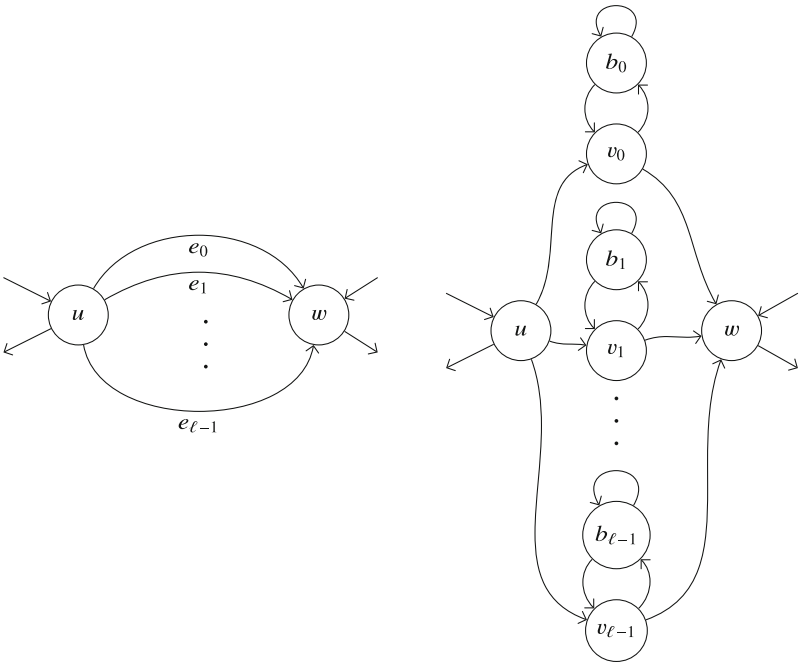


Fig. 4 A subgraph consisting of ℓ parallel edges from u to w with other incoming and outgoing edges before applying g (left-hand side) and after applying g (right-hand side). This gadget introduces loops again, but these loops are simple

Proof Let C be a cycle cover of G. If C does not contain any of the edges between u and w, we cover all the internal nodes of the gadget by the 2-cycles (b_i, v_i), (v_i, b_i) and this is the only way to cover the internal nodes, cf. Fig. 5. If C contains one of the edges between u and w, say e_i, then we connect u and w in G' via v_i, cover b_i by a self loop and the other inner nodes of the gadgets by 2-cycles as before. See Fig. 6 for an illustration. In both cases, if C has i cycles, then the new cover C' has $\ell + i$ cycles. Moreover, the mapping b is clearly injective.

Conversely, take any cycle cover C' of G'. Note that we get G back from G' by removing all nodes b_i and adjacent edges as well as contracting all edges (u, v_i) (or equivalently, all (v_i, w)). This transforms C' back into a cycle cover of G in such a way that distinct cycle covers of G' are mapped to distinct cycle covers of G. Thus the mapping b is a bijection. □

Let $k := \sum_{\substack{(u,v)\in E \\ \ell_{u,v}>1}} \ell_{u,v}$ be the total number of edges that are not simple.

Lemma 5 *There is a bijection b between cycle covers of G and cycle covers of $g(G)$, such that if C is a cycle cover of G with i cycles, then $b(C)$ is a cycle cover of $f(G)$ with $k + i$ cycles.*

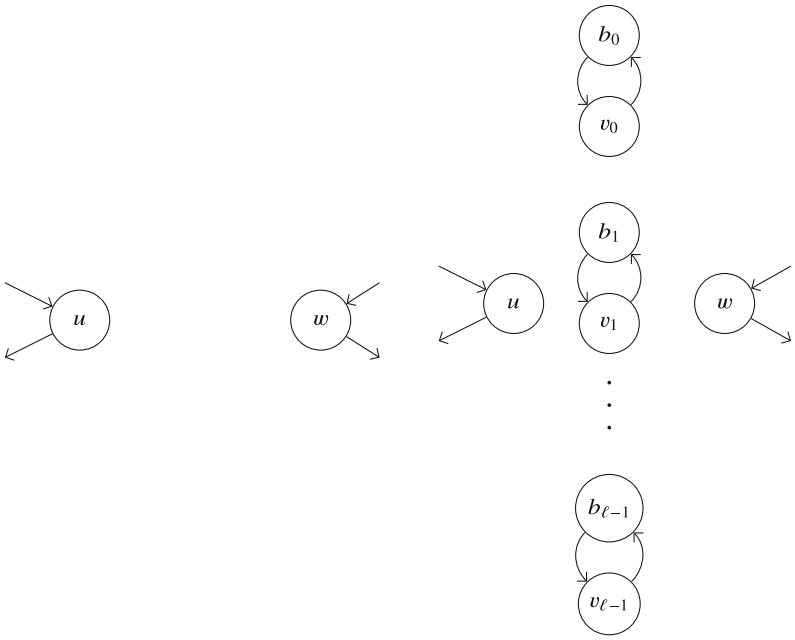


Fig. 5 A cycle cover that does not take any edges from u to w before applying g (left-hand side) and after applying g (right-hand side)

Proof The proof is by induction on k. If $k = 0$, then there is nothing to do.

Let therefore $k > 0$. Let u, w be a pair of vertices such that there are parallel edges from u to w. Let $\ell > 1$ be the number of parallel edges from u to w. Let G' be the graph obtained from G by just replacing the parallel edges between u and w by our gadget. Note that the number of parallel edges in G' is $k' := k - \ell < k$. Thus by the induction hypothesis, there is a bijection of the cycle covers of G' with i' cycles and the cycle covers of $g(G')$ with $i' + k'$ cycles. By Lemma 4, there is a bijection between the cycle covers of G with i cycles, and the cycle covers of G' with $i + \ell$ cycles. Thus there is a bijection between the cycle covers of G with i cycles and of $g(G')$ with $i + \ell + k' = i + k$ cycles. The claim follows, as $g(G') = g(G)$. □

Lemma 6 *For all $y \in \mathbb{Q}$, $y \neq 0$, and graphs G with k parallel edges in total, we get that $C_G(0, y) = \frac{1}{y^k} C_{g(G)}(0, y)$*

Proof This follows directly from Lemma 5. We get the factor $1/y^k$, because g introduces k new cycles. □

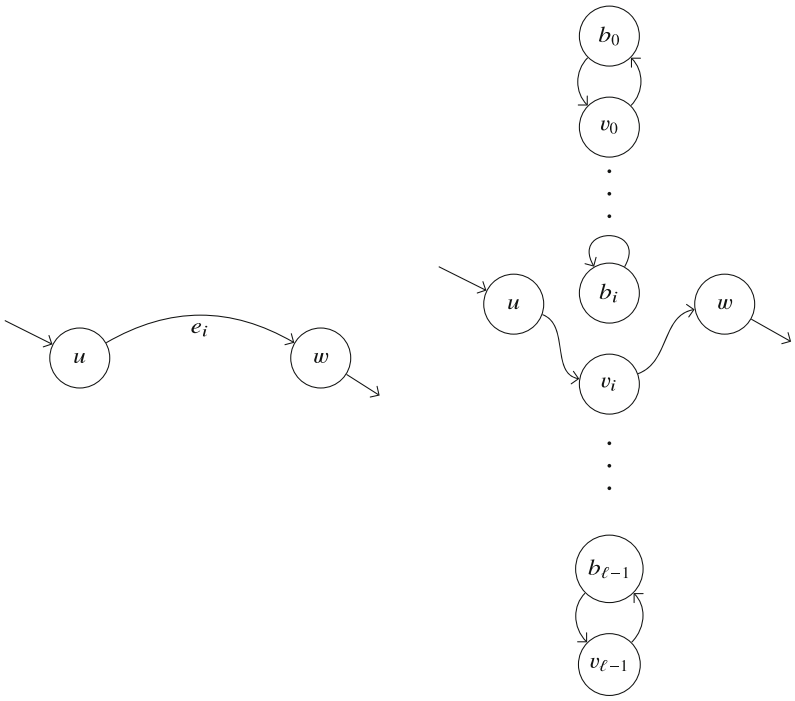


Fig. 6 A cycle cover when it uses an edge from u to w before applying g (left-hand side) and after (right-hand side)

4 Reductions

Using f and g, we get several results:

- There is a parsimonious reduction from the cover polynomial for graphs with loops to the cover polynomial for graphs without loops using f as the reduction.
- There is a many-one reduction from graphs with parallel edges to graphs without parallel edges using g as the reduction.

Lemma 7 *For all $y \in \mathbb{Q}$, $y \neq 0$, and all graphs G, there is simple graph G', such that $C_G(0, y)$ can be computed in polynomial time with one oracle call to $C_{G'}(0, y)$ and G' can be obtained from G in polynomial time. Moreover, if G is planar, then G' is also planar.*

Proof Let $y \in \mathbb{Q}$. The reduction is given by $f \circ g \circ f$: First we get rid of the loops by applying f. This reduction results in a graph with parallel edges, if G has nodes with parallel self-loops. Then we remove the parallel edges using g. Here, g introduces self-loops, but at most one per node. Therefore, the second application of f removes these self-loops without creating new parallel edges and the resulting graph is simple.

If G was planar, then every replacement made by f and g is planar locally and thus also globally.

Let G be a graph and $K := |\{(u, w) \in E \mid \ell_{u,w} > 1\}|$ be the number of edges with at least two parallel edges in G. Using our lemmas, we get that

$$C_{f(g(f(G)))}(0, y) = C_{g(f(G))} = y^K C_{f(G)}(0, y) = y^K C_G(0, y),$$

So, we can construct G' by applying f, then g, then again f, so $G' = f(g(f(G)))$.

Now, we can compute $C_G(0, y)$ as follows: Given a graph, count the number of ordered pairs (u, v) with $\ell_{u,v} > 1$ and call this number K. Then, compute $C_{f(g(f(G)))}(0, y)$ with the oracle and output the result divided by y^K. As this is obviously polynomial time bounded and we only query a single time, we have a polynomial-time many-one reduction. □

Now, in order to prove our main theorem, we need to look at path-cycle covers instead of cycle-covers: Using a result by Chung and Graham [7], we get the horizontal reduction as in [3]: For a graph G, let $G^{(r)}$ be the graph obtained by adding r isolated vertices to G. Then, we get that

$$C_{G^{(r)}}(x, y) = x^r C_G(x - r, y).$$

Using this, we can prove the last lemma needed, as the graph used for the oracle query is (planar and) simple if the given graph is (planar and) simple:

Lemma 8 *For all $x, y \in \mathbb{Q}$, if $x \in \mathbb{N}$ then we have $C(x', y) \leq_P^m C(x, y)$ for all $x' \leq x$ with $x' \in \mathbb{N}$. Otherwise, we have $C(x', y) \leq_P^T C(x, y)$ for all $x' \in \mathbb{Q}$.*

A proof of this reduction can be found in [3]. Now we come to the proof of one of our main results, the hardness of the cover polynomial for planar simple graphs.

Theorem 2 *Evaluating the cover polynomial $C(x, y)$ is #P-hard for planar simple graphs except for $\{(x, y) \mid x = y = 0 \vee y = -1\}$.*

Proof From [2], we know that evaluating $C(0, y)$ for $y \in \mathbb{Q} \setminus \{0, -1\}$ is #P-hard for planar graphs G.

By Lemma 7, there is a planar simple graph G', such that $C_G(0, y)$ can be computed given $C_{G'}(0, y)$. Thus, the #P-hardness carries over to planar simple graphs.

Using the reduction from Lemma 8, it follows that $C(x, y)$ is #P-hard for planar simple graphs for $y \notin \{0, -1\}$.

If $y = 0$, we know that $C(1, 0)$ counts the number of Hamiltonian paths, which is #P-hard even for planar graphs [14]. Using the last Lemma 8 again, we can stretch this hardness to $C(x, 0)$ with $x \neq 0$. □

For graphs, which are not planar, we can get a complete hardness classification.

Lemma 9 *There is a reduction $C(1, 0) \leq_P^T C(2, -1)$ that maps simple graphs to simple graphs.*

A proof of this reduction can be found in [3]. It is easy to verify that the resulting graph is simple if the original graph was simple. This proves Theorem 1, stated at the beginning of Sect. 3.

5 Consequences for Related Polynomials

We have seen that the cover polynomial is #P-hard for planar simple graphs. We can translate these results to the drop polynomial as well as to the geometric cover polynomial.

5.1 Translation of the Hardness to the Drop Polynomial

There is a 2-variable drop polynomial $B_G(x, y)$ that generalizes the univariate drop polynomial $B_G(x)$ that we defined earlier. It has two very interesting properties:

1. $B_G(x) = B_G(x, 1)$, that is, it is indeed a generalization of $B_G(x)$.
2. For all simple graph G, we have that $B_G(x, y) = C_G(x, y)$ for all x, y.

Using both of these properties, we get that $B_G(x) = B_G(x, 1) = C_G(x, 1)$ for simple graphs G. By Theorem 2, we know that the cover polynomial is #P-hard for planar simple graphs except for $\{(x, y) \mid x = y = 0 \vee y = -1\}$. Note that this set and the line $y = 1$ are disjoint, so they will not be relevant for the drop polynomial. Using this and the equality above, this property also follows for the drop polynomial:

Corollary 1 *The drop polynomial is #P-hard everywhere for simple graphs as well as for planar simple graphs.*

5.2 Translation of the Hardness to the Geometric Cover Polynomial

We also get similar results for the geometric cover polynomial, as $C^{\text{geo}}(0, y) = C(0, y)$ for all $y \in \mathbb{Q}$. We have the following horizontal reduction from [6]: For a graph G, the α-thickening $G^{\alpha\text{-thick}}$ is the graph obtained by replacing each edge with α copies, $\alpha \in \mathbb{N} \setminus \{0\}$.

Lemma 10 *For all $x, y \in \mathbb{Q}$ and $\alpha \in \mathbb{N} \setminus \{0\}$, $C^{\text{geo}}_{G^{\alpha\text{-thick}}}(x, y) = \alpha^n C^{\text{geo}}_G(x/\alpha, y)$.*

One might be tempted to use our gadget from Fig. 4 to remove the parallel edges introduced by the thickening. However, this does not seem to work, since our analysis uses the fact that the given cover only has cycles. If there are also paths,

then the local covering of the gadget depends on whether there ends a path in u (or whether there starts a path in w).

However, [2, Cor. 1] shows directly that $C^{\text{geo}}(x, y)$ is #P-hard for all $x \neq 0$, even for simple planar graphs. Thus, we get a full classification together with our results for the line $x = 0$ for the factorial cover polynomial, since $C^{\text{geo}}(0, y) = C(0, y)$ and $C(0, y)$ is #P-hard for (planar) simple graphs on the line $x = 0$.

Corollary 2 *For* $(x, y) \in \mathbb{Q}^2 \setminus \{(0, 0), (0, -1)\}$, $C^{\text{geo}}(x, y)$ *is* #P*-hard for planar simple graphs.*

References

1. Averbouch, I., Godlin, B., Makowsky, J.A.: An extension of the bivariate chromatic polynomial. Eur. J. Comb. **31**(1), 1–17 (2010)
2. Bläser, M., Curticapean, R.: The complexity of the cover polynomials for planar graphs of bounded degree. In: Murlak, F., Sankowski, P. (eds.) Mathematical Foundations of Computer Science 2011, pp. 96–107. Springer Berlin Heidelberg, Berlin/Heidelberg (2011)
3. Bläser, M., Dell, H.: Complexity of the cover polynomial. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) Automata, Languages and Programming, pp. 801–812. Springer Berlin Heidelberg, Berlin/Heidelberg (2007)
4. Bläser, M., Hoffmann, C.: On the complexity of the interlace polynomial. In: Albers S., Weil, P. (eds.) STACS 2008, 25th Annual Symposium on Theoretical Aspects of Computer Science, Bordeaux, France, February 21-23, 2008, Proceedings. LIPIcs, vol. 1, pp. 97–108. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany (2008)
5. Bläser, M., Dell, H., Makowsky, J.A.: Complexity of the Bollobás-Riordan polynomial. Exceptional points and uniform reductions. Theory Comput. Syst. **46**(4), 690–706 (2010)
6. Bläser, M., Dell, H., Fouz, M.: Complexity and approximability of the cover polynomial. Comput. Complex. **21**(3), 359–419 (2012)
7. Chung, F.R.K., Graham, R.L.: On the cover polynomial of a digraph. J. Comb. Theory Series B **65**(2), 273–290 (1995)
8. Chung, F., Graham, R.: The drop polynomial of a weighted digraph. J. Comb. Theory Series B **126**, 62–82 (2017)
9. D'Antona, O.M., Munarini, E.: The cycle-path indicator polynomial of a digraph. Adv. Appl. Math. **25**(1), 41–56 (2000)
10. Ellis-Monaghan, J.A., Moffatt, I. (eds.): Handbook of the Tutte Polynomial and Related Topics, 1st edn. Chapman and Hall/CRC, Boca Raton (2022)
11. Goodall, A.J., Hermann, M., Kotek, T., Makowsky, J.A., Noble, S.D.: On the complexity of generalized chromatic polynomials. Adv. Appl. Math. **94**, 71–102 (2018)
12. Hoffmann, C.: A most general edge elimination polynomial – Thickening of edges. Fundam. Informaticae **98**(4), 373–378 (2010)
13. Jaeger, F., Vertigan, D.L., Welsh, D.J.A.: On the computational complexity of the Jones and Tutte polynomials. Math. Proc. Camb. Philos. Soc. **108**, 35–53 (1990)
14. Liskiewicz, M., Ogihara, M., Toda, S.: The complexity of counting self-avoiding walks in subgraphs of two-dimensional grids and hypercubes. Theor. Comput. Sci. **304**(1–3), 129–156 (2003)
15. Makowsky, J.A.: From a zoo to a zoology: towards a general theory of graph polynomials. Theory Comput. Syst. **43**(3–4), 542–562 (2008)

Polynomial Threshold Functions of Bounded Tree-Width: Some Explainability and Complexity Aspects

Karine Chubarian, Johnny Joyce, and György Turán

Dedicated to Janos Makowsky on the occasion of his 75th birthday.

Abstract The tree-width of a multivariate polynomial is the tree-width of the hypergraph with hyperedges corresponding to its terms. Multivariate polynomials of bounded tree-width have been studied by Makowsky and Meer as a new sparsity condition that allows for polynomial solvability of problems which are intractable in general. We consider a variation on this theme for Boolean variables. A representation of a Boolean function as the sign of a polynomial is called a polynomial threshold representation. We discuss Boolean functions representable as polynomial threshold functions of bounded tree-width and present two applications to Bayesian network classifiers, a probabilistic graphical model. Both applications are in Explainable Artificial Intelligence (XAI), the research area dealing with the black-box nature of many recent machine learning models. We also give a separation result between the representational power of positive and general polynomial threshold functions.

1 Introduction

The tree-width of a multivariate polynomial is the tree-width of the hypergraph with hyperedges corresponding to its terms. *Multivariate polynomials of bounded tree-width* have been studied by Makowsky and Meer [32, 33] as a new sparsity

K. Chubarian · J. Joyce
University of Illinois at Chicago, Chicago, IL, USA
e-mail: kchuba2@uic.edu; jjoyce22@uic.edu

G. Turán (✉)
University of Illinois at Chicago, Chicago, IL, USA

HUN-REN-SZTE Research Group on AI, Szeged, Hungary
e-mail: gyt@uic.edu

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_11

condition that allows for polynomial solvability of problems which are intractable in general. The paper is closely related to Courcelle et al. [7] and to Fischer et al. [14]. The topics covered are *polynomials, graphs, widths, counting* and *logic*. We discuss a variation on this theme, in the context of Boolean functions. Polynomials, graphs, widths, counting and (propositional) logic all appear, although the context is different. The problems considered are from Explainable Artificial Intelligence (XAI) and complexity theory.

Boolean functions $f : \{0,1\}^n \rightarrow \{0,1\}$ can be represented by multivariate polynomials exactly or approximately in a many different ways over $GF(2)$ or the reals. In this paper we consider polynomials over the reals. As $x^2 = x$ for 0 and 1, polynomials can be assumed to be multilinear. Let

$$p(x_1, \ldots, x_n) = \sum_{I \in \mathcal{I}} \beta_I x_I \tag{1}$$

be a multilinear polynomial, where $\mathcal{I}$ is a family of subsets of $[n]$, $\beta_I \in \mathbb{R}$ and $x_I = \prod_{i \in I} x_i$. The degree of the polynomial is the maximal number of variables in a term, and its size is the number of terms. One can consider the Boolean function

$$\text{sgn}(p(x_1, \ldots, x_n)) : \{0,1\}^n \rightarrow \{0,1\},$$

where sgn is the sign function (1 for nonnegative values and 0 otherwise). For example,

$$\text{sgn}(x_1 + x_2 + x_3 + x_4 - x_1x_2 - x_3x_4 - 2) \tag{2}$$

represents the 4-variable Boolean function which has value 1 iff (x_1, x_2, x_3, x_4) contains at least three 1s, or both (x_1, x_2) and (x_3, x_4) contain exactly one 1.

Every Boolean function can be represented in this form, called a *polynomial threshold function (PTF) representation* (see Sect. 2.1). The smallest degree and size of polynomials representing a Boolean function are important measures of its complexity. A Boolean function is a degree-d polynomial threshold function if it can be written as the sign of a degree-d polynomial. Linear threshold functions (LTF or perceptrons) played an important role since the beginning of machine learning.

We consider the *minimal tree-width of polynomial sign-representations* as a measure of complexity of a Boolean functions. Consider a polynomial p as in (1). The *term-hypergraph* H_p of polynomial p has vertex set $[n]$ and edge set $\mathcal{I}$. The tree-width of polynomial p is the tree-width of its term-hypergraph H_p. A Boolean function is a tree-width-k polynomial threshold function if it can be written as the sign of a tree-width-k polynomial. Thus, for example, the Boolean function (2) has tree-width 1.

Polynomial threshold representations are useful in many different areas. One of these is a probabilistic graphical model, the *Bayesian network classifier (BNC)* [15], a generalization of the Naive Bayes classifier. A Bayesian network classifier can

be represented by a polynomial threshold function, where the terms reflect the graphical structure of the network (see Sect. 3). Polynomials representing the Naive Bayes classifier are linear. *Tree Augmented Bayesian network classifiers (TAN)* have additional edges forming a forest [15].[1] They correspond to quadratic polynomial threshold functions (QTF), where the quadratic terms correspond to the edges of the forest. Bayesian network classifiers with a bounded tree-width network structure correspond to polynomials of bounded tree-width. Bounded tree-width Bayesian networks form a tractable subclass for several inference and learning problems [8, 24]. In this paper we consider some aspects of Bayesian network classifiers related to their explainability.

Explainability Recent machine learning models, in particular deep neural networks, are often black boxes in the sense that they do not provide an explanation for the output produced, and this hinders the development of trustworthy AI. Standard examples are that a rejected loan applicant needs to know the reason for the rejection, and a physician needs to know reasons for a suggested diagnosis. The need for explanations has a long history in machine learning,[2] but the recent developments amplified the relevance of this issue and brought about Explainable AI as a separate field of research [35].

How to define what is an explanation is a difficult question, which has a long history in the philosophy of science. The type of explanation relevant in a particular machine learning application is context-dependent. A *post-hoc* explanation provides an explainable model[3] corresponding (exactly or approximately) to a black-box model. Post-hoc explanations can be *global* or *local*, depending on whether the whole model is explained or only its behavior on a specific input. Even if there is an agreed upon notion of explanation, it is usually not clear how to *evaluate* the explanations provided, one reason being the lack of *ground truth* (e.g., what is the reason for classifying an image as a cat?).

A neural network is usually considered non-explainable and a decision tree is usually considered explainable (but see, for example, [28] for a discussion of this distinction). Probabilistic graphical models are somewhere in between, having both explainable and non-explainable features. A Bayesian network, on one hand, gives explicit graphical information about conditional independencies. On the other hand, probabilistic inference is hard, and usually not explainable for the user. This holds even for the Naive Bayes classifier. Therefore, explaining Bayesian networks has been studied for a long time [27].

[1] Friedman et al. [15] considers trees only, but here one can use forests as well.

[2] For example, the MYCIN expert system contained an "Explanation System, which understands simple English questions and answers them in order to justify its decisions or instruct the user" [45].

[3] In the literature an explainable model, such as a decision tree, is usually referred to as an interpretable model. As interpretations are used in logic in a different sense, and also for simplifying terminology, we use explainability for this purpose as well.

Explaining Bayesian Network Classifiers by Approximate OBDD A post-hoc approach to global explanations for Naive Bayes classifiers, proposed by [5], is to compile them into *Ordered Binary Decision Diagrams (OBDD)* [48]. This is an instance of *knowledge compilation*, transforming a knowledge representation into another which is more suitable for a given application [10]. Explainability aspects of knowledge compilation are considered in [12].

Besides being explainable in the local sense similarly to decision trees, OBDDs can also be considered as global explanations, as they provide a useful data structure for Boolean functions. There are efficient algorithms for deciding their properties and performing operations on them that are relevant for propositional reasoning about a model [9, 31, 48]. Thus, if a system has a component which computes a Boolean function, then having an OBDD representing this function can be useful for reasoning about how this component works as part of the system. In this sense, OBDD is a "reasonable" representation.

It was shown in [21] that linear threshold functions require exponential size OBDDs in the worst case, so such compilation algorithms have to be exponential. Therefore, for efficient compilation approximations need to be considered. As learned models are supposed to be approximations, this requirement seems natural.

In [6] we gave an approximation scheme compiling Bayesian network classifiers of bounded tree-width into OBDD. Bounded tree-width is well-known to be a useful parameterization for efficient computation. Here we consider its relevance for finding explanations efficiently, which is a different aspect. An outline of the algorithm is given in Sect. 4. From the point of view of knowledge representation in AI, the result is an example of *approximate knowledge compilation*. Negative results for approximate knowledge compilation are given in [13].

The algorithm is based on first translating the classifier into a PTF of bounded tree-width, and then translating this PTF into an approximate OBDD. The second step generalizes the approximate deterministic counting algorithm of [16] (an FPTAS, fully polynomial-time approximation scheme) for the knapsack solution counting problem, corresponding to the LTF case. Error is measured with respect to the input distribution generated by the classifier, and not the uniform distribution.

Deterministic approximation algorithm schemes for counting the number of satisfying truth assignments for a class of Boolean functions work as follows. Given an $\epsilon > 0$ and a function f from the class, an estimate is computed for the probability $P(f(x) = 1)$, where x is uniformly distributed over $\{0, 1\}^n$. *Multiplicative* (resp., *additive*) approximation algorithms compute an estimate within a multiplicative error $1 \pm \epsilon$ (resp., an additive error $\pm\epsilon$). If the satisfiability problem for the class (i.e., deciding whether there is an x such that $f(x) = 1$) is NP-complete then, assuming $P \neq NP$, polynomial time multiplicative approximation algorithms cannot exist, as those could be used to solve the satisfiability problem in polynomial time. Hence in such cases additive approximation has to be considered. In contrast to the linear case, for degree at least 2 the PTF satisfiability problem is NP-complete, so only additive approximation algorithms can be hoped for (see [42], and [11] for the QTF case). Tree-width provides a different parameterization.

Explainability Evaluation for Generalized Additive Models with Interactions In Sect. 5 we discuss another application of polynomial threshold functions to explainability. A *generalized additive model (GAM)* is a generalization of logistic regression. It can be extended further to include interaction terms between the variables. InterpretML [36] contains an implementation of the GA^2M algorithm of [30], which is applied, for example, in the medical domain [4]. Its accuracy is competitive with more complex models, and it is also explainable in the practical, informal sense that the interaction terms are often meaningful for medical experts.

We discuss an experiment aimed at a *qualitative evaluation* of the explainability of the GA^2M algorithm. Bayesian network classifiers provide a class of instances with ground truth. The applicability of BNC in this context is due to the connection between BNC and polynomial threshold functions mentioned above. This connection implies that a BNC can be viewed as a generalized additive model with interactions, with the structure of the network corresponding to interaction terms. The ground truth is structured in the sense that terms not included in the basic representation also have a probabilistic meaning in the network. The basic setup of GA^2M is to handle pairwise interactions, therefore so far we have only considered TAN in our experiments.

We present experimental results for a small synthetic example, using InterpretML [36]. BNC is a generative model, so one can train a GA^2M on random examples generated from the input distribution of the BNC. The ground truth given by the polynomial threshold function representing the classifier can be compared with the polynomial produced by the GA^2M in different ways, perhaps the simplest being comparing the terms in the ground truth and learned polynomials. Besides achieving almost optimal accuracy on the example, the learned terms show a good correspondence to the ground truth.

Positive Polynomial Threshold Functions A main research direction of computational complexity theory is to compare the computational power of various models of computation.

A Boolean circuit computes a Boolean function using $\wedge$, $\vee$ and $\neg$ gates. The circuit is *monotone* if it contains only $\wedge$, $\vee$ gates and no negation. Such circuits can compute all monotone Boolean functions and only those. Comparing the computational power of monotone and general circuits and other computational models has been studied in complexity theory for a long time [39]. The problem is also referred to as "monotone versus positive" ([1], where it is shown that Lyndon's theorem fails for finite models by a circuit lower bound). Superpolynomial separations between monotone and general circuits were given in [40, 41], strengthened to exponential separation in [46]. A survey of results on monotone circuit complexity is given in [22].

As far as we know the related problems for PTF have not been considered before. A PTF is *positive* if it is of the form $p(x_1, \ldots, x_n) \geq t$, where p is a polynomial with nonnegative coefficients and no constant term, and t is nonnegative. Positive PTF represent monotone functions and only those, so one can ask about comparing the representational power of the positive and general versions for monotone Boolean

functions. In Sect. 6 we give an example of a QTF of linear size (and tree-width 1) for which every positive QTF has size $\Omega(n^2)$ (and thus tree-width $\Omega(n)$). There are several open problems in this direction and we mention some of those as well.

2 Preliminaries

In this section we give some background on Boolean functions, the notions of width considered and Bayesian network classifiers.

2.1 Boolean Functions

A Boolean function is of the form $f : \{0,1\}^n \to \{0,1\}$. For $a = (a_1, \ldots, a_n) \in \{0,1\}^n$ consider the polynomial $p_a(x) = \prod_{i=1}^n I_{a_i}(x_i)$, where for $\ell \in \{0,1\}$

$$I_\ell(z) = \begin{cases} z & \text{if } \ell = 1 \\ 1 - z & \text{if } \ell = 0 \end{cases} \tag{3}$$

over the reals. Then $p_a(a) = 1$ and $p_a(b) = 0$ for every $b \in \{0,1\}^n$ different from a. Thus $p_f(x) = \sum_{a\,:\,f(a)=1} p_a(x)$ is an exact polynomial representation of f, which is in fact unique among multilinear polynomials.

The polynomial $p_f(x) - 1/2$ is a polynomial sign-representation of f. Thus every Boolean function has a sign-representation of degree at most n and size at most 2^n. Random Boolean functions require linear degree and exponential size [38].

Truth values can also be represented by $\{\pm 1\}$ instead of $\{0,1\}$. In this case $x^2 = 1$ and so polynomials can be assumed to be multilinear as well.[4] The linear mapping $x_i' = 1 - 2x_i$ for $i = 1, \ldots, n$ provides a transformation between the two representations. The unique $\{\pm 1\}$ multilinear representation of a Boolean function is its Fourier representation [37]. Comparing both exact and polynomial threshold representations, degrees are preserved, but the number of terms changes. The parity function $x_1 \oplus \ldots \oplus x_n$ (where $\oplus$ is addition in $GF(2)$) requires exponentially many terms over $\{0,1\}$ even for a polynomial threshold representation [26], but it is simply the product over $\{\pm 1\}$. Being a tree-width-k polynomial threshold function is the same property over $\{0,1\}$ and $\{\pm 1\}$. The transformations between the two representations do not increase the tree-width of the term-hypergraph, as every new edge is a subset of an already existing edge.

[4] It is also of interest to consider other truth values, e.g. $\{1, 2\}$ (see [18, 19]), and then degrees can matter.

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is monotone if $x \leq y$ implies $f(x) \leq f(y)$, where $x = (x_1, \ldots, x_n) \leq y = (y_1, \ldots, y_n)$ iff $x_i \leq y_i$ for every i. A PTF $p(x) \geq t$ is *positive* if every coefficient of p is nonnegative, p has no constant term, and t is also nonnegative. A Boolean function is monotone iff it has a positive PTF representation. Every monotone function can be written as a monotone DNF [22]; replacing conjunctions with products, disjunctions with sums and using threshold $1/2$ gives a positive PTF. The other direction follows directly from the definitions.

2.2 Widths

For an undirected graph $G = (V, E)$, a *tree-decomposition* of G is given by a tree T and bags $B_t \subseteq V$ for every vertex t of T, such that for every $(u, v) \in E$ there exists t such that $u, v \in B_t$ and for every $v \in V$ the vertices t such that $v \in B_t$ form a subtree of T. The *width* of a tree decomposition is $\max_{t \in V(T)} |B_t| - 1$ and the *tree-width* $\mathsf{tw}(G)$ of G is the minimal width of tree-decompositions of G. The *path-width* $\mathsf{pw}(G)$ of G is defined by trees restricted to paths. There are many other notions of width, including those for directed graphs and hypergraphs [20].

The *moral graph* of a directed acyclic graph (DAG) G is the undirected graph $\mathsf{MG}(G)$ obtained from G by adding undirected edges between co-parents and disregarding the direction of the original directed edges. The tree-width of G is $\mathsf{tw}(G) = \mathsf{tw}(\mathsf{MG}(G))$ and its path-width is $\mathsf{pw}(G) = \mathsf{pw}(\mathsf{MG}(G))$. This definition is motivated by its use in probabilistic inference in Bayesian networks [8, 24].

The *primal graph* $\mathsf{PG}(H)$ of a hypergraph $H = (V, E)$ with $E \subseteq P(V)$ is the undirected graph obtained from H by replacing every hyperedge by a clique. The tree-width of H is $\mathsf{tw}(H) = \mathsf{tw}(\mathsf{PG}(H))$ and its path-width is $\mathsf{pw}(H) = \mathsf{pw}(\mathsf{PG}(H))$.

The *term-hypergraph* H_p of the PTF p has vertex set $[n]$ and edge set $\mathcal{I}$. The tree-width of a PTF is the tree-width of its term-hypergraph, and similarly for path-width.

2.3 Bayesian Network Classifiers

We use X_i, x_i, resp., a_i, to denote binary random variables, Boolean variables, resp., Boolean constants. The restriction of a vector $x = (x_1, \ldots, x_n)$ to a subset I of its coordinates is denoted by x^I.

A *Bayesian network classifier (BNC)* N is a DAG G_N over binary *input variables* $X_1, \ldots, X_n$ and a binary *classifier variable* C, with local conditional probabilities given for each vertex. These specify the conditional probability distribution of the variable corresponding to that vertex, depending on the values of its parents in the

DAG. It is assumed that (C, X_i) is an edge for every $i = 1, \ldots, n$. The set of additional edges is denoted by E_N. In a *Naive Bayes Classifier* $E_N = \emptyset$. In a *Tree Augmented Naive Bayes Classifier (TAN)* E_N is an in-forest, i.e., every node has at most one parent. Figure 2 shows a TAN with the specifications of the DAG and the conditional probabilities.

The BNC generates a probability distribution over the random variables C and $X_1, \ldots, X_n$ as follows. First a value is generated for C according to its distribution (with no parents in this case), and then values are generated for $X_1, \ldots, X_n$ proceeding down the DAG, always using the conditional probability distribution corresponding to the values generated for the parents. For the TAN of Fig. 2 the random values are generated moving down the tree.

Let $\Pi_i = \{j : X_j \text{ is a parent of } X_i\}$ be the set of parents of X_i other than C. The *family* of i is $\{i\} \cup \Pi_i$. Let $d_N = 1 + \max_i |\Pi_i|$ be the maximal size of families in G_N, referred to as the *degree* of the Bayesian network. The tree-width of N is $\mathsf{tw}(N) = \mathsf{tw}(\mathsf{MG}(G_N \setminus \{C\}))$, its path-width is $\mathsf{pw}(N) = \mathsf{pw}(\mathsf{MG}(G_N \setminus \{C\}))$.

The local conditional probabilities are

$$p_c^0 = P_N(C = c), \qquad p^i_{(a_i, a_{\Pi_i}, c)} = P_N(X_i = a_i \mid X_{\Pi_i} = a_{\Pi_i}, C = c),$$

for $i = 1, \ldots, n$.

The joint distribution of the variables is

$$P_N(X_1 = a_1, \ldots, X_n = a_n, C = c) = p_c^0 \prod_{i=1}^{n} p^i_{(a_i, a_{\Pi_i}, c)}. \tag{4}$$

The marginal distribution over the input variables, also referred to as the *input distribution*, is

$$P_{N,X}(X_1 = a_1, \ldots, X_n = a_n) = \sum_{c=0}^{1} P_N(X_1 = a_1, \ldots, X_n = a_n, C = c).$$

The Bayesian network classifier corresponding to N is a Boolean function $f_N(x_1, \ldots, x_n)$ where $f_N(a_1, \ldots, a_n) = 1$ iff

$$P_N(a_1, \ldots, a_n, 1) \geq P_N(a_1, \ldots, a_n, 0).$$

Bayesian network inference and learning problems have been studied in great detail (see [8, 24]). In this paper we consider some explainability aspects.

3 Bayesian Network Classifiers and Polynomial Threshold Functions

In this section we formulate the representation of BNC as PTF, which is used in the next two sections. A unified presentation of work on this connection, going back to [34], is given in [47].

Proposition 1 *Let N be a Bayesian network classifier with non-zero conditional probabilities. Then there is a polynomial p such that*

$$\log \frac{P_N(C=1 \mid X=x)}{P_N(C=0 \mid X=x)} = p(x),$$

where p is of degree at most d_N and every term is a subset of a family. Thus f_N is a PTF of degree at most d_N.

Proof It holds that

$$P_N(X_i = x_i \mid X_{\Pi_i} = x_{\Pi_i}, C = c) = \prod_{(a_i, a_{\Pi_i})} p^{i}_{(a_i, a_{\Pi_i}, c)}{}^{I_{a_i}(x_i) \prod_{j \in \Pi_i} I_{a_j}(x_j)}.$$

From (4) we get

$$\begin{aligned} &P_N(X_1 = x_1, \ldots, X_n = x_n, C = c) \\ &\quad = p^0_c \prod_{i=1}^{n} \prod_{(a_i, a_{\Pi_i})} \left(p^{i}_{(a_i, a_{\Pi_i}, c)} \right)^{I_{a_i}(x_i) \prod_{j \in \Pi_i} I_{a_j}(x_j)}. \end{aligned}$$

Then for $x = (x_1, \ldots, x_n)$ it holds that

$$\frac{P_N(C=1 \mid X=x)}{P_N(C=0 \mid X=x)} = \frac{p^0_1}{p^0_0} \prod_{i=1}^{n} \prod_{(a_i, a_{\Pi_i})} \left(\frac{p^{i}_{(a_i, a_{\Pi_i}, 1)}}{p^{i}_{(a_i, a_{\Pi_i}, 0)}} \right)^{I_{a_i}(x_i) \prod_{j \in \Pi_i} I_{a_j}(x_j)}.$$

Taking logarithms

$$\begin{aligned} \log \frac{P_N(C=1 \mid X=x)}{P_N(C=0 \mid X=x)} = \log \frac{p^0_1}{p^0_0} + \sum_{i=1}^{n} \sum_{(a_i, a_{\Pi_i})} \\ \times \log \left(\frac{p^{i}_{(a_i, a_{\Pi_i}, 1)}}{p^{i}_{(a_i, a_{\Pi_i}, 0)}} \right) I_{a_i}(x_i) \prod_{j \in \Pi_i} I_{a_j}(x_j). \end{aligned}$$

□

Corollary 1 *For a TAN with non-zero conditional probabilities f_N is a QTF of tree-width 1, with quadratic terms corresponding to E_N.*

Corollary 2 *For BNC of tree-width k with non-zero conditional probabilities f_N is a PTF of path-width $O(k \log n)$.*

Proof For every edge of the primal graph of the term-hypergraph H_p for the PTF constructed above there is a family of N containing that edge, which then belongs to the moral graph of the classifier. The statement then follows from the fact that $\mathsf{pw}(G) = O(\mathsf{tw}(G) \log n)$ for undirected graphs [25]. □

The primal graph may be a proper subset of the moral graph due to cancellations.

4 Approximating Bayesian Network Classifiers with OBDD

In this section we show that Bayesian network classifiers of bounded tree-width can be approximated by polynomial size OBDD. We first introduce OBDD, then formulate the result and outline its proof.

4.1 Ordered Binary Decision Diagrams

An *ordered binary decision diagram (OBDD)* over Boolean variables $x_1, \ldots, x_n$ computes a Boolean function f. An OBDD is a DAG with two sinks labeled 0 and 1, and the other nodes labeled with variables. The DAG is assumed here to be layered, with directed edges going from a layer to the next layer, and sinks on the last layer. There are $n+1$ layers and a permutation $\pi(i)$ of $[n]$ such that nodes on the i'th layer are labeled with variable $x_{\pi(i)}$. On the first layer there is a single start node labeled $x_{\pi(1)}$. Every non-sink node has two outgoing edges, labeled with 0, resp., 1. For every truth assignment $a = (a_1, \ldots, a_n)$, $f(a)$ is the label of the sink reached by following edge labels corresponding to the bits in a, evaluated in the order given by the labels of the layers. The *width* of an OBDD is the maximal number of nodes in a layer.[5]

Given an ordering π of the variables, every Boolean function has a unique minimal layered OBDD, where every node of the i'th layer correspond to a subfunction obtained by fixing the variables above the layer. This can further be simplified to the *reduced* OBDD, where edges can "jump" layers, by deleting nodes with both outgoing edges leading to the same successor node, and redirecting their incoming edges to the successor node. The left side of Fig. 1 shows the minimal

[5] Width is a relevant parameter for the computational power of OBDD: there is a jump at width 5 [3].

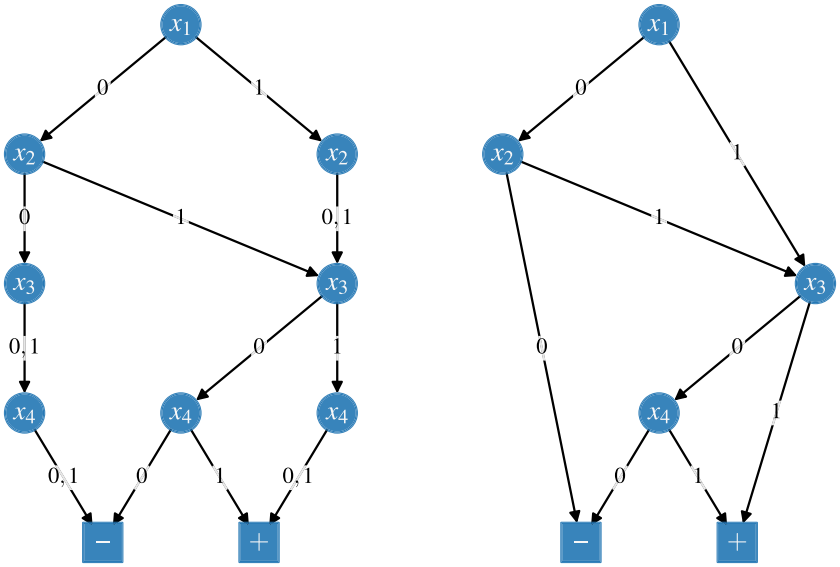


Fig. 1 OBDDs for function (2). (**a**) Minimal layered OBDD. (**b**) Reduced OBDD

layered OBDD of the function (2) and the right side shows its reduced OBDD, with respect to the identity permutation.

A *generator OBDD (GOBDD) D* generates a probability distribution over $\{0,1\}^n$. It is similar to a layered OBDD, except edges are also labeled with probabilities, and there is a single sink. A probability p_u is associated with every non-sink vertex u, and the 0-edge (resp. 1-edge) leaving u is labeled $p_u^0 = p_u$ (resp., $p_u^1 = 1 - p_u$). For every truth assignment $a = (a_1, \ldots, a_n)$, the GOBDD determines a path from the source to the sink, and $P_D(a)$ is the product of edge probabilities along the path. The *width* of a GOBDD is the width of the underlying OBDD. The *width* of a distribution is the minimal width of GOBDDs generating it. Product distributions have width one.

4.2 Result and Proof Outline

We now state the result on approximating Bayesian network classifiers of bounded tree-width with OBDD over the input distribution of the classifier. As discussed in the introduction, the goal is to provide a "small" approximate representation of bounded tree-width Bayesian network classifiers which allows for efficient reasoning. Besides the parameters discussed earlier, the bounds also depend on the bit-precision q of the conditional probabilities.

Theorem 2 ([6]) *For every Bayesian network classifier of tree-width k having n Boolean variables and q-bit non-zero conditional probabilities, and every $\varepsilon > 0$ there is an OBDD of size* $\mathsf{poly}\left(n^k, q, 1/\varepsilon\right)$ *approximating the classifier with multiplicative error at most ε with respect to the input distribution of the classifier. The OBDD can be constructed in time* $\mathsf{poly}\left(n^k, q, 1/\varepsilon\right)$.

Proof Let N be a BNC with the properties given in the theorem. Consider the polynomial p provided by Proposition 1 and let $P_{N,X}$ be the input distribution of N. We need to construct an approximate OBDD for $sgn(p(x))$, where error is measured with respect to the input distribution.

Consider first the following (not necessarily minimal) layered OBDD for the variable ordering $x_1, \ldots, x_n$, computing $sgn(p(x))$ exactly. A partial truth assignment $a = (a_1, \ldots, a_\ell)$ to the variables $x_1, \ldots, x_\ell$ will lead to a node labeled by (s, b), where s is the sum of the values of terms of p which contain only variables from $x_1, \ldots, x_\ell$, and b is the set of bits from a which occur in so far undetermined terms containing some variables from $x_1, \ldots, x_\ell$. Thus partial truth assignments having the same (s, b) lead to the same node, as those are equivalent for the rest of the computation.

This OBDD can have exponential size. In order to decrease its size, the following issues need to be taken into consideration:

1. The variable ordering should be chosen so that $|b|$ is small.
2. Partial sums should be computed approximately by some kind of binning, to get an approximate result with small error.
3. As error is measured with respect to the input distribution, the binning process needs to have access to the relevant marginals at each point of the computation.

Item 1 is related to tree-width, more precisely to path-width, as the variable ordering is represented by a path. The ordering used is the one given by Corollary 2 for the primal graph of the term-hypergraph H_p. Item 3 requires an OBDD-like computational representation of the input distribution: the GOBDD defined above. However, it turns out that the GOBDD corresponding to the good path-width ordering of the variables is not necessarily small. This requires another approximation, this time of the input distribution by a small-width GOBDD.

Assume w.l.o.g. that variables are numbered according to the small path-width ordering and let $G = ([n], E)$ be the primal graph. The *separator* of vertex ℓ is

$$\mathsf{S}_\ell = \{j \,:\, j \leq \ell \text{ and } (j,k) \in E \text{ for some } k > \ell\}.$$

The *vertex separation number* $\mathsf{vs}(G)$ of G is the minimum of $\max_{\ell \leq n-1} |\mathsf{S}_\ell|$ over all orderings of the vertices. It holds that $\mathsf{pw}(G) = \mathsf{vs}(G)$ (see [23]) .

Lemma 1 *For any assignment $\{a_1, \ldots, a_n, c\} \in \{0,1\}^{n+1}$ and any $\ell \leq n$ it holds that*

$$P_N\left(a_\ell \,|\, a_1, \ldots, a_{\ell-1}, c\right) = P_N\left(a_\ell \,|\, a_{\mathsf{S}_{\ell-1}}, c\right).$$

Note that the primal graph is undirected, but conditional probabilities are based upon the directed edges of the network, and so the proof of this lemma involves the notion of d-separation in Bayesian networks [8, 24].

Thus the joint distribution can be written as

$$P_N(X_1 = a_1, \ldots, X_n = a_n, C = c) = p_c^0 \prod_{\ell=2}^{n} P_N\left(a_\ell \mid a_{\mathsf{S}_{\ell-1}}, c\right).$$

Lemma 2 *The joint distribution $P_N(X_1, \ldots, X_n, C)$ and the conditional distributions $P_N(X_1, \ldots, X_n|C)$ have width $n^{O(k)}$.*

The input distribution $P_{N,X}(a_1, \ldots, a_n)$ can be written as

$$P_N(a_1, \ldots, a_n \mid 0)\, P_N(0) + P_N(a_1, \ldots, a_n \mid 1)\, P_N(1),$$

so by Lemma 2 it is a mixture of two polynomial-width distributions. However, it is not necessarily of polynomial width itself [44]. Nevertheless, it can be approximated by a polynomial-width distribution. The edge probabilities in a GOBDD for the input distribution can be written as

$$P_N(a_\ell \mid a_1, \ldots, a_{\ell-1}) = \frac{P_N(a_1, \ldots, a_\ell)}{P_N(a_1, \ldots, a_{\ell-1})} = \frac{\sum_{c=0}^{1} P_N(a_1, \ldots, a_\ell, c)}{\sum_{c=0}^{1} P_N(a_1, \ldots, a_{\ell-1}, c)}.$$

The partial truth assignments in the last expression correspond to paths in the GOBDD of Lemma 2, beginning with the start node. This suggests approximating $P_N(a_\ell \mid a_1, \ldots, a_{\ell-1})$ by approximating the terms on the right. This can be achieved by splitting the nodes of the GOBDD, in order to encode an approximation of the conditional probabilities along paths leading to that node. This gives the following approximation of the input distribution.

Lemma 3 *There is a distribution $D(X_1, \ldots, X_n)$ of width $\mathsf{poly}\left(n^k, q, 1/\varepsilon\right)$ such that for every $a = (a_1, \ldots, a_n)$ it holds that*

$$(1 - \varepsilon) P_D(a) \leq P_{N,X}(a) \leq (1 + \varepsilon) P_D(a).$$

The next step is to construct a "product" OBDD computing the PTF *exactly* combining the layered OBDD for the PTF and the small-width approximate GOBBD constructed in the previous lemma. This OBDD has exponential size. It also computes, for every node v, acceptance probabilities of random assignments to the remaining variables, starting at v. As every node corresponds to a sum s of the terms already evaluated, these probabilities are monotone functions of s.

The last step is to compress the OBDD to polynomial size. Nodes of the final OBDD are a set of polynomially many distinguished nodes selected on each level, and other nodes are merged into these nodes. Distinguished nodes are found by binary search based on the s values, using the monotonicity property of

the acceptance probabilities to guarantee small error. The compression process is "virtual", so the intermediate OBDD does not have to be constructed. □

5 Bayesian Network Classifiers for Evaluating the Explainability of Generalized Additive Models with Interactions

Logistic regression for Boolean variables is a probabilistic model of the form

$$\log \frac{P(C=1 \mid X=x)}{P(C=0 \mid X=x)} = \alpha + \sum_{i=1}^{n} \beta_i x_i,$$

where the coefficients are to be learned. A *generalized additive model with interactions* (GA^2M) has the more general form

$$g(P(C=1 \mid X=x)) = \sum_{I \in \mathcal{I}} f_I(x^I),$$

where g is a given function, $\mathcal{I}$ is a family of subsets of $[n]$ and the functions f_I are to be learned.

By Proposition 1 a BNC can be viewed as *logistic regression model with interactions*. In this case the functions f_I are products. In particular, by Corollary 1, TAN can be written as

$$\log \frac{p_1^0}{p_0^0} + \sum_{i \text{ root}} \left(\log \frac{p_{01}^i}{p_{00}^i}(1-x_i) + \log \frac{p_{11}^i}{p_{10}^i} x_i \right)$$

$$+ \sum_{i \text{ non-root}} \left(\left(\log \frac{p_{001}^i}{p_{000}^i}(1-x_i)(1-x_{f(i)}) \right) \right.$$

$$\left. + \left(\log \frac{p_{011}^i}{p_{010}^i}(1-x_i)x_{f(i)} \right) + \left(\log \frac{p_{101}^i}{p_{100}^i} x_i (1-x_{f(i)}) \right) + \left(\log \frac{p_{111}^i}{p_{110}^i} x_i x_{f(i)} \right) \right),$$

where the p^is are the local conditional probabilities and $f(i)$ is the parent of i.

BNC is a generative model, so one can generate sets of random training and test examples from the joint distribution starting at the classifier node and proceeding down the forests, choosing nodes according to the local conditional probabilities. Using such samples, the GA^2M algorithm can learn a logistic regression model $q(x)$ with pairwise interactions.

The accuracy of a (deterministic) classifier f over the whole domain is

$$Acc(f) = P_{(X,C)\sim P_N}(C = f(X)) = \sum_x P_N(x, f(x)). \tag{5}$$

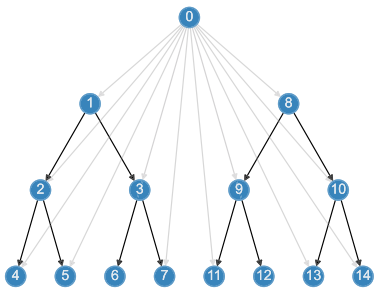
By its definition the polynomial p is an *optimal* classifier.

A GA^2M is considered to be an explainable model by analyzing the polynomial produced. For example, one can consider the set of terms $\mathcal{I}$ and the functions f_I. In our case this corresponds to considering the terms and their coefficients. As there is a ground truth, one can consider the *similarity* of the target polynomial p and the learned polynomial q. The relevant notion of similarity is to be specified. Here we consider what is perhaps the simplest possibility, the *fraction of common interaction terms*.

5.1 Experimental Results on a Small Example

In this section we present a small synthetic example of a target TAN and some experimental results on evaluating its similarity to the learned models. This is a snapshot of ongoing experiments.

The TAN structure considered is shown in Fig. 2a, and the conditional probabilities are listed in Fig. 2b. Here, for each node X_i that depends on nodes X_j and C, each conditional probability $P(X_i = 1 \mid X_j = x_j, C = c)$ (for $x_j \in \{0, 1\}$ and for $c \in \{0, 1\}$) was selected uniformly at random from the union of intervals $(\frac{1}{7}, \frac{2}{7}) \cup (\frac{3}{7}, \frac{4}{7}) \cup (\frac{5}{7}, \frac{6}{7})$. For nodes that depend only on C, $P(X_i = 1 \mid C = c)$ was selected in the same way. Finally, we set $P(C = 1) = \frac{1}{2}$.



(a)

X	Parent	$P(1 \mid 0,0)$	$P(1 \mid 0,1)$	$P(1 \mid 1,0)$	$P(1 \mid 1,1)$
1	-	0.234	-	-	0.732
2	1	0.472	0.164	0.156	0.455
3	1	0.506	0.203	0.812	0.458
4	2	0.810	0.202	0.508	0.734
5	2	0.281	0.759	0.527	0.554
6	3	0.148	0.453	0.268	0.157
7	3	0.743	0.470	0.449	0.540
8	-	0.235	-	-	0.479
9	8	0.284	0.535	0.469	0.827
10	8	0.844	0.471	0.184	0.733
11	9	0.559	0.471	0.798	0.224
12	9	0.176	0.790	0.850	0.785
13	10	0.149	0.163	0.542	0.433
14	10	0.483	0.220	0.236	0.194

(b)

Fig. 2 The TAN used in the experiment. (**a**) Graph structure. 0 is the source node C and i is the node for variable X_i. (**b**) Conditional probabilities. $P(a \mid b, c)$ is shorthand for $P(X_i = a \mid \text{Parent}(X_i) = b, C = c)$. Values have been rounded to 3dp

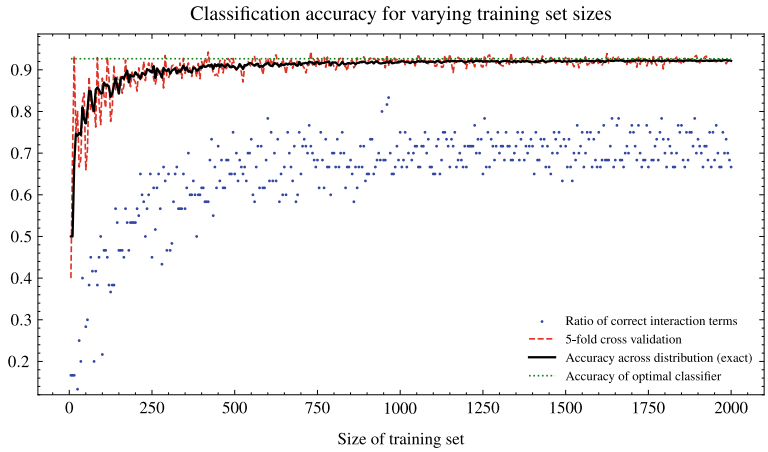


Fig. 3 Optimal classifier versus learned classifiers: accuracy and term overlap

We use the InterpretML software package [36] Python implementation of GA^2M, referred to as explainable boosting machines (EBM).[6] The number of interaction terms is specified to be 12, the number of edges between the X-variables. All other settings were left as default. To ensure accurate arithmetic with the TAN, particularly with repeated multiplication, all operations were performed with Python's decimal module. Pgmpy [2] was used to build the graph structure of the TAN, and NetworkX [17] was used for visualization. Classifiers were trained on samples of sizes up to 2000 (out of 2^{14} possible inputs), with step size 5, using fivefold cross validation. Accuracies are shown by the dashed curve.

Accuracies over the whole domain (as in (5)) are also shown on the top part of Fig. 3. The accuracy of the target classifier $p(x)$ is 0.9266, corresponding the top line. The accuracies of the learned classifiers $q(x)$ (averaged over the fivefolds) are shown by the thick middle curve.

The explanation quality of the classifiers produced is measured by the fraction of interaction terms which are common to p and q, for every sample size averaged over the fivefolds. The percentages are shown by the set of points in the lower half of Fig. 3. The percentages are growing with sample size and stabilize around 70%.

In order to get more detailed information, we present the coefficients of the optimal classifier p and three classifiers q_1, q_2, q_3 obtained from sample size 1600 in Tables 1 and 2 showing the affine, resp., quadratic terms. The accuracy of the classifiers is 0.9215, 0.9212 and 0.9198, respectively.

Based on Tables 1 and 2, we can make the following observations.

- The signs of the constant term do not always agree. The signs of the linear terms always agree, with the exceptions of x_{11} and x_{13}.

[6] GA^2M returns functions f_i, f_{ij} as step functions and are rewritten in polynomial form.

Table 1 Coefficients of *affine* terms of BNC polynomial (p) and some example GA^2M polynomials (q_1, q_2, and q_3)

	Const	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
p	0.11	1.87	−2.53	−1.71	−2.82	2.08	1.56	−1.17	−1.93	3.96	−1.94	−0.35	2.87	0.10	−1.20
q_1	−0.46	1.84	−1.90	−1.48	−2.03	1.27	1.41	−1.21	−0.92	3.69	−1.35	−0.39	2.92	−0.37	−0.46
q_2	0.45	2.18	−2.97	−2.36	−2.86	2.19	1.14	−0.88	−0.85	3.61	−0.99	−0.50	2.36	0.35	−0.50
q_3	−0.41	2.27	−2.26	−1.68	−2.69	2.09	1.39	−1.37	−1.61	4.02	−1.88	0.29	4.05	−0.66	−1.07

- Of the 12 interaction terms present in p, 8 always appear in the GA^2M polynomials. Their sign is always correct. The intersection sizes are 8, 9, 10, respectively.
- The coefficients of terms of p missing or having incorrect sign are smaller than other coefficients.
- The accuracy of the affine parts are 0.7525 for p, and 0.7648, 0.7654, 0.7640 for q_1, q_2, q_3, respectively, so the affine parts of p are worse than those of q_1, q_2, q_3. This seems to be due to the fact that GA^2M, in a greedy manner, produces the affine terms first [29].
- A comparison of the terms missing from p with the new terms added in q_1, q_2, q_3 shows that in several cases these can be paired up in such a way that a missing (parent, child) pair corresponds to a (grandparent, grandchild) or a (sibling1, sibling2) pair. For example, in q_1, the term x_1x_3 is missing, but x_1x_7 is added.

In summary, on this small example, using a simple evaluation, the GA^2M polynomials seem to have good explainability properties. It seems to be of interest to have a theoretical understanding of these properties. Extensions to Bayesian networks for real-world applications (see, e.g., [43]) could also be considered.

6 A Complexity Lower Bound for Positive Polynomial Threshold Representations

In this section we give a separation result between the sizes of positive and general QTF representations of an explicitly defined monotone Boolean function. The weight of $x \in \{0,1\}^n$ is $|x| = \sum_{i=1}^{n} x_i$. If $|x| = \ell$ then $|x|$ is on level ℓ. The Boolean threshold function $Th_\ell^n(x)$ has value 1 iff $|x| \geq \ell$.

Let $n = 2k$ and

$$f_n(x) = Th_{k+1}(x) \vee \left(Th_k(x) \wedge \bigwedge_{j=1}^{k} (\bar{x}_{2j-1} \vee \bar{x}_{2j}) \right).$$

A PTF representation of the function $f_4(x)$ is given in (2).

Table 2 Coefficients of *quadratic* (interaction) terms of BNC polynomial (p) and some example GA^2M polynomials (q_1, q_2, and q_3)

	Terms present in p												Terms not present in p					
	x_1x_2	x_1x_3	x_2x_4	x_2x_5	x_3x_6	x_3x_7	x_8x_9	x_8x_{10}	x_9x_{11}	x_9x_{12}	$x_{10}x_{13}$	$x_{10}x_{14}$	x_1x_6	x_1x_7	x_2x_3	x_8x_{12}	x_8x_{13}	$x_{11}x_{12}$
p	3.03	−0.24	3.80	−1.98	−2.24	1.55	0.62	4.31	−2.27	−3.31	−0.54	0.95						
q_1	1.88		2.57	−1.18	−1.88	1.06		3.39	−1.48	−3.07				0.44		−0.51	0.67	−0.39
q_2	3.45		3.39	−1.71	−1.75	1.65		3.66	−1.79	−2.97	−1.03		−0.33	−0.05	0.45			
q_3	2.80	−0.86	3.51	−2.07	−2.38	1.79		4.14	−2.00	−3.57		0.94					0.21	−1.23

Theorem 3 *The function* f_n

(a) is a monotone Boolean function,
(b) is a QTF of size $O(n)$ *(and tree-width 1),*
(c) has a positive QTF representation of size $O(n^2)$*,*
(d) every positive QTF representation of f_n *has size* $\Omega(n^2)$ *(and so tree-width* $\Omega(n)$*).*

Proof For *(a)* note that f_n is a *slice function*, i.e., for some level ℓ it is 0 on level $\ell - 1$ and below, it is 1 on level $\ell + 1$ and above, and so the only non-constant level is level ℓ. Every slice function is monotone.

Part *(b)* follows by considering the polynomial

$$p(x) = \sum_{i=1}^{n} x_i - \left(\frac{1}{k} \sum_{j=1}^{k} x_{2j-1} x_{2j} \right) - k.$$

The hyperedges of the term hypergraph are the singletons and the matching $M = \{(2j-1, 2j) : 1 \leq j \leq k\}$.

Part *(c)* follows by considering

$$\sum_{i=1}^{n} x_i \; + \; \frac{1}{\binom{k}{2}} \sum_{(i,j) \notin M} x_i x_j \geq k + 1.$$

For Part *(d)* assume that $p(x) \geq t$ is a positive QTF representation of f_n. The claim follows if we show that for every i, j, where $1 \leq i, j \leq k$, the polynomial p contains a mixed term $x_{2i-a} x_{2j-b}$ for some $a, b \in \{0, 1\}$ between the i'th and j'th blocks of odd and even bits. Assume w.l.o.g. that $i = 1$, $j = 2$ and there is no mixed term between the first two blocks. Let $z = (0, 1, 0, 1, \ldots)$ be the alternating truth assignment to the variables $x_5, \ldots, x_n$. Then by definition

$$p(0,0,1,1,z) < t, \quad p(1,1,0,0,z) < t, \quad p(1,0,1,0,z) \geq t, \quad p(0,1,0,1,z) \geq t.$$

Let $\gamma_{i,j}$ be the coefficient of $x_i x_j$. Then it holds that $\gamma_{1,3} = \gamma_{2,4} = 0$. Thus

$$p(0,0,1,1,z) + p(1,1,0,0,z) = p(1,0,1,0,z) + p(0,1,0,1,z) + \gamma_{1,2} + \gamma_{3,4},$$

a contradiction. □

A first open question is to determine the size of higher degree positive PTF representations of f_n. More generally, larger separations between positive and general PTF complexities would be of interest.

Acknowledgments We would like to thank Sebastian Bordt, Rich Caruana, Péter Hajnal, Lisa Hellerstein, Michal Moshkovitz and Lev Reyzin for discussions. The third author is partially supported by NSF grants 2217023 and 2240532, and by the Artificial Intelligence National Laboratory Program (RRF-2.3.1-21-2022-00004). Part of this work was done while visiting the Simons Institute for the Theory of Computing.

References

1. Ajtai, M., Gurevich, Y.: Monotone versus positive. J. ACM **34**(4), 1004–1015, 1987.
2. Ankan, A., Panda, A.: pgmpy: probabilistic graphical models using Python. In: Proceedings of the 14th Python in Science Conference (SciPy 2015), vol. 10. Citeseer (2015)
3. Barrington, D.A.M.: Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. J. Comput. Syst. Sci. **38**(1), 150–164 (1989)
4. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N.: Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1721–1730. ACM (2015)
5. Chan, H., Darwiche, A.: Reasoning about Bayesian network classifiers. In: UAI '03, Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence, pp. 107–115 (2003)
6. Chubarian, K., Turán, G.: Approximating bounded tree-width Bayesian network classifiers with OBDD. In: Proceedings of Machine Learning Research, vol. 138, pp. 113–124. PMLR (2020)
7. Courcelle, B., Makowsky, J.A., Rotics, U.: On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. Discret. Appl. Math. **108**(1–2), 23–52 (2001)
8. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press, Cambridge (2009)
9. Darwiche, A.: Logic for Explainable AI. In 38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS, pp. 1–11. IEEE (2023)
10. Darwiche, A., Marquis, P.: A knowledge compilation map. J. Artif. Intell. Res. **17**, 229–264 (2002)
11. De, A., Diakonikolas, I., Servedio, R.A.: Deterministic approximate counting for degree-2 polynomial threshold functions (2013). CoRR, abs/1311.7105
12. de Colnet, A., Marquis, P.: On translations between ML models for XAI purposes. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, pp. 3158–3166 (2023)
13. de Colnet, A., Mengel, S.: Lower bounds for approximate knowledge compilation. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, pp. 1834–1840 (2020)
14. Fischer, E., Makowsky, J.A., Ravve, E.V.: Counting truth assignments of formulas of bounded tree-width or clique-width. Discret. Appl. Math. **156**(4),, 511–529 (2008)
15. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn. **29**(2–3), 131–163 (1997)
16. Gopalan, P., Klivans, A.R., Meka, R., Stefankovic, D., Vempala, S., Vigoda,. E.: An FPTAS for #knapsack and related counting problems. In: IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, pp. 817–826 (2011)
17. Hagberg, A., Swart, P., Chult, D.S.: Exploring network structure, dynamics, and function using NetworkX. Technical Report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008
18. Hajnal, P., Liu, Z., Turán, G.: Nearest neighbor representations of Boolean functions. Inf. Comput. **285**(Part), 104879 (2022)

19. Hansen, K.A., Podolskii, V.V.: Polynomial threshold functions and Boolean threshold circuits. Inf. Comput. **240**, 56–73 (2015)
20. Hlinený, P., Oum, S., Seese, D., Gottlob, G.: Width parameters beyond tree-width and their applications. Comput. J. **51**(3), 326–362 (2008)
21. Hosaka, K., Takenaga, Y., Kaneda, T., Yajima, S.: Size of ordered binary decision diagrams representing threshold functions. Theor. Comput. Sci. **180**(1–2), 47–60 (1997)
22. Jukna, S.: Boolean Function Complexity - Advances and Frontiers. Algorithms and Combinatorics, vol. 27. Springer, Berlin (2012)
23. Kinnersley, N.G.: The vertex separation number of a graph equals its path-width. Inf. Process. Lett. **42**(6), 345–350 (1992)
24. Koller, D., Friedman, N.: Probabilistic Graphical Models - Principles and Techniques. MIT Press, Cambridge (2009)
25. Korach, E., Solel, N.: Tree-width, path-width, and cutwidth. Discret. Appl. Math. **43**(1), 97–101 (1993)
26. Krause, M., Pudlák, P.: Computing Boolean functions by polynomials and threshold circuits. Comput. Complex. **7**(4), 346–370 (1998)
27. Lacave, C., Díez, F.J.: A review of explanation methods for Bayesian networks. Knowl. Eng. Rev. **17**(2), 107–127 (2002)
28. Lipton, Z.C.: The mythos of model interpretability. Commun. ACM **61**(10), 36–43 (2018)
29. Lou, Y., Caruana, R., Gehrke, J.: Intelligible models for classification and regression. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 150–158 (2012)
30. Lou, Y., Caruana, R., Gehrke, J., Hooker, G.: Accurate intelligible models with pairwise interactions. In: The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, pp. 623–631 (2013)
31. Madre, J.C., Coudert, O.: A logically complete reasoning maintenance system based on a logical constraint solver. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence, pp. 294–299 (1991)
32. Makowsky, J., Meer, K.: Polynomials of bounded tree-width. In: Formal Power Series and Algebraic Combinatorics, pp. 692–703. Springer, Berlin (2000)
33. Makowsky, J.A., Meer, K.: Polynomials of bounded tree-width. In: Cucker, F., Rojas, M. (eds.) Foundations of Computational Mathematics, Proceedings of the Smalefest 2000, pp. 211–250. World Scientific, Singapore (2002)
34. Minsky, M.: Steps toward artificial intelligence. Proc. IRE **49**, 8–30 (1961)
35. Molnar, C.: Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2nd edn. Lulu.com (2022). https://christophm.github.io/interpretable-ml-book
36. Nori, H., Jenkins, S., Koch, P., Caruana, R.: InterpretML: A unified framework for machine learning interpretability (2019). CoRR, abs/1909.09223
37. O'Donnell, R.: Analysis of Boolean Functions. Cambridge University Press, Cambridge (2014)
38. O'Donnell, R., Servedio, R.A.: Extremal properties of polynomial threshold functions. J. Comput. Syst. Sci. **74**(3), 298–312 (2008)
39. Pratt, V.R.: The power of negative thinking in multiplying Boolean matrices. SIAM J. Comput. **4**(3), 326–330 (1975)
40. Razborov, A.A.: A lower bound for the monotone network complexity of the logical permanent. Mat. Zametki **37**, 887–900 (1985)
41. Razborov, A.A.: Lower bounds on the monotone complexity of some Boolean functions. Doklady Akademii Nauk SSSR **281**, 798–801 (1985)
42. Servedio, R.A., Tan, L.-Y.: Deterministic approximate counting of polynomial threshold functions via a derandomized regularity lemma. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021. LIPIcs, vol. 207, pp. 37:1–37:18 (2021)
43. Sesen, M.B., Nicholson, A.E., Banares-Alcantara, R., Kadir, T., Brady, M.: Bayesian networks for clinical decision support in lung cancer care. PloS One **8**(12), e82349 (2013)

44. Shen, Y., Choi, A., Darwiche, A.: Tractable operations for arithmetic circuits of probabilistic models. In: NIPS, 2016, pp. 3936–3944 (2016)
45. Shortliffe, E.H.: A rule-based computer program for advising physicians regarding antimicrobial therapy selection. In: Proceedings of the 1974 ACM Annual Conference, p. 739. ACM (1974)
46. Tardos, É.: The gap between monotone and non-monotone circuit complexity is exponential. Combinatorica **8**(1), 141–142 (1988)
47. Varando, G., Bielza, C., Larrañaga, P.: Decision boundary for discrete Bayesian network classifiers. J. Mach. Learn. Res. **16**, 2725–2749 (2015)
48. Wegener, I.: Branching Programs and Binary Decision Diagrams. SIAM, Philadelphia (2000)

Some Equalities are More Equal than Others

Ariel Cohen

All particles of matter were, in the beginning, equal to one another both in size and motion; and I leave no inequality in the universe except for...

— *Descartes,* Principles of Philosophy

Abstract Equality is an extremely useful relation: once we know that $a = b$, we can draw a great deal of important conclusions. However, it is rarely possible to conclude logically that, indeed, $a = b$. In this essay I discuss the notion of *sameness by default*: that it is sometimes beneficial to assume that two individuals are the same unless proved to be different. I demonstrate how an informal and rather procedural presentation of the idea turned, under the supervision of, and later, in collaboration with Prof. János Makowsky, into two rigorously formalized relations with important theoretical and applied implications.

1 A Personal Note

My acquaintance with Prof. János Makowsky's goes back to 1990, when I was an undergraduate student in computer science at the Technion. My main interest was computational linguistics, but of course there weren't too many courses on the topic, so I looked for courses that might sound interesting. A course taught by János caught my eye: *Automated Theorem Proving*.

For the final project of this course I was to pick a relevant article and critically evaluate it. I chose a paper that related automated theorem proving to natural language processing. While the paper described an interesting application, it provided little formal analysis. So, to be honest, the paper was not really appropriate

A. Cohen (✉)
Ben-Gurion University of the Negev, Beer-Sheva, Israel
e-mail: arikc@bgu.ac.il

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_12

for the course, and János would have been perfectly within his rights if he had refused to approve it.

But he did the exact opposite: he accepted the paper with enthusiasm, and took it upon himself to supervise my work on it. Under his guidance I was able to see the formal implications of the paper and to investigate them rigorously. Eventually, what started as a modest term paper became a research article on which we collaborated, together with Prof. Michael Kaminski.

This was my first foray, as a young student, into "real" research. I learned a great deal from János about how academic research is done and, more importantly, ought to be done. And later, what I have learned enabled me to take these results and apply them to computational linguistics, thus, in a sense, coming back a full circle. In this brief essay I would like to give you a taste of this journey.

2 Equality

Equality is, perhaps, the most fundamental and most intuitive of relations. Formally, it is an equivalence relation:

- Reflexive: $\forall x\ x = x$
- Symmetric: $\forall x \forall y\ (x = y \rightarrow y = x)$
- Transitive: $\forall x \forall y \forall z\ ((x = y \wedge y = z) \rightarrow x = z)$

In addition, equality is *substitutive*: for every predicate P:

- $\forall x \forall y\ ((P(x) \wedge x = y) \rightarrow P(y))$.

So, equality is an extremely powerful relation. If we know that $a = b$, we can conclude a great deal. But how do we know that $a = b$ in the first place? On what basis can we conclude this?

Leibniz famously proposed his principle of the *Identity of Indiscernibles*: if a and b share all their properties, then they are equal. In essence, this principle states that substitutivity is not only a necessary condition for equality, but it is also a sufficient one.

Leibniz's principle implies a possible procedure for ascertaining that a and b are equal. We start by assuming that a and b are different, and we try to determine in what way they are different: we try to find a predicate that holds of a but not of b (or vice versa). Only if we try and fail, can we conclude that a and b are equal. Note that, in order to apply this procedure, we have to know all the properties of a and b. If we are not sure about the truth value of $P(a)$ or of $P(b)$ for some predicate P, we cannot conclude that $a = b$. This, of course, is an extremely strong requirement. So, in practice, we can very rarely use Leibiniz's principle: we only know that $a = b$ when we are simply told this.

But there is an alternative. We can approach the problem from the opposite direction: assume that two expressions are equal, unless we can show that they are different. Let us call this idea, informally, *sameness by default*.

As far as I know, this idea was originally presented by Charniak [8]. Charniak's point of origin is a specific problem: analyzing the motivation for actions described in natural language texts. For example: "If we were told that someone wanted to smoke a cigarette, and we then saw her light a match, we would want to understand that she lighted the match in order to light the cigarette, and she wanted to do that in order to smoke it" (p. 275). The procedure he proposes for motivation analysis is, very roughly, as follows. Plans about typical activities are stored in the system. For example, the plan for smoking a cigarette includes igniting a light source, and using it to light the cigarette. Then, when we are told that the person lit a match, we have to identify the match as the light source in the plan. Since the description we are given rarely makes such an identification explicit, we cannot conclude logically that this was the object. But Charniak proposes that, unless we know something to the contrary, we should nonetheless conclude that it is.

So, according to Charniak, if we cannot conclude that two terms are different, we assume that they are equal. He calls the relation that holds between two such terms *nonmonotonic equality*. However, it should be pointed out that Charniak's paper is concerned with the implementation of nonmontonic equality for the specific problem of motivation analysis, and says little about the logical properties of this relation.

When I presented this paper in class, Prof. Makowsky was quick to point out this fact. Under his guidance and supervision, we started to formalize this relation, and identify its logical properties; at a later date Prof. Michael Kaminski joined us.

We have been able to come up with two alternative ways to formalize sameness by default, and have been able to identify and prove a number of interesting properties of these formalizations. Our results have been presented in [13], and formalized more fully in [14] and [15]. But before presenting these results, a few words about the logical apparatus we used are in order.

3 Default Logic

As the tool for formalizing this notion, we chose one of the better known and most widely studied logics for defeasible reasoning, Default Logic [30]. A substantial body of theoretical work has been devoted to Default Logic, and a number of theorem provers have been implemented.

A *default theory* is a pair (D, A), where D is a set of defaults and A is a set of first-order sentences. Defaults are expressions of the form

$$\frac{\alpha(\boldsymbol{x}) : \beta_1(\boldsymbol{x}), \ldots, \beta_m(\boldsymbol{x})}{\gamma(\boldsymbol{x})}, \tag{1}$$

where $\alpha(\boldsymbol{x})$, $\beta_1(\boldsymbol{x}), \ldots, \beta_m(\boldsymbol{x})$, and $\gamma(\boldsymbol{x})$ are formulas of first-order logic whose free variables are among $\boldsymbol{x} = x_1, \ldots, x_n$. Note that the presence of $\alpha(\boldsymbol{x})$ is optional. The formula $\alpha(\boldsymbol{x})$ is called the *prerequisite* of the default rule, the formulas $\beta_1(\boldsymbol{x}), \ldots, \beta_m(\boldsymbol{x})$ are called the *justifications*, and the formula $\gamma(\boldsymbol{x})$ is called the *conclusion*. A default is *closed* if none of $\alpha, \beta_1, \ldots, \beta_m$, and γ contains a free variable. Otherwise it is *open*.

The intuitive meaning of (1) is this: for every n-tuple of objects $\boldsymbol{t} = t_1, \ldots, t_n$, if $\alpha(\boldsymbol{t})$ is believed, and the $\beta_i(\boldsymbol{t})$s are consistent with one's beliefs, then one is permitted to deduce $\gamma(\boldsymbol{t})$.

For example, the following rule says that if something is a bird, and you don't know anything to the contrary, you may believe that it flies:

$$\frac{\mathbf{bird}(x) : \mathbf{fly}(x)}{\mathbf{fly}(x)} \tag{2}$$

A useful feature of some formalizations of Default Logic (e.g., [5]) is the possibility of assigning priorities to defaults. Intuitively, this means that if default d_1 outranks default d_2, then it applies first. While ranking is a very useful device, specifically for our purposes here, it is important to emphasize that it does not add to the formal power of the system: for every ranked default theory, an equivalent unranked default theory can be constructed [17].

We wanted to handle terms in their full generality, including terms containing variables. Hence, the default theory needed was *open*. Whereas the semantics of closed defaults has been quite thoroughly investigated, much less is known about open ones. For our semantics, we used *Herbrand models*, originally used to provide semantics for first order logic. Suppose we have a first order language $\mathcal{L}$, and we augment it with a set of new constants, b, calling the resulting language $\mathcal{L}_b$. The set of all closed terms of the language $\mathcal{L}_b$ is called the *Herbrand universe* of $\mathcal{L}_b$ and is denoted $\boldsymbol{T}_{\mathcal{L}_b}$. A *Herbrand b-model* is a set of closed atomic formulas of $\mathcal{L}_b$. This definition can be extended to Default Logic, following [23] and [18].

Like other default theories, Herbrand models can provide a semantics for sameness by default. A clarification, however, is in order. Since the Herbrand universe of a language $\mathcal{L}_b$ is the set of all closed terms of $\mathcal{L}_b$, then, by definition, in a Herbrand model no two terms are identical. But in our default theory, two terms may be considered the same by default. Is this a contradiction? The answer is no. We took equality, which forms the basis of sameness, to be *any* relation that satisfies the equality axioms, and not necessarily identity.[1] Hence, there is no problem about two terms being *equal*, even though they are not *identical*.

Crucial to the interpretation of Default Logic is the notion of an *extension*. Roughly speaking, an extension of a default theory is a set of statements containing all the logical entailments of the theory, plus as many of its default consequences as

[1] Of course, we can a have a non-Herbrand model where equality *is* identity—such models are called *normal*, see [25, p. 100] for details.

can be consistently believed. A default theory may have more than one extension, as in the well known *Nixon diamond*. Suppose we have the following set of defaults:

$$\left\{ \frac{\mathbf{Quaker}(x) : \mathbf{pacifist}(x)}{\mathbf{pacifist}(x)}, \frac{\mathbf{Republican}(x) : \neg\mathbf{pacifist}(x)}{\neg\mathbf{pacifist}(x)} \right\}. \tag{3}$$

The first rule says that Quakers are pacifist by default, and the second rule says that, by default, Republicans are not pacifist. If Nixon is both a Quaker and a Republican, in one extension he will be a pacifist, and in another he won't be. So, is Nixon a pacifist or isn't he?

When faced with multiple extensions, there are two general strategies we can use to decide which conclusions to accept: skeptical or credulous reasoning. Skeptical reasoning means accepting only what is true in all extensions. So, we will believe neither that Nixon is a pacifist, nor that he is not a pacifist. Credulous reasoning means picking one extension, based on whatever principles one deems appropriate, and accepting its conclusions. This means we will pick one extension, perhaps using our knowledge of Nixon's statements and actions, and based on this extension, conclude whether he is a pacifist or not.

Using Default Logic, we proposed two formalizations of sameness be default: Indistinguishability by Default and Equality by Default.

4 Indistinguishhability by Default

One formalization is called *Indistinguishability by Default*, indicated by $\sim$, a binary predicate symbol that is new to the underlying language, and defined by the following two defaults:

Definition 1 Indistinguishability by Default

$$\left\{ \frac{: x = y}{x \sim y}, \frac{x \neq y :}{x \not\sim y} \right\}. \tag{4}$$

Note that in the first default, the equality is a *justification*, while in the second default, the inequality is a *prerequisite*.

Indistinguishability by default has several desirable properties. One of them is that a default theory representing Indistinguishability by Default has a unique extension. This means that it is clear what the conclusions of the theory are in any given case, and there is no need to decide between credulous and skeptical reasoning—both are equivalent.

Moreover, note that the relation is part of the language; hence, it can appear as part of formulas, and, of course, be in the scope of quantifiers. It has additional interesting formal properties, such as reflexivity and symmetry, and while it is not

transitive, it is *semi-transitive*: if, for three terms t_1, t_2, and t_3, we cannot conclude $t_1 \neq t_2 \vee t_2 \neq t_3$, we can conclude $t_1 \sim t_3$.[2]

5 Equality by Default

An alternative we have proposed and formalized is *Equality by Default*. This, second formalization is an extension (modification) of the ordinary language equality by a similar default: two objects are *equal* if and only if their non-equality cannot be proved from the known facts. Formally, Equality by Default is defined by the following default rule:

Definition 2 Equality by Default

$$\frac{: x = y}{x = y}. \tag{5}$$

Unlike Indistinguishability by Default, Equality by Default is really an equality relation: in each extension it satisfies the equality axioms—it is reflexive, symmetric, transitive, and substitutive. This has the immediate consequence that, unlike the case of Indistinguishability by Default, the existence of a unique extension is not guaranteed.

6 Anaphora

6.1 Linguistic Background

In this section I would like to demonstrate the applicability of the two formalizations of the relation to a specific practical problem: anaphora resolution in natural language processing.

Often, the reference of an expression in natural language cannot be determined on its own, but only through the interpretation of another expression. For example, in (1), the pronoun *he* can only be interpreted once we realize that it refers back to the man who came in.

(1) A man came in. *He* sat down.

This phenomenon, where one expression refers to another, is called *anaphora*, and the problem of identifying the antecedent which an anaphoric trigger refers to is called *anaphora resolution*. Since anaphora is pervasive in natural language, any

[2] Obviously, this property would hold if $\sim$ were transitive.

computer program that attempts to understand natural language should use some method of anaphora resolution.

It must be noted that sometimes a pronoun has no antecedent in the discourse:

(2) Look out, *he* is going to hit you!

In this case, the reference of the pronoun is not given by the discourse, and the hearer must figure it out from the non-linguistic context. Such uses of pronouns are called *deictic*.

However, sometimes even when a candidate antecedent is provided by the discourse, it is not appropriate, for a variety of reasons. For example, consider the following discourse:

(3) A man came in. *He* saw *the man*.

For syntactic reasons, the referents of *he* and *the man* must be different: only one of them can refer to the man who came in—and the other must refer to some other individual.

It appears, then, that we try to identify a pronoun with an antecedent, unless there is an indication that they are different—which is precisely the notion underlying sameness by default.

This notion is necessary for the following reason. Research in anaphora resolution has raised a number of heuristics, usually called *preferences*, which suggest that a particular candidate is likely to be the correct antecedent.

For example, a syntactic subject is more likely to be the correct antecedent:

(4) The customer lost patience and called the waiter. *He* didn't want to listen to reason.

In principle, the antecedent of the pronoun could be either *the customer* or *the waiter*. But the former is much more likely; and note that it is the subject of the sentence.

Another such preference is topicality:

(5) Scientists have proven that temperature does not affect the nutritional composition of milk: babies can drink cold milk. If the baby does not thrive on cold milk, boil *it*.

Note that, in this case, the subject *the baby* is not a likely antecedent, and a much more plausible candidate is *cold milk*. Although *cold milk* is not a subject, it is a topic, since the passage is about milk and its proper temperature.

Sometimes the important factor is not the syntactic position of the candidate itself, but whether it has the same syntactic position as that of the pronoun:

(6) a. The cook combined chocolate with vanilla, but he had combined *it* with strawberry last time.

b. The cook combined chocolate with vanilla, but he had combined strawberry with *it* last time.

In (6a), the pronoun *it* is a direct object, and its plausible antecedent is also a direct object, *chocolate*; in (6b), it is an indirect object, and its most likely antecedent is the indirect object *vanilla*.

The point is that there are various preferences suggesting plausible antecedents. In the early period of anaphora resolution, which lasted until the early 1990s, such preferences were explicitly programmed; more recently, data-driven systems have been developed, which, using machine learning techniques, acquired the preferences automatically, and they were, in general, not transparently represented in the computer.[3]

Sometimes, however, the correct antecedent is not suggested by any of the preferences: it is neither topical, nor a subject, nor does its syntactic position parallel that of the trigger, etc. In such cases, the antecedent is simply chosen as a last resort, since the other potential candidates are ruled out. Without this "last resort" rule, no antecedent would be chosen.

However, it is not clear how to formalize this "last resort" rule in such a way that it could be implemented; although some linguists have used it for their study of anaphora, they have failed to formalize it.

For example, such a rule has been proposed in [31, p. 603], where it is called "Don't Overlook Anaphoric Possibilities (DOAP)". This rule says essentially that, when we encounter an anaphoric trigger, we must try to find an antecedent. However, neither DOAP, nor other similar rules in the linguistics literature, have ever been formalized.

6.2 Sameness by Default and Anaphora

An anaphora resolution system based on an appropriate formalization of sameness by default will have important advantages, as discussed below. However, it might be objected that sameness by default is too liberal, in that it allows too many elements to be the same by default. This, however, is not the case. Sameness by default does not apply in isolation; any reasonable system drawing inferences from natural language will require many more defaults, some of which deal specifically with anaphora, while others don't. We will assign low priority to sameness by default, so that, if other defaults can apply, they will; then, inappropriate equalities will be ruled out, and rather few equalities will remain.

In other words, sameness by default is a principle of *last resort*: it will not apply if another rule suggests an antecedent. It is, therefore, an attractive way to formalize rules like DOAP.

[3] See [27] and [24] for good overviews.

Consider the following examples:

(7) a. John saw Bill. *He* greeted *him*.
b. John hates *him*.
c. John doesn't have a car. *It* is red.
d. A man came into the bar. *She* was upset.

The most likely interpretation of (7a) is that *he* refers to John, and *him* refers to Bill, hence they are not the same. This interpretation may be generated by a preference for antecedents that share the grammatical position of the pronoun, as we have seen above. Since in (7a) a preference rule applies, sameness by default will not apply, and we are in no danger of concluding erroneously that the referents of the two pronouns are the same. Sentence (7b) does not have an interpretation where *him* is equated with John, for syntactic reasons. In (7c), the pronoun *it* should not be equated with the discourse referent representing the indefinite *a car*, because, according to a well established linguistic theory [19], the indefinite is not accessible to the pronoun. The discourse in (7d) is an example where the pronoun cannot be associated with the antecedent because of a gender mismatch. If all such constraints are formalized, as indeed they must be for any anaphora resolution system (either explicitly programmed or automatically acquired), and given a higher priority than sameness by default, inadmissible antecedents will be ruled out.

Thus, although sameness by default appears very permissive, in fact it allows rather few elements to be equal by default. These are intended to be anaphoric triggers and their potential antecedents, when no antecedent is suggested by an anaphora resolution factor.

6.3 The Importance of Formalization

But still, one may ask: why is it so important to formalize this relation? Why can't we simply implement it without the formalization?

Indeed, model builders are a case in point. The simplest way to show that a theory is consistent is to construct a model for it, and this is what model builders do. For reasons of economy, model builders attempt[4] to build a *minimal model*, i.e., a model that does not postulate the existence of more entities than is necessary.[5]

Take, for example, the following set of formulas: $A = \{P(a), P(b), \neg P(c)\}$. The task of a model builder is to construct a model for A. It will start with the smallest possible domain—a singleton. Clearly, no such model will be able to satisfy all the formulas in A. The model builder will then attempt to construct a model with two elements, and this time it will succeed. For example, the builder might construct a

[4] Though they do not always succeed, see, e.g., [6] for discussion.

[5] Very often, model builders also attempt to generate a model that is minimal in another sense, namely in terms of the statements that are true in the model, but this does not concern us here.

model whose domain consists of the elements d_1 and d_2 and whose interpretation function I has the following values: $I(a) = I(b) = d_1$, $I(c) = d_2$, $I(P) = \{d_1\}$. This model will clearly satisfy the set of formulas A.

Note that, in this model, the constants a and b are mapped onto the same domain element d_1. Of course, we could also have a model with three elements, where a and b have different interpretations. This, indeed, is the model expected under the *unique name assumption* [29], stating that two objects are different unless they can be proved to be equal.

Such a model, however, is not minimal, and will not generally be generated by a model builder. In contrast with the unique name assumption, in a minimal model we assume that two elements are the same, unless they can be proved to be different. Thus, a and c are different, because one of them has a property, namely P, which the other lacks; but a and b are assumed to be the same element, since no property distinguishes between them.

One might say, then, that the notion of sameness by default is implicit in the workings of model builders, without the need for properly formalizing it. Indeed, model builders, and the minimal models they generate, have been used to resolve anaphora [4, 20, 21]. Why is there a need for the rigorous formalization we have performed?

One obvious reason is theoretical: formalization allows one to define precisely what the relation does, when it can apply, and what conclusions can be drawn from it.

But even for the those who are only interested in a practical application, formalization may turn out to be indispensable. This is certainly the case for the application of anaphora resolution.

The reason is that model builders can only generate first order models; in particular, they cannot be applied to nonmonotonic logics. But it is well known that most, perhaps all rules for anaphora resolution are defeasible. For example (from [3]):

(8) The Vice-President entered the President's office. *He* was nervous and clutching his briefcase. After all, he couldn't fire the Vice-President without making trouble for himself with the chairman of the board.

The pronoun in the second sentence has two potential antecedents: *the Vice-President* or *the President*. Clearly *the Vice-President* is preferred: it has the same syntactic position (subject) as the pronoun, and it is more salient. However, by the time the third sentence is processed, it is clear that this choice is wrong, and the intended antecedent is, in fact, *the President*.

Such examples abound; and they indicate that all anaphora resolution factors, or almost all of them, are best thought of as defaults, which may be overridden. It is therefore attractive to model anaphora resolution as a system of prioritized defaults (e.g., [7, 22, 26, 28]). Hence, only a principled formalization of sameness by default in a logical nonmonotonic system, such as we have carried out, can actually be implemented in a practical system aimed at anaphora resolution.

6.4 Which Formalization to Choose?

We have been able to provide two different formalizations of sameness by default. But which one would be appropriate for anaphora resolution?

As described above, we have been able to prove several interesting properties of Indistinguishability by Default. An additional result we have proved, which is particularly relevant for anaphora resolution, is the existence of an element that is indistinguishable by default from every element.

This theorem provides an account of the availability of deictic readings, i.e., readings where the anaphoric trigger does not have an antecedent, but refers to some individual whom the speaker may indicate by, say, pointing. Since there is an element that is indistinguishable by default from every element, this means that there is always an element, about which nothing is said in the discourse, and yet can consistently be identified with any trigger.

At first sight, therefore, the relation of Indistinguishability by Default seems an attractive choice to apply to the problem of anaphora resolution. However, despite its formally attractive properties, Indistinguishability by Default may ultimately be inappropriate for this application.

The problem lies precisely in the fact that Indistinguishability by Default is a *new* relation and, consequently, is denoted by a *new* symbol. Therefore, no axiom is associated with this relation and knowing $t_1 \sim t_2$ tells us very little: there are few conclusions which can be drawn from this fact.

Also, since indistinguishability is not equality, we cannot maintain, even by default, that, if $t_1 \sim t_2$, then t_1 and t_2 denote the same individual.

This is particularly problematic for the application of anaphora resolution. Suppose we decided, by default, to identify pronoun u with individual a. Suppose that we also know of some property P that $P(a)$. We would like to conclude $P(u)$; however, this does not follow in our system, not even nonmonotonically.

For these reasons, Indistinguishability by Default, despite its formal elegance, is too weak to tackle the application we are considering. Therefore, Equality by Default, the more powerful relation, is better suited to represent the association of a pronoun with an antecedent [9–12].

But, one may ask, what about the deictic reading, where the pronoun refers to some individual not mentioned in the discourse? When we considered the relation between pronoun and antecedent to be *Indistinguishability* by Default, we could use a proved result to help us; there is no counterpart of such a proposition for *Equality* by Default, and yet we can still account for the deictic reading in the following way.

According to the semantics of open default theories we used [18], the theory domain is a Herbrand universe of the original language augmented with a set of new objects. Because of these new elements, deictic interpretations of pronouns are always possible: each of the new elements may be equal by default to any pronoun. Hence, we have a logical explanation for a linguistic phenomenon—the universal availability of deictic readings.

6.5 Dealing with Multiple Extensions

Resolving anaphora, using this formalization, becomes the problem of inferring equality between two elements. We will draw this inference by default, using Equality by Default.

Recall that there are two logical strategies for dealing with multiple extensions: credulous and skeptical reasoning. It turns out that these two approaches indicate two different strategies for dealing with unresolved anaphora. If there is exactly one appropriate candidate, the system should identify it. But what if there is more than one?

Automatic systems for anaphora resolution usually assign a score to each candidate antecedent. The score is based on the combined effect of the constraints and preferences—which, like mentioned above, used to be explicitly programmed and these days are usually acquired automatically. Then, the candidate with the highest score that is above a certain threshold is selected. What happens if no candidate receives a score that is above the threshold? Some systems pick the candidate with the highest scope anyway; other systems leave the anaphora unresolved.

When these choices are formalized in terms of Equality by Default, they reduce to credulous and skeptical reasoning, respectively. This has an important advantage.

To see this, consider the following example:

(9) John and Bill met at the ice cream parlor. *He* was upset.

Our default theory will have two extensions: in one of them, the pronoun is equated with *John*, and in the other—with *Bill*.

How do we deal with these extensions? We may decide to force a decision for one or the other; for example, we can decide that the most recent antecedent (Bill) is appropriate, or that the first one mentioned (John) is more prominent, hence preferred. So, in effect, we would apply credulous reasoning and pick one extension.

Note that, by the axioms of equality, once such a choice is made, any property of the antecedent becomes also a property of the trigger. For example, if we choose the extension where *he* is equated with Bill, and if Bill is bald, it will immediately follow that *he* is bald.

There are cases, however, when the anaphora is genuinely ambiguous, and we may have no reason to prefer one reading over the other. But even if we decide not to resolve the anaphora, there are still inferences we can make. For example, given (9), we want to conclude that whoever the pronoun refers to was at the ice cream parlor.

In this case, it makes sense to apply skeptical reasoning, and accept only what is true in all extensions. In one extension, the pronoun is equated with John; but we know, from the first sentence, that John was at the ice cream parlor. In the other extension, the pronoun is equated with Bill; but again, we know, from the first sentences, that Bill was at the ice cream parlor. Hence, we will generate the desired inference, since in both extensions, the pronoun has the properties that its antecedent

has, and in both extensions, these properties include the property of being at the ice cream parlor.

Computational linguists have for a while realized that, for many applications, it is not necessary to provide a complete analysis of the text. A lot of useful inferences can be generated even when the text receives only a partial processing, and some aspects of it remain unspecified. To this end, a number of systems of unspecified representations have been proposed: first for explicit, knowledge-based systems (e.g., [2, 16]), and, more recently, for statistical, machine-learning systems (e.g., [1]). When Equality by Default is integrated into an anaphora resolution system, it provides the benefit of underspecification for free, without the need to devise and implement such special representations.

7 Conclusion

In this essay I described the transformation of an informal idea to a rigorously defined relation, and the use of this relation for a specific application: anaphora resolution.

The moral of the story, and, perhaps the most important lesson I have learned from János, is the necessity for formalization: whatever concept you are considering, even if your goal is a practical application rather than a theoretical result, it is absolutely necessary to define your terms in a rigorous way, so that you can actually prove the properties of your proposal, what it does, and what follows from it.

References

1. Alshawi, H., Chang, P.C., Ringgaard, M.: Deterministic statistical mapping of sentences to underspecified semantics. In: Computing Meaning, vol. 4. Springer, Dordrecht (2004)
2. Alshawi, H., Crouch, R.: Monotonic semantic interpretation. In: Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics, pp. 32–39. Newark, Delaware (1992)
3. Asher, N.: Linguistic understanding and non-monotonic reasoning. In: Proceedings of the 1st International Workshop on Nonmonotonic Reasoning. New Paltz (1984)
4. Baumgartner, P., Kühn, M.: Abducing coreference by model construction. J. Language Comput. **1**(2), 175–190 (2000)
5. Brewka, G.: Adding priorities and specificity to Default Logic. In: Pereira, L., Pearce, D. (eds.) Proceedings of the 4th European Workshop on Logics in Articial Intelligence (JELIA-94), pp. 247–260 (1994)
6. Bry, F., Yahya, A.: Positive unit hyperresolution tableaux and their application to minimal model generation. J. Automated Reasoning **25**, 35–82 (2000)
7. Byron, D., Gegg-Harrison, W.: Evaluating optimality theory for pronoun resolution algorithm specification. In: Proceedings of the Discourse Anaphora and Reference Resolution Conference (DAARC2004), pp. 27–32 (2004)
8. Charniak, E.: Motivation analysis, abductive unification and nonmonotonic equality. Artif. Intell. **34**, 275–295 (1988)

9. Cohen, A.: Anaphora resolution and minimal models. In: Bos, J., Koller, A. (eds.) Proceedings of the 5th International Conference on Inference in Computational Semantics—ICOS–5, pp. 7–16 (2006)
10. Cohen, A.: An application of Default Logic to anaphora resolution. Presented at The Israeli Seminar in Computational Linguistics—Iscol (2006)
11. Cohen, A.: Anaphora resolution as equality by default. In: Branco, A. (ed.) Anaphora: Analysis, Algorithms and Applications. Lecture Notes in Artificial Intelligence, pp. 44–58. Springer-Verlag, Berlin (2007)
12. Cohen, A.: Anaphora resolution by default. In: Proceedings of IWCS-7 (the 7th International Workshop on Computational Semantics), pp. 53–64. Tilburg, The Netherlands (2007)
13. Cohen, A., Makowsky, J.A.: Two approaches to nonmonotonic equality. Technical Report CIS-9317, Technion—Israel Institute of Technology, 1993
14. Cohen, A., Kaminski, M., Makowsky, J.A.: Indistinguishability by default. In: Artemov, S., Barringer, H., d'Avila Garcez, A.S., Lamb, L.C., Woods, J. (eds.) We Will Show Them: Essays in Honour of Dov Gabbay, pp. 415–428. College Publications (2005)
15. Cohen, A., Kaminski, M., Makowsky, J.A.: Notions of sameness by default and their application to anaphora, vagueness, and uncertain reasoning. J. Logic Language Inf. 285–306 (2008)
16. Copestake, A., Flickinger, D., Sag, I., Pollard, C.: Minimal recursion semantics: an introduction. Res. Language Comput. **3**(23), 281–332 (2005)
17. Delgrande, J.P., Schaub, T.: Expressing preferences in Default Logic. Artif. Intell. **123**, 41–87 (2000)
18. Kaminski, M.: A comparative study of open default theories. Artif. Intell. **77**, 285–319 (1995)
19. Kamp, H., Reyle, U.: From Discourse to Logic. Kluwer Academic Publishers, Dordrecht (1993)
20. Kohlhase, M.: Model generation for discourse representation theory. In: Horn, W. (ed.) Proceedings of the 14th European Conference on Artifical Intelligence, pp. 441–445 (2000)
21. Konrad, K.: Model Generation for Natural Language Interpretation and Analysis. Ph.D. Thesis, University of Saarlandes, Saarbrücken, 2000
22. Lascarides, A., Asher, N.: Temporal interpretation, discourse relations and common sense entailments. Linguist. Philos. **16**, 437–493 (1993)
23. Lifschitz, V.: On open defaults. In: Lloyd, J.W. (ed.) Computational Logic: Symposium Proceedings, pp. 80–95. Springer–Verlag, Berlin (1990)
24. Poesio, M., Stuckardt, R., Versley, Y.: Anaphora Resolution: Algorithms, Resources and Applications. Springer, Berlin (2016)
25. Mendelson, E.: Introduction to Mathematical Logic. Chapman and Hall, London (1997)
26. Mitkov, R.: An uncertainty reasoning approach for anaphora resolution. In: Proceedings of the Natural Language Processing Pacific Rim Symposium (NLPRS'95), pp. 149–154. Seoul, Korea (1995)
27. Mitkov, R.: Anaphora Resolution. Longman, London (2002)
28. Poesio, M.: Semantic ambiguity and perceived ambiguity. In: van Deemter, K., Peters, S. (eds.) Semantic Ambiguity and Underspecification, pp. 159–201. CSLI, Stanford (1996)
29. Reiter, R.: Equality and domain closure in first order databases. J. ACM **27**, 235–249 (1980)
30. Reiter, R.: A logic for default reasoning. Artif. Intell. **13**, 81–132 (1980)
31. Williams, E.: Blocking and anaphora. Linguist. Inquiry **28**, 577–628 (1997)

On the Bipartition Polynomials for Rooted Caterpillars

Bruno Courcelle and Irène Durand

Abstract The bipartition polynomial $B(G, u, v, w)$ associated with a finite simple undirected graph G has been defined by Dod, Kotek, Preen and Tittmann. It is a common generalization of the domination polynomial, the Ising polynomial and a few others. These authors conjecture that for a finite tree T, $B(T, u, v, w)$ determines T up to isomorphism. We prove a weak form of the conjecture for caterpillars and we propose a method for extending the proof to rooted trees. A caterpillar is a path with additional pendent edges. For that, we introduce an alternative polynomial $Q(G, u, v, w)$ that is symmetric in the sense that $Q(G, u, v, w) = Q(G, v, u, w)$ and is equivalent to $B(G, u, v, w)$ (up to algebraic transformations). We can thus work on the conjecture with Q instead of B. This polynomial can be computed by a finite automaton on binary terms denoting finite rooted trees.

1 Introduction

In this article graphs are finite, undirected without loops or parallel edges. The bipartition polynomial $B(G, u, v, w)$ associated with such a graph G has been defined in [2]. It is a common generalization of the domination polynomial, the Ising polynomial and a few others.

A monomial $u^p v^q w^m$ in $B(G, u, v, w)$ says that there is a bipartite subgraph having m edges whose two sets of vertices are X and Y, such that X is included in a set X' of p vertices that is disjoint with Y of cardinality q, furthermore, each vertex of $X \cup Y$ is an end of some edge of the considered subgraph.

It is conjectured in [2] that for a tree T, the polynomial $B(T, u, v, w)$ determines T up to isomorphism. We prove a weak form of the conjecture for caterpillars. A caterpillar is a path with additional pendent edges.

B. Courcelle (✉) · I. Durand
LaBRI, Bordeaux University, Bordeaux, France
e-mail: courcell@labri.fr; idurand@labri.fr

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_13

For that, we introduce an alternative polynomial $Q(G, u, v, w)$ that is symmetric in the sense that $Q(G, u, v, w) = Q(G, v, u, w)$ and is equivalent to $B(G, u, v, w)$. It is defined like B except that we require that $X = X'$. As X and Y play symmetric roles in the definition of Q, we have $Q(G, u, v, w) = Q(G, v, u, w)$.

Easy algebraic transformations relate $Q(G, u, v, w)$ and $B(G, u, v, w)$ for G without isolated vertices, or having a single vertex. Hence all properties of B proved in [2] can be translated for Q. In particular, one can determine from $Q(G, u, v, w)$ the number of vertices, the number of edges, the degree sequence, the number of connected components, and the set of numbers of vertices of the connected components of G, but not G itself (up to isomorphism).

We consider the conjecture from [2] that $B(G)$ (that is $B(G, u, v, w)$ in short) determines G if it is a tree, hence, equivalently that $Q(G)$ determines G in this case.

We study it for rooted trees. We express $Q(T)$ as the sum $Q_0(T) + Q_1(T) + Q_2(T)$ where $Q_0(T)$, $Q_1(T)$ and $Q_2(T)$ are associated respectively with the cases where the root is not in $X \cup Y$, is in X or is in Y.

We establish recursive definitions and we show how these three subpolynomials of $Q(T)$ can be computed by a finite automaton on binary terms that define rooted trees, by using the methods of [1].

For helping to solve the main conjecture, we define a subpolynomial of $Q(T)$ corresponding to counting the subtrees of T as opposed to the subforests. As it has less monomials, understanding its behaviour may be easier.

Our main results are as follows:

1. for a rooted caterpillar T, the polynomial $Q_1(T)$ determines it,
2. for every rooted tree T, the computations of $Q(T)$, $Q_0(T)$, $Q_1(T)$ and $Q_2(T)$ can be done by using a deterministic automaton having 6 states, that takes as input a binary term defining T.

2 Bipartition Polynomials

We recall from the introduction that graphs are undirected, finite and without loops and parallel edges. The number of vertices of the current graph or tree is always n.

Definition 2.1 (Bipartition Polynomial) For a graph $G = (V, E)$ we define $B(G, u, v, w) := \sum_{\alpha(X,Y,F)} u^{|X|} v^{|Y|} w^{|F|}$ where $\alpha(X, Y, F)$ means that:

X is a set of vertices,
F is a set of edges between X and $V - X$,
and Y is the set of ends not in X of the edges in F.

We write $B(G)$ for $B(G, u, v, w)$.

Examples 2.2 $B(\bullet) = 1 + u$ where $\bullet$ denotes an isolated vertex.
$B(e) = 1 + 2u + u^2 + 2uvw$ for an edge e.

Proposition 2.3 *The following values concerning G can be computed from $B(G)$:*

(1) *the number of vertices,*
(2) *the number of edges,*
(3) *the degree sequence, whence, the number of isolated vertices,*
(4) *the number of connected components, and*
(5) *the set of numbers of vertices of the connected components.*

Proofs See [2]. □

Fact 2.4 ([2]) There exist two graphs having both 22 vertices, the same bipartition polynomial and that are not isomorphic.

Conjecture 2.5 ([2]) Two trees are isomorphic if they have the same bipartition polynomials.

It has been checked for trees up to 14 vertices.

2.1 An Equivalent Polynomial

Definition 2.6 (An Alternative Bipartition Polynomial) We define $Q(G, u, v, w) := \sum_{\beta(X,Y,F)} u^{|X|} v^{|Y|} w^{|F|}$ where $\beta(X, Y, F)$ means $\alpha(X, Y, F) \wedge$ "every vertex in X is the end of an edge in F".

The coefficient of a monomial $u^p v^q w^m$ is the number of triples (X, Y, F) such that $\beta(X, Y, F)$ holds and $(p, q, m) = (|X|, |Y|, |F|)$. We write $Q(G)$ for $Q(G, u, v, w)$ if no ambiguity can occur.

Remarks and Examples 2.7

(1) If F is empty, so must be X and Y.
(2) $Q(\bullet) = 1$ (F must be empty) and $Q(e) = 1 + 2uvw$.
(3) $Q(G) = Q(G \cup \bullet)$ for every graph G.
(4) The monomials of $Q(G)$ are either 1 or $u^p v^q w^m$ where $p + q \leq \min(n, 2m)$. For trees, we also have $m + 1 \leq p + q$.
(5) As $\beta(X, Y, F) \iff \beta(Y, X, F)$, we have $Q(G, u, v, w) = Q(G, v, u, w)$. Moreover $Q(G)$ is 1 + a sum of polynomials of the form $(u^p v^q + u^q v^p) w^m$. Hence, for each $m > 0$, there is in $Q(G)$ an even number of monomials of the form $u^p v^q w^m$. We consider $2uv = uv + uv$ as consisting of 2 monomials.
(6) The number of vertices that are not isolated can be determined from $Q(G)$: it is the maximun number $p + q$ such that $u^p v^q w^m$ is a monomial for some m. The corresponding values of m are the numbers of edges of the spanning bipartite subgraphs of G.

Proposition 2.8 *For a graph G having n vertices, $B(G)$ and $Q(G)$ can be computed from each other by the following formulas:*

(1) $B(G, u, v, w) = (1+u)^n Q(G, \frac{u}{1+u}, \frac{v}{1+u}, w)$
(2) $Q(G, u, v, w) = (1-u)^n B(G, \frac{u}{1-u}, \frac{v}{1-u}, w)$

Proof The polynomial B can be expressed as follows (compare with Q):

$B(G, u, v, w) := \sum_{\beta(X,Y,F) \wedge \delta(X,Y,W)} u^{|X|+|W|} v^{|Y|} w^{|F|}$ where $\beta(X, Y, F)$ is as in Definition 2.6 and $\delta(X, Y, W) \Longleftrightarrow W \cap (X \cup Y) = \varnothing$.

Hence, we obtain $B(G)$ by replacing in $Q(G)$ each monomial $u^p v^q w^r$ by $u^p(1+u)^{n-p-q} v^q w^r$. Formula (1) follows.

Conversely, by using (1) we get

$(1-u)^n B(G, \frac{u}{1-u}, \frac{v}{1-u}, w) = (1-u)^n (1+\frac{u}{1-u})^n Q(G, \frac{u/(1-u)}{1+u/(1-u)}, \frac{v}{1+u/(1-u)}, w)$ which is equal to $Q(G, u, v, w)$. □

For investigating Conjecture 2.5, we can use Q instead of B.

A Q-polynomial is one of the form $Q(G)$ for some graph G.

Proposition 2.9

(1) If $G_1, ..., G_k$ are the connected components of G, then $Q(G) = Q(G_1) \times ... \times Q(G_k)$.
(2) If G is connected and bipartite, then $Q(G)$ is not the product of two Q-polynomials not equal to 1.

We need some facts. Note that we consider polynomials with positive integer coefficients.

Definition 2.10 For a polynomial $P(u, v, w)$ we define $Max_w(P)$ as the sum of monomials of P where w has maximal degree. Hence $Max_w(PR) = Max_w(P) \times Max_w(R)$.

Examples and Remarks 2.11

(1) $Q(K_3) = 1+6uvw+3uv^2w^2+3u^2vw^2$. Hence, $Max_w(Q(K_3)) = 3uv^2w^2+3u^2vw^2$.

We let S_p denotes the star with p leaves connected to the root. Then, $Q(S_3) = 1 + u((1 + vw)^3 - 1) + v((1 + uw)^3 - 1)$ and $Max_w(Q(S_3)) = uv^3w^3 + u^3vw^3$.

(2) The number of edges m of a graph G is half the number of monomials in $Q(G)$ of the form uvw. A graph G is bipartite if and only if the maximum degree of w in $Q(G)$ is m. Hence, from $Q(G)$, one can determine if G is bipartite.

For a connected bipartite graph $G = (V_1, V_2, E)$ where $|E| = m \neq 0$, $|V_1| = p$, $|V_2| = q$, we have $Max_w(Q(G)) = w^m(u^p v^q + u^q v^p)$. For a bipartite graph G without isolated vertices that is not connected, $Max_w(Q(G))$ is not of this form as it must contain at least 4 monomials.

(3) For any Q-polynomial P, we have $Max_w(P) = w^{m_1}P'$ where P' is a symmetric polynomial in u, v having at least 2 monomials.
(4) For a tree T we have : $Q(T, 1/2, 1/2, w-1) = 2^{1-n}(1+w)^{n-1}$.

Proof Based on [4] proving that a connected graph G is a tree if and only if $B(G, 1, 1, w-1) = 2(1+w)^{n-1}$. With Proposition 2.8, we get the result.

Conversely, by the same computation and for G without isolated vertices, if $Q(G, 1/2, 1/2, w-1) = 2^{1-n}(1+w)^{n-1}$ then G is a tree. □

Proof of Proposition 2.9:

(1) is clear from the definition.
(2) Let G be bipartite and connected.

Assume for a contradiction that $Q(G) = PR$ for two Q-polynomials in $\mathbb{N}[u, v, w]$ not equal to 1. Then $Max_w(Q(G)) = w^m(u^p v^q + u^q v^p) = Max_w(P) \times Max_w(R) = w^{m_1}P'w^{m_2}R'$ where $m = m_1 + m_2$, $Max_w(P) = w^{m_1}P'$ and $Max_w(R) = w^{m_2}R'$, cf. Remark 2.11(3).

Hence $u^p v^q + u^q v^p = P' \times R'$. Its only possible factorizations in polynomials in $\mathbb{N}[u, v, w]$ are with $P' = u^a v^b$ and $R' = u^{p-a}v^{q-b} + u^{q-a}v^{p-b}$ (or *vice-versa*). Then $P = 1 + ... + w^{m_1}u^a v^b$ and $R = 1 + \ldots + w^{m_2}(u^{p-a}v^{q-b} + u^{q-a}v^{p-b})$ but then P is not a Q-polynomial. □

3 Trees

We call *nodes* the vertices of a tree.

Definition 3.1 (Splitting $Q(T)$) For a rooted tree T with root r, and $i = 0, 1, 2$, we define

$Q_i(T, u, v, w) := \sum_{\beta i(X,Y,F)} u^{|X|}v^{|Y|}w^{|F|}$ where
$\beta 0(X, Y, F) :\Longleftrightarrow \beta(X, Y, F) \wedge r \notin X \cup Y$,
$\beta 1(X, Y, F) :\Longleftrightarrow \beta(X, Y, F) \wedge r \in X$,
$\beta 2(X, Y, F) :\Longleftrightarrow \beta(X, Y, F) \wedge r \in Y$.

Hence : $Q_1(T, u, v, w) = Q_2(T, v, u, w)$ and $Q(T, u, v, w) = Q_0(T, u, v, w) + Q_1(T, u, v, w) + Q_2(T, u, v, w)$.

Lemma 3.2 *The degree of the root of T is p if and only if $Q_1(T)$ has exactly p monomials where w has degree 1 (these monomials must be all of the form uvw).*

Facts 3.3 If r is the root of T, then $Q_0(T, u, v, w) = Q(T - r, u, v, w)$ and it is a product of p Q-polynomials.

Definition 3.4 (Rooted Trees Defined by Binary Terms) The nullary symbol $\bullet$ denotes S_0, a single node that is the root. If A and B are disjoint rooted trees, then $A \square B$ denotes $A \cup B$ augmented with one edge between the roots of A and B; the root of $A \square B$ is defined as that of A.

It is clear that every rooted tree T is denoted by a term t written with $\Box$ and $\bullet$. The nodes of T are in bijection with the occurrences in t of $\bullet$ and its edges with those of $\Box$.

Examples and Remarks 3.5

(1) If B is a rooted tree, then $\bullet \Box B$ is B with a new edge $x - root_B$ such that x is the root of $\bullet \Box B$. Similarly, $B \Box \bullet$ defines B with a new edge $x - root_B$ and $root_B$ is the root of the new tree.
(2) We have the following partial commutativity rule:
$(A \Box B) \Box C = (A \Box C) \Box B$.
(3) Every rooted tree T can be expressed as follows:
$T = (\ldots((\bullet \Box A_1) \Box A_2) \Box \ldots) \Box A_p$
where p is the degree of the root. This expression is unique up to the partial commutativity rule (2).
(4) $Q(A \Box B) = Q(B \Box A)$.

Proposition 3.6 (Inductive Computations) *We recall that $Q_2(T)$ is obtained from $Q_1(T)$ by exchanging u and v.*

(1) $Q_1(\bullet) = 0$ *and* $Q_0(\bullet) = 1$.
(2) *For rooted rooted trees A and B, we have*

$$Q_1(A \Box B) = Q_1(A)Q(B) + w(Q_1(A) + uQ_0(A))(Q_2(B) + vQ_0(B)),$$
$$Q_0(A \Box B) = Q_0(A)Q(B),$$

where $Q(B) = Q_0(B) + Q_1(B) + Q_2(B)$.
(3) *For every $p > 0$ and rooted tree B, we also have:*

$$Q_1(S_p) = u((1 + vw)^p - 1),$$
$$Q_0(S_p) = 1,$$
$$Q(S_p) = u((1 + vw)^p - 1) + v((1 + uw)^p - 1) + 1.$$
$$Q_1(S_p \Box B) = u((1 + vw)^p - 1)Q(B) + uw(1 + vw)^p(Q_2(B) + vQ_0(B)),$$
$$Q_0(S_p \Box B) = Q(B) = Q_0(B) + Q_1(B) + Q_2(B).$$

Proof

(1) Easy to obtain from the definitions.
(2) Can be derived from (1) or checked directly from the definitions. □

Remark 3.7 (1) If $T = ((\bullet \Box A) \Box B) \Box C$, we have

$$Q_0(T) = Q(A)Q(B)Q(C).$$
$$Q_1(T) + uQ_0(T) =$$
$$u(Q(A) + wQ_2(A) + vwQ_0(A))[Q(B) + wQ_2(B) + vwQ_0(B)]$$
$$(Q(C) + wQ_2(C) + vwQ_0(C)).$$

From $Q_1(T)$, we can get $Q_2(T)$ and so, $Q_0(T) = Q(T) - Q_1(T) - Q_2(T)$.

If the factorization of $Q_0(T)$ is unique, we can get $Q(A)$, $Q(B)$ and $Q(C)$ from it. Then, from $Q_1(T)$ we may get the factors $Q(A) + wQ_2(A) + vwQ_0(A)$, $Q(B) +$

$wQ_2(B)+vwQ_0(B)$ and $Q(C)+wQ_2(C)+vwQ_0(C)$ if, again, we have a unique factorization.

These observations suggest a method to prove the following:

Conjecture 3.8 For a rooted tree T, the pair $(Q(T), Q_1(T))$ determines T up to isomorphism.

Recall that knowing $Q(T)$ and $Q_1(T)$ is equivalent to knowing $Q_0(T)$, $Q_1(T)$ and $Q_2(T)$.

Remark 3.9 Let us give construction rules for $Q_0(T)$, $Q_1(T)$ and $Q_2(T)$ if $T := A//B//C$ defined as the union of disjoint rooted trees A, B, C whose three roots are fused to form the root of T. (The operation $//$ is not expressible by a term written only with the operation $\square$.)

$Q_0(T) = Q_0(A)Q_0(B)Q_0(C)$.
$u^2(Q_1(T) + uQ_0(T)) =$
$\quad (Q_1(A) + uQ_0(A))[Q_1(B) + uQ_0(B)](Q_1(C) + uQ_2(C))$.
More generally:
$Q_0(A_1//.../A_p) = Q_0(A_1) \times ... \times Q_0(A_p)$.
$u^{p-1}(Q_1(A_1//.../A_p) + uQ_0(A_1//.../A_p)) =$
$\quad (Q_1(A_1) + uQ_0(A_1)) \times ... \times (Q_1(A_p) + uQ_0(A_p))$.
Since the expression of $T = A//B//C$ is not unique (the operation $//$ is associative), so neither is the factorization $Q_0(A)Q_0(B)Q_0(C)$ of $Q_0(T)$. □

4 Caterpillars

Definition 4.1 (Rooted Caterpillars) A *rooted caterpillar* C is a rooted tree consisting of a nonempty path P (having at least one edge) and, possibly, additional pendent edges attached to some nodes of this path. The root is an end of the path P.

More precisely, we denote C by $(d_k, d_{k-1},... ,d_1)$ where $k > 0$, $d_i > 0$ for $i = 1,...,k$. The path P is $x_k - x_{k-1} - ... - x_1$, the root is x_1, a node x_j has $d_j - 1$ pendent edges outside of P if $j < k$, and d_k such edges if $j = k$. The number of nodes is thus $d_1 + d_2 + ... + d_k + 1$. If $k = 1$, then C is the star S_{d_1} consisting of one node adjacent to d_1 leaves.

The *length* $\ell en(C)$ of C is the length of a longest path from the root. Hence it is k. See Example 4.2 and Fig. 1 for an example.

We denote by $C \restriction p$ where $p > 0$ the rooted subcaterpillar that is the union of the paths from the root of C having length at most p. Its root is x_1. Hence, $C \restriction \ell en(C) = C$.

A *rooted subcaterpillar* $C' = (V', F)$ of $C = (V, E)$ can be defined by a triple (X, Y, F) such that $X \cup Y = V' \subseteq V$, $F \subseteq E$, $\beta 1(X, Y, F) \vee \beta 2(X, Y, F)$ holds (cf. Definition 3.1 for $\beta 1$ and $\beta 2$) and $|F| - 1 = |X| + |Y|$. The latter condition ensures the connectedness of the corresponding subgraph. These triples yield the monomials of $Q_1(C) + Q_2(C)$ of the form $u^a v^b w^{a+b-1}$, $a, b > 0$. Hence, from

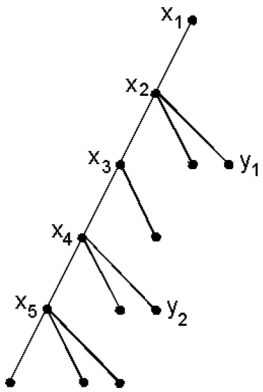


Fig. 1 Example of a caterpillar of length 5. It is discussed in Example 4.2

$Q_1(C) + Q_2(C)$, one can determine the number of subcaterpillars having a given number of edges.

Example 4.2 Figure 1 shows $C = (3, 3, 2, 3, 1)$ of length $k = 5$. Then $C \restriction 3 = (2, 3, 1)$. It consists of nodes x_1, x_2, x_3, x_4 and 3 nodes adjacent to x_2 and x_3.

The subgraph C' induced on $\{x_1, x_2, x_3, x_4, x_5, y_1, y_2\}$ is a subcaterpillar, denoted by $(2, 1, 2, 1)$. The corresponding monomial in $Q_1(C)$ is $u^5v^2w^6$. Let us remove from C' the edge $x_3 - x_4$, so defining D. This graph is not connected, hence is not a subcaterpillar. It yields the two monomials $u^4v^3w^5$ and $u^5v^2w^5$ of $Q_1(C)$ because x_4 can be put either in X or in Y.

Proposition 4.3 *The length of a rooted caterpillar C can be determined from $Q(C)$.*

Proof From $Q(C)$, one can check whether C is S_p. If this is not the case, then C is denoted by $(d_k, d_{k-1}, ..., d_1)$ where $k > 1$. Its length is k. Its number n of nodes is $d_1+d_2+...+d_k+1$. Its number n_1 of nodes of degree 1 is $d_1+d_2+...+d_k+1-k$. The numbers n and n_1 can be determined from $Q(C)$ by Proposition 2.3. So can be the length equal to $n - n_1$. □

Lemma 4.4 *Let B be a rooted tree and $p \geq 0$. From $Q_1(S_p \square B)$ and $Q(S_p \square B)$, one can determine $Q_1(B)$ and $Q(B)$.*

Proof From $Q_1(S_p \square B)$ one can get p by Lemma 3.2 and $Q_2(S_p \square B)$ by Definition 3.1. Hence one gets $Q_0(S_p \square B) = Q(S_p \square B) - Q_1(S_p \square B) - Q_2(S_p \square B)$.

We have $uwQ_2(B)(1+wv)^p = Q_1(S_p \square B) - uQ_0(S_p \square B)((1+wv)^p - 1)$ and $Q(B) = Q_0(S_p \square B)$, which gives $Q_2(B)$, whence also $Q_1(B)$. □

Proposition 4.5 *Two rooted caterpillars C and D are isomorphic if and only if $Q(C) = Q(D)$ and $Q_1(C) = Q_1(D)$.*

Proof Assume $Q(C) = Q(D)$ and $Q_1(C) = Q_1(D)$. One can check if one of C or D is a star. If so, the other is isomorphic. Otherwise, let us express them as $C = S_p \square C'$ and $D = S_q \square D'$. Since $Q(C) = Q(D)$, we have $p = q$. By

Lemma 4.4, we have $Q(C') = Q(D')$ and $Q_1(C') = Q_1(D')$. The result follows by induction. □

This gives a special case of Conjecture 3.8. Actually, we can do better.

Theorem 4.6 *Two rooted caterpillars C and D are isomorphic if and only if $Q_1(C) = Q_1(D)$.*

Lemma 4.7 *Let C be a caterpillar $(d_k, ..., d_1)$ of which we know (only) $Q_1(C)$.*

(1) We can determine n, d_1 and check if $k = 1$.
(2) For $p \leq k$ and from the knowledge of the top part $C \restriction p = (d_p, ..., d_1)$, we can determine if $p = k$ (in this case we know C) and, otherwise, we can determine d_{p+1}.
(3) We can determine successively $d_1, ..., d_k$ hence C.

Proof

(1) The number n of nodes is the maximum value of $p + q$ such that $u^p v^q w^m$ is a monomial of $Q_1(G)$ for some m. By Lemma 3.2, we get the degree d_1 of the root. If $n = d_1 + 1$, then $k = 1$ and $C = S_{d_1}$.
(2) We know $Q_1(C)$. Assume that we know the top part $C \restriction p = (d_p, \ldots, d_1)$ for some $p \leq k$. If $n = d_p + \ldots + d_1 + 1$, then $C = (d_p, \ldots, d_1)$ and $p = k$.

Otherwise, the rooted subcaterpillars of C having $p + 1$ edges are of two forms:

(i) either a subcaterpillar of C consisting of a path $y - x_{p+1} - x_p - x_{p-1} - \ldots - x_1$ for some y adjacent to x_{p+1} and there are d_{p+1} such cases,
(ii) or a subcaterpillar of $C \restriction p = (d_p, \ldots, d_1)$.

As we know $C \restriction p$, we know $Q_1(C \restriction p)$, hence the number s_p of subcaterpillars of type (ii).

The monomials in $Q_1(C)$ of the form $u^a v^b w^{p+1}$ with $p + 1 = a + b - 1$ correspond to the rooted subcaterpillars having $p+1$ edges, hence their number is $s_p + d_{p+1}$. As we know k_p, we get d_{p+1}.
(3) We can determine d_1 by (1) and then $d_2, \ldots, d_k$, hence C. □

This lemma proves Theorem 4.6. In the example of Fig. 1, there are $d_5 = 3$ paths of type (i). The corresponding monomials are $u^3 v^3 w^5$. (Some subcaterpillars of type (ii) yield the same monomials).

5 Computing Q_0, Q_1, Q_2 by a Deterministic Automaton

We use the tools of [1] for computations in rooted trees, not only those defining caterpillars.

In a logical structure S, the *multispectrum* of a formula $\varphi(X, Y, Z)$ where X, Y, Z are set variables is the multiset of triples $(|X|, |Y|, |Z|)$ such that $S \models$

$\varphi(X, Y, Z)$. The associated polynomial $P_{\varphi,S}(u, v, w)$ is defined as the sum of the monomials $u^{|X|}v^{|Y|}w^{|Z|}$ such that $S \models \varphi(X, Y, Z)$. The coefficient of $u^{|X|}v^{|Y|}w^{|Z|}$ in $P_{\varphi,S}(u, v, w)$ is thus the number of triples $(|X|, |Y|, |Z|)$ such that (X, Y, Z) satisfies φ in S. Hence, we can see $P_{\varphi,S}$ as a compact notation of the multispectrum of φ in S. Logical descriptions of graph polynomials have been investigated by Makowsky *et al.* in [3].

Let us go back to rooted trees. Each of them is defined by a term t in $T(\{\Box, \bullet\})$, the set of finite terms written with the binary operation symbol $\Box$ and the nullary symbol $\bullet$, cf. Definition 3.4. For an example, the term $t = \Box(\bullet, \Box(\bullet, \bullet))$ defines the caterpillar (1,2) that is a path of length 2. Such an expression is not unique by Remark 3.5(2).

We let $val(t) = T$ be the tree defined by a term t. The nodes of T are the leaves of t, considered as an ordered tree in the standard way, hence the occurrences of $\bullet$; the root of T is the leftmost leaf of t. The edges of T are the occurrences of $\Box$, hence, the internal nodes of t.

The formulas $\beta i(X, Y, F)$, $i = 0, 1, 2$, are to be checked in trees $val(t)$. However they can be rewritten as $\gamma i(X, Y, F)$ to be checked in the logical structure representing t. Actually, we use logically expressed properties of terms rather than explicit logical formulas. We have:

$Q_i(val(t), u, v, w) = \sum_{\gamma i(X,Y,F)} u^{|X|}v^{|Y|}w^{|F|}$

where X, Y denote sets of leaves of t (seen as a tree) and F sets of internal nodes. Hence, $Q_i(val(t), u, v, w)$ is a compact notation for the multispectrum of γi in t (cf. [1]), as observed above.

Example 5.1 We have: $Q(K_3) = 1 + 6uvw + 3uv^2w^2 + 3u^2vw^2$. corresponding multispectrum of $\gamma 0(X, Y, F) \vee \gamma 1(X, Y, F) \vee \gamma 2(X, Y, F)$ is the multiset {(0,0,0):1, (1,1,1):6, (1,2,2):3, (2,1,2):3} where (1,2,2):3 means that the triple (1,2,2) has 3 occurrences, giving the term $3uv^2w^2$ in $Q(K_3)$.

Following the general method of [1], we represent the set variables X, Y, F in terms over $\{\Box, \bullet\}$ by Boolean labels. We label each leaf (holding $\bullet$) by a pair of Booleans as follows:

(1,0) to mean that the corresponding node is in $X - Y$,
(0,1) to mean that the corresponding node is in $Y - X$,
(0,0) to mean that the corresponding node is not in $X \cup Y$,
(1,1) means that the corresponding node is in $X \cap Y$.

However, each formula $\gamma i(X, Y, F)$ is false if $X \cap Y$ is not empty, hence, this last case has no reason to occur and only three node labels are actually useful.

We label an internal node (an occurrence of $\Box$) by a Boolean as follows:

0 to mean that the corresponding edge is not in F,
1 to mean that the corresponding edge is in F.

We define as follows a deterministic automaton $\mathcal{A}$ on labelled terms:

(a) Its states are $c0, c1, c2, d1, d2, Error$. Its accepting states are $c0, c1, c2$ whose corresponding multispectra define respectively Q_0, Q_1, Q_2.
(b) We want that, at a node x of a labelled term t (its labels encode a triple (X, Y, F)):

(b.1) the state is ci if and only if the subterm t/x satisfies $\gamma i(X', Y', F')$ where X', Y', F' are the restrictions of X, Y, F in t/x.
(b.2) The state is di, here $i = 1, 2$, if and only if the subterm t/x satisfies $\gamma i(X', Y', F')$ where X', Y', F' are as above *except that no edge in* F' is incident to the root of $val(t/x)$; this root belongs to X if $i = 1$ and to Y if $i = 2$.

If t/x is $\bullet$, the corresponding (isolated node) can be in X, without any incident edge, and the associated state is $d1$. Such a node may later on, by some labelled operation $(\square, 1)$, be made adjacent to a node in Y. Otherwise, an error will occur and the labelled term will not contribute to the construction of any polynomial. We have the rule:

$(\square, 0)[d2, d1] \to Error.$

The node holding the state $d1$ has no incident edge below it and cannot have any above. To the opposite, we have:

$(\square, 1)[d2, d1] \to c2.$

Here is the automaton. The transitions on nullaries are as follows:

$(\bullet, 00) \to c0$
$(\bullet, 10) \to d1$
$(\bullet, 01) \to d2$
$(\bullet, 11) \to Error.$

The last rule is of no use in an implementation where only the labels 00, 10 and 01 can be attached to nullaries. The transitions on $\square$ are as follows:

$(\square, 0)[c0, s] \to c0$ for $s = c0, c1, c2,$
$(\square, 0)[c1, s] \to c1$ for $s = c0, c1, c2,$
$(\square, 0)[c2, s] \to c2$ for $s = c0, c1, c2,$
$(\square, 0)[d1, s] \to d1$ for $s = c0, c1, c2,$
$(\square, 0)[d2, s] \to d2$ for $s = c0, c1, c2,$
$(\square, 1)[c1, c2] \to c1,$
$(\square, 1)[c1, d2] \to c1,$
$(\square, 1)[c2, c1] \to c2,$
$(\square, 1)[c2, d1] \to c2,$
$(\square, 1)[d1, c2] \to c1,$
$(\square, 1)[d1, d2] \to c1,$
$(\square, 1)[d2, c1] \to c2,$
$(\square, 1)[d2, d1] \to c2,$

All other transitions (an example is given above) yield $Error$, and $Error$ propagates bottom-up. There are 26 transitions not yielding $Error$. These transitions correspond to the inductive rules of Proposition 3.6.

For an example, the rule:

$$\begin{aligned} Q_1(A\Box B) = {} & Q_1(A)Q_0(B) + Q_1(A)Q_1(B) + Q_1(A)Q_2(B) + \\ & wQ_1(A)Q_2(B) + wQ_1(A)vQ_0(B) + \\ & wuQ_0(A)Q_2(B) + wuQ_0(A)vQ_0(B). \end{aligned}$$

yields the following transitions:

$(\Box, 0)[c1, s] \to c1$ for $s = c0, c1, c2$,
$(\Box, 1)[c1, c2] \to c1$,
$(\Box, 1)[c1, d2] \to c1$,
$(\Box, 1)[d1, c2] \to c1$,
$(\Box, 1)[d1, d2] \to c1$,

The *counting computation* of $\mathcal{A}$ with accepting state ci yields the multispectrum of γi hence $Q_i(val(t))$ for any t in $T(\{\Box, \bullet\})$, a term without Boolean labels.

Example 5.2 Let $t = \Box(\bullet, \Box(\bullet, \bullet))$ defining the caterpillar (1,2), a path of length 2. We give three examples of labellings of t and the corresponding runs of the automaton. The states at the root are in the second column.

$t =$	$\Box$	$\bullet$	$\Box$	$\bullet$	$\bullet$	Triple/monomial
Label	0	00	1	10	01	
State	c0	c0	c1	d1	d2	(1,1,1) / uvw in Q_0
Label	1	10	0	01	10	
State	$Error$	d1	$Error$	d2	d1	None
Label	1	10	1	01	10	
State	c1	d1	c2	d2	d1	(2,1,2) / u^2vw^2 in Q_1

6 Two Subpolynomials of Q

This section suggests research directions in view of proving the conjecture, at least for caterpillars. We consider two subpolynomials of Q for trees T.

6.1 Counting Full Sets of Edges

We let $Q_{full}(T)$ consist of the monomials $u^a v^b w^m$ of $Q(T)$ such that $a + b$ is the number of nodes. It corresponds to considering triples (X, Y, F) such that every node is an end of some edge in F. Hence it is a subpolynomial of $Q_1(T) + Q_2(T)$.

An automaton for computing it is obtained from $\mathcal{A}$ by replacing state $c0$ by $Error$. The accepting states are $c1$ and $c2$.

Question 6.1 For caterpillars T, does $Q_{full}(T)$ determine T ?

6.2 Counting Subtrees

We now restrict the polynomials Q to triples inducing connected subgraphs of trees, hence to subtrees by defining $Q_{conn}(T, u, v, w) := \sum_{\mu(X,Y,F)} u^{|X|}v^{|Y|}w^{|F|}$ where $\mu(X, Y, F)$ means that (X, Y, F) induces a subgraph as in Definition 2.6, except that this subgraph must now be connected and nonempty.

The coefficient of a monomial $u^p v^q w^m$ is the number of triples (X, Y, F) such that $\mu(X, Y, F)$ holds and $(p, q, m) = (|X|, |Y|, |F|)$. We must have $m = p+q-1$ as a connected subgraph is a tree.

Fact 6.2 $Q_{conn}(T, u, v, w)$ is the sum of the monomials $u^p v^q w^m$ of $Q(T, u, v, w)$ such that $m = p + q - 1$.

We do not know whether $Q(T)$ is computable from $Q_{conn}(T)$.

Conjecture 6.3 Two trees T and R are isomorphic if and only if $Q_{conn}(T) = Q_{conn}(R)$.

This conjecture implies Conjecture 2.5.

We define a deterministic automaton $\mathcal{B}$ on labelled terms intended to compute $Q_{conn}(T, u, v, w)$. It differs from $\mathcal{A}$ defined above (we name states in a different way although there are some similarities).

(1) Its states are $a1, a2, f1, f2, b, e$ and $Error$.
(2) Its accepting states are $a1, a2, b$ whose corresponding multispectra define collectively Q_{conn}.

The state e correspond to subtrees of T with no node in X or Y, no edge in F. The states $f1$, $f2$ correspond to isolated vertices in the subgraph (X', Y', F') that are in X' and Y' respectively. The states a_1, a_2 correspond to subtrees induced by (X', Y', F') whose root is that of the tree $val(t)$ and is in X', resp. Y'. The state b corresponds to the case where some subtree T' meets the required conditions but its root is not that of T.

Here is the automaton. The transitions on nullaries are:

$(\bullet, 00) \to e$
$(\bullet, 10) \to f1$
$(\bullet, 01) \to f2$
$(\bullet, 11) \to Error.$

The last one is of no use in an implementation where only the labels 00, 10 and 01 can be attached to nullaries. The transitions on $\square$ are:

$(\square, 0)[e, e] \to e$,
$(\square, 0)[e, s] \to b$ for $s = a1, a2, b$,
$(\square, 0)[s, e] \to s$ for $s = a1, a2, b, f1, f2$.
$(\square, 1)[a1, a2] \to a1$,
$(\square, 1)[a1, f2] \to a1$,
$(\square, 1)[a2, a1] \to a2$,
$(\square, 1)[a2, f1] \to a2$,
$(\square, 1)[f1, f2] \to a1$,
$(\square, 1)[f2, f1] \to a2$,
$(\square, 1)[f1, a2] \to a1$,
$(\square, 1)[f2, a1] \to a2$.

The *counting computation* of $\mathcal{B}$ with accepting states $a1, a2, b$ yields $Q_{conn}(val(t))$ for any t in $T(\{\square, \bullet\})$ (hence without the Boolean labels).

Example 6.4 Let $t = \square(\bullet, \square(\bullet, \bullet))$ that defines the caterpillar (1,2), a path of length 2.

$t =$	$\square$	$\bullet$	$\square$	$\bullet$	$\bullet$	Triple/monomial
Label	0	00	1	10	01	
State	b	e	$a1$	$f1$	$f2$	(1,1,1) / uvw in Q_{conn}
Label	1	10	0	01	10	
State	$Error$	$f1$	$Error$	$f2$	$f1$	None
Label	1	10	1	01	10	
State	$a1$	$f1$	$a2$	$f2$	$f1$	(2,1,2) / u^2vw^2 in Q_{conn}

The advantage of $Q_{conn}(T)$ over $Q(T)$ is that it has much less monomials, hence it is easier to compare $Q_{conn}(T)$ and $Q_{conn}(R)$ in concrete cases.

Examples 6.5 We compare $C := (1, 2, 1, 1, 1)$ and $D := (2, 1, 1, 1, 1)$.

$Q_{conn}(C) = 12uvw + 6uv^2w^2 + 6u^2vw^2 + uv^3w^3 + u^3vw^3 + 10u^2v^2w^3 +$
$5u^2v^3w^4 + 5u^3v^2w^4 + u^2v^4w^5 + u^4v^2w^5 + 4u^3v^3w^5 + u^3v^4w^6 + u^4v^3w^6$.

The polynomial $Q(C)$ contains $1+36+22+22+4+4+24+8+8+6+6 = 141$ additional monomials (where $36u^2v^2w^2$ counts for 36).

We have $Q_{conn}(C) - Q_{conn}(D) = 2u^2v^2w^3 + u^2v^3w^4 + u^3v^2w^4$, which shows in particular that $Q_{conn}(D)$ is a subpolynomial of $Q_{conn}(C)$. □

The restrictions Q_{full} and Q_{conn} are somewhat orthogonal: combining them yields a polynomial with two monomials that carries almost no information.

7 Conclusion

We made small progress towards the proofs of Conjecture 2.5 by Dod et al. [2], and some related ones.

Acknowledgments We thank Peter Tittmann and the referee for useful comments.

References

1. Courcelle, B., Durand, I.: Computations by fly-automata beyond monadic second-order logic. Theor. Comput. Sci. **619**, 32–67 (2016)
2. Dod, M., Kotek, T., Preen, J., Tittmann, P.: Bipartition polynomials, the ising model, and domination in graphs. Discussiones Mathematicae, Graph Theory **35**(2) (2015). https://doi.org/10.7151/dmgt.1808
3. Makowsky, J., Ravve, E., Kotek, T.: A logician's view of graph polynomials. Ann. Pure Appl. Log. **170**(9), 1030–1069 (2019)
4. Tittmann, P.: Graph Polynomials: The Eternal Book, (2003). https://www.researchgate.net/publication/377572474_Graph_Polynomials_The_Eternal_Book

NP-completeness by First-order and Quantifier-free Interpretations and Related Topics

Elias Dahlhaus

Abstract We consider two signatures (sets of relational symbols) L_1 and L_2. A map I that maps each finite L_1-structure to a finite L_2-structure is called an *interpretation* if for each L_1-structure M, the domain of $I(M)$ is the disjoint union of powers of the domain of M and the relations of $I(M)$ are defined by first-order formulas over the relations of M (the formulas do not depend on the special M). It has been shown that many reductions to NP-complete decision problems are interpretations. As examples, the problems Satisfiability, Clique, and Hamilton Cycle all are NP-complete via interpretations. Here, we show that also Graph Colouring and Exact Cover are complete in NP with respect to interpretations. It also could be shown that 3-Satisfiability is not NP-complete with respect to interpretations if it is described in a certain way as a class of finite structures If we model 3-Satisfiability the same way as the satisfiability problem then it is not trivially clear that it is not NP-complete through interpretations as reductions. In this article, we will show that 3-Satisfiability under this representation neither is NP-complete by interpretations. On the other hand, 3-Satisfiability can be modelled as a class of finite models that is defined by an existential second order formula with a universal first-order part. It will be shown that 3-Satisfiability is complete by quantifier-free interpretations in such classes of finite models. As consequence, also 3-colouring is complete in that class by quantifier-free interpretations. Also some variations of 3-Satisfiability are complete in that class by interpretations. It will be mentioned that interpretations can be used for trivial proofs that many NP-complete decision problems cannot be formulated in logics like Least-Fixed-Point.

E. Dahlhaus (✉)
Algorithms Group, Department of Computer Science, Darmstadt University of Technology, Darmstadt, Germany
e-mail: dahlhaus@algo.informatik.tu-darmstadt.de

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_14

1 Introduction

A still unsolved research problem in theoretical computer science is whether nondeterministic polynomial time NP and deterministic polynomial time P are identical classes. Certain decision problems in NP, like the satisfiability problem for propositional formulas (SAT for short) or the travelling salesman problem (TSP) have the property that each decision problem being in NP can be reduced to each of them by a polynomial time algorithm. Such decision problems are also called NP-complete (see [19]). Here, the NP-completeness of SAT is proved by constructing, for each nondeterministic Turing machine that has a polynomial time bound, a polynomial time computable map that constructs, for each input, a propositional formula that is satisfiable if and only if there is a computation path that leads to an accepting state.

To prove other NP-completeness results, one reduces a known NP-complete decision problem to the decision problem for which NP-completeness has to be shown. Fagin [13] considered decision problems in NP as classes of finite models of a fixed similarity type. For example, decision problems on graphs can be considered as classes of finite models of the similarity type containing one binary relation symbol. Fagin now showed that an isomorphism closed class of finite models is in NP if and only if it is the class of finite models satisfying a certain existential second order formula. It might be interesting to consider the fine structure of reductions to NP-complete decision problems. Lovász and Gács [21] were the first showing that every class of finite models that can be defined by an existential second order formula (being in NP) can be reduced to SAT by a first-order definable reduction (called *elementary reduction* in [21], here it is called an *interpretation*). In [7, 8] it has been shown that CNF-SAT (the satisfiability problem for conjunctive normal forms), CLIQUE, and Hamilton Cycle all are NP-complete by quantifier-free definable reductions (also called *quantifier-free interpretation*). On the other hand, 3-SAT (the restriction of the satisfiability problem of conjunctive normal forms with 3 literals per clause) is not NP-complete with respect to first-order definable interpretations. In [18], Immerman showed that certain decision problems are complete in NL and P with respect to interpretations if a total ordering on the domain belongs to the similarity type of the decision problem to be reduced. Interpretations are also used to show that Least-Fixed-Point formulas can be transformed into a so-called Skolem normal form [9], see also [20]. A certain decision problem was shown to be complete in Least-Fixed-Point by quantifier-free interpretations, and the formulation of this decision problem by a Skolem normal form leads to a Skolem normal form for all decision problems that can be formulated as a Least-Fixed-Point problem. For further readings on Least-Fixed-Points, we refer to [16]. Here, we consider both, decision problems that are and that are not NP-complete with respect to interpretations. An interesting connection between interpretations and generalized quantifiers is discussed in great detail in [22] and also in [10], linking our approach to finite model theory. The paper is intended to give an overview on the usefulness of interpretations. At the same time, it solves some minor open research

problems. The main goal is to show that interpretations are a useful tool to classify NP-complete problems and also to simplify proofs in finite model theory.

1.1 Outline of the Paper

In Sect. 2, we first collect decision problems known to be NP-complete with respect to interpretations. These are CNF-SAT, CLIQUE and Hamilton Cycle. Next, NP-completeness of Graph colouring with respect to quantifier-free interpretations will be shown. Furthermore, some more decision problems being NP-complete by interpretations are listed. In Section 3, we consider decision problems that are NP-complete but not NP-complete with respect to interpretations. It is shown that 3-SAT is not NP-complete with respect to interpretations, also if we model 3-SAT as a general CNF-SAT with 3 literals per clause and not as in [7, 8], where the clauses were modelled by ternary relations on the variable set. This solves a problem of I.A. Stewart [26]. It is also shown that 3-SAT as in the formulation in [7, 8] is complete for decision problems expressible by existential second-order formulas with a universal first-order body in prenex normal form by quantifier-free interpretations. This is also true both for 3-colouring (the question whether a certain graph is colourable by three colours?) and for generalized satisfiability problems satisfying certain conditions [24]. Finally, it will be mentioned that the status of 3-Dimensional Matching is open. In Sect. 4, we argue how interpretations can be used to show that none of the decisions problem such as 3-SAT, 3-colouring and also any decision problem being NP-complete by interpretations can be expressed in logics related to P such as the Least-Fixed-Point. In particular, the simplicity of the corresponding proof is interesting.

2 Interpretations

2.1 Basic Definitions

A *similarity type* or signature is a set $L = \{(R_1, n_1), \ldots, (R_q, n_q)\}$ of relation symbols R_i together with their arity n_i. The class of structures of a signature L is denoted by $Mod(L)$. Here, we are interested in finite structures of a fixed similarity type. We denote the class of finite structures of a similarity type L by $Mod_{fin}(L)$. A map $I : Mod(L) \to Mod(L')$ is called an *interpretation* if each structure $(M, R_1, \ldots, R_k)$ of signature L maps to a structure $(M', R'_1, \ldots, R'_{k'})$ that satisfies the following properties:

1. M' is the disjoint union of first-order definable subsets of powers of M, i.e., $M' = \bigcup_i \{(i, x_1, \ldots, x_{k_i}) : x_j \in M, j = 1, \ldots, k_i, \phi_i(x_1, \ldots, x_{k_i})$ is valid in $(M, R_1, \ldots, R_k)\}$. Here, each $\phi_i(x_1, \ldots, x_{k_i})$ is a first-order formula on L.

In most cases, M' is the disjoint union of powers of M.

2. The relations R'_i are defined by first-order formulas over L using the relations $R_1, \ldots, R_k$. More precisely, for each relation symbol R'_j with arity m and each $i_1, \ldots, i_m$, there is a first-order formula $\psi_{i_1,\ldots,i_m}(y_1, \ldots, y_m))$ that is satisfied in $(M, R_1, \ldots, R_k)$ if and only if $R'_j((i_1, y_1), \ldots, (i_m, y_m))$ is satisfied in $(M', R'_1, \ldots, R'_{k'})$. The $y_1, \ldots, y_m$ are tuples of elements of the domain M.

If the formulas defining an interpretation I are quantifier-free then we call I a *quantifier-free interpretation*.

Remark 1 Any interpretation can be computed in logarithmic space if we restrict it to finite structures.

Theorem 1 ([13]) *A class of finite structures of a fixed similarity type and being closed under isomorphisms is in NP if and only if it is the class of finite structures satisfying a certain existential second order formula. The formula begins with existential quantifiers over relations followed by a first-order formula.*

We now consider the question which NP-complete decision problems are NP-complete by (quantifier-free) interpretations, i.e., to which NP-complete decision problems one can reduce any decision problem in NP by an interpretation. Here, we say that an interpretation I *reduces* a decision problem $\mathbf{M} \subseteq Mod_{fin}(L)$ given as set of finite structures over L, to a decision problem $\mathbf{M}' \subseteq Mod_{fin}(L')$ if the following holds: For any $(M, R_1, \ldots, R_k) \in Mod_{fin}(L)$, we have $(M, R_1, \ldots, R_k) \in \mathbf{M}$ if and only if $I(M, R_1, \ldots, R_k) \in \mathbf{M}'$. We then also call $\mathbf{M}$ the *source problem* and $\mathbf{M}'$ the *target problem* of the interpretation I.

2.2 Known Completeness Results

First results dealing with interpretations as reductions to NP-complete decision problems were given by Lovász and Gács [21]. They described the satisfiability problem for propositional formulas as a class of finite structures and showed its completeness by interpretations. In a short remark, they mentioned that the satisfiability problem of propositional formula (as switching circuits) can easily be reduced to satisfiability of conjunctive normal forms. In [7, 8], the problems SAT, CLIQUE and the (directed) Hamiltonian Cycle problem were described as classes of finite models that could be shown to be NP-complete by quantifier-free interpretations. We describe once more these decision problems as classes of finite structures. We assume that the reader is familiar with the classical formulation of these decision problems as given, for example, in [19].

SAT

1. *Unary relation symbols*: C as a relation symbol for the clauses, V as a relation symbol for the variables
2. *Binary relation symbols*: $P(c, v)$ meaning that v appears as a non-negated variable in clause c and $N(c, v)$ meaning that v appears negated in c.
   A formulation of SAT by an existential second order formula then is as follows:

$$\exists W \forall c(C(c) \rightarrow \exists x(V(x) \wedge ((P(c, x) \wedge W(x)) \vee (N(c, x) \wedge \neg W(x))$$

CLIQUE

1. We are given two unary relation symbols V and N. V represents the set of vertices. N represents the size of the clique we are looking for.
2. A binary relation symbol E represents the adjacency of two vertices.
   A formulation by an existential second order formula then is as follows:
   $\exists I : [\forall n\,(N(n) \rightarrow \exists! v(V(v) \wedge I(n, v)))] \wedge$
   $\forall n_1, n_2, v_1, v_2\,[(I(n_1, v_1) \wedge I(n_2, v_2) \wedge n_1 \neq n_2) \rightarrow (v_1 \neq v_2 \wedge E(v_1, v_2))]$.

Hamilton Cycle is a class of directed graphs modelled as finite structures over the signature that contains one binary relation symbol and nothing else. The formulation by an existential second order formula is left to the interested reader.

Theorem 2 ([7, 8]) *SAT, CLIQUE, and Hamilton Cycle are NP-complete with respect to quantifier-free interpretations.*

2.3 Further Decision Problems being NP-complete by Interpretations

We consider some additional NP-complete decision problems that are also complete by interpretations.

2.3.1 Graph Colouring (Chromatic Number)

As decision problem, it is the question whether given a graph $G = (V, E)$ and natural number n, can the vertices of G be coloured by at most n colours, such that adjacent vertices have different colours?

Again, we formulate the problem as one for a class of finite structures.

1. Introduce two unary relation symbols V and N representing the vertices and the available colours, respectively.
2. A binary relation symbol E represents adjacent vertices.

Graph Colouring now can be expressed by the existential second-order formula
$\exists C \forall v\,[V(v) \rightarrow \exists! c(N(c) \wedge C(v,c))] \wedge$
$\forall v, w, c, d\,[(E(v,w) \wedge v \neq w \wedge C(v,c) \wedge C(w,d)) \rightarrow c \neq d]$.
As exemplification of typical reasoning in the area we present a full proof of

Theorem 3 *Graph Colouring is NP-complete by quantifier-free interpretations.*

Proof As starting remark it seems that the reduction from 3-SAT to Graph Colouring given in [19] is not correct. So we design a reduction by our own. This reduction moreover is a quantifier-free interpretation. Let an instance (M, V, C, P, N) of SAT be given.

We may assume that there exist no clause c containing a variable v such that both $N(c, v)$ and $P(c, v)$ are satisfied. Otherwise this clause is inherently satisfied and we could erase it. The graph to be coloured and the number of colours are defined as follows:

1. We introduce two vertices T and F (for true and false) and a copy of V as a set of indifferent vertices Ind. All these vertices are made pairwise adjacent.
2. For each variable v, we introduce a vertex v^+ and a vertex v^- representing v and its negation, respectively. v^+ and v^- are adjacent, and both are adjacent to all vertices in Ind.
3. For each clause c, we provide a copy $Ind_c \cup \{T_c, F_c\}$ of $Ind \cup \{T, F\}$. The vertices of each copy are pairwise adjacent. If $(c, v) \in P$ then for the corresponding vertex v_c in Ind_c an edge $v^+ v_c$ is provided. If $(c, v) \in N$, then an edge $v^- v_c$ is provided. All other v_c and F_c are adjacent to F. T_c is adjacent to all vertices in $Ind \cup \{F\}$.
4. The set of colours is $Ind \cup \{T, F\}$, so its number is the number of variables plus 2.

From a satisfying assignment of (M, V, C, P, N) one gets a colouring as follows. If a variable v is assigned to be true then v^+ gets the same colour as T and v^- gets the same colour as F. Otherwise, v^+ gets the same colour as F and v^- gets the same colour as T. Note that v^+ and v^- can only be coloured by the colours of T or F, because they are both adjacent to all vertices in Ind. T_c must have the same colour as T. $Ind_c \cup \{T_c, F_c\}$ can be coloured consistently if and only if there is an edge $v^+ v_c$ or an edge $v^- v_c$ such that v^+ and v^- has the colour of T, respectively. For each clause c, pick a v such that $(c, v) \in P$ or $(c, v) \in N$; in the first case, v is true and in the second case v is false. In both cases, v_c gets the same colour as F. The remaining vertices in $Ind_c \cup \{F_c\}$ get the remaining colours in Ind arbitrarily.

Vice versa, assume that the above graph can be coloured by as many colours as elements of $Ind \cup \{T, F\}$. Then v^+ and v^- get both the colours of T or F, because they are both adjacent to all vertices in Ind. One of v^+ and v^- has the colour of T, the other has the colour of F. Since T_c is adjacent to all vertices in $Ind \cup \{F\}$, it has the same colour as T. Since F_c is adjacent to F, F_c has not the same colour as F. Therefore, one of the vertices in Ind_c has the colour of F. The variable v corresponding to such an element in Ind_c must appear in clause c, because only in that case, v_c is not adjacent to F. The particular node among v^+ and v^- being

adjacent to v_c must have the colour of T. This is true for each clause, and therefore we get a satisfying assignment.

It remains to show that the above reduction is a quantifier-free interpretation. Let (M, V, C, P, N) be an instance of SAT and let (M', V', E', N') be the instance of Graph Colouring obtained by the above reduction. Then the domain M' is the union of V' and N'. V' is the disjoint union of Ind, the set of vertices v^+, the set of vertices v^- , the Ind_c , the set of all T_c, the set of all F_c, and $\{T\}$ and $\{F\}$. It is therefore the disjoint union of three copies of V, one copy of $C \times V$ (the union of all Ind_c), two copies of C, and two copies of M^0 (which is a one element set). Therefore, V' is a disjoint union of quantifier-free definable subsets of powers of M. Here, N' is a subset of V'. So the domain M' is identical to V'. Thus, M' satisfies the conditions of a quantifier-free interpretation.

Clearly, V' is definable by quantifier-free formulas in the language of SAT. For each copy of a power of M, it is a formula that is always true. N' is defined in Ind by a unary formula and in $\{T\}$ and $\{F\}$ by 0-ary formulas that are always true. For all other copies of powers of M, we provide formulas that are always false.

It remains to show that E' can be defined by quantifier-free formulas in the language of SAT. For each pair of copies of powers of M, we define a quantifier-free formula that is valid for x in the one copy and each y in the other copy if and only if xy is in E'. Note that x and y are tuples of elements of M. For example, if K is the copy of M that contains Ind then for x and y in K, xy is in E' if and only if $V(x) \wedge V(y) \wedge x \neq y$. Let K_1 be the copy of M containing the v^+ and let K_2 be copy of M containing the v^-. Then for v_1^+ in K_1 and v_2^- in K_2, $v_1^+ v_2^-$ is in E' if and only if $v_2 = v_1$ and $V(v_2)$ and $V(v_1)$. For the vertices v^+ (or v^-) and w in Ind, only $V(v)$ and $V(w)$ have to be satisfied to be joined by an edge in E'. The requirement that $Ind_c \cup \{T_c, F_c\}$ is a complete subgraph can be expressed as follows:

- $v_c w_d \in E'$ if and only if $c = d \wedge v \neq w \wedge V(v) \wedge V(w) \wedge C(c) \wedge C(d)$.
- $T_c F_d \in E'$ if and only if $c = d \wedge C(c) \wedge C(d)$
- $v_c F_d \in E'$ if and only if $c = d \wedge V(v) \wedge C(c) \wedge C(d)$

The edges related to the appearance of a variable or negated variable in a clause are defined as follows:

- $v^+ w_c \in E'$ if and only if $v = w \wedge P(c, v)$
- $v^- w_c \in E'$ if and only if $v = w \wedge N(c, v)$
- $w_c F \in E'$ if and only if $V(w) \wedge C(c) \wedge \neg P(c, w) \wedge \neg N(c, w)$
- $F_c F \in E'$ is always true.

This finishes the proof. □

2.3.2 Reductions Known from Literature

One can easily verify that the reduction from Graph Colouring to Exact Cover given in [21] is a quantifier-free interpretation. As consequence, Exact Cover is complete in NP by quantifier-free interpretations. The following decision problems

share this property: Set Packing, Node Cover, Set Covering, Feedback Node Set, Feedback Arc Set, Clique Cover, Hitting Set, Undirected Hamilton Circuit. For all these problems, the reductions mentioned in [19] can be verified to be quantifier-free interpretations in the same way as done in Theorem 1. Note that the proof of the NP-completeness of Directed Hamilton Cycle in [19] gives a reduction from Vertex Cover that uses the order of the nodes. But as mentioned before, Directed Hamilton Cycle is NP-complete by quantifier-free interpretations. The reduction from Directed Hamilton Cycle to Undirected Hamilton Cycle in [19] is a quantifier-free interpretation, and therefore also Undirected Hamilton Cycle is NP-complete by quantifier-free interpretations.

2.3.3 Homogeneous Interpretations

In many references an interpretation maps a structure $(M, R_1, \ldots, R_k)$ of L to a structure $(M', R'_1, \ldots, R'_l)$ that satisfies the following properties:

1. M' is a first-order definable subset of a power of M and
2. the relations R'_i are defined by first-order formulas over L.

In the finite model theory community there is a consensus that any interpretation can be transformed into a homogeneous interpretation. However, nowhere in the literature, we were able to find a proof that this always is possible. Most reductions to NP-complete decision problems are done by local replacements (see[15]). In many cases it can be recognized immediately that such a reduction is an interpretation, in many cases, however, not a homogeneous interpretation. On the other hand, one often immediately can show that if an homogeneous interpretation reduces one decision problem to another and if the latter is expressible in a certain logic then also the former problem is expressible in that logic.

Call an interpretation *semi-homogeneous* if the domain M' is the disjoint union of first-order definable subsets of the same power of M. Under the condition that no M^0 is a part of the domain M' of $I((M, R_1, \ldots, R_k)$ one can show that the interpretation I can be transformed into a semi-homogeneous interpretation. In order to make a subset of M^q a subset of M^r with $r > q$, for each tuple in this subset we add $r - q$ copies of the first component to the tuple. First-order definability and also quantifier-free definability are preserved.

We now show how to transform an interpretation that reduces to an NP-complete decision problem into a homogeneous one.

1. Consider the reduction to SAT and replace the domain of the SAT-instance by the Cartesian product with $(x_1, x_2) : x_1, x_2 \in M, x_1 \neq x_2$ and obtain $n(n-1)$ copies of the original SAT-instance. Here, n is the number of elements of the domain M of the instance of the decision problem to be reduced. This extended SAT-instance is satisfiable if and only if the original SAT-instance is satisfiable.

2. Next, reduce the extended SAT-instance to the original decision problem and make the concatenation of the reduction to the extended SAT-instance and of the reduction to the other NP-complete decision problem semi-homogeneous.
3. Note that in the concatenation of these two reductions, the domain of the target instance is the disjoint union

$$\bigcup_i \{(i, x_1, \ldots, x_k) : x_j \in M, j = 1, \ldots, k, (M, R_1, \ldots, R_k) \text{ satisfies } \phi_i(x_1, \ldots, x_k)\}$$

There are always two components x_p and x_q that are unequal. Without loss of generality, we may assume $x_1 \neq x_2$. If the domain of the target instance is the union of m powers of M, we replace in any $(i, x_1, \ldots, x_k)$ the entry x_1 by i copies of x_1 and the entry x_2 by $m - i$ copies of x_2. We may remove the entry i. The first m entries of any tuple give all the information to which i it belongs. This defines a homogeneous interpretation.

3 NP-Complete Decision Problems that Are Not NP-Complete by Interpretations

3.1 3-SAT as a Class of Finite Models

One can model 3-SAT in different ways. One possibility is to consider those instances (M, V, C, P, N), such that for each clause c there are at most three variables v with $(c, v) \in P \cup N$. This version of 3-SAT is also called SAT(3). The other possibility is to model the clauses as triples of variables. More precisely, we introduce a relation symbol $P_0(x, y, z)$ meaning that all variables x, y, z appear non-negated, a relation symbol $P_1(x, y, z)$ meaning that x appears as a negated and y, z appear non-negated, a relation symbol $P_2(x, y, z)$ meaning that x, y appear negated and z appears non-negated, and a relation symbol $P_3(x, y, z)$ meaning that all variables appear negated. An existential second order formulation of 3-SAT then reads

$$\begin{aligned}\exists V' \forall x, y, z : \{P_0(x, y, z) &\rightarrow (V'(x) \vee V'(y) \vee V'(z))\} \wedge \\ \{P_1(x, y, z) &\rightarrow (\neg V'(x) \vee V'(y) \vee V'(z))\} \wedge \\ \{P_2(x, y, z) &\rightarrow (\neg V'(x) \vee \neg V'(y) \vee V'(z))\} \wedge \\ \{P_3(x, y, z) &\rightarrow (\neg V'(x) \vee \neg V'(y) \vee \neg V'(z))\}\end{aligned}$$

This version of 3-SAT is called 3-SAT-triples.

Theorem 4 ([7, 8]) *3-SAT-triples is not complete in NP by (first-order) interpretations.*

The main proof argument is that 3-SAT-triples has an existential second order formulation that has a purely universally quantified first-order part in prenex normal form, i.e., can be expressed by a $\Sigma_1^1\Pi_1^0$-formula. The class EVEN of finite models with a domain of even size cannot be reduced to 3-SAT-triples, because any interpretation from EVEN to 3-SAT-triples can be transformed to a quantifier-free interpretation that coincides with the given first-order interpretation up to finitely many domain cardinalities. This is true because every first-order formula on a similarity type that contains only unary relation symbols can be transformed to a quantifier-free formula by a quantifier elimination argument, see for example [23]. Therefore, EVEN coincides with the class of finite domains satisfying a certain $\Sigma_1^1\Pi_1^0$-formula up to finitely many cardinalities. Consequently, EVEN is closed by sub-objects up to finitely many exceptions, yielding a contradiction.

Theorem 5 *There is a (first-order) interpretation that reduces SAT(3) to 3-SAT-triples.*

Proof The reduction can be done as follows. Consider any clause c. If c has more than three literals then c is ignored. If c has one literal that is a non-negated variable x then we put (x, x, x) into P_0. If c contains one literal and that literal is the negation of variable x then (x, x, x) is put into P_3. If c contains two literals and they are both non-negated variables x and y, or both negation of these variables, we put (x, x, y) and (y, y, x) into P_0 and P_3, respectively. If c contains the non-negated variable x and the negated variable y, then (y, x, x) is put into P_1. If c contains three non-negated variables as literals, or three negated variables as literals, say x, y, z, then any permutation of (x, y, z) is put into P_0 and P_3, respectively. If c contains one negated variable x and two non negated variables y and z then (x, y, z) and (x, z, y) are put into P_1. If c contains two negated variables x and y and one non-negated variable z then (x, y, z) and (y, x, z) are put into P_2. It is easily seen that for instances of SAT with at most 3 literals per clause the reduction maps to 3-SAT-triples instances with the same meaning. It is also easily seen that the reduction is a first-order interpretation. Note, however, that this is not a quantifier-free interpretation. □

An immediate consequence is

Corollary 1 *SAT(3) is not NP-complete by (first-order) interpretations.*

Otherwise, 3-SAT-triples would be NP-complete by first-order interpretations.

We call a class of finite models expressible by a $\Sigma_1^1\Pi_1^0$-formula to be $\Sigma_1^1\Pi_1^0$-*complete* if every class of finite models expressible by a $\Sigma_1^1\Pi_1^0$-formula can be reduced to it by a quantifier-free interpretation.

Theorem 6 *Each class of finite structures expressible by a $\Sigma_1^1\Pi_1^0$-formula can be reduced to 3-SAT-triples by a quantifier-free interpretation.*

Proof Consider a $\Sigma_1^1\Pi_1^0$-formula $\exists X_1 \ldots, X_k \forall y_1 \ldots, y_l \phi$. We may assume that ϕ is in conjunctive normal form, say $\phi = \phi_1 \wedge \ldots \wedge \phi_m$ and each $\phi_i = \bigvee\{L_i{}^j, j = 1, \ldots, p_i\}$ is a disjunction of elementary formulas and negated elementary formulas.

Each ϕ_i can be transformed into a formula $\psi_i \rightarrow \psi_i'$, where ψ_i is a conjunction of elementary formulas and negated elementary formulas containing only fixed relation symbols, and ψ_i' is a disjunction of elementary formulas and negated elementary formulas containing only existentially quantified relation symbols. We may assume that $\psi_i' = \bigvee\{{L_i}^j, j = 1, \ldots, q_i\}$. As in the reduction of SAT to 3-SAT [19], we may replace ψ_i' by a conjunction of clauses with at most three elementary or negated elementary formulas. We add, for each ψ_i', $q_i - 1$ additional existentially quantified relation symbols $R_i^j(y_1, \ldots, y_l)$ and replace ψ_i' with $q_i > 3$ by the conjunction of clauses C_i^j with $C_i^1 = L_i^1 \vee R_i^1(y_1, \ldots, y_l)$, $C_i^{q_i} = \neg R_i^{q_i-1}(y_1, \ldots, y_l) \vee L_i^{q_i}$, and $C_i^j = \neg R_i^{j-1}(y_1, \ldots, y_l) \vee L_l^j \vee R_i^j(y_1, \ldots, y_l)$.

As in the proof of the NP-completeness of SAT by interpretations [7, 8], we get a quantifier-free interpretation that reduces to 3-SAT-triples. If $V_1, \ldots, V_m$ are the existentially quantified relation symbols then the domain of 3-SAT-triples is the disjoint union $\bigcup_i M^{n_i}$. M is the domain of the source decision problem, and n_i is the arity of V_i. For each $y_1, \ldots, y_l$ and each clause C_i^j, one gets a triple of individuals in $P_0, \ldots, P_3$, if the precondition ψ_i is satisfied. □

The 3-Colouring problem can be expressed by a $\Sigma_1^1\Pi_1^0$-formula and consequently is not NP-complete by interpretations. The same holds for the problem 1-in-3-SAT: Given a set X of variables and a set C of triples of elements of X, is there an assignment of the elements of X by true and false such that for each $c = (x, y, z) \in C$, exactly one of the variables x,y,z is assigned to be true?

Both 1-in-3-SAT and 3-SAT-triples can be seen in a more general context (see[24], and also [1] for further considerations). Let S be a finite set of (logical) relations on $\{0, 1\}$ of any arity. An S-clause is an $R(x_1, \ldots, x_k)$, where k is the arity of the relation R that belongs to S and $x_1, \ldots, x_k$ are variables. An S-formula is a conjunction of S-clauses, and SAT(S) consists of all S-formulas that are satisfiable.

Theorem 7 *3-Colouring is $\Sigma_1^1\Pi_1^0$-complete.*

Proof Consider the reduction from 3-SAT to 3-Colouring in [14]. We design a reduction from 3-SAT-triples to 3-Colouring and show that this reduction is a quantifier-free interpretation. Let (V, P_0, P_1, P_2P_3) be an instance of 3-SAT-triples. Then we can construct a graph that is 3-colourable if and only if the instance of 3-SAT-triples is satisfiable:

1. Provide three vertices T, F, B that are pairwise joined by an edge. T stands for true, F stands for false, and B stands for base.
2. For each variable v, we provide v and its negation $\neg v$ as vertices. Both are joined by an edge to B, and v and $\neg v$ are joined by an edge. In a 3-Colouring, v and $\neg v$ can only have the colours of T or of F.
3. For each clause six vertices a, b, c, d, e, f are introduced, where a, b, c and d, e, f each are triangles of pairwise adjacent vertices. Furthermore, a and the first literal (variable or negated variable) of the clause are adjacent, b and the second literal of the clause are adjacent, e and the third literal of the clause are adjacent, c and d are adjacent, and f is adjacent to F and B. In a 3-Colouring,

if the first and the second literal have the colour of F then one of a or b has the colour of T and the other has the colour of B, c has the colour of F. If one of the first two literals has the colour of T then it is possible to colour c with the colour of T. By the same argument, f can get the colour of T if and only if the third literal or c has the colour of T. This is equivalent to the statement that one of the literals has the colour of T.

Now, the vertex set V_{col} is defined as

$$\{T, F, B\} \cup V \cup \{\neg v, v \in V\} \cup$$
$$\{a, b, c, d, e, f\} \times (\{(0, x, y, z), (0, x, y, z); P_0(x, y, z)\} \cup$$
$$\{(1, x, y, z) : P_1(x, y, z)\} \cup \{(2, x, y, z) : P_2(x, y, z)\} \cup \{(3, x, y, z) : P_3(x, y, z)\}.$$

The edge set E_{col} is defined as

$$\{TF, TB, FB\} \cup \{v\neg v, v \in V\} \cup \{Bv : v \in V\} \cup \{B\neg v, v \in V\} \cup$$
$$\{(a, i, x, y, z)(b, i, x, y, z), (a, i, x, y, z)(c, i, x, y, z), (b, i, x, y, z)(c, i, x, y, z),$$
$$(c, i, x, y, z)(d, i, x, y, z), (d, i, x, y, z)(e, i, x, y, z), (d, i, x, y, z)(f, i, x, y, z) :$$
$$\text{where } P_i(x, y, z), i = 0, \ldots 3\} \cup$$
$$\{(a, 0, x, y, z)x : P_0(x, y, z)\} \cup \{(a, i, x, y, z)\neg x : P_i(x, y, z), i \neq 0\} \cup$$
$$\{(b, i, x, y, z)y : P_i(x, y, z), i \leq 1\} \cup \{(b, i, x, y, z)\neg y : P_i(x, y, z), i > 1\} \cup$$
$$\{(e, i, x, y, z)z : P_i(x, y, z), i \leq 2\} \cup \{(e, i, x, y, z)\neg y : P_i(x, y, z), i > 2\} \cup$$
$$\{F(f, i, x, y, z), B(f, i, x, y, z) : P_i(x, y, z)\}$$

The reader may verify that this graph is exactly the graph that is constructed in [14] to reduce 3-SAT to 3-Colouring. It is also easily verified that the vertex set and the edge set can be defined by quantifier-free formulas of the language of 3-SAT-triples. □

If $F(x_1, \ldots, x_k, y_1, \ldots, y_l)$ is an S-formula with the Boolean variables $x_1, \ldots, x_k, y_1, \ldots, y_l$, then $\exists y_1, \ldots, y_l : F(x_1, \ldots, x_k, y_1, \ldots, y_l)$ is called an *existential S-formula*. Schaefer [24] proved the following cases of SAT(S) to be polynomial time solvable:

1. Every relation in S is valid if all entries are 0.
2. Every relation in S is valid if all entries are 1.
3. Every relation in S is weakly positive, i.e., is equivalent to a conjunction of clauses with at most one negated variable.
4. Every relation in S is weakly negative, i.e., is equivalent to a conjunction of clauses with at most one non-negated variable.
5. Every relation in S is bijunctive, i.e., equivalent to a conjunction of clauses with at most two literals per clause.
6. Every relation in S is affine, i.e., equivalent to a conjunction of linear equations over the finite field with two elements.

The so-called Dichotomy Theorem now states

Theorem 8 ([24]) *If none of the conditions 1–6 is satisfied for S then each logical relation can be expressed by an existential S-formula.*

Corollary 2 *If none of the conditions 1–6 is satisfied for S then SAT(S) is* $\Sigma_1^1\Pi_1^0$*-complete.*

Proof The NP-completeness proof in [24] presents a reduction that can be transformed into a quantifier-free interpretation that reduces 3-SAT-triples to SAT(S). Consider an instance (V, P_0, P_1, P_2, P_3) of 3-SAT-triples. Each $P_i(x_1, x_2, x_3)$ as Boolean function is equivalent to an S-formula $\exists y_1^i, \ldots, y_k^i F_i(x_1, x_2, x_3, y_1^i, \ldots, y_{k_i}^i)$, where $F_i(x_1, x_2, x_3, y_1^i, \ldots, y_{k_i}^i)$ is a conjunction of S-clauses in variables $x_1, x_2, x_3, y_1^i, \ldots, y_{k_i}^i$. Each such S-clause is an $s(z_1, \ldots, z_l)$ and is represented by a relation $R_s(z_1, \ldots, z_l)$ in the signature of SAT(S). We now can construct an interpretation reducing 3-SAT-triples to SAT(S):

1. For all three variables v_1, v_2, v_3 in V with $P_i(v_1, v_2, v_3)$, we introduce additional variables $y_1^{i,(v_1,v_2,v_3)}, \ldots, y_{k_i}^{i,(v_1,v_2,v_3)}$. These are $k_1 + k_2 + k_3$ copies of V^3 and they are definable quantifier-freely.
2. For each $s(z_1, \ldots, z_l)$ present in the definition of P_i, $R_s(z_1^{(v_1,v_2,v_3)}, \ldots, z_1^{(v_1,v_2,v_3)})$ is satisfied if $P_i(v_1, v_2, v_3)$ is satisfied, $z_q^{(v_1,v_2,v_3)}$ is v_j if $z_i = x_j$ and is $y_j^{i,(v_1,v_2,v_3)}$ if $z_q = y_j^i$.

It is easily checked that this defines a quantifier-free interpretation. □

Corollary 3 *1-in-3-SAT is* $\Sigma_1^1\Pi_1^0$*-complete.*

3.2 What Is the Status of 3-Dimensional Matching?

The 3-Dimensional Matching 3DM problem [19] asks for three given sets A, B, C of the same size and a ternary relation $R \subseteq A \times B \times C$, whether there exists a subrelation $R' \subset R$, such that each $x \in A \cup B \cup C$ appears in exactly one tuple in R'. The reduction to 3DM in [19] showing its NP-completeness is not an interpretation, because it uses more than only the relations of the source decision problem. On the other hand, 3DM is not in $\Sigma_1^1\Pi_1^0$. This can easily verified by the fact that 3DM is not closed by substructures. Any known reduction from SAT or Exact Cover to it can not be transformed into a first-order interpretation. But it remains an open research problem to prove (or disprove) whether 3DM is NP-complete with respect to first-order interpretations.

A decision problem that is related to 3DM is Exact Cover by 3-Sets, which is the restriction of Exact Cover to sets of size 3. As with 3-SAT, it can be modelled in two ways as a class of finite models. One such way is to consider a structure consisting of a ground set together with a ternary relation. The other possibility is to restrict

the model for Exact Cover to sets of size 3. 3DM can be seen as a special case of Exact Cover modelled as a ground set with a ternary relation. It can also be reduced to Exact cover by an quantifier-free interpretation. Here, the domain of the Exact Cover instance is $A \cup B \cup C$ and each triple in R is made a set of size 3.

Exact Cover can be seen as a special kind of a satisfiability problem called *1-in-SAT*: Given a set of variables and a set of clauses, find an assignment of the variables such that each clause contains exactly one true variable. A transformation to Exact Cover can be designed via

1. making the clauses corresponding to the domain of the Exact Cover instance and
2. letting variables correspond to the sets of the Exact Cover instance. A clause is considered to be in the set corresponding to a variable if the variable is in the clause.

Exact Cover by 3-sets can be considered as the dual decision problem of 1-in-3-SAT: While for 1-in-3-SAT the number of variables per clauses is restricted, in Exact Cover by 3-Sets the number of clauses per variables is restricted.

4 Conclusions

Interpretations can be used to prove that certain decision problems (classes of finite models) cannot be expressed in a certain logic, i.e., using a particular type of logic formulas. In [3] it has been shown that there is no term in Least Fixed-Point with Counting that $7/8 + \epsilon$-approximates Max-3-SAT.

Let $L^{\omega}_{\infty,\omega}$ be the closure of first-order logic by possibly infinite conjunction and possibly infinite disjunction restricted to formulas that use only a finite number of variables. Note that $L^{\omega}_{\infty,\omega}$ includes Least-Fixed-Point. Dawar mentioned in [11] that 3-colouring is not only not $L^{\omega}_{\infty,\omega}$-expressible [4] but neither expressible in $L^{\omega}_{\infty,\omega}$ with counting. For more details see [4, 17]. We can state the following result.

Theorem 9 *No decision problem being either $\Sigma^1_1\Pi^0_1$-complete or complete in NP by interpretations can be expressed in $L^{\omega}_{\infty,\omega}$ with counting, and therefore neither can be expressed in Least-Fixed-Point with counting or in Least-Fixed-Point logic.*

Problems covered by the previous theorem include 3-Colouring, 3-SAT, SAT, CLIQUE, Hamilton Cycle, Graph Colouring, and Exact Cover. For the decision problems that are complete in NP by quantifier-free interpretations, an easy proof of not being expressible in Least-Fixed-Point might have been reachable already in 1983 by finding one decision problem being in NP that is not Least-Fixed-Point expressible, combining the results in [5, 6] with the results of [7, 8]. Nevertheless, a proof that Hamiltonian Cycle is not Least-Fixed-Point expressible based on Ehrenfeucht games was published in 1987 [12]. For decision problems that are $\Sigma^1_1\Pi^0_1$ -complete, a $\Sigma^1_1\Pi^0_1$-expressible decision problem had to be found, where it was possible to show that it is not expressible in $L^{\omega}_{\infty,\omega}$ with counting [4, 7, 17].

Note that $L^{\omega}_{\infty,\omega}$ includes all decision problems expressible by a symmetric linear program. An immediate consequence is that no $\Sigma^1_1\Pi^0_1$-complete decision problem and no decision problem that is NP.complete by interpretations can be solved by a symmetric linear program.

Interpretations for a long time played only a minor role in finite model theory. Interpretations as reductions were known since 1977 [21]. But the pioneering paper by Lovász and Gács did not attract that interest as it should have. I would like to mention that I was not aware of the paper of Lovász and Gács [21] when I submitted my doctoral Dissertation [7], although I asked several experts who were familiar with the topic of finite model theory. The paper of Immerman [18] was one of the first papers after the one by Lovász and Gács [21] that deals with interpretations. In the later 1990'ies, Dawar published two significant papers showing that interpretations are quite a useful tool. In [11], a short proof is presented that Hamiltonian Cycle is not in $L^{\omega}_{\infty,\omega}$. The previous theorem was not formulated explicitly. But I could imagine that experts in finite model theory are aware of such result. For further readings on interpretations and generalized quantifiers, we refer also to [22] and [10]. The major result at the end are necessary and sufficient conditions that a complexity class closed under interpretations contains decision problems that are complete in that complexity class by interpretations.

Acknowledgments I am grateful to Ulrike Mosel and Julian Harbarth for helping me to improve my English and to discover some typos. I am grateful to J.A. Makowsky for useful hints concerning the literature. I am grateful to Klaus Meer helping me to improve the quality of the paper significantly.

References

1. Allender, E., Bauland, M., Immerman, N., Schnoor, H., Vollmer, H.: The Complexity of Satisfiability Problems: Refining Schaefer's Theorem. J. Comput. Syst. Sci. **75**, 243–254 (2009)
2. Anderson, M., Dawar, A., Holm, B.: Maximum matching and linear programming in fixed-point logic with counting. 28th Annual ACM-ÌEEE Symposium on Logic in Computer Science, pp.173–182 (2013)
3. Atserias, A., Dawar, A.: Definable inapproximability: new challenges for duplicator. J. Logic Comput. **29**(8), 1185–1210 (2019)
4. Cai, J.Y., Fürer, M., Immerman, N.: An optimal lower bound on the number of variables for graph identification. Combinatorica **12**(4), 389–410 (1992)
5. Chandra, A.K., Harel, D.: Computational queries for relational data bases. J. Comput. Syst. Sci. **21**, 156–178 (1980)
6. Chandra, A.K., Harel, D.: Structure and complexity of relational queries. J. Comput. Syst. Sci. **25**(1), 99–128 (1982)
7. Dahlhaus, E.: Kombinatorische und logische Eigenschaften von Reduktionen auf einige vollständige Probleme in NP und NL. Doctoral Thesis, TU Berlin, 1982
8. Dahlhaus, E.: Reductions to NP-complete Problems by Interpretations, Logic and Machines: Decision Problems and Complexity. LNCS, vol. 171, pp. 357–365. Springer, Berlin (1983)

9. Dahlhaus, E.: Skolem Normal Forms Concerning the Least Fixpoint, Computation Theory and Logic. LNCS, vol. 270, pp. 101–106. Springer, Berlin (1987)
10. Dawar, A.: Generalized quantifiers and logical reductions. J. Logic Comput. **5**, 213–226 (1995)
11. Dawar, A.: A restricted second-order logic for finite structures. Inf. Comput. **143**(2), 154–174 (1998)
12. De Rougemont, M.: Second order and inductive definability on finite structures. Zeitschrift für Mathematische Logik und Grundl. der Mathematik **33**(8), 47–63 (1987)
13. Fagin, R.: Generalized first-order spectra and polynomial-time recognizable sets. In: Karp, R.M. (ed.) Complexity of Computation. SIAM-AMS Proceedings, vol. 7, pp. 43–73 (1974)
14. Garey, M., Johnson, D.S., Stockmeyer, L.: Some simplified NP-complete graph problems. Theor. Comput. Sci. **2**, 237–267 (1976)
15. Garey, M., Johnson, D.S.: Computers and Intractability. W.H. Freeman and Company, New York (1979)
16. Gurevich, Y., Shelah, S.: Fixed point extensions of first-order logic. Ann. Pure Appl. Logic **32**, 265–280 (1986)
17. Hella, L.: Logical hierarchies in P-Time. Inf. Comput. **129**, 1–19 (1996)
18. Immerman, N.: Languages which capture complexity classes. SIAM J. Comput. **16**, 760–778 (1987)
19. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.F., Thatcher, J.W. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum, New York (1972)
20. Kolaitis, P.: The expressive power of stratified logic programs. Inf. Comput. **90**(3), 50–66 (1991)
21. Lovász, L., Gács, P.: Some Remarks on generalized spectra. Zeitschrift für Mathematische Logik und Grundl. der Mathematik **23**, 547–554 (1977)
22. Makowsky, J.A., Pnueli, Y.B.: Oracles and quantifiers. In: Börger, E., Gurevich, Y., Meinke, K. (eds.) Proceedings of the Computer Science Logic: 7th Workshop, CSL'93 Swansea. Springer Lecture Notes in Computer Science, vol. 832, pp. 189–222 (1994)
23. Monk, J.D.: Mathematical Logic. Graduate Texts in Mathematics. Springer, New York (1976)
24. Schaefer, T.J.: The complexity of satisfiability problems. STOC '78: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing, pp. 216–226 (1978)
25. Stewart, I.A.: Methods for proving completeness via logical reductions. Theor. Comput. Sci. **118**, 193–229 (1993)
26. Stewart, I.A.: On completeness for NP via projection translations. Math. Syst. Theory **27**, 125–157 (1994)

Bounded Languages Over Infinite Alphabets

Yoav Danieli and Michael Kaminski

Dedicated to Johann Makowsky
on the occasion of his 75^{th} *birthday.*

Abstract It is shown that the problem whether there is a bound on the lengths of the words in the language accepted by an alternating one-window finite-memory automaton is decidable. The decision procedure relies on a kind of the pumping lemma for languages over infinite alphabets.

1 Introduction

Finite-memory automata [5, 6] are a generalization of the classical Rabin-Scott finite-state automata [11] to *infinite* alphabets. In addition to a finite set of "proper" states, finite-memory automata are equipped with a finite set of registers, called in [5, 6] and in this paper *windows*, which at any stage of a computation (automaton's run) contain a symbol from the infinite alphabet. By restricting the power of the automaton to comparing the input symbol with the contents of the registers and copying the input symbol to a register, without the ability to perform *any* operations, the automaton is only able to "remember" a finite set of input symbols. Thus, the languages accepted by finite-memory automata possess many of the desirable properties of regular languages and therefore, are referred to as *quasi-regular* languages.

An important facet of finite-memory automata is a certain *indistinguishability* view of the infinite alphabet embedded in the modus operandi of the automaton. The language accepted by an automaton is invariant under automorphisms of the infinite alphabet. Thus, the actual symbols occurring in the input are of no real significance.

Y. Danieli (✉) · M. Kaminski
The Henry and Marilyn Taub Faculty of Computer Science, Technion – Israel Institute of Technology, Haifa, Israel
e-mail: yoavd@campus.technion.ac.il; kaminski@cs.technion.ac.il

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_15

Only the initial and repetition patterns matter. This follows from the nature of the restriction to copying and comparison (reminiscent of *term-unification*). If a new symbol (i.e., one not in any window) is copied and later successfully compared, any other new symbol, appearing in the same position, would cause the *same* transitions.

This paper deals with the *alternating* versions of finite-memory automata [9, 10]. These automata are tightly related to the linear and branching temporal logics with the freeze quantifier, see [1, 2]. Whereas the emptiness problem for alternating two-window finite-memory automata is undecidable, see [10, Theorem 5.1], it was shown in [1, 2] that the emptiness problem for alternating one-window finite-memory automata is decidable.[1] The proofs in [1, 2] are based on reductions to other decision problems and are very involved. An alternative, relatively simple, and self-contained proof was presented in [3].

In this paper, we modify the decision procedure from [3] to decide whether the language accepted by a one-window alternating finite-memory automaton is bounded, i.e., whether there is a bound on the lengths of the words in that language.

Theorem 1 *The boundness problem for alternating one-window finite-memory automata is decidable.*

Whereas, over finite alphabets, a language is bounded if and only if it is finite, over infinite alphabets, bounded languages are not necessarily finite, e.g., the language consisting of all words of length one. Thus, boundness can be thought of as an infinite alphabet counterpart of the finiteness of ordinary regular languages.

Before addressing the boundness problem, we briefly recall the decision procedure of emptiness from [3]. For an alternating one-window finite-memory automaton $\boldsymbol{A}$, this procedure computes a positive integer N such that for each word $\boldsymbol{\sigma} \in L(\boldsymbol{A})$ longer than N, $L(\boldsymbol{A})$ contains a word shorter than $\boldsymbol{\sigma}$, see Lemma 1 in the next section.

Thus, $L(\boldsymbol{A}) \neq \emptyset$ if and only if $L(\boldsymbol{A})$ contains a word shorter than N. Then the emptiness of $L(\boldsymbol{A})$ is reduced to the emptiness of the restriction of $L(\boldsymbol{A})$ to an N-symbol alphabet that is regular.

In some sense, our approach for deciding the boundness problem is "reversal" to that of [3]. Namely, for an alternating one-window finite-memory automaton $\boldsymbol{A}$, our procedure computes a positive integer $N_{\boldsymbol{A}}$ such that for each word $\boldsymbol{\sigma} \in L(\boldsymbol{A})$ longer than $N_{\boldsymbol{A}}$, $L(\boldsymbol{A})$ contains a word *longer* than $\boldsymbol{\sigma}$. Thus, $L(\boldsymbol{A})$ is unbounded if and only if contains a word longer than $N_{\boldsymbol{A}}$, which, as shown in the sequel, is decidable. We elaborate what we mean by shortening and lengthening in Sect. 3, after giving necessary definitions in Sect. 2.

Namely, this paper is organized as follows. In the next section we recall the definition of alternating finite-memory automata and prove some of its basic properties needed for our decision procedure. The decision procedure itself is presented at the end of Sect. 3, whereas in the beginning of this section we compare

[1] Decidability of this problem implies decidability of the emptiness problem for *top-view* k-pebble automata, see [12, 13].

our approach to that of [3]. Section 4 contains the proof of the "lengthening" lemma (Lemma 2) stated in Sect. 3. Finally, in Sect. 5, we show that the boundness problem for alternating finite-memory automata with two windows is undecidable.

2 Alternating Finite-Memory Automata

In this section, we recall the definition of alternating finite-memory automata and prove its basic properties needed for our decision procedure.

We restrict ourselves to one-register automata with which we deal in this paper. Like in the classical finite alphabet case, these automata have a finite set of proper states in which the "real computation" is done. In addition, the automaton is equipped with a register storing a symbol from the infinite alphabet. We refer to the register as the "*window*," see [5, 6].

Throughout this paper we use the following conventions.

- Σ is a fixed infinite alphabet.
- Symbols in Σ are denoted by σ, θ, or δ (sometimes indexed or primed).
- Bold lower-case Greek letters $\boldsymbol{\sigma}$, $\boldsymbol{\tau}$, and $\boldsymbol{\upsilon}$ denote words over Σ.
- Symbols which occur in a word denoted by a boldface letter are always denoted by the same *non-boldface* letter with an appropriate subscript. For example, symbols which occur in $\boldsymbol{\sigma}'$ are denoted by σ'_i.

Definition 1 (Cf. [6, Definition 1]) An *alternating one-window finite-memory automaton* (over Σ) is a system $\boldsymbol{A} = \langle S, s_0, F, \Delta, \delta, \mu_\Delta, \mu_=, \mu_{\neq}\rangle$ whose components are as follows.

- S is a finite set of *states*.
- $s_0 \in S$ is the *initial state*.
- $F \subseteq S$ is the set of *accepting* states.
- $\Delta \subset \Sigma$ is a finite set of *distinguished* symbols.
- $\delta \in \Sigma \setminus \Delta$ is the *initial window assignment*.
- $\mu_\Delta : S \times \Delta \to 2^{2^S}$, $\mu_= : S \to 2^{2^S}$, and $\mu_{\neq} : S \to 2^{(2^S)^2}$ are the *transition* functions.

  The intuitive meaning of these functions is as follows. Let $\boldsymbol{A}$ read a symbol σ being in state s with a symbol θ stored in the window.

  - If $\sigma \in \Delta$, then, for some $Q \in \mu_\Delta(s, \sigma)$, the computation splits to the computations from all states in Q with θ in the window.
  - If $\sigma = \theta$, then, for some $Q \in \mu_=(s)$, the computation splits to the computations from all states in Q with the same θ in the window.
  - If $\sigma \notin \Delta \cup \{\theta\}$, then, for some $(Q', Q'') \in \mu_{\neq}(s)$, the computation splits to the computations from all states in Q' with θ in the window and the computations from all states in Q'' with the current input symbol σ in the window.

Remark 1 Note that symbols from Δ are recognized by μ_Δ, and, therefore, never appear in the window.

In accordance with the above intuitive meaning of the transition functions, an actual state of $\boldsymbol{A}$ is an element of $S\times(\Sigma\setminus\Delta)$, where the state component of the pair is the current state and the symbol component is the symbol stored in the window. Thus $\boldsymbol{A}$ has infinitely many states, which are pairs (s,σ), where $s\in S$ and $\sigma\in(\Sigma\setminus\Delta)$. These are called *configurations* of $\boldsymbol{A}$.

The transition functions μ_Δ, $\mu_=$, and $\mu_\neq$ induce the following transition function $\mu^{\boldsymbol{c}}:(S\times(\Sigma\setminus\Delta))\times\Sigma\to 2^{2^{(S\times(\Sigma\setminus\Delta))}}$.

- If $\sigma\in\Delta$, then

$$\mu^{\boldsymbol{c}}((s,\theta),\sigma)=\{Q\times\{\theta\}:Q\in\mu_\Delta(s,\sigma)\}.$$

- If $\sigma=\theta$, then

$$\mu^{\boldsymbol{c}}((s,\theta),\sigma)=\{Q\times\{\theta\}:Q\in\mu_=(s)\}\,..$$

- If $\sigma\notin\Delta\cup\{\theta\}$, then

$$\mu^{\boldsymbol{c}}((s,\theta),\sigma)=\{Q'\times\{\theta\}\cup Q''\times\{\sigma\}:(Q',Q'')\in\mu_\neq(s)\}.$$

Remark 2 Note that $\mu^{\boldsymbol{c}}((s,\theta),\sigma)$ is a finite collection of finite sets of configurations.

Next, we extend $\mu^{\boldsymbol{c}}$ to finite sets of configurations in the standard "alternating" manner. Let $C=\{c_1,c_2,\ldots,c_k\}$ be a set of configurations, $\sigma\in\Sigma$, and let $\mu^{\boldsymbol{c}}(c_i,\sigma)=\{C_{i,1},C_{i,2},\ldots,C_{i,m_i}\}$, $i=1,2,\ldots,k$. Then $\mu^{\boldsymbol{c}}(C,\sigma)$ consists of all finite sets of configurations of the form $\bigcup_{i=1}^{k}C_{i,j_i}$, $j_i=1,2,\ldots,m_i$. That is, $C'\in\mu^{\boldsymbol{c}}(C,\sigma)$, if there is a collection of finite sets of configurations $\{C_{1,j_1},C_{2,j_2},\ldots,C_{k,j_k}\}$, $C_{i,j_i}\in\mu^{\boldsymbol{c}}(c_i,\sigma)$, $i=1,2,\ldots,k$, such that $C'=\bigcup_{i=1}^{k}C_{i,j_i}$.

Now, we can define the notion of a *run* of $\boldsymbol{A}$. Let C be a finite set of configurations and let $\boldsymbol{\sigma}=\sigma_1\sigma_2\cdots\sigma_n\in\Sigma^*$. A *C-run* (or just a run, if C is clear from the context) of $\boldsymbol{A}$ on $\boldsymbol{\sigma}$ is a sequence of finite sets of configurations $C_0,C_1,\ldots,C_n$, where

- $C_0=C$ and
- $C_{i+1}\in\mu^{\boldsymbol{c}}(C_i,\sigma_{i+1})$, $i=0,1,\ldots,n-1$.

This run is *accepting*, if $C_n\subseteq F\times\Sigma$, i.e., C_n is a set of *accepting* configurations, and the automaton *C-accepts* a word $\boldsymbol{\sigma}\in\Sigma^*$, if there is an accepting C-run of $\boldsymbol{A}$ on $\boldsymbol{\sigma}$.

The set of all words C-accepted by $\boldsymbol{A}$ is denoted by $L(\boldsymbol{A}_C)$.

For a one-element set of configurations $\{c\}$, we write just $\boldsymbol{A}_c$ for $\boldsymbol{A}_{\{c\}}$. It immediately follows form the definition that

$$L(\boldsymbol{A}_C) = \bigcap_{c \in C} L(\boldsymbol{A}_c) \,. \tag{1}$$

Finally, we define the language $L(\boldsymbol{A})$ of $\boldsymbol{A}$ as the language $L\left(\boldsymbol{A}_{(s_0,\delta)}\right)$.
Proposition 1 below immediately follows from the definition of acceptance.

Proposition 1 (Cf. [6, Lemma A.2]) *Let C and C' be finite sets of configurations such that $C \subseteq C'$. Then $L(\boldsymbol{A}_{C'}) \subseteq L(\boldsymbol{A}_C)$.*

The proof of Theorem 1 is based on the following closure property of accepted languages. To state it we need Definitions 2 and 3 below.

Definition 2 Let α be an automorphism of Σ. We extend α to configurations by $\alpha(s, \sigma) = (s, \alpha(\sigma))$ and then to finite sets of configurations by

$$\alpha(C) = \{\alpha(c) : c \in C\} \,.$$

Definition 3 Automorphisms of Σ which are invariant on Δ are called Δ-*automorphisms*.

Proposition 2 (Cf. [6, Lemma A.4]) *Let $\boldsymbol{A}$ be an alternating one-window finite-memory automaton, $\boldsymbol{\sigma} = \sigma_1\sigma_2 \cdots \sigma_n \in \Sigma^*$, C be a finite set of configurations of $\boldsymbol{A}$, $C_0, C_1, \ldots, C_n$ be a C-run of $\boldsymbol{A}$ on $\boldsymbol{\sigma}$, and let α be a Δ-automorphism of Σ. Then $\alpha(C_0), \alpha(C_1), \ldots, \alpha(C_n)$ is an $\alpha(C)$-run of $\boldsymbol{A}$ on $\alpha(\boldsymbol{\sigma})$.*

The proof of the proposition is similar to that of [6, Lemma 1] and is omitted.

Corollary 1 (Cf. [6, Lemma A.4]) *Let C be a finite set of configurations of $\boldsymbol{A}$ and let α be a Δ-automorphism of Σ. Then $\alpha(L(\boldsymbol{A}_C)) = L(\boldsymbol{A}_{\alpha(C)})$.*[2]

Remark 3 Proposition 2 and Corollary 1 show that emptiness is invariant under Δ-automorphisms. For this reason, in [1, 2] symbols of the infinite alphabet Σ are equivalently replaced with the sets of positions which they occupy in the input word.

Finally, the paragraph following Theorem 1 can be elaborated as follows. Consider a relation $\sim$ on Σ^* defined by $\boldsymbol{\sigma} \sim \boldsymbol{\tau}$ if and only if there exists an automorphism α of Σ such that $\boldsymbol{\sigma} = \alpha(\boldsymbol{\tau})$. Obviously, $\sim$ is an equivalence relation and a language L is bounded if and only if $L/\sim$ is finite.

3 Boundness vs. Emptiness and the Decision Procedure

We precede the decision procedure with a comparison of our approach to that in [3]. The latter is summarized in the following "shortening" lemma.

[2] In particular, if $C = \{(s_0, \delta)\}$ and $\alpha(\delta) = \delta$, then $\alpha(L(\boldsymbol{A})) = L(\boldsymbol{A})$.

Lemma 1 ([3]) *Let $\boldsymbol{A}$ be an alternating one-window finite-memory automaton. There is a computable constant $N_{\boldsymbol{A}}$ such that for every word $\boldsymbol{\sigma} \in L(\boldsymbol{A})$ that is longer than $N_{\boldsymbol{A}}$, there exist a decomposition $\boldsymbol{\sigma} = \boldsymbol{\tau}\boldsymbol{\upsilon}\boldsymbol{\varphi}$ of $\boldsymbol{\sigma}$ and a Δ-automorphism α of Σ satisfying*

- $|\boldsymbol{\tau}\boldsymbol{\upsilon}| \leq N_{\boldsymbol{A}}$,
- $|\boldsymbol{\upsilon}| > 0$, *and*
- $\boldsymbol{\tau}\alpha(\boldsymbol{\varphi}) \in L(\boldsymbol{A})$.

Roughly speaking, for the proof of Lemma 1, in [3], it has been shown that, for sufficiently long word $\boldsymbol{\sigma} = \sigma_1\sigma_2\cdots\sigma_n$ and a run $C_0, C_1, \ldots, C_n$ of $\boldsymbol{A}$ on $\boldsymbol{\sigma}$, there exist nonnegative integers i and j, $0 \leq i < j \leq n$, and a Δ-automorphism α of Σ such that $\alpha(C_i) \subseteq C_j$. Having such i, j, and α, the argument in [3] is as follows.

If $C_0, C_1, \ldots, C_n$ is an accepting run of $\boldsymbol{A}$ on $\boldsymbol{\sigma}$, then, $\sigma_j, \sigma_{j+1}\cdots\sigma_n \in L(\boldsymbol{A}_{C_j})$, implying, by Corollary 1, $\alpha^{-1}(\sigma_j\sigma_{j+1}\cdots\sigma_n) \in L(\boldsymbol{A}_{\alpha^{-1}(C_j)})$. This, together with $C_i \subseteq \alpha^{-1}(C_j)$, implies, by Proposition 1, $\alpha^{-1}(\sigma_j\sigma_{j+1}\cdots\sigma_n) \in L(\boldsymbol{A}_{C_i})$, that, in turn, implies $\sigma_1\sigma_2\cdots\sigma_i\alpha^{-1}(\sigma_j\sigma_{j+1}\cdots\sigma_n) \in L(\boldsymbol{A})$.

As we have mentioned in the introduction, our approach is a "reversal" of that of [3]. Namely, whereas the decision procedure in [3] is based on iteratively "shortening" sufficiently long words in the language, resulting in a word (if exists) shorter than a computable constant, in this paper we "lengthen" the language words longer than a computable constant. For this, we show that, for sufficiently long word $\boldsymbol{\sigma} = \sigma_1\sigma_2\cdots\sigma_n$ and a run $C_0, C_1, \ldots, C_n$ of $\boldsymbol{A}$ on $\boldsymbol{\sigma}$, there exist nonnegative integers i and j, $0 \leq i < j \leq n$, and a Δ-automorphism α of Σ such that $\alpha(C_j) \subseteq C_i$. Then, by Corollary 1, $\sigma_1\sigma_2\cdots\sigma_{j-1}\alpha^{-1}(\sigma_i\sigma_{i+1}\cdots\sigma_n) \in L(\boldsymbol{A})$.

Note that, whereas the decision procedure in [3] is based on iteratively "shortening" prefixes of sufficiently long words in the language, resulting in a word (if exists) shorter than a computable constant, in contrast, we "lengthen" suffices of the language words longer than a computable constant. The reason for passing from prefixes to suffices is illustrated by the example below showing that it is not always possible to extend every segment of a word while staying in the language.

This example involves the language

$$L_{\text{diff}} = \{\sigma_1\sigma_2\cdots\sigma_n \in \Sigma^* : \text{for all } i, j = 1, 2, \ldots, n \text{ such that } i \neq j,\ \sigma_i \neq \sigma_j\}$$

(see [6, Proposition 5]) and the following notation.

For word $\boldsymbol{\sigma} = \sigma_1\sigma_2\cdots\sigma_n \in \Sigma^*$, we denote by $[\boldsymbol{\sigma}]$ the set of symbols occurring in $\boldsymbol{\sigma}$:

$$[\boldsymbol{\sigma}] = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$$

and call this set the *contents* of $\boldsymbol{\sigma}$.

Example 1 Consider the language:

$$L_{\subseteq} = \{\boldsymbol{\sigma}\#\boldsymbol{\tau} : \boldsymbol{\sigma}, \boldsymbol{\tau} \in L_{\text{diff}} \text{ and } [\boldsymbol{\sigma}] \subseteq [\boldsymbol{\tau}] \subset \Sigma \setminus \{\#\}\}$$

(see [4, Example 3]).

This language is accepted by an alternating one-window finite-memory automaton, by verifying that # appears in the input word exactly once, the prefix before # and the suffix after # do not have more then one appearance of any symbol, and that every symbol appearing before # also appears after #.[3]

Let $\sigma\#\sigma \in L_{\subseteq}$. Then, the prefix σ before # cannot be extended such that the resulting word still remains in the language. This is because the symbols in the extension must differ from the contents of the suffix σ.

In contrast, any extension of the suffix σ with symbols not belonging to its contents, still keeps the resulting word in $L_{\subseteq}$.

A similar argument shows that deleting symbols from the prefix σ leaves the resulting word in the language, but deleting symbols from the suffix σ does not. In fact, symbol deletions in [3] are only in the bounded prefix.

Example 1 (that motivates our "reversal" approach) is generalized in the "lengthening" lemma below that focuses on an appropriate suffix of the word.

Lemma 2 *Let $\boldsymbol{A}$ be an alternating one-window finite-memory automaton. There is a computable constant $N_{\boldsymbol{A}}$ such that for every word $\sigma \in L(\boldsymbol{A})$ that is longer than $N_{\boldsymbol{A}}$, there exist a decomposition $\sigma = \tau\upsilon\varphi$ of σ and a Δ-automorphism α of Σ satisfying*

- $|\upsilon\varphi| \leq N_{\boldsymbol{A}}$,
- $|\upsilon| > 0$, *and*
- $\tau\upsilon\alpha^{-1}(\upsilon\varphi) \in L(\boldsymbol{A})$.

Based on this lemma, whose proof is postponed to the next section, the decision procedure for boundness is as follows.

Proof of Theorem 1 Let $\boldsymbol{A}$ be an alternating one-window finite-memory automaton and let $N_{\boldsymbol{A}}$ be the constant provided by Lemma 2. Then $L(\boldsymbol{A})$ is bounded if and only if

$$L(\boldsymbol{A}) \cap \Sigma^{N_{\boldsymbol{A}}}\Sigma^* = \emptyset.$$

Indeed, if the above intersection is empty, then $L(\boldsymbol{A})$ is bounded (by $N_{\boldsymbol{A}}$). Otherwise, $L(\boldsymbol{A})$ contains a word longer than $N_{\boldsymbol{A}}$, implying, by Lemma 2, that it contains infinitely many words with pairwise distinct lengths. That is, $L(\boldsymbol{A})$ is unbounded.

Obviously, $\Sigma^{N_{\boldsymbol{A}}}\Sigma^*$ is accepted by a one-window finite-memory automaton. Thus, the intersection $L(\boldsymbol{A}) \cap \Sigma^{N_{\boldsymbol{A}}}\Sigma^*$ is accepted by an alternating one-window finite-memory automaton, because languages accepted by an alternating one-window finite-memory automaton are closed under Boolean operations. That is, boundness is reduced to emptiness, that by Genkin et al. [3, Theorem 1] is decidable. □

[3] Note that the restriction of languages L_{diff} and $L_{\subseteq}$ to any finite alphabet is finite. Thus, the boundness problem cannot be trivially reduced to the ordinary finite alphabet counterpart.

4 Proof of Lemma 2

The proof of Lemma 2 consists of four parts. First, we introduce the necessary notation and prove a proposition underlying the motivation of the partial order defined in the second part. Based on this partial order, in the third part, we define the backward automaton's runs and establish their basic properties. The proof of the lemma itself is presented in the last part.

4.1 Notation and a Motivating Result

Let $\boldsymbol{A} = \langle S, s_0, F, \Delta, \delta, \mu_\Delta, \mu_=, \mu_{\neq} \rangle$ be an alternating one-window finite-memory automaton. Following [3], we introduce the notation below.

For a finite set C of configurations of $\boldsymbol{A}$, we denote by $S(C)$ and $\Sigma(C)$ the following set of states and the subset of $\Sigma \setminus \Delta$, respectively.

$$S(C) = \{s \in S : \text{ for some } \sigma \in \Sigma, \ (s, \sigma) \in C\} \tag{2}$$

and

$$\Sigma(C) = \{\sigma \in \Sigma \setminus \Delta : \text{ for some } s \in S, \ (s, \sigma) \in C\}. \tag{3}$$

That is, $S(C)$ and $\Sigma(C)$ consist of all states and all elements of $\Sigma \setminus \Delta$, respectively, which occur in configurations from C.

It immediately follows from (3) that for any Δ-automorphism α of Σ,

$$\alpha(\Sigma(C)) = \Sigma(\alpha(C)). \tag{4}$$

Consider the relation $\equiv_C$ on $\Sigma(C)$ such that $\sigma \equiv_C \sigma'$ if and only if the following holds.

- For each $s \in S$, $(s, \sigma) \in C$ if and only if $(s, \sigma') \in C$.

Obviously, $\equiv_C$ is an equivalence relation. The equivalence classes of $\equiv_C$ can be described as follows. Let $\sigma \in \Sigma \setminus \Delta$ and let the subset $S^\sigma(C)$ of S be defined by

$$S^\sigma(C) = \{s : (s, \sigma) \in C\}. \tag{5}$$

It follows from (2) that $S(C) = \bigcup_{\sigma \in \Sigma(C)} S^\sigma(C)$.

Then $\sigma \equiv_C \sigma'$ if and only if $S^\sigma(C) = S^{\sigma'}(C)$.

Next, let P be a nonempty subset of S and let the subset $\Sigma^P(C)$ of $\Sigma(C)$ be defined by

$$\Sigma^P(C) = \{\sigma : S^\sigma(C) = P\}. \tag{6}$$

Then, for any Δ-automorphism α of Σ,

$$\alpha(\Sigma^P(C)) = \Sigma^P(\alpha(C)), \tag{7}$$

cf. (4).

Remark 4 Note that, if $C' \subseteq C''$, then, for each be a nonempty subset P of S, $\Sigma^P(C') \subseteq \Sigma^P(C'')$.

It follows from (3) and (6) that

$$\Sigma(C) = \bigcup_{P \subseteq 2^S \setminus \{\emptyset\}} \Sigma^P(C) \tag{8}$$

and the equivalence classes of $\equiv_C$ are in one-to-one correspondence with those subsets P of S for which $\Sigma^P(C)$ is nonempty. Actually, in the sequel we deal with $\|\Sigma^P(C)\|$—the number of elements of $\Sigma^P(C)$, where P is a nonempty subset of S. These non-negative integers are the values of the function $f_C : 2^S \setminus \emptyset \to \mathbb{N}$, [4] defined by

$$f_C(P) = \|\Sigma^P(C)\|.$$

Remark 5 This function can be thought of as a $(2^{\|S\|} - 1)$-dimensional vector whose components correspond to the nonempty subsets of S: the component corresponding to subset P is $\|\Sigma^P(C)\|$.

Remark 6 Note that, by (7), for a Δ-automorphism α of Σ, $f_{\alpha(C)}(P) = f_C(P)$.

We fix $\boldsymbol{A}$ throughout the rest of this paper.

The idea lying behind the "backward" acceptance relies on the following proposition, stating that if a word $\boldsymbol{\sigma}$ of length n is accepted from a set of configurations C, then it is also accepted from any set of configurations that, roughly speaking, coincides with C on the symbols from $[\boldsymbol{\sigma}]$.

[4] This function is the component $f_C^{\bar{\Delta}}$ of the pair of functions $\left(f_C^{\Delta}, f_C^{\bar{\Delta}}\right)$, also denoted f_C, defined in [3], where the first component f_C^{Δ} addresses the configurations having a symbol from Δ in the window. However, it was wrong to introduce this component, because, by the definition of the automaton's run, symbols from Δ cannot appear in the window, see Remark 1. Removing f_C^{Δ} from the definition of f_C in [3] does not affect the decision procedure.

Proposition 3 *Let C' be a finite set of configurations, $\boldsymbol{\sigma} \in L(\boldsymbol{A}_{C'})$, $|\boldsymbol{\sigma}| = n \geq 1$, and let C'' be a finite set of configurations satisfying the conditions below.*

- *For all $\sigma \in [\boldsymbol{\sigma}] \setminus \Delta$,*

$$S^{\sigma}(C'') \subseteq S^{\sigma}(C'). \tag{9}$$

- *For all $P \subseteq S$ such that $\|\Sigma^P(C'')\| \leq n$,*

$$\Sigma^P\left(C''\right) \subseteq \Sigma^P\left(C'\right). \tag{10}$$

- *For all $P \subseteq S$ such that $\|\Sigma^P(C'')\| > n$,*

$$\|\Sigma^P(C')\| > n. \tag{11}$$

Then $\boldsymbol{\sigma} \in L(\boldsymbol{A}_{C''})$.

Proof By (1), we have to show that for each configuration $c'' = [s'', \sigma''] \in C''$, $\boldsymbol{\sigma} \in L(\boldsymbol{A}_{c''})$. We distinguish between the cases of $\sigma'' \in [\boldsymbol{\sigma}]$ and $\sigma'' \notin [\boldsymbol{\sigma}]$.

Assume $\sigma'' \in [\boldsymbol{\sigma}]$. Then, by (9),

$$s'' \in S^{\sigma''}(C'') \subseteq S^{\sigma''}(C'),$$

implying $c'' = [s'', \sigma''] \in C'$. Thus, $\boldsymbol{\sigma} \in L(\boldsymbol{A}_{c''})$.

Assume $\sigma'' \notin [\boldsymbol{\sigma}]$ and let $P = S^{\sigma''}(C'')$.

If $\|\Sigma^P(C'')\| \leq n$, then, by (10),

$$\sigma'' \in \Sigma^P(C'') \subseteq \Sigma^P(C'),$$

implying $c'' = [s'', \sigma''] \in C'$, because, by the definition of P, it contains s''. Thus, like above, $\boldsymbol{\sigma} \in L(\boldsymbol{A}_{c''})$.

Finally, if $\|\Sigma^P(C'')\| > n$, then, by (11),

$$\|\Sigma^P(C')\| > n,$$

as well. Therefore, there is $\sigma' \in \Sigma^P(C')$ that is not in $[\boldsymbol{\sigma}]$. Let α be the automorphism swapping σ'' with σ'. Since, by the definition of P, it contains s'', configuration $c' = [s'', \sigma']$ is in C', implying

$$\boldsymbol{\sigma} \in L(\boldsymbol{A}_{c'}).$$

Since both σ' and σ'' appear in the automaton window, they are not in Δ, implying that α is, indeed, a Δ-automorphism. Therefore, by Corollary 1,

$$\alpha(\boldsymbol{\sigma}) \in L(\boldsymbol{A}_{\alpha(c')}) = L(\boldsymbol{A}_{c''}).$$

Since $\sigma', \sigma'' \notin [\boldsymbol{\sigma}]$, $\alpha(\boldsymbol{\sigma}) = \boldsymbol{\sigma}$, and $\boldsymbol{\sigma} \in L(\boldsymbol{A}_{c''})$ follows. □

4.2 A Partial Order

By Proposition 3, when an automaton runs on a word of length n, the actual size of $\Sigma^P(C)$, if greater than n, does not matter. This motivates the definitions below, cf. the similar definitions in [7, Section IV A], where the infinite set of vectors with non-negative integer components are "converted" into a finite set of vectors with bounded components, by representing the values greater than the bound by a single symbol ω.

Definition 4 We extend the order $<$ from $\mathbb{N}$ to the set $\mathbb{N} \cup \{\omega\}$ by

$$n < \omega,$$

for all non-negative integers n, and, for a set X and two functions $f, g : X \to \mathbb{N} \cup \{\omega\}$, we write $f \leq g$, if for all $x \in X$, $f(x) \leq g(x)$.

Definition 5 For a function $f : X \to \mathbb{N}$ and a non-negative integer n, we define the function $[f \wedge n] : X \to \{0, 1, \ldots, n, \omega\}$ by

$$[f \wedge n](x) = \begin{cases} f(x), & \text{if } f(x) \leq n \\ \omega, & \text{if } f(x) > n \end{cases}.$$

Lemma 3 *For a function $f : X \to \mathbb{N}$ and non-negative integers m and n such that $m < n$, $[f \wedge n] \leq [f \wedge m]$.*

Proof Let $x \in X$. If $f(x) \leq m$, then both $[f \wedge m](x)$ and $[f \wedge n](x)$ are $f(x)$; and, if $m < f(x)$, then $[f \wedge m](x) = \omega$, implying

$$[f \wedge n](x) \leq [f \wedge m](x),$$

because ω is the maximal element of $\mathbb{N} \cup \{\omega\}$. □

Proposition 4 *Let C' and C'' be finite sets of configurations and let n be a non-negative integer such that*

$$[f_{C''} \wedge n] \leq [f_{C'} \wedge n]. \tag{12}$$

Then, for any finite subset Σ' of $\Sigma \setminus \Delta$ there exists a Δ-automorphism α such that

- *for all non-empty subsets P of S, for which $\|\Sigma^P(C'')\| \leq n$,*

$$\alpha(\Sigma^P(C'')) \subseteq \Sigma^P(C'), \tag{13}$$

- *for all non-empty subsets P of S, for which $\|\Sigma^P(C'')\| > n$, there exist $n+1$ pairwise distinct symbols $\{\theta_{P,1}, \ldots, \theta_{P,n+1}\} \subseteq \Sigma^P(C'')$ such that for all $i = 1, \ldots, n+1$,*

$$\alpha(\theta_{P,i}) \in \Sigma^P(C'), \tag{14}$$

and

- *for all $\sigma \in \Sigma'$,*

$$S^{\alpha^{-1}(\sigma)}(C'') \subseteq S^{\sigma}(C'). \tag{15}$$

Proof For every $P \subseteq S$ for which $\|\Sigma^P(C'')\| \leq n$, by the definition of $[f_{C''} \wedge n]$ and (12),

$$f_{C''}(P) = [f_{C''} \wedge n](P) \leq [f_{C'} \wedge n](P),$$

implying

$$\|\Sigma^P(C'')\| \leq \|\Sigma^P(C')\|.$$

Thus, there is a one-to-one mapping $\alpha_P : \Sigma^P(C'') \to \Sigma^P(C')$.

For $P \subseteq S$ such that $\|\Sigma^P(C'')\| > n$, by the definition of $[f_{C''} \wedge n]$, $[f_{C''} \wedge n](P) = \omega$ implying, by (12), $[f_{C'} \wedge n](P) = \omega$ as well. Thus, both $\|\Sigma^P(C')\|$ and $\|\Sigma^P(C'')\|$ are strictly greater than n and we can pick $n+1$ pairwise distinct symbols in $\Sigma^P(C'')$ which we map to some pairwise $n+1$ distinct symbols in $\Sigma^P(C')$. We denote such mapping by α_P.

Now, we define a mapping $\alpha' : \Sigma(C'') \to \Sigma(C')$ by

$$\alpha' = \bigcup_{P \in 2^S \setminus \{\emptyset\}} \alpha_P.$$

That is, for $\sigma \in \Sigma^P(C'')$, $\alpha'(\sigma) = \alpha_P(\sigma)$. Since all $\Sigma^P(C'')$ are mutually disjoint, by (8), α' is well defined and is injective.

Obviously, for all σ in the range of α', we have (15).

Finally, with each element σ of (a finite set) Σ', that is not in the range of α', we associate a symbol σ' not belonging to $\Sigma(C'')$, such that symbols associated with different elements of Σ' are different, and define $\alpha'(\sigma') = \sigma$.

Then

$$S^{\alpha'^{-1}(\sigma)}(C'') = S^{\sigma'}(C'') = \emptyset$$

and (15) follows immediately.

Since both $\Sigma(C'')$ and Σ' are finite and disjoint with Δ and $\Sigma \setminus \Delta$ is infinite, we can extend α' to a Δ-automorphism α of Σ. □

Theorem 2 (Cf. [3, Proposition 12]) *Let C' and C'' be finite sets of configurations and let $\boldsymbol{\sigma} \in L(\boldsymbol{A}_{C'})$, $|\boldsymbol{\sigma}| = n \geq 1$. If*

$$[f_{C''} \wedge n] \leq [f_{C'} \wedge n],$$

then $\alpha^{-1}(\boldsymbol{\sigma}) \in L(\boldsymbol{A}_{C''})$, where α is the Δ-automorphism of Σ provided by Proposition 4 for Σ' being $[\boldsymbol{\sigma}]$.

Proof It suffices to show that $\boldsymbol{\sigma}$ and the sets of configurations C' and $\alpha(C'')$ satisfy the prerequisites of Proposition 3 (for $\alpha(C'')$ instead of C''), which would imply $\boldsymbol{\sigma} \in L(A_{\alpha(C'')})$, that, in turn, would imply $\alpha^{-1}(\boldsymbol{\sigma}) \in L(A_{C''})$.

The prerequisites of Proposition 3 easily follow.

- Let $\sigma \in [\boldsymbol{\sigma}]$. Then

  $$S^{\sigma}(\alpha(C'')) = S^{\alpha^{-1}(\sigma)}(C'') \subseteq S^{\sigma}(C'),$$

  where the equality is by the definition of $\alpha(C'')$, see Definition 2, and the inclusion is by (15). This proves (9).
- Let $P \subseteq S$ such that $\|\Sigma^P(\alpha(C''))\| \leq n$. Then

  $$\Sigma^P(\alpha(C'')) = \alpha(\Sigma^P(C'')) \subseteq \Sigma^P(C'),$$

  where the equality is by the definition of $\alpha(C'')$, see Definition 2, and the inclusion is by (13). This proves (10).
- Finally, let $P \subseteq S$ such that $\|\Sigma^P(\alpha(C''))\| > n$. Then also $\|\Sigma^P(C'')\| > n$. Therefore, there are pairwise distinct symbols $\{\theta_{P,1}, \ldots, \theta_{P,n+1}\} \subseteq \Sigma^P(C'')$ such that

  $$\alpha(\theta_{P,i}) \in \Sigma^P(C'),$$

  implying (11).

□

4.3 Backward Runs of A

The backward runs of $\boldsymbol{A}$ from an accepting ending in a set $C \subseteq F \times \Sigma$ of accepting configurations are gathered in a tree T_C whose nodes of height $\ell = 0, 1, \ldots$ are some of the pairs (f, ℓ), where

$$f : 2^S \setminus \{\emptyset\} \to \{0, 1, \ldots, \ell, \omega\}. \tag{16}$$

The root is the pair $([f_C \wedge 0], 0)$, and the nodes of height $\ell + 1$ are the successors of all nodes of height ℓ, where the successor (parent-child) relation is a subrelation of a relation $\to$ that is defined as follows.

Let i and j be nonnegative integers and let f and g be functions from $2^S \setminus \{\emptyset\}$ to $\{0, 1, \ldots, i, \omega\}$ and $\{0, 1, \ldots, j, \omega\}$, respectively. Then $(f, i) \to (g, j)$, if $j = i+1$ and if there exist two finite sets of configuration C' and C'' and a symbol $\sigma \in \Sigma$ such that

- $C'' \in \mu^{\mathfrak{c}}(C', \sigma)$,
- $g = [f_{C'} \wedge j]$, and
- $f = [f_{C''} \wedge i]$.

That is, for nodes (f, i) and (g, j) of T_C, (g, j) is a successor of (f, i), if $(f, i) \to (g, j)$.

Remark 7 Note that the relation $\to$ is defined on the set of *all* pairs (f, ℓ).

Remark 8 Note that T_C depends on $[f_C \wedge 0]$ only. Thus, the set of such trees T_C is finite, because

$$\|\{T_C : C \subseteq F \times \Sigma\}\| = \|\{[f_C \wedge 0] : C \subseteq F \times \Sigma\}\| = 2^{\|F\|} - 1.$$

Remark 9 Since the set of functions (16) is finite, T_C is of a finite branching degree.

Remark 10 It follows from the definition of T_C that for a word $\boldsymbol{\sigma} = \sigma_1\sigma_2 \cdots \sigma_\ell \in \Sigma$ and a run $C_0, C_1, \ldots, C_\ell = C$ of $\boldsymbol{A}$ on $\boldsymbol{\sigma}$,

$$([f_{C_\ell} \wedge 0], 0) \to ([f_{C_{\ell-1}} \wedge 1], 1) \to ([f_{C_0} \wedge \ell], \ell)$$

is a path in T_C.

Theorem 3 below is the first step in the proof of Lemma 2.

Theorem 3 *There exists a positive integer N_C such that for all paths*

$$(f_0, 0) \to (f_1, 1) \to \cdots \to (f_\ell, \ell)$$

with $\ell \geq N$, there exist i and j, $0 \leq i < j \leq N_C$, such that $f_i \leq f_j$.

Proof Assume to the contrary that there is no such N_C. Then, for each $\ell = 1, 2, \ldots$ there exists a path

$$\pi_\ell = (f_{\ell,0}, 0) \to (f_{\ell,1}, 1) \to \cdots \to (f_{\ell,\ell}, \ell)$$

such that for no i and j, $0 \le i < j \le \ell$, $f_{\ell,i} \le f_{\ell,j}$.

Let T'_C be the subtree of T_C consisting of all nodes appearing in the above paths. This is an infinite tree of a finite branching degree, see Remark 9. By König's Infinity Lemma [8], it contains an infinite path

$$\pi = (f_0, 0) \to (f_1, 1) \to \cdots \to (f_\ell, \ell) \to \cdots .$$

This path is of the form

$$\pi = \lim_{k=1}^{\infty} \pi_{\ell_k},$$

where $\ell_1, \ell_2, \ldots, \ell_k, \ldots$ is an increasing sequence of positive integers. Therefore, for all i and j such that $0 \le i < j$, $f_{\ell,i} \not\le f_{\ell,j}$.

Since S is finite, there is a (finite) nonempty subset $\boldsymbol{P}$ of 2^S and there is an infinite subsequence $f_{m_0}, f_{m_1}, \ldots$ of $f_0, f_1, \ldots$ such that for each nonempty subset P of S, $f_{m_k}(P) = \omega$ if and only if $P \in \boldsymbol{P}$, $k = 0, 1, \ldots$. Then, for all i and j such that $0 \le i < j$, $f_{m_i}|_{\overline{\boldsymbol{P}}} \not\le f_{m_j}|_{\overline{\boldsymbol{P}}}$, where $f_{m_i}|_{\overline{\boldsymbol{P}}}$ and $f_{m_j}|_{\overline{\boldsymbol{P}}}$ are the restrictions of f_{m_i} and f_{m_j} to $(2^S \setminus \{\emptyset\}) \setminus \boldsymbol{P}$, respectively. However, by Remark 5, this contradicts [7, Lemma 4.1] stating that exactly the opposite holds. □

Proposition 5 *The relation* $\to$ *is decidable.*[5]

Recall that $\to$ is defined for any two pairs (f, ℓ) and $(g, \ell + 1)$, independently from T_C.

Lemma 4 *Let C' and C'' be finite sets of configurations and let $\sigma \in \Sigma$ be such that $C'' \in \mu^{\mathbf{c}}(C', \sigma)$. Then, for every nonnegative integer ℓ there exist sets of configurations C_ℓ and $C_{\ell+1}$ such that*

$$C_\ell \in \mu^{\mathbf{c}}(C_{\ell+1}, \sigma), \tag{17}$$

$$\|C_{\ell+1}\| \le (2\ell + 3)\left(2^{\|S\|} - 1\right), \tag{18}$$

$$[f_{C_\ell} \wedge \ell] = [f_{C''} \wedge \ell], \tag{19}$$

and

$$[f_{C_{\ell+1}} \wedge (\ell + 1)] = [f_{C'} \wedge (\ell + 1)]. \tag{20}$$

[5] That is, for pairs (f, ℓ) and $(g, \ell + 1)$, it is decidable whether $(f, \ell) \to (g, \ell + 1)$.

Proof Let $C' = \{c'_1, c'_2, \cdots, c'_k\}$ and let $C''_i \in \mu^{\mathbf{c}}(c'_i, \sigma)$, $i = 1, 2, \ldots, k$, be such that

$$C'' = \bigcup_{i=1}^{k} C''_i.$$

Let the sets of configurations $\widehat{\widehat{C}}$ and $\widehat{C}$ be defined by

$$\widehat{\widehat{C}} = \bigcup_{\|\Sigma^P(C'')\| \leq \ell} P \times \Sigma^P(C'') \ \cup \bigcup_{\|\Sigma^P(C'')\| \geq \ell+1} P \times \{\theta_{P,1}, \ldots, \theta_{P,\ell+1}\},$$

where for $\|\Sigma^P(C'')\| \geq \ell + 1$, $\{\theta_{P,1}, \ldots, \theta_{P,\ell+1}\}$ are pairwise different elements of $\Sigma^P(C'')$, and

$$\widehat{C} = \bigcup_{\|\Sigma^P(C')\| \leq \ell+1} P \times \Sigma^P(C') \ \cup \bigcup_{\|\Sigma^P(C')\| \geq \ell+2} P \times \{\theta_{P,1}, \ldots, \theta_{P,\ell+2}\},$$

where for $\|\Sigma^P(C')\| \geq \ell + 2$, $\{\theta_{P,1}, \ldots, \theta_{P,\ell+2}\}$ are pairwise different elements of $\Sigma^P(C')$.

For each configuration

$$c'' \in \widehat{\widehat{C}} \subseteq C'',$$

let $i_{c''} \in \{1, 2, \ldots, k\}$ be such that $c'' \in C''_{i_{c''}}$.[6]

We contend that the sets of configurations

$$C_{\ell+1} = \widehat{C} \cup \{c'_{i_{c''}} : c'' \in \widehat{\widehat{C}}\} = \{c'_{j_1}, c'_{j_2}, \ldots, c'_{j_{k'}}\} \tag{21}$$

and

$$C_\ell = \bigcup_{i=1}^{k'} C''_{j_i} \tag{22}$$

satisfy the lemma requirements.

Note that

$$\widehat{\widehat{C}} \subseteq C_\ell \subseteq C'', \tag{23}$$

where the first inclusion is by (21) and (22) and the second inclusion is immediate.

[6] Such an $i_{c''}$ is not necessarily unique.

Since, $C''_{j_i} \in \mu^{\mathbf{c}}(c'_{j_i}, \sigma)$, by the definition of $\mu^{\mathbf{c}}$, we have (17).
It follows from the definitions of $C_{\ell+1}$ and C_ℓ that

$$\begin{aligned}\|C_{\ell+1}\| &\leq \|\widehat{C}\| + \|\{c'_{i_{c''}} : c'' \in \widehat{\widehat{C}}\}\| \\ &\leq (\ell+2)(2^{\|S\|}-1) + (\ell+1)(2^{\|S\|}-1) \\ &= (2\ell+3)\left(2^{\|S\|}-1\right)\end{aligned}$$

which proves (18).

For the proof of (19), let P be a nonempty subset of S. It follows from (23), by Remark 4, that

$$\Sigma^P(\widehat{\widehat{C}}) \subseteq \Sigma^P(C_\ell) \subseteq \Sigma^P(C''),$$

implying

$$[f_{\widehat{\widehat{C}}} \wedge \ell] \leq [f_{C_\ell} \wedge \ell] \leq [f_{C''} \wedge \ell],$$

that, in turn, implies (19), because, by the definition of $\widehat{\widehat{C}}$, $\Sigma^P(\widehat{\widehat{C}}) = \Sigma^P(C'')$, if $\|\Sigma^P(\widehat{\widehat{C}})\| \leq \ell$.

The proof of (20) is similar to that of (19). Let P be a nonempty subset of S. Since

$$\widehat{C} \subseteq C_{\ell+1} \subseteq C',$$

by Remark 4

$$\Sigma^P(\widehat{C}) \subseteq \Sigma^P(C_{\ell+1}) \subseteq \Sigma^P(C'),$$

implying

$$[f_{\widehat{C}} \wedge \ell + 1] \leq [f_{C_{\ell+1}} \wedge \ell + 1] \leq [f_{C'} \wedge \ell + 1],$$

that, in turn, implies (20), because, by the definition of $\widehat{C}$, $\Sigma^P(\widehat{C}) = \Sigma^P(C')$. if $\|\Sigma^P(\widehat{C})\| \leq \ell + 1$. □

Proof of Proposition 5 For two pairs (f, ℓ) and $(g, \ell + 1)$ we have to decide whether there exist two finite sets of configuration C' and C'' and a symbol $\sigma \in \Sigma$ such that

- $C'' \in \mu^{\mathbf{c}}(C', \sigma)$,
- $g = [f_{C'} \wedge \ell + 1]$, and
- $f = [f_{C''} \wedge \ell]$.

By Lemma 4, we may restrict ourselves to sets of configurations C' such that

$$\|C'\| \leq (2\ell + 3)\left(2^{\|S\|} - 1\right). \tag{24}$$

That is for each subset $\widehat{C}$ of $S(C') \times \Sigma(C')$ we have to check whether

$$g = [f_{\widehat{C}} \wedge \ell + 1] \tag{25}$$

and then to look for a symbol $\sigma \in \Sigma$ and a set of configurations $\widehat{\widehat{C}} \in \mu^{\mathbf{c}}(\widehat{C}, \sigma)$ such that

$$f = [f_{\widehat{\widehat{C}}} \wedge \ell]. \tag{26}$$

It follows from (24) that

$$\|\Sigma(C')\| \leq \|C'\| \leq (2\ell + 3)\left(2^{\|S\|} - 1\right). \tag{27}$$

Let

$$M = (2\ell + 3)\left(2^{\|S\|} - 1\right) + 1$$

and let

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_M, \theta_{M+1}\}$$

be a set of pairwise distinct symbols from $\Sigma \setminus \Delta$.

Let α be a Δ-automorphism of Σ such that

$$\alpha(\Sigma(C')) \subseteq \{\theta_1, \theta_2, \ldots, \theta_M, \theta_{M+1}\}.$$

Since, by Proposition 2, $\alpha(C'') \in \mu^{\mathbf{c}}(\alpha(C'), \alpha(\sigma))$, and, by Remark 6, $f_{\alpha(C')} = f_{C'}$ and $f_{\alpha(C'')} = f_{C''}$, instead of configurations from $S(C') \times \Sigma(C')$ we may consider configurations from $S(C') \times \Theta$. We have also to distinguish between the cases of $\sigma \in \Delta$ and $\sigma \notin \Delta$.

- For the case of $\sigma \in \Delta$, for each subset C' of $S(C') \times \Theta$ satisfying (25) and for each $\sigma \in \Delta$ we compute all C'' in $\mu^{\mathbf{c}}(C', \sigma)$ and check whether one of them satisfies (26).
- For the case of $\sigma \notin \Sigma(C') \cup \Delta$, assuming, without loss of generality, that $\alpha(\sigma) = \theta_{M+1}$, for each subset C' of $S(C') \times \Theta$ satisfying (25) and, we compute all C'' in $\mu^{\mathbf{c}}(C', \theta_{M+1})$ and check whether one of them satisfies (26). By (27), this covers both the case of $\sigma \in \Sigma(C')$ and the case of $\sigma \notin \Sigma(C')$.

□

Corollary 2 *The set of successors of a node (f, ℓ) of T_C is computable.*

Proof For each function $g : 2^S \setminus \{\emptyset\} \to \{0, 1, \ldots, \ell + 1, \omega\}$, using Proposition 5, we just check whether $(f, \ell) \to (g, \ell + 1)$. Since the set of such functions is finite, the corollary follows. □

Corollary 3 (Cf. [6, Lemma A.5] and [3, Lemma 18]) *A positive integer N_C, whose existence is provided by Theorem 3, is computable.*

Proof Since T_C is of a finite branching degree, see Remark 9, and, by Corollary 2, the set of successors of each node in T_C is computable, for each $\ell = 1, 2 \ldots$, we can, consecutively, check all paths of length ℓ for the existence of i and j provided by Theorem 3. This process will terminate when we arrive at $\ell = N_C$. □

4.4 Proof of Lemma 2

Let

$$N_{\boldsymbol{A}} = \max\{N_C : C \subseteq F \times \Sigma\}.$$

Since, by Remark 8, the number of trees T_C is finite, $N_{\boldsymbol{A}}$ is well defined. Thus, it is computable, because by Corollary 3, each N_C is computable.

Let $\boldsymbol{\sigma} \in L(\boldsymbol{A})$, $|\boldsymbol{\sigma}| = n$ be such that $n \geq N_{\boldsymbol{A}}$ and let $C_0, C_1, \ldots, C_n$ be the accepting run of A on $\boldsymbol{\sigma}$. By Remark 10,

$$([f_{C_n} \wedge 0], 0), ([f_{C_{n-1}} \wedge 1], 1), \ldots, ([f_{C_0} \wedge n], n)$$

is a path in T_{C_n}.

By Theorem 3, there exist i and j, $0 \leq i < j \leq N_{\boldsymbol{A}}$, such that

$$[f_{C_{n-i}} \wedge i] \leq [f_{C_{n-j}} \wedge j] \tag{28}$$

and we define

- $\boldsymbol{\tau} = \sigma_1 \cdots \sigma_{n-j}$,
- $\boldsymbol{\upsilon} = \sigma_{n-j+1} \cdots \sigma_{n-i}$, and
- $\boldsymbol{\varphi} = \sigma_{n-i+1} \cdots \sigma_n$.

Since $i < j$, by Lemma 3 and (28),

$$[f_{C_{n-i}} \wedge j] \leq [f_{C_{n-i}} \wedge i] \leq [f_{C_{n-j}} \wedge j].$$

Since, by definition, $\boldsymbol{v\varphi} \in L(A_{C_{n-j}})$, by Theorem 2, there is a Δ-automorphism α such that $\alpha^{-1}(\boldsymbol{v\varphi}) \in L(A_{C_{n-i}})$, which, together with that $C_0, C_1, \ldots, C_{n-i}$ is a run of A on $\boldsymbol{\tau v}$, imply the desired containment

$$\boldsymbol{\tau v}\alpha^{-1}(\boldsymbol{v\varphi}) \in L(\boldsymbol{A}).$$

5 The Boundness Problem for Alternating Finite-Memory Automata with Two Windows

In this section, reducing the undecidable emptiness problem for two-window alternating finite-memory automata to the boundness problem for two-window alternating finite-memory automata, we show that the latter is undecidable.

Below, $\boldsymbol{A}$ is an r-window alternating finite-memory automaton with the set of states S, the set of accepting states F, and the set of distinguished symbols Δ.

The reduction is as follows.

Lemma 5 *Let* $\$ \notin \Delta$. *Then* $L(\boldsymbol{A}) \neq \emptyset$ *if and only if* $L(\boldsymbol{A}) \cap (\Sigma \setminus \{\$\})^* \neq \emptyset$.

Proof The "if" direction is immediate and, for the proof of the "only if" direction, assume that $L(\boldsymbol{A}) \neq \emptyset$ and let $\boldsymbol{\sigma}$ be a word in $L(\boldsymbol{A})$. If $\$ \notin [\boldsymbol{\sigma}]$, then $\boldsymbol{\sigma} \in L(\boldsymbol{A}) \cap (\Sigma \setminus \{\$\})^* \neq \emptyset$. Otherwise, let $\sigma \notin [\boldsymbol{\sigma}] \cup \Delta$ and let α be a Δ-automorphism of Σ that interchanges $\$$ and σ and leave all other symbols in Σ intact. Then, it follows from Corollary 1 that $\alpha(\boldsymbol{\sigma}) \in L(\boldsymbol{A}) \cap (\Sigma \setminus \{\$\})^*$. □

The language intersection $L(\boldsymbol{A}) \cap (\Sigma \setminus \{\$\})^*$ is accepted by an r-window alternating finite-memory automata $\boldsymbol{A}_1$ that is obtained from $\boldsymbol{A}$ by extending the set of states S with a new sink state q, extending the set of distinguished symbols Δ with $\$$, and extending the transition function μ_Δ (now, $\mu_{\Delta\cup\{\$\}}$) with $\mu_{\Delta\cup\{\$\}}(s, \$) = \{\{q\}\}$, for all states $s \in S \cup \{q\}$.

Corollary 4 $L(\boldsymbol{A})$ *is nonempty if and only if* $L(\boldsymbol{A}_1)\{\$\}^*$ *is unbounded.*

Proof Assume that $L(\boldsymbol{A}) \neq \emptyset$. Then, by Lemma 5, $L(\boldsymbol{A}_1) \neq \emptyset$ as well, and, for all $\boldsymbol{\sigma} \in L(\boldsymbol{A}_1)$ and all $k = 0, 1, \ldots, \boldsymbol{\sigma}\$^k \in L(\boldsymbol{A}_1)\{\$\}^*$, implying that $L(\boldsymbol{A}_1)\{\$\}^*$, is unbounded.

Conversely, assume that $L(\boldsymbol{A}_1)\{\$\}^*$, is unbounded. Then, $L(\boldsymbol{A}_1)\{\$\}^* \neq \emptyset$. Let $\boldsymbol{\sigma} \in L(\boldsymbol{A}_1)\{\$\}^*$. Then for some $\boldsymbol{\sigma}' \in L(\boldsymbol{A}_1)$ and some nonnegative integer k, $\boldsymbol{\sigma} = \boldsymbol{\sigma}'\$^k$. In particular, $\boldsymbol{\sigma}' \in L(\boldsymbol{A}_1)$, implying $L(\boldsymbol{A}_1) \neq \emptyset$. □

The language concatenation $L(\boldsymbol{A}_1)\{\$\}^*$ is accepted by an r-window alternating finite-memory automata $\boldsymbol{A}_2$ that is obtained from $\boldsymbol{A}_1$ by extending the set of accepting F states with a new state f and extending the transition function $\mu_{\Delta\cup\{\$\}}$ with $\mu_{\Delta\cup\{\$\}}(s, \$) = \{\{f\}\}$, for all accepting states $s \in F \cup \{f\}$.

It follows that $L(\boldsymbol{A})$ is nonempty if and only if $L(\boldsymbol{A}_2)$ is unbounded.

It has been shown in [10, Section 5.1] that the universality problem for finite-memory automata is undecidable. A routine inspection of the proof shows that it is undecidable already for two-window automata. Since the class of languages accepted by alternating finite-memory automata includes the class of languages accepted by ordinary finite-memory automata and the former is closed under complement, the emptiness problem for two-window alternating finite-memory automata is undecidable. Therefore, the boundness problem for two-window alternating finite-memory automata is undecidable either.

Finally, it follows from the construction of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ that the emptiness problem for alternating finite-memory automata reduces to the boundness one in linear time. Since the former is not primitive recursive, see [1, 2], the latter is not primitive recursive either.

References

1. Demri, S., Lazić, R.: LTL with the freeze quantifier and register automata. In: Proceedings of the 21th IEEE Symposium on Logic in Computer Science LICS 2006, pp. 17–26 (2006)
2. Demri, S., Lazić, R.: LTL with the freeze quantifier and register automata. ACM Trans. Comput. Log. **10**, 16 (2009)
3. Genkin, D., Kaminski, M., Peterfreund, L.: A note on the emptiness problem for alternating finite-memory automata. Theor. Comput. Sci. **526**, 97–107 (2014)
4. Genkin, D., Kaminski, M., Peterfreund, L.: Closure under reversal of languages over infinite alphabets. In Computer Science - Theory And Applications - 13th International Computer Science Symposium In Russia, CSR 2018. Lecture Notes in Computer Science, vol. 10846, pp. 145–156 (2018)
5. Kaminski, M., Francez, N.: Finite-memory automata. In: Proceedings Of The 31th Annual IEEE Symposium on Foundations of Computer Science FOCS 1990, pp. 683–688 (1990)
6. Kaminski, M., Francez, N.: Finite-memory automata. Theor. Comput. Sci. **134**, 329–363 (1994)
7. Karp, R., Miller, R.: Parallel program schemata. J. Comput. Syst. Sci. **3**, 147–195 (1969)
8. König, D.: Sur les correspondences multivoques des ensembles. Fundam. Math. **8**, 114–134 (1926)
9. Neven, F., Schwentick, T., Vianu, V.: Towards regular languages over infinite alphabets. In: Mathematical Foundations Of Computer Science 2001, 26th International Symposium, MFCS 2001. Lecture Notes in Computer Science, vol. 2136, pp. 560–572 (2001)
10. Neven, F., Schwentick, T., Vianu, V.: Finite state machines for strings over infinite alphabets. ACM Trans. Comput. Log. **5**, 403–435 (2004)
11. Rabin, M., Scott, D.: Finite automata and their decision problems. IBM J. Res. Dev. **3**, 114–125 (1959)
12. Tan, T.: On pebble automata for data languages with decidable emptiness problem. In: Mathematical Foundations Of Computer Science 2009, 34th International Symposium, MFCS 2009. Lecture Notes in Computer Science, vol. 5734, pp. 712–723 (2009)
13. Tan, T.: On pebble automata for data languages with decidable emptiness problem. J. Comput. Syst. Sci. **76**, 778–791 (2010)

Linear Algebraic Quantifiers

Anuj Dawar

Abstract The long-standing research question of whether there is a logic expressing exactly the polynomial-time decidable properties of finite structures has motivated, in recent years, the exploration of logics with linear-algebriac operators. There have been a number of significant recent results on the expressive power of such logics. This paper surveys some of these results and places them within the context of the general theory of Lindström quantifiers, identifying the key closure properties of these quantifiers and relating them to earlier work on arity hierarchies. It provides pointers to the detailed technical proofs of the results on their expressive power.

1 Introduction

Generalized quantifiers (also known as Lindström quantifiers) emerged as objects of study in the context of abstract model theory (see [16]). Classical model theory [22] is centred around first-order logic and its infinitary extensions. As a study of mathematical logic, this is based on a view that divides mathematical practice into those operations that are inherently logical (such as the Boolean connectives and first-order quantifiers) on the one hand and those that are part of the mathematical objects or structures described by the logic, on the other. In contrast, abstract model theory is much more permissive in the kinds of mathematical constructions that it allows within the umbrella of logic. Indeed, any mathematical property can be turned into a logical construct and the main purpose of the subject is to compare the different logics that arise.

A Lindström quantifier is a way of describing extensions of first-order logic (or, indeed, infinitary logic) in an abstract fashion. It gives a way of defining a minimal extension $L(P)$ of a logic L that can also express some fixed property P, while still

A. Dawar (✉)
Department of Computer Science and Technology, University of Cambridge, Cambridge, UK
e-mail: anuj.dawar@cl.cam.ac.uk

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_16

retaining natural closure properties of the logic. Here the property P, which forms the quantifier, can be any isomorphsim-closed class of structures.

While its origins lie in the realm of abstract model theory, the study of Lindström quantifiers acquired an important place in the field of finite model theory and *descriptive complexity*. In this context, we are interested in logics whose expressive power is related to computational complexity classes. Unlike in classical model theory, first-order logic has never played a central role in finite model theory (see [8] for a discussion), and the focus has been on extensions of this logic and on comparing their expressive power. Logics with Lindström quantifiers provide a very general mechanism for defining and comparing such logics. Indeed, comparing the expressive power of logics $L(P)$ and $L(Q)$ for distinct classes of structures P and Q is closely related to comparing the descriptive complexity of the classes themselves. In a precise sense, any reasonable complexity class with certain closure properties can be characterized by a logic that is the extension of first-order logic with a family of Lindström quantifiers (see [7]). This has provided an important motivation for the study of these quantifiers.

János Makowsky's work played an important role in the transfer of knowledge from abstract model theory to finite model theory. Indeed, he was a central figure in the development of abstract model theory in the 1980s (see [30–32]) and made vital contributions, in joint work with Yitzhak Pnueli in the 1990s, developing the study of generalized quantifiers in the context of descriptive complexity theory (see [33–35]). In particular, he formalized the analogy which views Lindström quantifiers as playing the same role in abstract logics that is played by *oracles* in the context of Turing machines.

A central open question in descriptive complexity theory, often described as *the* motivating question, is whether there is a logic for P. That is, a logic in which one can express all, and only, the properties of finite structures that are decidable by polynomial-time algorithms and which permits a computable translation from its formulas to such polynomial-time algorithms. The existence of a logic for P in this sense was formalized by Gurevich [19] and remains one of the major open questions of theoretical computer science. It is known [7] that there is a logic for P if, and only if, there is one that is an extension of first-order logic by means of a *vectorized* family of generalized quantifiers.

While the existence of a logic for P remains an open question, in the field of descriptive complexity a number of logics have been identified that form natural subclasses within P. What's more, for many of them, we have developed sophisticated methods for proving inexpressibility results. That is, methods for showing that some properties are *not* definable in the logic. These methods can be seen as establishing complexity lower bounds for certain natural subclasses of P. This development of new lower bound techniques is one of the most significant contributions of descriptive complexity to the wider field of complexity theory. And, many of these techniques are naturally formulated as methods for proving inexpressibility results for infinitary logics with generalized quantifiers.

Specifically, the starting point of the study of logics for P is the result of Immerman and Vardi [25, 38] to the effect that the extension of first-order logic

with a least fixed-point operation (FP) captures exactly P on ordered structures (such as words). However, there are simple polynomial-time decidable properties of unordered finite structures that are not expressible in FP. One elegant way of demonstrating these impossibility results is the use of pebble games that characterize definability in $\mathsf{L}^{\omega}_{\infty\omega}$, into which FP can be translated (see [28]). Then, it is natural to consider extensions of FP by means of families of generalized quantifiers or indeed other forms of logical operators. One widely studied extension of FP is fixed-point logic with counting (FPC). This was proposed by Immerman [25] as a possible candidate logic for P and has since been extensively studied: see [36] for a book-length treatment and [9] for a more recent review. It was shown by Cai et al. [4] that the expressive power of FPC is strictly weaker than P. In other words, there are polynomial-time decidable classes of finite structures that are not definable in FPC. The method for proving this is again by embedding FPC in the much richer infinitary logic $\mathsf{C}^{\omega}_{\infty\omega}$ with generalized counting quantifiers and using a suitable pebble game to prove inexpressibility in this logic. The proof required the sophisticated construction of graph classes on which the games are played.

One research direction that has emerged in recent years, seeking to understand the gap between FPC and P, has been the study of extensions of FP by means of *algebraic* operators [13, 18], and specifically ones based on linear algebra. This is because such operations provide a rich source of examples of polynomial-time computations that are not expressible in FPC. It turns out again that the resulting logics are most effectively analysed by embedding them in an infinitary logic with generalized quantifiers [11, 14]. A considerable amount of algebraic machinery has since been developed and deployed to proving inexpressibility results for these logics (see [15, 29]).

The aim of the present contribution is to explain the significance of linear algebraic quantifiers within the broader theory of logics with generalized quantifiers. While the literature on linear algebraic quantifiers in finite model theory has been growing, it is often focused on new results of great technical complexity and it may become difficult to discern the bigger picture from this literature. In this paper, as well as presenting this bigger picture, I aim to show how the quantifiers arise naturally from considering closure conditions of classes of structures that are stronger than isomorphism. Moreover, I explain why these stronger conditions are necessary to consider given what we know about the power and limitations of classes of generalized quantifiers based on their arity. This account then places the study of linear algebraic quantifiers squarely in its natural place in the developing story of Lindström quantifiers in finite model theory.

2 Logics and Quantifiers

In the bulk of this article, we are interested in extensions of infinitary logics (with finitely many variables per formula) by means of Lindström quantifiers. In this section I give a quick introduction to these logics to fix notation and a more detailed

presentation can be found, for instance, in the textbook of [17]. The reason for considering infinitary logics is that they subsume fixed-point logics and provide the most effective means of proving limitations on the expressive power of these logics. This is done by considering equivalence relations induced by the infinitary logics (or, more precisely indexed families of equivalence relations) which we review in Sect. 2.3.

2.1 Basic Logics

For convenience, we consider just purely relational vocabularies and finite structures. Thus, a vocabulary σ is a finite sequence of relation symbols R, each with an associated arity $\mathrm{ar}(R)$. A σ-structure $\mathbb{A}$ is a finite set A along with an interpretation $R^{\mathbb{A}} \subseteq A^{\mathrm{ar}(R)}$ of each symbol R in σ. We are interested in the definability of classes of structures in a logic. A *class* of structures always means an isomorphism-closed class.

We write FO or $\mathsf{L}_{\omega\omega}$ for *first-order logic*. For a fixed vocabulary σ, the collection of formulas of FO over σ is given by the closure of atomic formulas (i.e. formulas of the form $R(x_1, \ldots, x_{\mathrm{ar}(R)})$ and $x = y$) under *finitary* Boolean operations and first-order quantification.

The infinitary logic $\mathsf{L}_{\infty\omega}$ is obtained by closing FO under the operations of infinitary conjunction and disjunction. That is, for any set S of formulas of $\mathsf{L}_{\infty\omega}$, we have formulas $\bigvee S$ and $\bigwedge S$ that are the disjunction and conjunction respectively of the set of formulas S. As we are only interested in finite structures, it suffices to restrict ourselves to countable sets. Let $\mathsf{L}_{\omega_1\omega}$ denote the logic obtained by closing FO under the operations $\bigvee S$ and $\bigwedge S$ for *countable* sets S. Then, over finite structures, the expresive power of $\mathsf{L}_{\omega_1\omega}$ is *complete*. I make this notion precise with the following definition.

Definition 1 For a logic L and a class of structures C, we say that the expressive power of L is *complete* on C if for every class $\mathcal{K}$ of structures, there is a formula $\phi_{\mathcal{K}}$ of L that defines $\mathcal{K}$ within C: that is, for all structures $\mathbb{A}$ in C, $\mathbb{A} \models \phi_{\mathcal{K}}$ if, and only if, $\mathbb{A}$ is in $\mathcal{K}$.

We can now state the completeness result.

Proposition 1 *For any vocabulary σ, the expressive power of $\mathsf{L}_{\omega_1\omega}$ is complete on the class of finite σ-structures.*

The proposition follows immediately from the fact that for each finite structure $\mathbb{A}$ there is a sentence $\phi_{\mathbb{A}}$ of FO which defines $\mathbb{A}$ up to isomorphism (see, for example, [8, p. 94] for a construction). Then, a class $\mathcal{K}$ of finite σ-structures is defined by the countable disjunction $\bigvee\{\phi_{\mathbb{A}} \mid \mathbb{A} \in \mathcal{K}\}$ or equivalently, the countable conjunction $\bigwedge\{\neg\phi_{\mathbb{A}} \mid \mathbb{A} \notin \mathcal{K}\}$.

Proposition 1 shows that $\mathsf{L}_{\omega_1\omega}$ is much too rich a logic to study in the context of finite model theory. There are no limitations one can establish on its expressive

power. And, for us, the point of model-theoretic study is precisely to establish what are the limits of what is expressible, and to develop the necessary tools to do so. An interesting restriction of $\mathsf{L}_{\omega_1\omega}$, first proposed by Barwise [2], is to restrict the total number of variables (free or bound) that can appear in a single formula. The resulting logics, the finite-variable infinitary logics were introduced in the subject of finite model theory through the work of [28], and they form the base logics whose extensions we study in the present article.

Definition 2 For any positive integer k, $\mathsf{L}^k_{\infty\omega}$ denotes the collection of formulas ϕ of $\mathsf{L}_{\infty\omega}$ such that ϕ contains at most k distinct variables, free or bound.
$\mathsf{L}^\omega_{\infty\omega}$ denotes the collection $\bigcup_{k\in\omega} \mathsf{L}^k_{\infty\omega}$.

Sometimes the logic $\mathsf{L}^k_{\infty\omega}$ is defined as the collection of formulas of $\mathsf{L}_{\infty\omega}$ built using only the variables $x_1, \ldots, x_k$. As we are interested in the expressive power rather than the syntax of the logic, we can treat these alternative definitions as equivalent. Indeed, an objection sometimes raised to both versions is that it is a strange logic whose collection of formulas is not closed under the operation of renaming bound variables. To address this we could adopt a more liberal definition which takes $\mathsf{L}^k_{\infty\omega}$ to be the collection of those formulas ϕ of $\mathsf{L}_{\infty\omega}$ such that no subformula of ϕ contains more than k free variables. This is again equivalent to our definition in terms of expressive power and has the advantage that it highlights the key reason for limiting the number of variables: it limits the number of distinct elements of a structure that can be simultaneously addressed by a subformula, and this is the key resource limitation that we want to study.

Another important logic that is much studied in finite model theory is the extension of $\mathsf{L}^\omega_{\infty\omega}$ with *counting quantifiers*, denoted $\mathsf{C}^\omega_{\infty\omega}$. We take this up in the next subsection in the context of a more general discussion of generalized quantifiers.

2.2 Generalized Quantifiers

We can associate a generalized quantifier with any class $\mathcal{K}$ of structures. In this article, we do not distinguish notationally between the class $\mathcal{K}$ and the quantifier it defines. For a logic L, we write $L(\mathcal{K})$ to denote the extension of L with the quantifier $\mathcal{K}$. More generally, if Q is a collection of quantifiers, we write $L(Q)$ for the extension of L with *all* quantifiers in Q. The base logic L that we consider will always be one of the logics introduced in Sect. 2.1 above: $\mathsf{L}_{\omega\omega}$, $\mathsf{L}^k_{\infty\omega}$ or $\mathsf{L}^\omega_{\infty\omega}$. We now define these notions more formally.

Let σ, τ be relational vocabularies with $\tau = \{R_1, \ldots, R_m\}$, and $\mathrm{ar}(R_i) = r_i$ for each $i \in [m]$. A *simple interpretation* $\mathcal{I}$ of τ in σ with parameters $\mathbf{z}$ is a tuple of σ-formulas $(\psi_1, \ldots, \psi_m)$ along with tuples $\mathbf{y}_1, \ldots, \mathbf{y}_m$ of variables with $|\mathbf{y}_i| = r_i$ for $i \in [m]$, such that the free variables of ψ_i are among $\mathbf{y}_i\mathbf{z}$. Such a simple interpretation defines a mapping that takes a σ-structure $\mathbb{A}$, along with an interpretation α of the parameters $\mathbf{z}$ in $\mathbb{A}$ to a τ-structure $\mathbb{B}$ as follows. The universe

of $\mathbb{B}$ is A, and the relations $R_i \in \tau$ are interpreted in $\mathbb{B}$ by $R_i^{\mathbb{B}} = \{\mathbf{b} \in A^{r_i} \mid (\mathbb{A}, \alpha[\mathbf{b}/\mathbf{y}_i]) \models \psi_i\}$. We denote this mapping also $\mathcal{I}$.

Let L be a logic and $\mathcal{K}$ a class of τ-structures. The extension $L(\mathcal{K})$ of L by the *generalized quantifier* $\mathcal{K}$ is obtained by extending the syntax of L by the following formula formation rule:

> For $\mathcal{I} = (\psi_1, \ldots, \psi_m)$ a simple interpretation of τ in σ with parameters $\mathbf{z}$, $\psi(\mathbf{z}) = \mathcal{K}\mathbf{y}_1, \ldots, \mathbf{y}_m\mathcal{I}$ is a formula over the signature σ, with free variables $\mathbf{z}$. The semantics of the formula is given by $(\mathbb{A}, \alpha) \models \psi(\mathbf{z})$, if, and only if, $\mathcal{I}(\mathbb{A}, \alpha)$ is in the class $\mathcal{K}$.

The extension $L(Q)$ of L by a collection Q of generalized quantifiers is defined by adding the rules above to L for each $\mathcal{K} \in Q$ simultaneously.

The *type* of the quantifier $\mathcal{K}$ is $(r_1, \ldots, r_m)$, and the *arity* of $\mathcal{K}$ is $\max\{r_1, \ldots, r_m\}$.

Example 1

1. The *existential quantifier* has type (1) and consists of the class of structures (A, U) where $U \subseteq A$ is non-empty.
2. For any positive integer m, the m-counting quantifier (often denoted $\exists^{\geq m}$) has type (1) and consists of the class of structures (A, U) where $|U| \geq m$.
3. The *connectivity* or *transitive closure* quantifier has type (2, 1, 1) and consists of structures (A, E, S, T) where for every $s \in S$ and $t \in T$, there is a path from s to t in the directed graph (A, E).
4. The *Hamiltonicity* quantifier has type (2) and consists of directed graphs (V, E) that contain a Hamiltonian cycle.

It is common in the literature to consider interpretations more general than the ones I have defined here. In particular, for the interpretations $\mathcal{I}$ that I have defined, it is always the case that the universe of $\mathcal{I}(\mathbb{A})$ is the same as that of $\mathbb{A}$. In more generous notions of interpretations, which allow for vectorization, relativization and quotienting, we can interpret structures from $\mathbb{A}$ whose universe is a quotient (under a suitable congruence) of a subset of a power of A. I review some of these general notions of interepretation in Sect. 3 below. For the purpose of defining quantifiers, the restricted notion suffices.

2.3 Induced Equivalence

The key reason for studying infinitary logics of the form $\mathsf{L}^{\omega}_{\infty\omega}$ and its extensions by means of generalized quantifiers is twofold. On the one hand, the expressive power of these logics subsumes that of fixed-point logics (and their corresponding extensions by means of general operators) and on the other hand, there are powerful combinatorial tools to study the expressive power of the infinitary logics. The limit on the number of variables is what gives us the handle to define these combinatorial tools, which often take the form of games. The expressive power of $\mathsf{L}^{k}_{\infty\omega}$ is

characterized by a k-pebble game, first introduced in the form of a *back-and-forth* system by Barwise [2] (see also [24, 37]). Extensions of these pebble games play an important role in the story of generalized quantifiers in finite model theory. They are aimed at characterizing elementary equivalence in the logics $\mathsf{L}^{\omega}_{\infty\omega}(Q)$ for various families of quantifiers Q. Of particular interest to us are the bijection game of [20] and the invertible map game of [11] which we review in Sect. 4.2.

Definition 3 For a collection Q of quantifiers and a positive integer k, we write $\mathbb{A} \equiv^k_Q \mathbb{B}$ to denote that for every sentence ϕ of $\mathsf{L}^k_{\infty\omega}(Q)$, $\mathbb{A} \models \phi$ if, and only if, $\mathbb{B} \models \phi$.

As usual, we write $\mathbb{A} \cong \mathbb{B}$ to mean that the structures $\mathbb{A}$ and $\mathbb{B}$ are isomorphic.

In dealing with infinitary logic on finite structures, there is a tight correspondence between the expressive power of the logic and the equivalence relation it induces. To make this precise, we introduce the following definition.

Definition 4 For a relational vocabulary τ we say that an equivalence relation $\sim$ on τ-structures is *discrete* if $\mathbb{A} \sim \mathbb{B}$ if, and only if, $\mathbb{A} \cong \mathbb{B}$.

We are interested in the question of when $\equiv^k_Q$ is discrete. We begin with the following observation.

Theorem 1 *The following are equivalent for any set of quantifiers Q and any finite relational vocabulary τ.*

1. *There is some k such that $\equiv^k_Q$ is discrete on τ-structures.*
2. *The expressive power of $\mathsf{L}^{\omega}_{\infty\omega}(Q)$ is complete on τ-structures.*

Proof For the first direction, assume $\equiv^k_Q$ is discrete and let $\mathcal{K}$ be any isomorphism-closed class of finite τ-structures. Note that since τ is finite, there are, up to isomorphism, only countably many finite τ-structures. By the discreteness of $\equiv^k_Q$, for every pair $\mathbb{A}, \mathbb{B}$ of non-isomorphic structures there is a sentence $\delta_{\mathbb{A},\mathbb{B}}$ of $\mathsf{L}^k_{\infty\omega}(Q)$ which is true in $\mathbb{A}$ and false in $\mathbb{B}$. Then $\mathcal{K}$ is defined by the following sentence:

$$\bigvee_{\mathbb{A}\in\mathcal{K}} \bigwedge_{\mathbb{B}\notin\mathcal{K}} \delta_{\mathbb{A},\mathbb{B}}.$$

For the other direction, assume $\equiv^k_Q$ is not discrete for any k. This implies that we can find, for each k, a pair of structures $\mathbb{A}_k, \mathbb{B}_k$ that are not isomorphic, but for which we have $\mathbb{A}_k \equiv^k_Q \mathbb{B}_k$. We can then construct a class of structures that, for infinitely many different values of k contains exactly one of the pair of structures $\mathbb{A}_k$ and $\mathbb{B}_k$. We construct this class $\mathcal{K}$ as follows.

Suppose first that there are only finitely many structures (up to isomorphism) among $\{\mathbb{A}_k, \mathbb{B}_k \mid k \in \omega\}$. Then, it must be that there is a single pair $\mathbb{A}, \mathbb{B}$ of non-isomorphic structures such that $\mathbb{A} \equiv^k_Q \mathbb{B}$ for infinitely many values of k. Then, taking $\mathcal{K}$ to be the class of structures isomorphic to $\mathbb{A}$ we see that $\mathcal{K}$ is not definable in $\mathsf{L}^{\omega}_{\infty\omega}(Q)$. Indeed, if it were, it would be definable by a sentence α in $\mathsf{L}^{l}_{\infty\omega}(Q)$ for

some $l \in \omega$. Since $\mathbb{A} \equiv^k_Q \mathbb{B}$ for some $k > l$, and $\mathbb{A} \models \alpha$ we have $\mathbb{B} \models \alpha$. However, $\mathbb{B}$ is not isomorphic to $\mathbb{A}$ and hence not in $\mathcal{K}$ and we have a contradiction.

Suppose then that there are infinitely many distinct pairs of structures among $\{(\mathbb{A}_k, \mathbb{B}_k) \mid k \in \omega\}$. We now define two disjoint classes of structures $\mathcal{K}^+$ and $\mathcal{K}^-$ in stages indexed by the natural numbers. Let $\mathcal{K}^+_1$ be the collection of all structures isomorphic to $\mathbb{A}_1$ and $\mathcal{K}^-_1$ be the collection of all structures isomorphic to $\mathbb{B}_1$. Inductively, assume that $\mathcal{K}^+_k$ and $\mathcal{K}^-_k$ have been defined and consider the pair of structures $\mathbb{A}_{k+1}$ and $\mathbb{B}_{k+1}$. If both these structures are already in $\mathcal{K}^+_k \cup \mathcal{K}^-_k$, then $\mathcal{K}^+_{k+1} = \mathcal{K}^+_k$ and $\mathcal{K}^-_{k+1} = \mathcal{K}^-_k$. Otherwise, suppose exactly one out of $\mathbb{A}_{k+1}$ and $\mathbb{B}_{k+1}$ is in $\mathcal{K}^+_k \cup \mathcal{K}^-_k$, say without loss of generality that it is $\mathbb{A}_{k+1}$ that is in $\mathcal{K}^-_k$. Then, we let $\mathcal{K}^-_{k+1} = \mathcal{K}^-_k$ and $\mathcal{K}^+_{k+1}$ be the isomorphism closure of $\mathcal{K}^+_k \cup \{\mathbb{B}_{k+1}\}$. Finally, if neither of $\mathbb{A}_{k+1}$ and $\mathbb{B}_{k+1}$ is in $\mathcal{K}^+_k \cup \mathcal{K}^-_k$, we let $\mathcal{K}^+_{k+1} = \mathcal{K}^+_k \cup \{\mathbb{A}_{k+1}\}$ and $\mathcal{K}^-_{k+1} = \mathcal{K}^-_k \cup \{\mathbb{B}_{k+1}\}$, again closing under isomorphisms. Finally, we let $\mathcal{K} = \bigcup_k \mathcal{K}^+_k$. Since, at each stage k, there are only finitely many structures in $\mathcal{K}^+_k \cup \mathcal{K}^-_k$ up to isomorphism, it follows that infinitely often we add a structure to one or the other and therefore for infinitely many values of k, $\mathcal{K}$ contains exactly one of $\mathbb{A}_k$ and $\mathbb{B}_k$ as desired.

It then follows that $\mathcal{K}$ is not definable in $\mathsf{L}^\omega_{\infty\omega}(Q)$. Indeed, if it were, it would be definable by a sentence ϕ of $\mathsf{L}^k_{\infty\omega}(Q)$ for some k. Choose an $l > k$ for which $\mathcal{K}$ contains exactly one of $\mathbb{A}_l$ and $\mathbb{B}_l$ and we see that $\mathbb{A}_l \models \phi$ if, and only if, $\mathbb{B}_l \models \phi$, giving a contradiction □

3 Arity Hierarchies

In the study of generalized quantifiers in the context of finite model theory, among the most important early results are the classification of quantifiers by their *arity*. Recall that the arity of a quantifier $\mathcal{K}$, given as a class of τ-structures, is the largest arity of any relation in τ. Limitations on the power of such quantifiers were established by Hella [20] and these results opened the door to the study of families of quantifiers of unbounded arity in the context of descriptive complexity. I review these results and show why they lead naturally to the consideration of families of quantifiers with restrictive closure conditions.

Write Q_n for the collection of *all* quantifiers of arity at most n. Hella [20] proved that there are, for every n, classes of structures that are not definable in $\mathsf{L}^\omega_{\infty\omega}(Q_n)$. A key contribution of his work is the introduction of the *bijection game*, a Spoiler-Duplicator game that characterizes exactly the equivalence relation $\equiv^k_{Q_n}$. Specifically, the game is characterized by two parameters k and n. It is played on a pair of structures $\mathbb{A}$ and $\mathbb{B}$ with k pairs of pebbles, with up to n pebbles moving in each round. At each move, *Duplicator* chooses a bijection h between $\mathbb{A}$ and $\mathbb{B}$ and *Spoiler* chooses a tuple $\mathbf{a}$ of length at most n of elements of $\mathbb{A}$ and the pebbles that are moving are placed on $\mathbf{a}$ and $h(\mathbf{a})$. Paraphrasing the main result of the paper into the terminology we have adopted in this paper, it can be stated as follows.

Theorem 2 (Hella) *For every positive integer n, there is a vocabulary τ such that $\equiv^k_{Q_n}$ is not discrete on τ-structures for any k.*

Moreover, the class of structures that is shown not definable in $\mathsf{L}^\omega_{\infty\omega}(Q_n)$ can be chosen to be decidable in polynomial time. A consequence of the result, therefore, is that the class P cannot be captured by any extension of fixed-point logic with quantifiers of bounded arity.

It is important to note the dependence, in the statement of Theorem 2, of τ on n. In particular, the vocabulary τ necessarily contains relations of arity $n+1$ or greater. Indeed, it is clear that if τ contains only relations of arity n or less, then the expressive power of $\mathsf{L}^\omega_{\infty\omega}(Q_n)$ on the class of finite τ-structures is complete. Indeed, any class $\mathcal{K}$ of τ-structures is trivially defined by the sentence $\mathcal{K}\mathbf{x}\mathcal{I}$ where $\mathcal{I}(\mathbf{x})$ is the identity interpretation taking a structure to itself. Thus, by Theorem 1, it follows that $\equiv^k_{Q_n}$ is discrete on this class of structures when $k \geq n$.

Thus, in particular, the expressive power of $\mathsf{L}^\omega_{\infty\omega}(Q_2)$ is complete on the class of finite graphs. On the other hand, we know that there is a logic exactly capturing P if, and only if, there is one that captures P on graphs. And, Theorem 2 does not rule out the possibility that a logic capturing P on graphs might be an extension of fixed-point logic with a collection of quantifiers of bounded arity. Thus, what exactly Theorem 2 tells us about quantifiers needed to capture P is subtle and worth exploring further. We begin by looking at quantifiers of arity one and two.

3.1 Unary and Binary Quantifiers

Quantifiers of arity one, also known as *unary* quantifiers are the simplest to characterize and have been the object of much study. Of the example quantifiers in Example 1, the existential quantifier and the m-counting quantifiers are unary. Indeed, the counting countifiers play a special role in the development of the theory of quantifiers. Recall that, for any positive integer m, the m-counting quantifier is given by the class of structures (A, U) where $|U| \geq m$. The quantifier is usually written $\exists^{\geq m}$ so that the formula $\exists^{\geq m} x\phi$ is to be read as "there are at least m elements x such that ϕ." We write C for the collection of quantifiers $\{\exists^{\geq m} \mid m \in \omega\}$. It is common to write $\mathsf{C}^k_{\infty\omega}$ and $\mathsf{C}^\omega_{\infty\omega}$ for the logics $\mathsf{L}^k_{\infty\omega}(C)$ and $\mathsf{L}^\omega_{\infty\omega}(C)$ respectively and we adopt this convention.

It is not difficult to show that every unary quantifier is definable in $\mathsf{C}^\omega_{\infty\omega}$ (see [20, 27]).

Proposition 2 *Every formula of $\mathsf{L}^k_{\infty\omega}(Q_1)$ is equivalent to a formula of $\mathsf{C}^{k+1}_{\infty\omega}$.*

In particular, this means that Hella's k, 1-bijection game can be used to characterize the equivalence relation $\equiv^{k+1}_C$. This equivalence relation is much studied in the context of graph isomorphism under the name k-dimensional Weisfeiler-Leman equivalence (see [26]). It is an approximation of isomorphism that is decidable in polynomial time (to be precise in time $n^{O(k)}$ for n-vertex graphs). This equivalence

relation is used to understand the limits of the expressive power of FPC, the extension of fixed-point logic with counting terms. In particular, the famous construction of [4] gives a polynomial-time decidable class of graphs that is not definable in $\mathsf{C}^{\omega}_{\infty\omega}$ and, *a fortiori*, not in FPC. It is this construction that Hella generalizes to establish Theorem 2. While the result of Cai et al. showed that the properties definable in FPC form a strict subclass of P, it is a subclass of great interest (see [9]) and the structure of the equivalence relations $\equiv^{k+1}_{C}$ remains key to its analysis.

Turning now to binary quantifiers, it is immediately clear that $\mathsf{L}^{\omega}_{\infty\omega}(Q_2)$ can express all properties of graphs.

Proposition 3 *On graphs, the expressive power of* $\mathsf{L}^{\omega}_{\infty\omega}(Q_2)$ *is complete.*

Recall that much of the interest in studying the expressive power of infinitary logics with generalized quantifiers comes from the search for extensions of fixed-point logics that might characterize P, particularly on graphs. Proposition 3, together with Theorem 2 suggest that limiting quantifiers by just their arity is not the most productive way of approaching the problem. We could consider an extension of fixed-point logic with all quantifiers in Q_2 which are polynomial-time decidable. This can certainly express all graph properties in P but fails (by Theorem 2) to express all of P on richer signatures.[1] A key reason is that it is not closed under *first-order reductions*. That is to say, there are classes of structures (say involving a vocabulary with a ternary relation) that are polynomial-time definable that are reducible by means of a first-order reduction to a polynomial-time quantifier in Q_2 but not themselves definable in $\mathsf{L}^{\omega}_{\infty\omega}(Q_2)$. We now turn to the topic of reductions.

3.2 Interpretations

In Sect. 2.2, we saw the definition of a *simple interpretation*. This is a means of defining a function from σ-structures to τ-structures using formulas (of a suitable logic). One strong limitation of the functions so defined is that the universe of the target structure is the same as that of the source. In particular, we cannot use such functions to map a σ-structure to a τ-structure that is larger in size. For this we need interpretations of higher dimension which we now define.

As before we fix vocabularies σ and τ with $\tau = (R_1, \ldots, R_m)$, and $\mathrm{ar}(R_i) = r_i$ for each $i \in [m]$ and let L be a logic. A *d-ary L-interpretation of τ in σ* is a sequence of L-formulas in vocabulary σ consisting of: (i) a formula $\delta(\mathbf{x})$; (ii) a formula $\varepsilon(\mathbf{x}, \mathbf{y})$; and (iii) for each relation symbol $R_i \in \tau$ a formula $\phi_{R_i}(\mathbf{x}_1, \ldots, \mathbf{x}_{r_i})$, where each $\mathbf{x}$, $\mathbf{y}$ or $\mathbf{x}_j$ is a d-tuple of variables. We call d the *dimension* of the interpretation. We say that an interpretation $\mathcal{I}$ associates a τ-structure $\mathbb{B}$ to a σ-structure $\mathbb{A}$ if there is a map h from $\{\mathbf{a} \in A^d \mid \mathbb{A} \models \delta[\mathbf{a}]\}$ to the universe B of $\mathbb{B}$

[1] It also fails a key criterion of being a logic for P as it does not have an effective syntax.

such that: (i) h is surjective onto B; (ii) $h(\mathbf{a}_1) = h(\mathbf{a}_2)$ if, and only if, $\mathbb{A} \models \varepsilon[\mathbf{a}_1, \mathbf{a}_2]$; and (iii) $R_i^{\mathbb{B}}(h(\mathbf{a}_1), \ldots, h(\mathbf{a}_{r_i}))$ if, and only if, $\mathbb{A} \models \phi_{R_i}[\mathbf{a}_1, \ldots, \mathbf{a}_{r_i}]$. Note that an interpretation $\mathcal{I}$ associates a τ-structure with $\mathbb{A}$ only if ε defines an equivalence relation on A^d that is a congruence with respect to the relations defined by the formulae ϕ_R. In such cases, however, $\mathbb{B}$ is uniquely defined up to isomorphism and we write $\mathcal{I}(\mathbb{A}) = \mathbb{B}$. It is also worth noting that the size of $\mathbb{B}$ is at most n^d, if $\mathbb{A}$ is of size n.

If $\mathcal{C}$ and $\mathcal{D}$ are two classes of structures and $\mathcal{I}$ is an L-interpretation such that for any structure $\mathbb{A}$ we have $\mathbb{A} \in \mathcal{C}$ if, and only if, $\mathcal{I}(\mathbb{A}) \in \mathcal{D}$ then we say that $\mathcal{I}$ is an L-reduction of $\mathcal{C}$ to $\mathcal{D}$. If there is an L-reduction from $\mathcal{C}$ to $\mathcal{D}$, we say that $\mathcal{C}$ is L-reducible to $\mathcal{D}$ and write this $\mathcal{C} \leq_L \mathcal{D}$.

Of particular interest in the field of descriptive complexity are FO-reductions. Say that a logic L is closed under FO-reductions if, whenever $\mathcal{D}$ is definable in L, and $\mathcal{C} \leq_{\mathrm{FO}} \mathcal{D}$, then $\mathcal{C}$ is also definable in L. These are, in complexity-theoretic terms, a particularly weak form of reduction and almost any reasonable complexity class is closed under FO-reductions. Thus, we are especially interested in logics that are also closed under FO-reductions. In particular, since the complexity class P is closed under FO-reductions, any logic that captures it must also be. Moreover, it is known that there is a logic for P in the sense of Gurevich if, and only if, there is a problem in P which is complete under FO-reductions [7].

We know that $\mathsf{L}^{\omega}_{\infty\omega}(Q_n)$ is not closed under FO-reductions for any fixed n. Thus, it is natural to look to families of quantifiers of unbounded arity. At the same time, we do not want to consider including *all* quantifiers as we are interested in limiting expressive power. This leads us to constructions that generate specific families of quantifiers of unbounded arity. Of particular interest are *vectorized* quantifiers which are just the closure of a quantifier under FO-reductions.

3.3 Vectorized Quantifiers

Let $\tau = (R_1, \ldots, R_m)$ be a relational vocabulary and $\mathcal{K}$ a class of σ-structures. For a logic L, we obtain a minimal extension of L that can express $\mathcal{K}$ *and* is closed under FO-reductions by closing L under the family $\overline{\mathcal{K}} = \{\mathcal{K}_d \mid d \in \omega\}$ of quantifiers, where $\mathcal{K}_d$ is a quantifier in the vocabulary

$$\tau_d = (U_d, \sim_d, (R_{i,d})_{i\in[m]}),$$

where $\mathrm{ar}(U_d) = d$, $\mathrm{ar}(\sim_d) = 2d$ and $\mathrm{ar}(R_{i,d}) = d \cdot \mathrm{ar}(R_i)$. It is defined by

$$\mathbb{A} \in \mathcal{K}_d \quad \text{if, and only if,} \quad (U_d^{\mathbb{A}} / \sim_d^{\mathbb{A}}, (R_{i,d}^{\mathbb{A}})_{i\in[m]}) \in \mathcal{K}.$$

Here, $(U_d^{\mathbb{A}} / \sim_d^{\mathbb{A}}, (R_{i,d}^{\mathbb{A}})_{i\in[m]})$ is a τ-structure that is defined if, and only if, the relation $\sim_d^{\mathbb{A}}$ defines an equivalence relation on the set of d-tuples of $\mathbb{A}$ in the relation

$U_d^{\mathbb{A}}$ and moreover, $\sim_d^{\mathbb{A}}$ is a congruence with respect to each of the relations $R_{i,d}^{\mathbb{A}}$. Then, the universe of the structure is $U_d^{\mathbb{A}}/\sim_d^{\mathbb{A}}$, i.e. the quotient of the set of d-tuples in $U_d^{\mathbb{A}}$ under the equivalence relation $\sim_d^{\mathbb{A}}$, and each relation symbol R_i in τ is interpreted as the collection of equivalence classes which are included in $R_{i,d}^{\mathbb{A}}$.

We call $\mathcal{K}_d$ the dth *vectorization* of the quantifier $\mathcal{K}$ and we call the family $\overline{\mathcal{K}}$ the *family of vectorized quantifiers* generated by $\mathcal{K}$. Sometimes, by an abuse of terminology, we call this infinite family *the vectorized quantifier* given by $\mathcal{K}$. It should be clear from the definitions that if $\mathcal{I}(\mathbf{x})$ is a d-ary interpretation of τ in a vocabulary σ, then the formula $\mathcal{K}_d\mathbf{x}\mathcal{I}$ is satisfied in a σ-structure $\mathbb{A}$ if, and only if, $\mathcal{I}(\mathbb{A}) \in \mathcal{K}$. This ensures that if $\mathcal{C} \leq_L \mathcal{D}$ for classes of structures $\mathcal{C}$ and $\mathcal{D}$, then $\mathcal{C}$ is definable in $L(\overline{\mathcal{D}})$. See [17, Chapter 12] for a detailed treatment.

It is worth noting that $\overline{\mathcal{K}}$ is not included in Q_n for any fixed $n \in \omega$ and thus the inexpressiveness results for the logics $\mathsf{L}^{\omega}_{\infty\omega}(Q_n)$ established by Hella do not yield any limitations on the power of $\mathsf{L}^{\omega}_{\infty\omega}(\overline{\mathcal{K}})$. Let us write $\overline{Q_n}$ for the collection of vectorizations of all quantifiers in Q_n. More generally, for any collection S of quantifiers, we write $\overline{S}$ for the collection containing all vectorizations of quantifiers in S. We now take a closer look at the expressive power of the collections $\overline{Q_1}$ and $\overline{Q_2}$.

Vectorizations of Unary Quantifiers

We noted above (see Proposition 2) that every unary quantifier is definable in $\mathsf{C}^{\omega}_{\infty\omega}$, i.e. using only counting quantifiers. It follows immediately that vectorizations of quantifiers in Q_1 are definable using vectorizations of counting quantifiers. But, we can say something stronger. All vectorizations of unary quantifiers are definable in $\mathsf{C}^{\omega}_{\infty\omega}$. That is, we can state the following strengthening of Proposition 2

Theorem 3 *Every formula of* $\mathsf{L}^{\omega}_{\infty\omega}(\overline{Q_1})$ *is equivalent to one of* $\mathsf{C}^{\omega}_{\infty\omega}$.

To prove Theorem 3, it suffices to show that each quantifier in $\overline{Q_1}$ is itself definable in $\mathsf{C}^{\omega}_{\infty\omega}$. The proof of Proposition 2 is easily adapted to show that the dth vectorization of any quantifier in Q_1 is definable using just the dth vectorizations of the counting quantifiers. Let $C_{d,t}$ denote the quantifier which is the dth vectorization of the counting quantifier $\exists^{\geq t}$. In other words, it is the collection of structures $(A, U, \sim, S)$, where U and S are d-ary relations and $\sim$ is a $2d$-ary relation, with the properties:

1. $\sim$ is an equivalence relation on the d-tuples in U;
2. S is a union of $\sim$-equivalence classes; and
3. S contains at least t distinct equivalence classes.

Thus, we just need to show that each of these three conditions is itself expressible in $\mathsf{C}^{\omega}_{\infty\omega}$, with a number of variables that is independent of t.

It is clear how to express the first two in first-order logic in the standard way. To show that the third can be expressed in $\mathsf{C}^{\omega}_{\infty\omega}$, we establish two facts. First that there is a formula of $\mathsf{C}^{2}_{\infty\omega}$ that, interpreted in a structure with an equivalence relation E, expresses that there are at least t distinct E-equivalence classes. Secondly that

counting d-tuples can be done by just counting single elements: there is a formula of $\mathsf{C}^k_{\infty\omega}$ that expresses that there are at least t distinct d-tuples $\mathbf{x}$ for which $\phi(\mathbf{x})$ holds, where k may depend on d but is independent of t. Both of these are easy exercises in coding in $\mathsf{C}^\omega_{\infty\omega}$ for which we give a sketch for the sake of completeness.

To write down the formulas, we use the abbreviation $\exists^{=m} x\phi$ as usual for $\exists^{\geq m} x\phi \wedge \neg\exists^{\geq m+1} x\phi$. For any positive integer n, let $\mathcal{P}_{n,t}$ denote the collection of *partitions* of n into t or more parts, that is the collection of multisets P of positive integers with at least t elements such that $\sum P = n$. For P in $\mathcal{P}_{n,t}$ and a positive integer j, we write $m_P(j)$ for the multiplicity (which may be 0) with which j appears in P. Then, a formula of $\mathsf{C}^2_{\infty\omega}$ that says that there are exactly t E-equivalence classes is the following:

$$\bigvee_{n\in\omega}\left(\exists^{=n} x(x=x) \wedge \bigvee_{P\in\mathcal{P}_{n,t}} \bigwedge_{1\leq j\leq n} \exists^{=j\cdot m_P(j)} x\exists^{=j} yE(x,y)\right).$$

By replacing the variables x and y by d-tuples of variables, we obtain a formula $\eta_{d,t}$, using d-ary counting quantifiers, and no more than $2d$ variables, that expresses the existence of at least t E-equivalence classes, when E is an equivalence relation on d-tuples.

For the second claim, suppose we have a formula, $\exists^{\geq t}\mathbf{x}\phi$ where $\mathbf{x} = (x_1, \ldots, x_d)$ using a d-ary counting quantifier. We aim to show that this can be expressed using only counting quantifiers of arity 1, without increasing the number of variables. We proceed by induction on d. The base case $d = 1$ is trivial. Inductively, let $\mathcal{P}_t$ denote all partitions of t. Then the formula $\exists^{\geq t}\mathbf{x}\phi$ where $\mathbf{x} = (x_1, \ldots, x_{d+1})$ is equivalent to

$$\bigvee_{P\in\mathcal{P}_t} \bigwedge_{1\leq j\leq t} \exists^{\geq j} x_{d+1}\exists^{\geq m_P(j)}(x_1, \ldots, x_d)\phi,$$

where, by induction, the subformula $\exists^{\geq m_P(j)}(x_1, \ldots, x_d)\phi$ can be expressed using only unary counting quantifiers.

Taken together, this establishes that the dth vectorization of each counting quantifier is indeed definable in $\mathsf{C}^\omega_{\infty\omega}$, and thus Theorem 3.

Vectorizations of Binary Quantifiers

In contrast to the situation with unary quantifiers, taking vectorizations of quantifiers of arity 2 adds considerable expressive power to the logic. Indeed, it makes the expressive power as rich as can possibly be.

Theorem 4 *The expressive power of $FO(\overline{Q_2})$ is complete on the class of finite structures.*

The theorem follows from the fact that for any vocabulary σ, there is a first-order definable *bi-interpretation* to the vocabulary τ_2 with one binary relation [22, Theorem 5.5.1]. That is to say, there are interpretations $\mathcal{I}$ of τ_2 in σ and $\mathcal{J}$ of σ in τ_2

such that for any σ-structure $\mathbb{A}$, the structure $\mathcal{J}(\mathcal{I}(\mathbb{A}))$ is isomorphic to $\mathbb{A}$. Note that the dimension d of the interpretation $\mathcal{I}$ depends on σ. It then follows that for any class $\mathcal{K}$ of σ-structures there is a class $\mathcal{G}$ of τ_2 structures, namely exactly those that are in the image of $\mathcal{K}$ under $\mathcal{I}$, such that for any σ-structure $\mathbb{A}$, we have $\mathcal{I}(\mathbb{A}) \in \mathcal{G}$ if, and only if, $\mathbb{A} \in \mathcal{K}$. In other words, $\mathcal{K}$ is defined by the $\mathrm{FO}(\overline{Q_2})$ sentence $\mathcal{G}\mathbf{x}\mathcal{I}$.

Thus, in contrast to Theorem 2 which shows that the arity hierarchy for quantifiers is infinite, we have that the arity hierarchy for *vectorized* quantifiers has just two levels. At the first level we have the well-studied logic $\mathsf{C}^{\omega}_{\infty\omega}$, whose limitations are well-known, and at the second level we have $\mathsf{L}^{\omega}_{\infty\omega}(\overline{Q_2})$ which can express everything. To get interesting classes of vectorized quantifiers beyond the unary case, we need to consider proper subclasses of $\overline{Q_2}$. One way of getting interesting classes is to strengthen the requirement that classes of quantifiers are invariant under isomorphism. We consider one such strengthening that gives us the *linear algebraic quantifiers*.

4 Linear Algebraic Quantifiers

We consider quantifiers in Q_2. These are classes of structures in a vocabulary with only unary and binary relations. Without loss of generality, we can assume that all relations are binary, as unary relations can always be coded as binary relations. So, let us fix for the rest of this section a vocabulary $\sigma = (R_1, \ldots, R_m)$ where each R_i has arity 2.

The classic way to define isomorphism of σ-structures is to say that two σ-structures $\mathbb{A} = (A, R_1^{\mathbb{A}}, \ldots, R_m^{\mathbb{A}})$ and $\mathbb{B} = (B, R_1^{\mathbb{B}}, \ldots, R_m^{\mathbb{B}})$ are isomorphic if there is a bijection $\beta : A \to B$ with $\beta(R_i^{\mathbb{A}}) = R_i^{\mathbb{B}}$ for all i. To motivate the strengthenings of isomorphism that we want to consider, it is useful to re-phrase this in a different way.

Any binary relation R over the set $\{1, \ldots, n\}$ can be seen as a $n \times n$ matrix with entries in $\{0, 1\}$. Namely, the entry (i, j) in the matrix is 1 if, and only if, $(i, j) \in R$. If $\mathbb{A}$ and $\mathbb{B}$ both have n elements, fix bijections between A and $\{1, \ldots, n\}$ and B and $\{1, \ldots, n\}$ respectively and we can regard each relation $R_i^{\mathbb{A}}$ and $R_i^{\mathbb{B}}$ as an $n \times n$ $\{0, 1\}$-matrix in this way. We do not distinguish notationally between the relation and the matrix. Then, an isomorphism between $\mathbb{A}$ and $\mathbb{B}$ is an $n \times n$ *permutation matrix* P such that for each $i \in [m]$: $PR_i^{\mathbb{A}}P^{-1} = R_i^{\mathbb{B}}$. Recall that a permutation matrix is a square matrix with entries in $\{0, 1\}$ such that each row and each column has exactly one non-zero entry. By relaxing the requirement that P is a permutation matrix, we get equivalence relations on structures that are relaxations of isomorphism. One we are particularly interested in is linear algebraic equivalence, as defined in [14].

4.1 Linear Algebraic Equivalence

Definition 5 Let $\mathbb{F}$ be a field. We say that two σ-structures $\mathbb{A} = (A, R_1^{\mathbb{A}}, \ldots, R_m^{\mathbb{A}})$ and $\mathbb{B} = (B, R_1^{\mathbb{B}}, \ldots, R_m^{\mathbb{B}})$ with n elements each are $\mathbb{F}$-*linear algebraically equivalent* if there is an invertible linear map $I \in \mathrm{GL}_n(\mathbb{F})$ such that, for each $i \in [m]$

$$I R_i^{\mathbb{A}} I^{-1} = R_i^{\mathbb{B}}.$$

Since a permutation matrix is necessarily invertible, two isomorphic structures are always $\mathbb{F}$-linear algebraically equivalent for any $\mathbb{F}$, but the converse may fail. In general, $\mathbb{F}$-linear algebraic equivalence is a relaxation of isomorphism. We now establish some useful facts about these equivalence relations. We write Ch for the set $\{0\} \cup$ Primes, where Primes denotes the set of prime integers, and for any $p \in$ Ch, we write $\mathbb{F}_p$ for the *prime field* of characteristic p.

Theorem 5 *For any field $\mathbb{F}$ of characteristic p, two structures $\mathbb{A} = (A, R_1^{\mathbb{A}}, \ldots, R_m^{\mathbb{A}})$ and $\mathbb{B} = (B, R_1^{\mathbb{B}}, \ldots, R_m^{\mathbb{B}})$ are $\mathbb{F}$-linear algebraically equivalent if, and only if, they are $\mathbb{F}_p$-linear algebraically equivalent.*

Proof In one direction since $\mathbb{F}_p$ is a subfield of $\mathbb{F}$, if $I \in \mathrm{GL}_n(\mathbb{F}_p)$ is such that $I R_i^{\mathbb{A}} I^{-1} = R_i^{\mathbb{B}}$ for all i, then I also defines an invertible linear map from $\mathrm{GL}_n(\mathbb{F})$ to itself that is a witness to the fact that $\mathbb{A}$ and $\mathbb{B}$ are $\mathbb{F}$-linear algebraically equivalent.

In the other direction, suppose $I \in \mathrm{GL}_n(\mathbb{F})$ witnesses that $\mathbb{A}$ and $\mathbb{B}$ are $\mathbb{F}$-linear algebraically equivalent. Then, since $R_1^{\mathbb{A}}, \ldots, R_m^{\mathbb{A}}$ and $R_1^{\mathbb{B}}, \ldots, R_m^{\mathbb{B}}$ are $\{0, 1\}$-matrices, they are contained in $\mathbb{F}_p^{n \times n}$. Let $S_{\mathbb{A}}$ and $S_{\mathbb{B}}$ be the subspaces of $\mathbb{F}_p^{n \times n}$ generated by $R_1^{\mathbb{A}}, \ldots, R_m^{\mathbb{A}}$ and $R_1^{\mathbb{B}}, \ldots, R_m^{\mathbb{B}}$ respectively. Then, the fact that $I R_i^{\mathbb{A}} I^{-1} = R_i^{\mathbb{B}}$ for all i means that the restriction I_1 of the conjugation action of I to $S_{\mathbb{A}}$ is an isomorphism from $S_{\mathbb{A}}$ to. $S_{\mathbb{B}}$. This can then be extended to an invertible linear map I_2 from $\mathbb{F}_p^{n \times n}$ to itself such that $I_2 R_i^{\mathbb{A}} I_2^{-1} = R_i^{\mathbb{B}}$ □

Theorem 5 establishes that the relation of $\mathbb{F}$-linear algebraic equivalence of structures in a binary vocabulary is completely determined by the characteristic of the field $\mathbb{F}$. With this in hand, we can write $\mathbb{A} \cong_p \mathbb{B}$ to denote that the two structures $\mathbb{A}$ and $\mathbb{B}$ are $\mathbb{F}_p$-linear algebraically equivalent, and hence $\mathbb{F}$-linear algebraically equivalent for *all* fields of characteristic p.

As we have noted, if $\mathbb{A}$ and $\mathbb{B}$ are isomorphic σ-structures, then $\mathbb{A} \cong_p \mathbb{B}$ for any $p \in$ Ch. On the other hand, for each p, we can find non-isomorphic structures which are $\cong_p$-equivalent. This is an immediate consequence of results we state in the next section. For now, let us note that the equivalence relation $\cong_p$ is decidable in polynomial time.

Theorem 6 (Chistov et al. [6]) *For each $p \in$ Ch, the equivalence relation $\cong_p$ is decidable in polynomial time*

Theorem 6 is not stated in this form in [6] but it is an immediate consequence of the fact proved there that the simultaneous similarity problem for tuples of matrices

is decidable in polynomial time. This result is itself a special case of the algorithm for the module isomorphism problem. It is worth seeing the connection between the equivalence relation $\cong_p$ and module isomorphism.

Recall that a *left module* over a ring $\mathcal{R}$ is an Abelian group $(M, +)$ with a left multiplication of $\mathcal{R}$ on M satisfying the natural coherence axioms. Associate with any structure $\mathbb{A} = (A, R_1^{\mathbb{A}}, \ldots, R_m^{\mathbb{A}})$ and any $p \in \mathrm{Ch}$ a left module $M_{\mathbb{A}}$ over the polynomial ring $\mathbb{F}_p[x_1, \ldots, x_m]$. The elements of $M_{\mathbb{A}}$ are the vectors in $\mathbb{F}_p^n$ where n is the number of elements of $\mathbb{A}$, and the group operation is standard vector addition. The left multiplication of a polynomial $p(x_1, \ldots, x_m)$ on a vector $v \in \mathbb{F}_p^n$ is defined as taking it to $p(R_1, \ldots, R_m)v$. It is then easily verified that for any pair $\mathbb{A}$, $\mathbb{B}$ of σ-structures, $\mathbb{A} \cong_p \mathbb{B}$ if, and only if, the modules $M_{\mathbb{A}}$ and $M_{\mathbb{B}}$ are isomorphic.

4.2 Linear Algebraic Logics

With the definition of the equivalence relations $\cong_p$ in place, we are ready to define the generalized quantifiers we are interested in. Write L_p for the collection of all quantifiers over vocabularies of binary relation symbols which are invariant under the relation $\cong_p$. We call L_p the collection of *linear algebraic quantifiers of characteristic* p. We call the collection $\bigcup_{p\in\mathrm{Ch}} L_p$ the collection of linear algebraic quantifiers. More generally, for any set $\Omega \subseteq \mathrm{Ch}$, we define the linear algebraic quantifiers with characteristic in Ω and denote it $L_\Omega = \bigcup_{p\in\Omega} L_p$. We are mostly interested in vectorizations of these quantifiers.

The motivating example of linear algebraic quantifiers are the *rank* quantifiers. For each $p \in \mathrm{Ch}$ and each positive integer t, let rk_p^t be the quantifier consisting of the class of structures (A, M) where $M \subseteq A \times A$ is a binary relation and the rank of M seen as a $\{0, 1\}$-matrix in $\mathbb{F}_p$ is at least t. Note that the rank of M is well-defined as it does not depend on the particular choice of a bijection between A and $[n]$ to get an $n \times n$ matrix. Moreover, since the rank of a matrix is preserved by invertible linear maps, rk_p^t is clearly a quantifier that is invariant under $\cong_p$. We write Rk_p to denote the collection of quantifiers $\{\mathrm{rk}_p^t \mid t \in \omega\}$ and Rk for $\bigcup_{p\in\mathrm{Ch}} \mathrm{Rk}_p$.

The logic $\mathsf{L}^{\omega}_{\infty\omega}(\overline{\mathrm{Rk}})$, known as *infinitary rank logic*, was introduced in [13] where it was denoted $\mathrm{R}^{\omega}_{\infty\omega}$. Its expressive power subsumes the extension of FPR, fixed-point logic with *rank operators*. This is true both of the original version of FPR as introduced in [13] and the more expressive version with variable rank operators introduced by Grädel and Pakusa[18].

More generally, we can consider logics with *all* linear algebraic quantifiers, and we introduce special notation for these. For any $\Omega \subseteq \mathrm{Ch}$, we write $\mathsf{LA}^k(\Omega)$ and $\mathsf{LA}^\omega(\Omega)$ to denote the logic $\mathsf{L}^k_{\infty\omega}(\overline{L_\Omega})$ and $\mathsf{L}^\omega_{\infty\omega}(\overline{L_\Omega})$ respectively. Finally, we write LA^k and LA^ω for the logics $\mathsf{LA}^k(\mathrm{Ch})$ and $\mathsf{LA}^\omega(\mathrm{Ch})$.

Note that two structures $\mathbb{A}$ and $\mathbb{B}$ are indistinguishable in the logic $\mathsf{LA}^k(\Omega)$ if, and only if, $\mathbb{A} \equiv^k_{L_\Omega} \mathbb{B}$. The relation $\equiv^k_{L_\Omega}$ is, for fixed values of k and finite Ω, decidable in polynomial time. Indeed, this is a consequence of Theorem 6 as the

relation $\equiv^k_{L_\Omega}$ can be determined through repeated (at most $n^{O(k)}$) applications of the algorithm evaluating the relation $\cong_p$ for $p \in \Omega$. This is similar to the Weisfeiler-Leman iteration where we obtain the partition of k-tuples in a structure $\mathbb{A}$ into equivalence classes under the relation $\equiv^k_C$. Just like the Weisfeiler-Leman equivalences are characterized by Hella's bijection game of arity 1, the relation $\equiv^k_{L_\Omega}$ can be characterized by a pebble game. This is the *invertible map game* and was introduced in [11].

Invertible Map Game The game is played between *Spoiler* and *Duplicator* on a pair of structures $\mathbb{A}$ and $\mathbb{B}$, with universes A and B respectively. We have, as usual, k pairs of pebbles $(s_i, t_i)_{i\in[k]}$. The pebbles s_i may be placed on elements of A and the pebbles t_i on elements of B. Thus, the position of the game at any point has at most k pebbles in A and at most k pebbles in B and this defines a partial map $f : A \rightharpoonup B$ taking the element pebbled by s_i to the element pebbled by t_i (we are assuming that distinct pebbles are always placed on distinct elements, which we can do without loss of generality). Play proceeds as follows.

1. *Spoiler* announces a set $j_1, \ldots, j_{2m} \in [k]$ of indices of pebbles that are going to move in this round;
2. *Spoiler* chooses a characteristic $p \in \Omega$;
3. *Duplicator* gives a partition of A^{2m} into parts $P_1, \ldots, P_t$ and a partition of B^{2m} into parts $Q_1, \ldots, Q_t$ such that

$$(A^m, P_1, \ldots, P_t) \cong_p (B^m, Q_1, \ldots, Q_t);$$

4. *Spoiler* chooses some $i \in \{1, \ldots, t\}$ and an $\mathbf{a} \in P_i$ and $\mathbf{b} \in Q_i$; and
5. the pebbles $s_{j_1}, \ldots, s_{j_{2m}}$ are placed on $\mathbf{a}$ and $t_{j_1}, \ldots, t_{j_{2m}}$ on $\mathbf{b}$.

If, at the end of any move, the partial map $f : A \rightharpoonup B$ given by the pebbled positions is not a partial isomorphism from $\mathbb{A}$ to $\mathbb{B}$, then *Spoiler* has won the game, otherwise it can continue. *Duplicator* has a winning strategy for playing forever if, and only if, $\mathbb{A} \equiv^k_{L_\Omega} \mathbb{B}$.

The game is rather more complicated than Hella's bijection game and can be correspondingly unwieldy to deploy. Nonetheless, Lichter's remarkable proof separating FPR from P does use the game. We review this result and others in the next section.

5 Expressiveness Results

We now review what is known about the expressive power of the linear algebraic logics introduced above. I have not included proofs of the results, which can be found in the papers referred to, but I give in each case a brief idea of what is involved in the construction, particularly where it involves a separating example. In all the

results below, when I write $L_1 \leq L_2$ for a pair of logics L_1 and L_2 it is to be read as meaning that all classes definable in L_1 are also expressible in L_2.

The first result of note is that linear algebraic quantifiers of characteristic zero can be simulated by counting quantifiers.

Theorem 7 (Holm [23] and Dawar and Vagnozzi [12])

$$\mathsf{LA}^{\omega}(\{0\}) \leq \mathsf{C}^{\omega}_{\infty\omega}.$$

Theorem 7 essentially follows from the fact that for any vocabulary σ of binary relations, if $\mathbb{A}$ and $\mathbb{B}$ are σ-structures then $\mathbb{A} \equiv^3_C \mathbb{B}$ implies $\mathbb{A} \cong_0 \mathbb{B}$. This is established in [12] by using the fact that $\equiv^3_C$ can be characterized in terms of *coherent algebras* (see [5]) and isomorphism of such algebras reduces to module isomorphism over $\mathbb{Q}$.

An earlier result, which was foundational in the study of linear algebraic logics shows that in characteristic two, the situation is starkly different from characteristic zero.

Theorem 8 (Dawar et al. [13])

$$\mathsf{L}^3_{\omega\omega}(L_2) \not\leq \mathsf{C}^{\omega}_{\infty\omega}.$$

In other words, with a linear algebraic quantifier of characteristic two, even without infinitary connectives and without vectorization, we can express a property that is not definable in $\mathsf{C}^{\omega}_{\infty\omega}$. The property that is definable is the classic construction of [4] who construct for each $k \in \omega$ a pair of graphs G_k and H_k such that $G_k \equiv^k_C H_k$ but $G_k \ncong H_k$. What is shown in [13] is that there is a sentence φ of $\mathsf{L}^3_{\omega\omega}(L_2)$ (indeed, of $\mathsf{L}^3_{\omega\omega}(\mathrm{Rk}_2)$) such that $G_k \models \varphi$ and $H_k \not\models \varphi$ for all k.

The graph construction of [4] can be seen as encoding the problem of deciding the solvability of systems of linear equations over the field $\mathbb{F}_2$ (see [1]) and thus, the logic $\mathsf{L}^3_{\omega\omega}(L_2)$ is perfectly suited to express it. This idea can be generalized to separating the expressive power of linear algebraic logics of distinct characteristics. The first such result is the following.

Theorem 9 (Dawar and Holm [11]) *For distinct prime numbers p and q,*

$$\mathsf{L}_{\omega\omega}(L_p) \not\leq \mathsf{L}^{\omega}_{\infty\omega}(L_q).$$

Theorem 9 is proved by constructing a class of structures $\mathrm{CFI}(p)$, generalizing the construction of [4], for each prime p which codes solvable systems of linear equations over $\mathbb{F}_p$. The problem is definable in $\mathsf{L}^3_{\omega\omega}(L_p)$ by construction. To show that it is not definable in $\mathsf{L}^{\omega}_{\infty\omega}(L_q)$, we use a simplified version of the invertible map game, one where the parameter m is always 1. In other words, *Spoiler* always moves two pebbles at each move.

The reason for considering the simplified game is that we are not considering *vectorized quantifiers*. Indeed, at the time of [11] it seemed very difficult to

provide an argument that used the full power of the invertible map game. The first inexpressibility result for a linear algebraic logic using vectorized quantifiers was the following result about rank logics.

Theorem 10 (Grädel and Pakusa [18]) *For any prime number p, if $\Omega = Ch \setminus \{p\}$, then*

$$\mathsf{L}_{\omega\omega}(Rk_p) \not\leq \mathsf{L}^{\omega}_{\infty\omega}(\bigcup_{q\in\Omega} \overline{Rk_q}).$$

The separating example is once again a version of the class of structures CFI(p). It is shown that these can be constructed in such a way that they are *homogeneous*, meaning that the orbits of the automorphism groups of the structures can be defined in $\mathsf{C}^k_{\infty\omega}$ for a fixed value of k. This is then used to obtain a quantifier elimination result, showing that on these structures the rank quantifiers, even when vectorized, can be defined in $\mathsf{C}^{\omega}_{\infty\omega}$. The theorem then follows from Theorem 8.

The next step improves Theorem 10 by replacing the right-hand side from just the logic with rank quantifiers to a logic with all linear-algebraic quantifiers with characteristics in Ω. I state it below with $\mathsf{LA}^{\omega}(\{p\})$ on the left-hand side, which is how it is formulated in the original source, but we could strengthen the statement by having $\mathsf{L}_{\omega\omega}(\mathrm{Rk}_p)$ on the left-hand side, since the separating example is exactly the one in Theorem 10 above.

Theorem 11 (Dawar et al. [14]) *For any prime number p, if $\Omega = Ch \setminus \{p\}$, then*

$$\mathsf{LA}^{\omega}(\{p\}) \not\leq \mathsf{LA}^{\omega}(\Omega).$$

This also implies that as long as Ω does not contain *all* the prime numbers, then the expressive power of $\mathsf{LA}^{\omega}(\Omega)$ is not complete. By virtue of Theorem 1, there is no k such that $\equiv^k_{L_\Omega}$ is discrete.

Theorem 11 requires substantial new machinery to prove. The essential idea is similar to the proof of Theorem 10. The separating example is, once again, the version of the class CFI(p) used in the proof of that theorem. While the proof of Theorem 10 showed that, on CFI(p), computing the rank of a definable matrix in fields of characteristic co-prime with p could be reduced to counting, we now need to do the same for *all* linear-algebraic quantifiers. Since the structures in CFI(p) are homogeneous, i.e. the orbits of l-tuples under automorphisms are $\mathsf{C}^k_{\infty\omega}$ definable, and moreover the automorphism group is an Abelian p-group, we are able to show that these orbits induce a coherent algebra (over any field of characteristic q co-prime with p) that is semi-simple. This is an application of Maschke's theorem. Semi-simple algebras are completely determined by counting the types of the simple algebras that divide them, and one can show that all of this can be carried out in $\mathsf{C}^k_{\infty\omega}$ for suitable k. This yields a quantifier-elimination process whereby all quantifiers in $\overline{L_\Omega}$ can be eliminated in favour of counting quantifiers.

The proof of Theorem 11 introduces significant new algebraic tools, in the form of modular representation theory, to the toolkit of finite model theory. The next step

is Lichter's dramatic result that rank quantifiers alone are not sufficient to express all polynomial-time decidable properties.

Theorem 12 (Lichter [29]) *There is a polynomial-time decidable property that is not definable in* $\mathsf{L}^{\omega}_{\infty\omega}(\overline{Rk})$.

The particular polynomial-time decidable property that is constructed in the proof of Theorem 12 is again based on the CFI construction. Again, it can be seen as encoding the solvability of systems of linear equations. Indeed, we can generalize the CFI construction to code not just equations over finite fields but more general finite rings or indeed finite Abelian groups (as in [1]). Lichter in his construction codes systems of equations over rings of the form $\mathbb{Z}/(2^m\mathbb{Z})$ for positive integers m, with m growing with the size of the structure. The proof that, in the resulting class of structures which we denote CFI($\mathbb{Z}m$), $\mathsf{L}^{\omega}_{\infty\omega}(\overline{\mathrm{Rk}})$ cannot distinguish the solvable from the unsolvable case rests on two pillars. The first is that the structures in CFI($\mathbb{Z}m$) are homogeneous, as in the proof of Theorem 10, and so on these structures we can eliminate all rank quantifiers of characteristic other than 2 in favour of counting quantifiers. Finally, to show that the class is not definable using rank quantifiers of characteristic 2, Lichter deploys the invertible map game with $\Omega = \{2\}$. The proof is a *tour de force* in the use of the game and Lichter develops significant tools for describing the *Duplicator* winning strategy.

The final result on the limitations of linear-algebraic logics is the following.

Theorem 13 (Dawar et al. [15]) *There is a polynomial-time decidable property that is not definable in* LA^{ω}.

This stengthens Theorem 12 and relies on the construction from its proof. Indeed, the proof is based on the observation that just as Maschke's theorem allowed us to strengthen Theorems 10 to 11, so the same ideas can be used also to eliminate all linear-algebraic quantifiers of characteristic other than 2 over the class CFI($\mathbb{Z}m$). Then, since the invertible-map game strategy developed by Lichter in the proof of Theorem 12 shows undefinability in $\mathsf{LA}^{\omega}(\{2\})$ and not just the rank logic, Theorem 13 follows. In particular, it shows that the expressive power of LA^{ω} is not complete and there is no k for which the equivalence relation $\equiv^{k}_{L_{\mathrm{Ch}}}$ is discrete.

6 Concluding Remarks

Linear algebraic quantifiers are a *natural* class of Lindström quantifiers to consider in the context of finite model theory. The known limitations of classes of quantifiers of bounded arity and the closure of complexity classes under logical reductions strongly motivate the study of *vectorized* quantifiers in descriptive complexity theory. However, vectorizations of all quantifiers of arity 2 lead to something far too powerful, able to express absolutely everything. This leads us to consider naturally restricted classes of quantifiers of unbounded arity generated by restricted

quantifiers of arity 2. The restriction obtained by strengthening the requirement of isomorphism closure to linear-algebraic closure turns out to be particularly fruitful. It gives us logics that properly extend the expressive power of counting quantifiers but still have nice algorithmic properties, such as polynomial-time decidable equivalence relations. Moreover, we have developed sophisticated algebraic machinery for analysing the expressive power of these logics and showing that it is not complete.

There are other ways of generating interesting classes of quantifiers of unbounded arity, other than the restriction to linear-algebraic equivalence. For an example motivated by the universal-algebraic study of constraint satisfaction problems (CSP), (see [10, 21]). The study of CSP also motivates an interesting open question, with which I will close. The so-called fixed-domain constraint satisfaction problems have been completely classified into those that are polynomial-time decidable and those that are NP-complete [3, 39]. Many of the examples that motivate the study of linear-algebraic quantifiers and serve as separating examples in the proof are indeed fixed-domain CSP. In particular the problem of solving systems of linear equations over a fixed finite field or indeed a fixed finite ring is a polynomial-time decidable CSP. However, the problem of solving systems of equations over the ring $\mathbb{Z}/(2^m\mathbb{Z})$, with m varying, which shows the limitations of the logic LA^ω in Theorem 13, is tractable but not a finite-domain CSP. This leaves open the intriguing question: are all tractable finite-domain CSP definable in linear algebraic logic?

References

1. Atserias, A., Bulatov, A., Dawar, A.: Affine systems of equations and counting infinitary logic. Theor. Comput. Sci. **410**(18), 1666–1683 (2009)
2. Barwise, J.: On Moschovakis closure ordinals. J. Symbol. Logic **42**, 292–296 (1977)
3. Bulatov, A.A.: A dichotomy theorem for nonuniform CSPs. In: 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS, pp 319–330 (2017). https://doi.org/10.1109/FOCS.2017.37
4. Cai, J.Y., Fürer, M., Immerman, N.: An optimal lower bound on the number of variables for graph identification. Combinatorica **12**(4), 389–410 (1992)
5. Chen, G., Ponomarenko, I.: Lectures on Coherent Configurations. Central China Normal Univ Press, Wuhan (2019)
6. Chistov, A., Ivanyos, G., Karpinski, M.: Polynomial time algorithms for modules over finite dimensional algebras. In: Proceedings of 1997 International Symposium on Symbolic and Algebraic Computation, pp. 68–74. ACM, New York (1997)
7. Dawar, A.: Generalized quantifiers and logical reducibilities. J. Logic Comput. **5**(2), 213–226 (1995)
8. Dawar, A.: Finite models and finitely many variables. In: Niwinski, D., Maron, R. (eds.), Logic, Algebra and Computer Science, Banach Center Publications, vol. 46, pp 93–117. Polish Academy of Sciences, Warsaw (1999)
9. Dawar, A.: The nature and power of fixed-point logic with counting. ACM SIGLOG News **2**(1), 8–21 (2015)
10. Dawar, A., Hella, L.: Quantifiers closed under partial polymorphisms. In: 32nd EACSL Annual Conference on Computer Science Logic, CSL, pp. 23:1–23:19 (2024). https://doi.org/10.4230/LIPICS.CSL.2024.23

11. Dawar, A., Holm, B.:Pebble games with algebraic rules. Fundam. Inform. **150**, 281–316 (2017)
12. Dawar, A., Vagnozzi, D.: Generalizations of k-Weisfeiler-Leman stabilization. Moscow J. Number Theory Combin. **9**, 229–252 (2020)
13. Dawar, A., Grohe, M., Holm, B., Laubner, B.: Logics with rank operators. In: Proceedings of 24th IEEE Symposium on Logic in Computer Science, pp. 113–122 (2009)
14. Dawar, A., Grädel, E., Pakusa, W.: Approximations of isomorphism and logics with linear-algebraic operators. In: 46th International Colloquium on Automata, Languages, and Programming, ICALP'19 (2019). https://doi.org/10.4230/LIPIcs.ICALP.2019.112
15. Dawar, A., Grädel, E., Lichter, M.: Limitations of the invertible-map equivalences. J. Log. Comput. **33**(5), 961–969 (2023). https://doi.org/10.1093/LOGCOM/EXAC058
16. Ebbinghaus, H.D. (1985). Extended logics: the general framework. In: Barwise, J., Feferman, S. (eds), Model-Theoretic Logics, pp. 25–76. Springer, Berlin (1985)
17. Ebbinghaus, H.D., Flum, J.: Finite Model Theory, 2nd edn. Springer, Berlin (1999)
18. Grädel, E., Pakusa, W.: Rank logic is dead, long live rank logic! J. Symbol. Logic **84**(1), 54–87 (2019). https://doi.org/10.1017/jsl.2018.33
19. Gurevich, Y.: Logic and the challenge of computer science. In: Börger, E. (ed) Current Trends in Theoretical Computer Science, pp 1–57. Computer Science Press. Rockville, MD, USA (1988)
20. Hella, L.: Logical hierarchies in PTIME. Inf. Comput. **129**, 1–19 (1996)
21. Hella, L.: The expressive power of CSP-quantifiers. In: 31st EACSL Annual Conference on Computer Science Logic, CSL, pp 25:1–25:19 (2023). https://doi.org/10.4230/LIPIcs.CSL.2023.25
22. Hodges, W.: Model Theory. Cambridge University Press, Cambridge (1993)
23. Holm, B.: Descriptive complexity of linear algebra. Ph.D. Thesis, University of Cambridge (2010)
24. Immerman, N.: Upper and lower bounds for first-order expressibility. J. Comput. Syst. Sci. **25**, 76–98 (1982)
25. Immerman, N.: Relational queries computable in polynomial time. Inf. Control **68**, 86–104 (1986)
26. Kiefer, S.: The Weisfeiler-Leman algorithm: an exploration of its power. ACM SIGLOG News **7**(3), 5–27 (2020). https://doi.org/10.1145/3436980.3436982
27. Kolaitis, P.G., Väänänen, J.A.: Generalized quantifiers and pebble games on finite structures. Ann. Pure Appl. Logic **74**(1), 23–75 (1995)
28. Kolaitis, P.G., Vardi, M.Y.: Infinitary logics and 0-1 laws. Inf. Comput. **98**(2), 258–294 (1992)
29. Lichter, M.: Separating rank logic from polynomial time. J. ACM **70**(2), 14:1–14:53 (2023). https://doi.org/10.1145/3572918
30. Makowsky, J.: Abstract embedding relations. In: Barwise, J., Feferman, S. (eds.), Model-Theoretic Logics, pp. 747–791. Springer, Berlin (1985)
31. Makowsky, J.: Compactness, embeddings and definability. In: Barwise, J, Feferman, S. (eds.), Model-Theoretic Logics, pp. 645–716. Springer, Berlin (1985)
32. Makowsky, J., Mundici, D.: Abstract equivalence relations. In: Barwise, J., Feferman, S. (eds.), Model-Theoretic Logics, pp. 717–746. Springer, Berlin (1985)
33. Makowsky, J., Pnueli, Y.: Computable quantifiers and logics over finite structures. In: Krynicki, M., Mostowski, M., Szczerba, L. (eds.), Quantifiers: Generalizations, Extensions and Variants of Elementary Logic. Kluwer Academic, Dordrecht (1993)
34. Makowsky, J.A.: Capturing complexity classes with Lindström quantifiers. In: Prívara, I., Rovan, B., Ruzicka, P. (eds.), Mathematical Foundations of Computer Science 1994. Lecture Notes in Computer Science, vol. 841, pp. 68–71. Springer, Berlin (1994). https://doi.org/10.1007/3-540-58338-6_59
35. Makowsky, J.A., Pnueli, Y.B.: Logics capturing relativized complexity classes uniformly. In: Leivant, D. (ed.), Logical and Computational Complexity. Springer Lecture Notes in Computer Science, vol. 960, pp. 463–479. Springer, Berlin (1994). https://doi.org/10.1007/3-540-60178-3_98

36. Otto, M.: Bounded Variable Logics and Counting — A Study in Finite Models. Lecture Notes in Logic, vol. 9. Springer, Berlin (1997)
37. Poizat, B.: Deux ou trois choses que je sais de L_n. J. Symbol. Logic **47**(3), 641–658 (1982)
38. Vardi, M.Y.: The complexity of relational query languages. In: Proceedings of the 14th ACM Symposium on the Theory of Computing, pp. 137–146 (1982)
39. Zhuk, D.: A proof of the CSP dichotomy conjecture. J. ACM **67**, 30:1–30:78 (2020). https://doi.org/10.1145/3402029

A Coarse Tutte Polynomial for Hypermaps

Joanna A. Ellis-Monaghan, Iain Moffatt, and Steven Noble

Abstract We give an analogue of the Tutte polynomial for hypermaps. This polynomial can be defined as either a sum over subhypermaps, or recursively through deletion-contraction reductions where the terminal forms consist of isolated vertices. Our Tutte polynomial extends the classical Tutte polynomial of a graph as well as the Tutte polynomial of an embedded graph (i.e., the ribbon graph polynomial), and it is a specialization of the transition polynomial via a medial map transformation. We give hypermap duality and partial duality identities for our polynomial, as well as some evaluations, and examine relations between our polynomial and other hypermap polynomials.

1 Introduction

In this paper we introduce a Tutte polynomial for hypermaps as a direct generalisation of the Tutte polynomials for abstract graphs and for maps (graphs cellularly embedded on surfaces).

An edge of a graph may be defined as a multiset containing exactly two (not necessarily distinct) vertices. Hypergraphs generalize graphs by allowing hyperedges which are multisets containing any number of vertices. A map may be thought of as a drawing of a graph on a surface (compact 2-manifold) so that the edges do not cross and so that each face is a region of the surface homeomorphic to

J. A. Ellis-Monaghan (✉)
Korteweg-de Vries Instituut voor Wiskunde, Universiteit van Amsterdam, Amsterdam, The Netherlands
e-mail: j.a.ellismonaghan@uva.nl

I. Moffatt
Department of Mathematics, Royal Holloway, University of London, Egham, UK
e-mail: iain.moffatt@rhul.ac.uk

S. Noble
School of Computer Science, University of Leeds, Leeds, UK
e-mail: s.d.noble@leeds.ac.uk

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_17

a disc. Similarly, a hypermap may be thought of as a hypergraph drawn on a surface. (See Sect. 2 for formal definitions.)

Hypergraph and hypermap models are attracting increasing attention as applications of traditional network models seek refinements through higher-order interactions, with [1, 2] providing particularly compelling overviews of the urgency for research in this direction. These higher-order interactions, which consider connections among multiple nodes instead of just pairwise connections, correspond to systems with hyperedges in place of simply edges. Applications, particularly in physics, have already led to extensions of the Potts model to hypergraphs, for example [7, 19], and efforts have begun to extend the Tutte polynomial to hypergraphs, for example in [3, 22].

Our interest here is in constructing a Tutte polynomial for hypermaps. To do this, since hypermaps generalize maps, it is natural to build on the existing theory of map polynomials. There is a rich literature on analogues of the Tutte polynomial for maps (see, e.g., [4, 13, 20, 23, 28] and the survey [8]). By adapting the approach described in [28], we begin by constructing a version of the dichromatic polynomial for hypermaps. Classical connections between the dichromatic and Tutte polynomials then lead us to a Tutte polynomial for hypermaps. The reason we take this approach is that it keeps deletion-contraction properties at its heart, and so the polynomials we construct can be defined by recursive deletion-contraction relations with a base case consisting of hyperedgeless hypermaps.

The hypermap deletion and contraction relations we use here are direct extensions of map operations. Deletion of a hyperedge removes the entire edge from the hypermap, and contraction takes the partial dual of a hyperedge (see [9, 29]) and then removes it. Since there are other, more refined, definitions of hyperedge deletion and contraction, such as those given in [10, 11], we describe our hypermap Tutte polynomial as 'coarse' because we use these 'whole hyperedge' operations. The coarse Tutte polynomial for hypermaps that emerges from this choice captures many desirable properties, and we give several identities and evaluations for it.

We also discuss various interconnections between our hypermap polynomial and other polynomials from the literature. We see that for maps, which are equivalent to ribbon graphs, the coarse Tutte polynomial coincides with the ribbon graph polynomial. We also show that, via a medial map construction, the coarse Tutte polynomial for hypermaps is a specialization of the transition polynomial. Finally, we make a comparison with the recent hypermap Whitney polynomial of Cori and Hetyei [11], which uses a more refined edge deletion, showing that neither of the two polynomials determine the other.

János Makowsky is a keen collector of graph polynomials for his zoo [26]. Hypermaps can provide many more such specimens, which we believe will enrich his zoo. Potentially they provide a fertile setting for the extension of many of his ideas, particularly his Monadic Second Order Logic framework for graph polynomials (see [25] and also [27]). We close by suggesting this extension as a possible direction for future research and discussing complexity issues.

2 Hypermaps

We allow graphs to have loops and parallel edges, and follow the terminology in [5].

An *embedded graph* or *map* consists of a closed surface Σ (not necessarily connected and possibly nonorientable), a set of distinct points on the surface (called *vertices*) and a set of simple paths (called *edges*) whose ends lie on vertices. Furthermore, an edge may only intersect a vertex at its ends, and edges may not intersect except at their ends. The vertices and edges divide the surface into regions, called *faces*, and we insist that each face is homeomorphic to a disc. (Thus we only consider *cellularly embedded graphs* here.) A consequence of this is that each component of the underlying graph must lie in a different connected component of the closed surface. In this paper we use both the terms embedded graph and map. We favour the term map but use embedded graph when it is the preferred term in the sources we are citing. (We also work with ribbon graphs in Sect. 4.1 for this reason.)

We shall define a *hypergraph* to be a bipartite graph $G = (V_v \sqcup V_e, E)$ in which multiple edges are allowed but no vertex of V_e is isolated. The set V_v forms the set of *hypervertices* of the hypergraph, and each vertex in V_e together with its incident edges forms a *hyperedge*. A *hypermap* is an embedded hypergraph.

Figure 1a shows a hypermap with 4 hypervertices (the black vertices) and 3 hyperedges (the white vertices and their incident edges) embedded in the sphere. Here $V_v = \{v_1, v_2, v_3, v_4\}$, $V_e = \{e_1, e_2, e_3\}$ and the faces are labelled $f_1, \ldots, f_4$.

It is convenient to describe a hypermap as a *graph encoded hypermap*, which we abbreviate as *gehm*. A gehm is a properly edge 3-coloured cubic graph in which edges are coloured from the set $\{b, g, r\}$ (standing for {blue, green, red}). In addition, we allow our gehms to have g-coloured edges which do not meet any vertices. We call these *isolates*. They appear in the gehm as isolated g-coloured circles. If each b–r-cycle in a gehm has length exactly four then we say it is a *graph encoded map* or *gem* (and it then describes an embedded graph [6, 24]). Gehms describe hypermaps and vice versa, but we must develop some terminology before giving this correspondence. Figures 1a, b show, respectively, a hypermap and a gehm which we will see later correspond to each other. The labels on the gehm indicate the correspondence and may be ignored for the moment. As figures can be misleading, we emphasise that gehms are (abstract) graphs and are not embedded.

For readability in both colour and black and white printing, we adopt the following convention in our figures. We use blue dashed lines to denote b-edges, green dotted lines to denote g-edges, and red solid lines to denote r-edges.

We set up the following terminology, trusting that the rationale for the names will become clear.

Terminology 1

- A *gehm-edge* is an edge of the cubic graph that forms the gehm.
- A *gehm-vertex* is a vertex of the cubic graph that forms the gehm.
- A b–r-cycle in the gehm is called a *hyperedge*: it represents an edge of the hypermap.

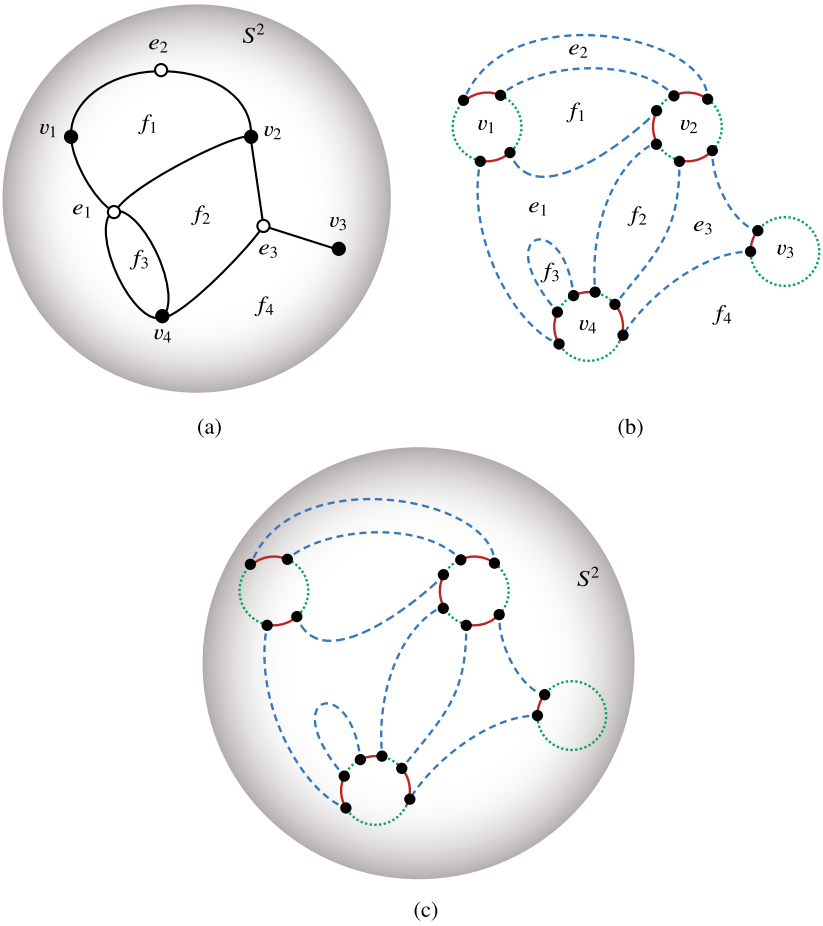


Fig. 1 A hypermap, its gehm and the natural embedding of the gehm. (**a**) A hypermap in the sphere. (**b**) The corresponding gehm. (**c**) The natural embedding of the gehm in the sphere

- A b–g-cycle in the gehm is called a *hyperface*: it represents a face of the hypermap.
- A g–r-cycle in the gehm is called a *hypervertex*: it represents a vertex of the hypermap.
- An *isolate* is both a *hyperface* and a *hypervertex*: it represents a component of a hypermap that consists of an isolated vertex embedded in the sphere.
- $E(\mathbb{H})$, $V(\mathbb{H})$ and $F(\mathbb{H})$ are, respectively, the sets of hyperedges, hypervertices and hyperfaces of the gehm $\mathbb{H}$, and $e(\mathbb{H}) = |E(\mathbb{H})|$, $v(\mathbb{H}) = |V(\mathbb{H})|$ and $f(\mathbb{H}) = |F(\mathbb{H})|$. Note that both $V(\mathbb{H})$ and $F(\mathbb{H})$ include all the isolates.
- $k(\mathbb{H})$ denotes the number of components of the gehm $\mathbb{H}$: each represents a component of the hypermap. Isolates contribute to $k(\mathbb{H})$.

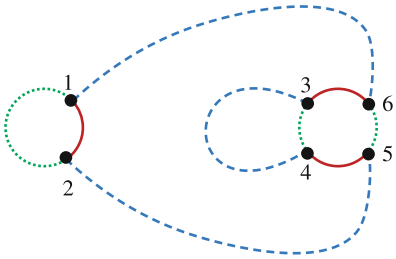


Fig. 2 An example of a gehm

- The *degree* $d(e)$ of a hyperedge e is half the number of edges in its b–r cycle, and this coincides with the degree of the corresponding vertex in the bipartite graph $G = (V_v \sqcup V_e, E)$ whose embedding gives the hypermap. We write $d(\mathbb{H}) = \sum_{e \in E(\mathbb{H})} d(e)$. Note that $d(\mathbb{H})$ is equal to the number of r-edges, the number of b-edges and the number of g-edges after excluding isolates.
- A gehm is *orientable* if it is bipartite, in which case an *orientation* is a choice of gehm-vertex 2-colouring. If a gehm is not bipartite then it is *non-orientable*.
- The *Euler genus*, $\gamma(\mathbb{H})$, is defined through *Euler's formula*

$$\gamma(\mathbb{H}) = 2k(\mathbb{H}) - v(\mathbb{H}) - e(\mathbb{H}) + d(\mathbb{H}) - f(\mathbb{H}).$$

- The *genus* of $\mathbb{H}$ is $\gamma(\mathbb{H})$ when $\mathbb{H}$ is non-orientable, and is $\gamma(\mathbb{H})/2$ when $\mathbb{H}$ is orientable.

For example, consider the gehm in Fig. 2. It has nine gehm-edges and six gehm-vertices which are labelled $1, \ldots, 6$. It has two hypervertices given by the g–r-cycles 1 2 1 and 3 4 5 6 3; one hyperedge of degree 3 given by the b–r-cycle 1 6 3 4 5 2 1; and two hyperfaces given by the b–g-cycles 3 4 3 and 1 6 5 2 1. The gehm is orientable and has genus zero.

The gehm in Fig. 1b has four hypervertices, three hyperedges and four hyperfaces. The degrees of its hyperedges are 2, 3 and 4. It has Euler genus 0 and is orientable. The labels e_i, f_i, and v_i indicate the coloured cycles corresponding to the hyperedges, hyperfaces and hypervertices. We emphasize the gehm is not embedded, so the labels refer to cycles in the gehm not faces in the drawing on the page.

Two gehms are *equivalent* if there is an isomorphism between them that preserves the edge-colouring. If the gehms are oriented the isomorphism should also preserve the gehm-vertex 2-colouring.

Each gehm has a natural embedding in a surface, as follows.

Construction 1 We construct a complex from a gehm. Ignore isolates for the moment. Consider the gehm as a 1-complex with gehm-vertex giving the 0-simplices and gehm-edges the 1-simplices (following the standard construction for graphs). Take one disc for each hypervertex (g–r-cycle) and identify the boundary of this disc with the hypervertex. Do the same for each hyperedge and hyperface.

Finally, consider each isolate as a copy of S^1 and embed each in a distinct sphere. (We think of the two hemispheres as corresponding to a vertex and a face.) Thus we have obtained a cellular embedding of the gehm in a surface. We call this the *natural embedding* of a gehm. The 2-cells of this natural embedding correspond to hypervertices, hyperedges and hyperfaces of the gehm.

The natural embedding of the gehm in Fig. 1b is shown in Fig. 1c.

From the natural embedding of a gehm we can obtain a hypermap in the obvious way by placing one vertex in each hypervertex-disc to get V_v, and one vertex in each hyperedge-disc to get V_e. For each intersection between a hypervertex-disc and a hyperedge-disc embed an edge between the corresponding vertices in the usual way (the edges should not intersect themselves or each other). For an isolate place one vertex of V_v in the sphere. This is clearly reversible and when combined with Construction 1 gives a correspondence between hypermaps and gehms. Note that this correspondence draws the natural embedding of the gehm and its corresponding hypermap in the same surface. Thus, a hypermap and the natural embedding of its corresponding gehm are always in homeomorphic surfaces.

All of the parameters and terminology given in Terminology 1 align with their standard hypermap usage. The only terms that perhaps require some comment are Euler genus and orientability. Let $\mathbb{H}$ be a gehm corresponding to a hypermap G given by the bipartite graph $(V_v \sqcup V_e, E)$ embedded in a surface Σ. The Euler genus $\gamma(G)$ of G is the Euler genus of Σ. The Euler genus of a disconnected surface is the sum of the Euler genera of its components. We can disregard isolates as they do not contribute to the Euler genus of either $\mathbb{H}$ or G. We see that the definition of $\gamma(\mathbb{H})$ is consistent with $\gamma(G)$ as follows. By Euler's formula, $\gamma(G) = 2k(G) - (|V_v| + |V_e|) + e(G) - f(G)$. However, $|V_v| = v(\mathbb{H})$ and $|V_e| = e(\mathbb{H})$, while $k(G) = k(\mathbb{H})$ and $f(G) = f(\mathbb{H})$, and finally $e(G) = d(\mathbb{H})$. With this, $\gamma(\mathbb{H}) = \gamma(G)$, and this common value is also the Euler genus of the surface created in constructing the natural embedding of the gehm.

For orientability, let $\mathbb{H}$ be a gehm, G be the corresponding hypermap, and $\mathbb{G}$ this hypermap described as a gem. (Recall that $\mathbb{G}$ is an embedded bipartite graph, so may be described by a gem, that is, a gehm in which $d(e) = 2$ for every hyperedge e.) By considering how $\mathbb{G}$ can be obtained directly from $\mathbb{H}$ it is clear that $\mathbb{H}$ is bipartite if and only if $\mathbb{G}$ is. Then by a standard result about gems (see e.g., [6, Theorem 4.3]) $\mathbb{G}$ is bipartite if and only if G is orientable and it follows that $\mathbb{H}$ is orientable if and only if G is.

It follows from the correspondence and standard properties of the Euler genus of a surface that $\gamma(\mathbb{H}) \geq 0$ and if $\mathbb{H}$ is orientable then $\gamma(\mathbb{H})$ is even.

3 A Tutte Polynomial for Hypermaps

3.1 Duality and Minors

There are six ways to permute the colours of the edges of a gehm, each of which corresponds to a natural duality or triality operation.

Definition 1 Let $\mathbb{H}$ be a gehm, and μ be a permutation on the set $\{b, g, r\}$. Then we use $\mathbb{H}^{\mu}$ to denote the gehm obtained from $\mathbb{H}$ by, for each $c \in \{b, g, r\}$ changing all c-coloured edges to $\mu(c)$ coloured edges.

If μ is of order 2, then $\mathbb{H}^{\mu}$ is said to be a *dual*. In particular, $\mathbb{H}^{(br)}$ is the usual geometric dual, denoted by $\mathbb{H}^*$, which interchanges the faces and vertices of a hypermap. The gehm $\mathbb{H}^{(bgr)}$ is the *trial* of $\mathbb{H}$, introduced by Tutte in [30]. In this paper we focus on geometric duals: a fuller study of duality (and minors) can be found in [12].

Definition 2 Let $\mathbb{H}$ be a gehm, and let e be a hyperedge (i.e., a b–r-cycle). The *partial dual* $\mathbb{H}^e$ is the gehm obtained by interchanging the colours of the gehm-edges in the cycle e.

Figures 3b, and c show, respectively, the dual and the partial dual with respect to the degree four hyperedge of the gehm from Fig. 3a.

Notice that in moving from Fig. 3a, to b, the colours b and r are swapped, so every b–g cycle becomes a g–r cycle and vice versa. Thus every hyperface becomes a hypervertex and vice versa. While $\mathbb{H}$ and $\mathbb{H}^*$ in the figure are of genus zero, the partial dual $\mathbb{H}^e$ is of genus two. This can be verified by using Euler's formula.

For distinct hyperedges e and f, it is straightforward to check that $(\mathbb{H}^e)^f = (\mathbb{H}^f)^e$. Thus we may unambiguously extend the definition of partial duality to sets of hyperedges. For a set A of hyperedges, the partial dual $\mathbb{H}^A$ is defined to be the result of computing the partial dual with respect to each edge of A in any order.

Partial duals for hypermaps were introduced independently in [9, 29]. We only consider partial duals of hyperedges here, however, as in [9], this definition is easily extended to hypervertex and hyperface partial duals.

We next consider operations of deletion and contraction. Because a hyperedge can in general be incident with many hypervertices, there are many possible definitions of deletion. Below, we use one of the coarsest possible definitions, and remove the entire hyperedge without removing its incident hypervertices. This is sometimes called 'weak hyperedge deletion', in contrast to 'strong hyperedge deletion', which deletes the incident hypervertices as well. There are also other hyperedge deletion models. For example, in [10] the cyclic order of the hypervertices about the hyperedge e in the embedding is used to define a non-crossing partition of the hypervertices incident with e and replace e with multiple smaller hyperedges corresponding to the blocks of the partition. We shall discuss this further in Sect. 4.3.

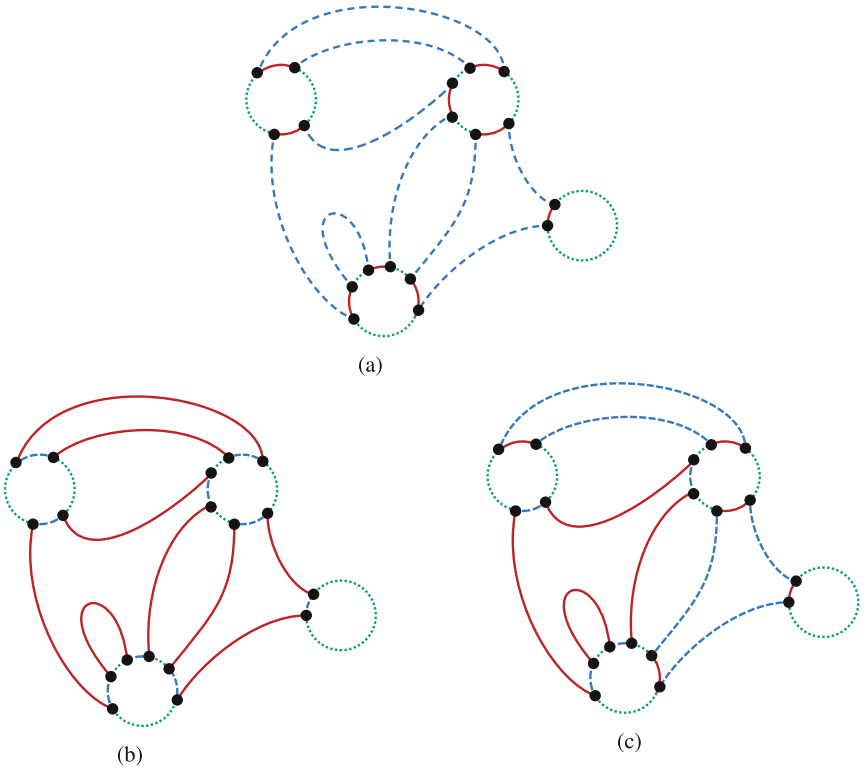


Fig. 3 The dual and a partial dual of a gehm $\mathbb{H}$. (**a**) The gehm $\mathbb{H}$. (**b**) The dual $\mathbb{H}^*$. (**c**) The partial dual $\mathbb{H}^e$ where e is the degree four edge

A common method of contracting the hyperedges of a hypergraph is often described as identifying a hyperedge and its incident hypervertices to form a new hypervertex. This agrees with our definition of hyperedge contraction in hypermaps below provided that all the hypervertices are distinct. If a hyperedge in a hypermap is incident with the same hypervertex multiple times, this does not hold, and multiple hypervertices may be created. (This is also what happens when contracting an orientable loop in a ribbon graph [15].)

Let v be a vertex of degree two in a graph. By *suppressing* v, we mean the following operation. If the only edge incident with v is a loop, then replace v and its incident edge with an isolated edge not adjacent to any vertex (in what follows this edge will be an isolate). Otherwise contract one of the edges incident with v.

Definition 3 Let $\mathbb{H}$ be a gehm, and let e be a hyperedge (i.e., a b–r-cycle) then

1. $\mathbb{H}$ *delete* e, denoted by $\mathbb{H} \setminus e$ is the gehm obtained from $\mathbb{H}$ by deleting the b-gehm-edges in the b–r-cycle e, contracting the r-gehm-edges in the b–r-cycle e and then suppressing the resulting vertices of degree two;

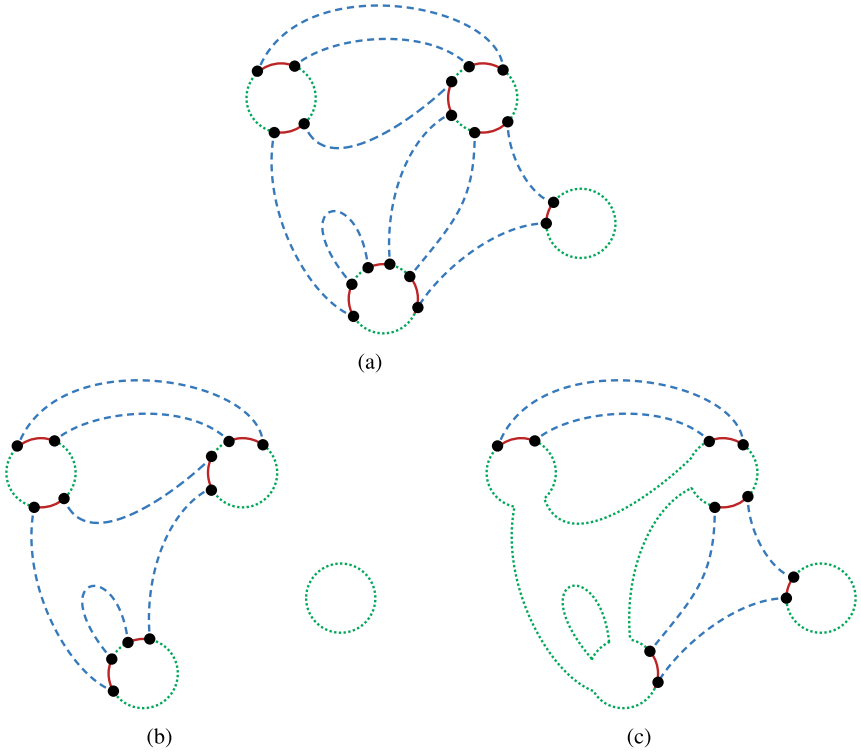


Fig. 4 Deleting and contracting a hyperedge in a gehm $\mathbb{H}$. (**a**) The gehm $\mathbb{H}$. (**b**) $\mathbb{H} \setminus e$ where e is the degree three hyperedge. (**c**) $\mathbb{H} / f$ where f is the degree four hyperedge

2. $\mathbb{H}$ *contract* e, denoted by $\mathbb{H} / e$ is the gehm obtained from $\mathbb{H}$ by deleting the r-gehm-edges in the b–r-cycle e, contracting the b-gehm-edges in the b–r-cycle e and then suppressing the resulting vertices of degree two.

Figure 4b shows the effect of deleting the edge with degree three from the gehm in Fig. 4a. Figure 4c shows the effect of contracting the edge with degree four from the gehm in Fig. 4a. Notice that in this example, both deletion and contraction create an additional component, here an isolate. Also, while the diagrams in Fig. 4a, b coincide with natural embeddings of the gehm in one sphere (Fig. 4a) or two spheres (Fig. 4b), the diagram in Fig. 4c does not. In a natural embedding, each isolate is in a separate spherical component.

The following lemma is straightforward.

Lemma 1 *Let $\mathbb{H}$ be a gehm, and let e and f be distinct hyperedges. Then the following hold.*

1. $\mathbb{H}^e \setminus e = \mathbb{H} / e$;
2. $(\mathbb{H} \setminus e) \setminus f = (\mathbb{H} \setminus f) \setminus e$;

3. $(\mathbb{H} / e) / f = (\mathbb{H} / f) / e$*;*
4. $(\mathbb{H} / e) \setminus f = (\mathbb{H} \setminus f) / e$*.*
5. If $\mathbb{H}$ *is orientable, then so are both* $\mathbb{H} \setminus e$ *and* $\mathbb{H} / e$*.*

It follows that we may carry out a sequence of deletion and contraction operations in any order without affecting the result. In particular, for a set A of hyperedges we may unambiguously define $\mathbb{H} \setminus A$ to be the result of deleting all the edges in A. We let $\mathbb{H}_{|A} = \mathbb{H} \setminus (E(\mathbb{H}) - A)$, $v(A) = v(\mathbb{H}_{|A}) = v(\mathbb{H})$, $k(A) = k(\mathbb{H}_{|A})$, $e(A) = e(\mathbb{H}_{|A}) = |A|$, $f(A) = f(\mathbb{H}_{|A})$ and $\gamma(A) = \gamma(\mathbb{H}_{|A})$. Finally, we let $d(A) = d(\mathbb{H}_{|A}) = \sum_{e \in A} d(e)$.

Hyperedge deletion for gehms and hypermaps correspond with each other. A description of deletion for maps can be found in, for example, [20, 28]. It acts as follows. If G is a map with an edge e then $G \setminus e$ is obtained by removing the edge e from the map together with its adjacent face or faces. This gives a surface with one boundary component. Next cap off the hole by identifying its boundary with the boundary of a disc, resulting in a map. (For readers familiar with ribbon graphs, this corresponds exactly to deleting an edge of a ribbon graph.)

Let $\mathbb{H}$ be a gehm and G be its corresponding hypermap (i.e., embedded bipartite graph). Suppose e is a hyperedge of $\mathbb{H}$ and v_e its corresponding vertex in G. Then the hypermap corresponding to $\mathbb{H} \setminus e$ is obtained by deleting all the edges incident with v_e then deleting v_e including the sphere it is embedded in.

3.2 Defining the Polynomials

In this section we introduce a Tutte polynomial for hypermaps. This polynomial is a direct generalisation of the well-studied Tutte polynomial for maps (which is also known as the ribbon graph polynomial) [4, 13, 20, 23, 28]. Although our polynomial is naturally defined in terms of deletion-contraction relations, it is more convenient to begin with an analogue of the dichromatic polynomial. This immediately generalizes to a multivariate version that facilitates partial duality identities which lead in turn to full duality formulas. A standard argument, albeit using hypermap properties, shows that the hypermap dichromatic polynomial has a deletion-contraction reduction. Our analogue of the dichromatic polynomial then leads to a hypermap analogue of the Tutte polynomial.

3.2.1 A Hypergraph Dichromatic Polynomial

Definition 4 The *dichromatic polynomial* $Z(\mathbb{H}; u, v)$ of a gehm $\mathbb{H}$ is defined as follows:

$$Z(\mathbb{H}; u, v) = \sum_{A \subseteq E(\mathbb{H})} u^{d(A)-|A|} v^{f(A)}.$$

It is straightforward to extend the dichromatic polynomial to a multivariate version, which will enable us to easily establish a duality relation. In the multivariate polynomial, the variable u is replaced by a family of commuting variables $\mathbf{u} := \{u_e\}_{e\in E(\mathbb{H})}$ indexed by the hyperedges of $\mathbb{H}$.

Definition 5 The *multivariate dichromatic polynomial* $Z(\mathbb{H}; \mathbf{u}, v)$ of a gehm $\mathbb{H}$ is defined as follows:

$$Z(\mathbb{H}; \mathbf{u}, v) = \sum_{A\subseteq E(\mathbb{H})} \Big(\prod_{e\in A} u_e^{d(e)-1} \Big)\, v^{f(A)}.$$

The multivariate dichromatic polynomial satisfies the following recurrence relation. For this we note that a gehm $\mathbb{H}$ with no hyperedges comprises $v(\mathbb{H}) = f(\mathbb{H})$ isolates.

Lemma 2 *Let $\mathbb{H}$ be a gehm. Then $Z(\mathbb{H}; \mathbf{u}, v) = v^{f(\mathbb{H})}$ if $\mathbb{H}$ has no hyperedges and otherwise for each hyperedge e,*

$$Z(\mathbb{H}; \mathbf{u}, v) = Z(\mathbb{H} \setminus e; \{u_e\}_{e\in E(\mathbb{H}\setminus e)}, v) + u_e^{d(e)-1} Z(\mathbb{H} / e; \{u_e\}_{e\in E(\mathbb{H}/e)}, v).$$

Proof The argument is very standard so we spare the reader the details beyond noting the key observation that for every subset A of $E(\mathbb{H}) - \{e\}$, we have $f(A \cup \{e\}) = f((\mathbb{H} / e)_{|A})$. □

From this we immediately deduce the following.

Corollary 1 *Let $\mathbb{H}$ be a gehm. Then $Z(\mathbb{H}; u, v) = v^{f(\mathbb{H})}$ if $\mathbb{H}$ has no hyperedges and otherwise for each hyperedge e,*

$$Z(\mathbb{H}; u, v) = Z(\mathbb{H} \setminus e; u, v) + u^{d(e)-1} Z(\mathbb{H} / e; u, v).$$

3.2.2 Duality

We first examine the effect of partial duality on $Z(\mathbb{H}; \mathbf{u}, v)$. Consider a gehm $\mathbb{H}$ and a subset X of its hyperedges. We identify the edges of $\mathbb{H}$ and $\mathbb{H}^X$ in the natural way. Given $\mathbf{u} = \{u_e\}_{e\in E(\mathbb{H})}$, we define $\mathbf{u}^X = \{u'_e\}_{e\in E(\mathbb{H}^X)}$ by

$$u'_e = \begin{cases} 1/u_e & \text{if } e \in X, \\ u_e & \text{if } e \notin X. \end{cases}$$

Proposition 1 *Let $\mathbb{H}$ be a gehm, and let X be a subset of its hyperedges. Then*

$$Z(\mathbb{H}; \mathbf{u}, v) = \Big(\prod_{e\in X} u_e^{d(e)-1} \Big)\, Z(\mathbb{H}^X; \mathbf{u}^X, v).$$

Proof For each hyperedge e and subset A of hyperedges of $\mathbb{H}$, we have $f(\mathbb{H}_{|A}) = f((\mathbb{H}^e)_{|A} \triangle \{e\})$, so

$$\begin{aligned} Z(\mathbb{H}; \mathbf{u}, v) &= \sum_{A \subseteq E(\mathbb{H}) - \{e\}} \Big(\prod_{h \in A} u_h^{d(h)-1} \Big) (v^{f(A)} + u_e^{d(e)-1} v^{f(A \cup \{e\})}) \\ &= \sum_{A \subseteq E(\mathbb{H}^e) - \{e\}} \Big(\prod_{h \in A} u_h^{d(h)-1} \Big) (v^{f((\mathbb{H}^e)_{|A \cup \{e\}})} + u_e^{d(e)-1} v^{f((\mathbb{H}^e)_{|A})}) \\ &= u_e^{d(e)-1} \sum_{A \subseteq E(\mathbb{H}^e) - \{e\}} \\ &\quad \Big(\prod_{h \in A} u_h^{d(h)-1} \Big) \Big(\frac{1}{u_e^{d(e)-1}} v^{f((\mathbb{H}^e)_{|A \cup \{e\}})} + v^{f((\mathbb{H}^e)_{|A})} \Big) \\ &= u_e^{d(e)-1} \, Z(\mathbb{H}^e; \mathbf{u}^{\{e\}}, v). \end{aligned}$$

The result now follows by induction on $|X|$. □

From this we deduce the following.

Corollary 2 *Let $\mathbb{H}$ be a gehm. Then*

$$Z(\mathbb{H}; u, v) = u^{d(\mathbb{H}) - e(\mathbb{H})} \, Z(\mathbb{H}^*; 1/u, v).$$

3.2.3 Translating to the Associated Hypergraph Tutte Polynomial

We wish to define an analogue of the Tutte polynomial for hypermaps. For this we take the approach described in [28] where it is explained how the classical connection between the dichromatic and Tutte polynomials of a graph (see, e.g., [16]) can be used to derive a Tutte polynomial of maps. The three key properties that our Tutte polynomial for hypermaps, $T(\mathbb{H}; x, y)$, should satisfy are: (1) it should be equivalent (up to change of variables and multiplication by simple prefactors) to the dichromatic polynomial; (2) it should satisfy the duality relation $T(\mathbb{H}; x, y) = T(\mathbb{H}^*; y, x)$; and (3) $T(\mathbb{H}; x, y)$ should coincide with the Tutte polynomial of a map [4, 13, 20, 23, 28] when $\mathbb{H}$ is a gem, that is a hypermap in which each hyperedge has degree two, and hence represents a map. For this we define

$$\rho(\mathbb{H}) = v(\mathbb{H}) - k(\mathbb{H}) + \tfrac{1}{2}\gamma(\mathbb{H}),$$

and for a set A of hyperedges of $\mathbb{H}$ we let

$$\begin{aligned} \rho(A) = \rho(\mathbb{H}_{|A}) &= v(A) - k(A) + \tfrac{1}{2}\gamma(A) \\ &= \frac{1}{2}(v(A) + d(A) - |A| - f(A)), \end{aligned}$$

where the last equality follows by Euler's Formula. It follows immediately from properties of γ that we have established that $\rho(A) \geq 0$, and if $\mathbb{H}$ is orientable then $\rho(A)$ is integral for each subset of its edges (since $\gamma(A)$ must be even).

We can now introduce our Tutte polynomial for hypermaps as a sum over sets of hyperedges. The reader will likely notice a striking similarity with the definition of the classical Tutte polynomial of a graph. As we shall see, this similarity is an important feature of the polynomial.

Definition 6 For a gehm $\mathbb{H}$, we define its *Tutte polynomial* by

$$T(\mathbb{H}; x, y) = \sum_{A \subseteq E(\mathbb{H})} (x-1)^{\rho(\mathbb{H})-\rho(A)} (y-1)^{d(A)-|A|-\rho(A)}. \tag{1}$$

Proposition 2 *The Tutte polynomial is a polynomial in* $\sqrt{x-1}$ *and* $\sqrt{y-1}$.

Proof Consider the hypermap (i.e., the embedded bipartite graph G) corresponding to the gehm $\mathbb{H}$. As deleting edges in a map cannot increase genus or decrease the number of components, $v(\mathbb{H}) - k(\mathbb{H}) \geq v(A) - k(A)$ and $\gamma(\mathbb{H}) \geq \gamma(A)$. Thus $\rho(\mathbb{H}) - \rho(A) \geq 0$ and the $(x-1)$ exponent is non-negative.

The $(y-1)$ exponent can be written as $d(A)-|A|-(v(A)-k(A)+\frac{1}{2}\gamma(A))$. Here $d(A) - (|A| + v(A)) + k(A)$ is the nullity of the bipartite graph G corresponding to $\mathbb{H}_{|A}$. In any map G, the Euler genus $\gamma(G)$ cannot be greater than the twice the nullity. To see this start with a spanning tree of each connected component of G embedded in the sphere. The number of remaining edges is equal to the nullity and adding these edges one at a time increases γ by at most two at each stage. Thus $d(A)-|A|-\rho(A) \geq 0$. (A more sophisticated argument, for example by considering the homology generators, will show $\gamma(G)$ cannot be greater than the nullity.) □

Since when $\mathbb{H}$ is orientable $\rho(A)$ is integral for each subset of edges A, the following holds.

Proposition 3 *If* $\mathbb{H}$ *is orientable, then* $T(\mathbb{H}; x, y)$ *can be expanded as a polynomial in* x *and* y.

The converse of Proposition 3 is false, as shown by the gehm shown in Fig. 5a which is non-orientable but has Tutte polynomial $x + y - 2$.

By comparing the exponents of the corresponding terms in the sums expressing T and Z we easily obtain the following translation between the two functions.

Proposition 4 *For a gehm* $\mathbb{H}$,

$$T(\mathbb{H}; x+1, y+1) = \sqrt{x}^{d(\mathbb{H})-e(\mathbb{H})-f(\mathbb{H})} \sqrt{y}^{-v(\mathbb{H})} Z\left(\mathbb{H}; \frac{\sqrt{y}}{\sqrt{x}}, \sqrt{x}\sqrt{y}\right).$$

This enables us to obtain the following deletion-contraction recurrence and duality relation for T.

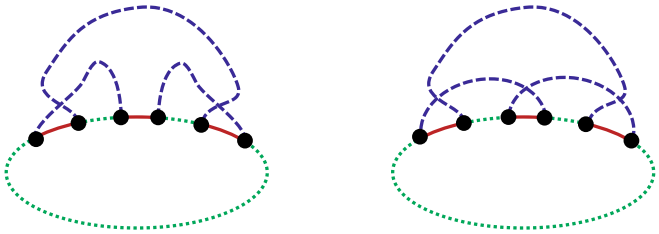


Fig. 5 Two gehms with the same Tutte polynomial $x + y - 2$. (**a**) A nonorientable gehm. (**b**) An orientable gehm

Theorem 1 *Let $\mathbb{H}$ be a gehm. Then $T(\mathbb{H}; x, y) = 1$ if $\mathbb{H}$ has no hyperedges and otherwise for each hyperedge e,*

$$T(\mathbb{H}; x, y) = \sqrt{x-1}^{\,f(\mathbb{H}\setminus e)-f(\mathbb{H})+d(e)-1} T(\mathbb{H} \setminus e; x, y) + \sqrt{y-1}^{\,v(\mathbb{H}/e)-v(\mathbb{H})+d(e)-1} T(\mathbb{H} / e; x, y).$$

Proof Using Proposition 4, Corollary 1 and then Proposition 4 again, we obtain

$$\begin{aligned} T(\mathbb{H}; x+1, y+1) &= \sqrt{x}^{\,d(\mathbb{H})-e(\mathbb{H})-f(\mathbb{H})} \sqrt{y}^{\,-v(\mathbb{H})} Z\left(\mathbb{H}; \tfrac{\sqrt{y}}{\sqrt{x}}, \sqrt{x}\sqrt{y}\right) \\ &= \sqrt{x}^{\,d(\mathbb{H})-e(\mathbb{H})-f(\mathbb{H})} \sqrt{y}^{\,-v(\mathbb{H})} \left(Z\left(\mathbb{H} \setminus e; \tfrac{\sqrt{y}}{\sqrt{x}}, \sqrt{x}\sqrt{y}\right)\right. \\ &\quad \left. + \left(\frac{\sqrt{y}}{\sqrt{x}}\right)^{d(e)-1} Z\left(\mathbb{H} / e; \tfrac{\sqrt{y}}{\sqrt{x}}, \sqrt{x}\sqrt{y}\right)\right) \\ &= \sqrt{x}^{\,d(e)-1+f(\mathbb{H}\setminus e)-f(\mathbb{H})} \sqrt{y}^{\,v(\mathbb{H}\setminus e)-v(\mathbb{H})} \\ &\quad T(\mathbb{H} \setminus e; x+1, y+1) \\ &\quad + \sqrt{x}^{\,f(\mathbb{H}/e)-f(\mathbb{H})} \sqrt{y}^{\,d(e)-1+v(\mathbb{H}/e)-v(\mathbb{H})} \\ &\quad T(\mathbb{H} / e; x+1, y+1). \end{aligned}$$

The result follows by noting that $v(\mathbb{H} \setminus e) = v(\mathbb{H})$ and dually that $f(\mathbb{H} / e) = f(\mathbb{H})$. □

Proposition 5 *For a gehm $\mathbb{H}$,*

$$T(\mathbb{H}^*; x, y) = T(\mathbb{H}; y, x).$$

Proof By using Proposition 4, Corollary 2 and then Proposition 4 again, we obtain

$$\begin{aligned} T(\mathbb{H}; x+1, y+1) &= \sqrt{x}^{\,d(\mathbb{H})-e(\mathbb{H})-f(\mathbb{H})} \sqrt{y}^{\,-v(\mathbb{H})} Z\left(\mathbb{H}; \frac{\sqrt{y}}{\sqrt{x}}, \sqrt{x}\sqrt{y}\right) \\ &= \sqrt{x}^{\,-f(\mathbb{H})} \sqrt{y}^{\,d(\mathbb{H})-e(\mathbb{H})-v(\mathbb{H})} Z\left(\mathbb{H}^*; \frac{\sqrt{x}}{\sqrt{y}}, \sqrt{x}\sqrt{y}\right) \\ &= \sqrt{x}^{\,-v(\mathbb{H}^*)} \sqrt{y}^{\,d(\mathbb{H}^*)-e(\mathbb{H}^*)-f(\mathbb{H}^*)} Z\left(\mathbb{H}^*; \frac{\sqrt{x}}{\sqrt{y}}, \sqrt{x}\sqrt{y}\right) \\ &= T(\mathbb{H}^*; y+1, x+1). \end{aligned}$$

□

It is worth emphasising that Proposition 5 holds for all gehms, not just those of genus 0, in contrast with the duality relation for the more refined hypermap polynomial of [11, Theorem 2.16] which only holds for genus 0 hypermaps.

As all the relevant parameters are additive over components, if $\mathbb{H}_1$ and $\mathbb{H}_2$ are disjoint gehms, we have $T(\mathbb{H}_1 \sqcup \mathbb{H}_2; x, y) = T(\mathbb{H}_1; x, y)\, T(\mathbb{H}_2; x, y)$.

Now let $\mathbb{H}_1$ and $\mathbb{H}_2$ be disjoint gehms such that for $i = 1, 2$, $\mathbb{H}_i$ includes a g-edge $e_i = x_i y_i$ which is not an isolate. Then a join of $\mathbb{H}_1$ and $\mathbb{H}_2$ along e_1 and e_2, denoted by $\mathbb{H}_1 \,{}_{e_1}\vee_{e_2}\, \mathbb{H}_2$, is obtained by forming the disjoint union of $\mathbb{H}_1$ and $\mathbb{H}_2$, and then replacing the g-edges e_1 and e_2 by g-edges x_1x_2 and y_1y_2.

Proposition 6 *Let $\mathbb{H}_1$ and $\mathbb{H}_2$ be disjoint gehms such that for $i = 1, 2$, $\mathbb{H}_i$ includes a g-edge $e_i = x_i y_i$ which is not an isolate. Then we have*

$$T(\mathbb{H}_1 \,{}_{e_1}\vee_{e_2}\, \mathbb{H}_2; x, y) = T(\mathbb{H}_1; x, y)\, T(\mathbb{H}_2; x, y).$$

Proof Let $\mathbb{H} = \mathbb{H}_1 \,{}_{e_1}\vee_{e_2}\, \mathbb{H}_2$. Observe that for any subset A of the hyperedges of $\mathbb{H}$, we have $v(\mathbb{H}_{|A}) = v((\mathbb{H}_1 \sqcup \mathbb{H}_2)_{|A}) - 1$ and $f(\mathbb{H}_{|A}) = f((\mathbb{H}_1 \sqcup \mathbb{H}_2)_{|A}) - 1$. Hence $\rho(\mathbb{H}_{|A}) = \rho((\mathbb{H}_1 \sqcup \mathbb{H}_2)_{|A})$. Thus

$$T(\mathbb{H}; x, y) = T(\mathbb{H}_1 \sqcup \mathbb{H}_2; x, y) = T(\mathbb{H}_1; x, y)\, T(\mathbb{H}_2; x, y).$$

□

In order to state some evaluations of T we make the following definitions. Notice that for a gehm $\mathbb{H}$ we have $v(\mathbb{H}) \leq d(\mathbb{H}) - e(\mathbb{H}) + k(\mathbb{H})$. (For any graph G we have $e(G) \geq v(G) - k(G)$. The gehm inequality follows by applying this to the underlying bipartite graph of $\mathbb{H}$ as a hypermap.) When equality holds we say that $\mathbb{H}$ is a *hyperforest*. Then a *hypertree* is a connected hyperforest. For example, deleting the degree four hyperedge of Fig. 1b gives a hypertree.

For a hypertree $\mathbb{H}$, we have

$$\gamma(\mathbb{H}) = 2 - f(\mathbb{H}) + d(\mathbb{H}) - e(\mathbb{H}) - v(\mathbb{H}) = 1 - f(\mathbb{H}).$$

As $\gamma(\mathbb{H}) \geq 0$ and $f(\mathbb{H}) \geq 1$, we deduce that $f(\mathbb{H}) = 1$ and $\gamma(\mathbb{H}) = 0$.

For a gehm $\mathbb{H}$, we define $t(\mathbb{H})$ to be its number of *spanning hypertrees*, that is, the number of subsets A of $E(\mathbb{H})$ so that $\mathbb{H}_{|A}$ is a hypertree.

Proposition 7 *Let $\mathbb{H}$ be a gehm. Then*

1. $T(\mathbb{H}; 2, 2) = 2^{e(\mathbb{H})}$.
2. *T does not detect orientability.*
3. *If $\mathbb{H}$ is connected, then*

$$T(\mathbb{H}; 1, 1) = \begin{cases} t(\mathbb{H}) & \text{if } \mathbb{H} \text{ has Euler genus } 0, \\ 0 & \text{otherwise.} \end{cases}$$

Proof

1. When $x = y = 2$, every term in the sum in the right-side of Eq. (1) equals 1, and the result follows easily.
2. For example, the gehms in Fig. 5 share the same Tutte polynomial, namely $x + y - 2$, but only the gehm in Fig. 5b is orientable.
3. Clearly $T(\mathbb{H}; 1, 1)$ is equal to the number of subsets A of $E(\mathbb{H})$ with $\rho(A) = \rho(\mathbb{H}) = d(A) - |A|$. As $\mathbb{H}$ is connected, we have $\rho(\mathbb{H}) = v(\mathbb{H}) - 1 + \gamma(\mathbb{H})/2$.

   Next suppose that $\gamma(\mathbb{H}) = 0$ and that $\mathbb{H}_{|A}$ is a hypertree. Then $\rho(A) = v(A) - 1$ and, from the definition of a hypertree, $d(A) - |A| = v(A) - 1$. Hence $\rho(A) = \rho(\mathbb{H}) = d(A) - |A|$ and A contributes one to $T(\mathbb{H}; 1, 1)$.

   Now let A be a subset of $E(\mathbb{H})$ with $\rho(A) = \rho(\mathbb{H}) = d(A) - |A|$. The condition $\rho(A) = d(A) - |A|$ is equivalent to

$$d(A) - |A| = v(A) - f(A). \tag{2}$$

   The non-negativity of $\gamma(A)$ gives

$$d(A) - |A| \geq f(A) + v(A) - 2k(A). \tag{3}$$

   By combining these two equations we get $f(A) \leq k(A)$ which gives $f(A) = k(A)$. Then Eq. (2) yields $v(A) = d(A) - |A| + k(A)$ which implies that $\mathbb{H}_{|A}$ is a hyperforest. Moreover, we also get

$$\rho(A) = d(A) - |A| = v(\mathbb{H}) - k(A).$$

   Thus the condition $\rho(A) = \rho(\mathbb{H})$ is equivalent to $v(\mathbb{H}) - k(A) = v(\mathbb{H}) - 1 + \gamma(\mathbb{H})/2$ which is only satisfied when $\gamma(\mathbb{H}) = 0$ and $k(A) = 1$, that is, when $\gamma(\mathbb{H}) = 0$ and $\mathbb{H}_{|A}$ is a hypertree.

□

4 Connections with Other Polynomials

4.1 Classical and Topological Tutte Polynomials

We begin by describing the coincidence of $T(\mathbb{H}; x, y)$ with the Tutte polynomials of graphs and maps. If $\mathbb{H}$ is a gem and therefore represents a graph embedded in a surface then $T(\mathbb{H}; x, y)$ coincides with the *ribbon graph polynomial*, also known as the *2-variable Bollobás–Riordan polynomial* or the *Tutte polynomial of cellularly embedded graphs*. The ribbon graph polynomial is an important and well-studied map analogue of the Tutte polynomial [13, 20, 23, 28]. When $\mathbb{H}$ is a gem, $T(\mathbb{H}; x, y)$ is also a specialisation of the *Bollobás–Riordan polynomial* of [4], with $T(\mathbb{H}; x, y) = (x-1)^{\gamma(\mathbb{H})/2} R(\mathbb{H}; x, y-1/\sqrt{(x-1)(y-1)}, 1)$. If the gem $\mathbb{H}$ represents a graph G embedded in the plane, then by Euler's Formula $\rho(A)$ equals the rank of the graph $G \setminus (E - A)$. It follows that in this case $T(\mathbb{H}; x, y) = T(G; x, y)$ where $T(G; x, y)$ is the classical Tutte polynomial of the graph G.

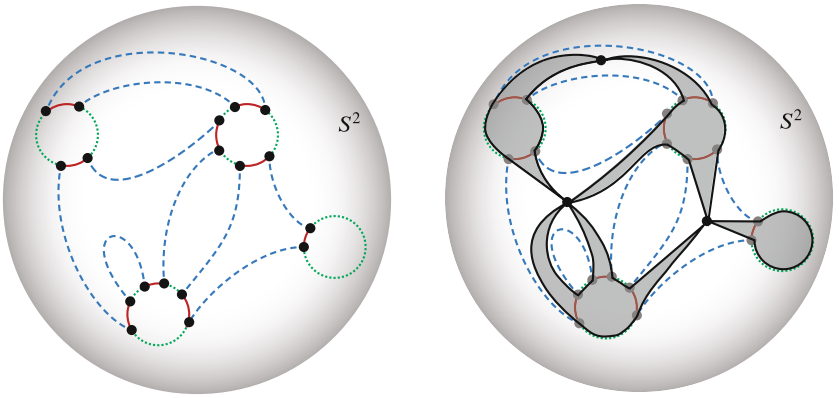
4.2 Transition Polynomials

Here we use the term *Eulerian graph* to mean a graph in which each vertex is of even degree (and in particular it need not be connected). In order to accommodate isolates, we also allow Eulerian graphs to have edges that are incident with no vertices. We call these *free loops*.

The multivariable dichromatic polynomial $Z(\mathbb{H}; \boldsymbol{u}, v)$ can be seen to be an evaluation of the transition polynomial of an Eulerian graph. To make this connection, we must first define the medial map of a gehm.

Let $\mathbb{H}$ be a gehm. Its *medial map* $\mathbb{H}_m$ is an Eulerian map defined as follows. Consider the natural embedding of the gehm in a surface Σ as given in Construction 1. Recall that each face in the embedding corresponds to a hypervertex, hyperedge or hyperface. Then $\mathbb{H}_m$ is the graph embedded in Σ constructed as follows. Retain all isolates, so each isolate becomes a circle embedded in a sphere. For all non-isolate components proceed as follows. For vertices of $\mathbb{H}_m$, place one vertex in the interior of each of the faces of the embedded gehm that corresponds to a hyperedge. Form the edges of $\mathbb{H}_m$ by embedding non-intersecting arcs as follows. For each vertex w_e of $\mathbb{H}_m$ corresponding to a hyperedge e, embed non-intersecting arcs from w_e to each gehm-vertex of the hyperedge e. Now for each g-edge in the gehm, join up the two arcs meeting its end-vertices to form an edge of $\mathbb{H}_m$, as in Fig. 6.

Note that each face of the map $\mathbb{H}_m$, other than those in components arising from isolates, corresponds to either a hypervertex or hyperface of $\mathbb{H}$. Colour the faces corresponding to hypervertices grey and those corresponding to hyperfaces white. For the components arising from isolates, assign one colour to each face. This results in a proper face 2-colouring. We call this a *natural checkerboard colouring* of $\mathbb{H}_m$.



(a) (b)

Fig. 6 Creating and face 2-colouring the medial map $\mathbb{H}_m$ of a gehm $\mathbb{H}$. (**a**) Natural embedding of the gehm $\mathbb{H}$ in the sphere. (**b**) The face 2-coloured medial map $\mathbb{H}_m$ in the sphere, superimposed on the natural embedding of the gehm

We now recall the generalized transition polynomial and define a specialisation of it that, when applied to the medial map, agrees with the dichromatic polynomial of the gehm. The generalised transition polynomial, $q(G; W, t)$, of [17] is a multivariate graph polynomial that generalises Jaeger's transition polynomial of [21]. In [14], the authors specialised the generalised transition polynomial to maps, calling this specialisation the *topological transition polynomial*. Our hypermap (or gehm) transition polynomial uses analogous ideas.

A *vertex state* at a vertex w of an Eulerian graph is a partition of the half-edges incident with w into pairs. The corresponding *smoothing* at a vertex w is the result of the following process for all half-edges that are paired. If (u, w) and (v, w) are two non-loop edges whose half-edges are paired at the vertex w, then we replace these two edges with a single edge (u, v). In the case of a loop, we temporarily insert an extra vertex of degree two on the loop, carry out the operation, and then suppress the temporary vertex.

A *graph state* of an Eulerian graph G is a choice of vertex state at each of its vertices. A set of free loops is obtained from a graph state s by smoothing each vertex state in it. We let $k(s)$ denote the number of free loops arising from s in this way. If G has no vertices, and so is either empty or a collection of free loops then its unique graph state is just itself. We let $\mathcal{S}(G)$ denote the set of all graph states of G.

We can now assign weights to these vertex and graph states.

Definition 7 Let $\mathcal{R}$ be a commutative ring with unity.

- A *pair weight* is an association of a value $p(e_w, f_w) \in \mathcal{R}$ to a pair of half-edges incident with a vertex w.

- A *weight system* of an Eulerian graph G, denoted $\Omega(G)$, or simply Ω when G is clear from context, is an assignment of a pair weight to every possible pair of adjacent half-edges of G.
- The *vertex state weight* of a vertex state of a vertex w is $\prod p(e_w, f_w)$ over all pairs (e_w, f_w) forming the vertex state.
- The *state weight* of a graph state s of a graph G with weight system Ω is $\omega(s) = \prod \omega(w, s)$ where $\omega(w, s)$ is the vertex state weight of the vertex state at w in the graph state s, and where the product is over all vertices of G.

Note that in many specialisations of the generalised transition polynomial, it is common to give just vertex state weights for particular vertex states. Thus, if a vertex has degree $2n$, then implicit in giving just a vertex state weight of say α for some vertex state is that all the pair weights for the edge pairs comprising that state are $\alpha^{1/n}$. This assignment of pair weights of course has to be consistent across all the vertex states. It is also common to use additional information, such as the cyclic order of the edges about a vertex in an embedding or a face colouring to determine vertex state weights.

Figure 7 shows a vertex v of degree 4 within a graph G. So a weight system for G would include 6 pair weights corresponding to the $\binom{4}{2}$ pairs of half-edges at v. There are three vertex states at v, corresponding to the three ways of partitioning the four half-edges into sets of size two. We are usually interested in the product of the pair weights corresponding to the pairing of half-edges in a vertex state, so as noted above it is common to specify the vertex state weights, providing these are consistent. Issues of consistency never arise for a vertex of degree four, and a potential set of the three vertex state weights at v is also shown in Fig. 7. In this case G is a two face-coloured map and the weights are determined from the colouring.

Figure 8a shows a face 2-coloured map $\mathbb{G}$ with two vertices, both of degree four. (The map is in the sphere with the drawing on the page indicating the embedding.) So $\mathbb{G}$ has nine graph states. One of these graph states is shown in Fig. 8b with the labels on the vertices indicating the vertex state weights following the scheme given in Fig. 7. Thus if s is this graph state, $\omega(s) = ab$ and $k(s) = 1$.

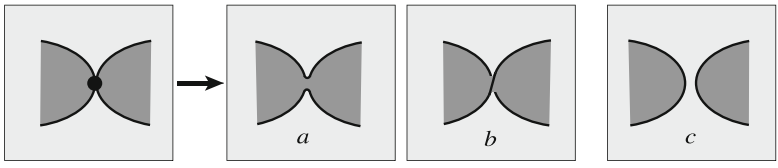
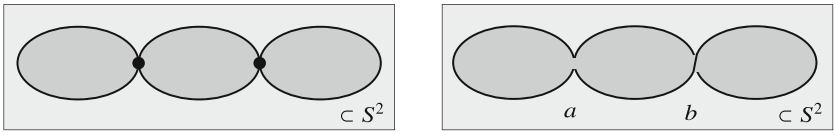


Fig. 7 A set of vertex state weights for all vertex states at a vertex of degree 4 in a face 2-coloured map (showing part of the surface), where $a, b, c \in \mathcal{R}$



(a) (b)

Fig. 8 An example of a map and a graph state. (**a**) A plane, face 2-coloured map $\mathbb{G}$. (**b**) One of the graph states of $\mathbb{G}$ showing its vertex state weights

The generalised transition polynomial is then defined as follows.

Definition 8 Let G be an Eulerian graph, with weight system Ω. Then the *generalised transition polynomial* is

$$q(G, \Omega; x) = \sum_{s \in \mathcal{S}(G)} \omega(s)\, x^{k(s)}.$$

We now apply the generalized transition polynomial to obtain a hypermap transition polynomial via the medial map. We will see that with appropriately chosen weight systems, both the coarse Tutte polynomial for hypermaps defined here and the topological transition polynomial of [14] are specialisations of the hypermap transition polynomial.

Definition 9 (The Hypermap Transition Polynomial) The *hypermap transition polynomial* $\Phi(\mathbb{H}, \Omega, t)$ is the specialization of the generalised transition polynomial to medial maps of gehms (i.e., hypermaps) given by

$$\Phi(\mathbb{H}, \Omega, t) = \sum_{s \in \mathcal{S}(\mathbb{H}_m)} \omega(s)\, t^{k(s)},$$

where Ω is a weight system for $\mathbb{H}_m$.

For our purposes, we need a particular specialisation of the hypermap transition polynomial, using only two types of vertex smoothings. If $\mathbb{H}$ is a gehm and $\mathbb{H}_m$ is its naturally checkerboard coloured medial map then we may distinguish two special vertex states at w. Travelling round w we see half-edges and faces in the cyclic order $h_0 f_{0,g} h_1 f_{1,w} \cdots f_{2d-1,w} h_0$ where h_i are the half-edges, $f_{i,g}$ grey faces and $f_{i,w}$ white faces. The *c-state* pairs $\{h_1, h_2\}, \{h_3, h_4\}, \ldots \{h_{2d-1}, h_0\}$; the *d-state* pairs $\{h_0, h_1\}, \{h_2, h_3\}, \ldots \{h_{2d-2}, h_{2d-1}\}$. Note that for vertices of degree 2 the c-state and d-state are identical. See Fig. 9.

Using the notation in the previous paragraph, we define pair weights by setting, for vertices of degree $2d \geq 4$, $p(h_1, h_2) = p(h_3, h_4) = \cdots = p(h_{2d-1}, h_0) = u^{1-1/d}$, and $p(h_0, h_1) = p(h_2, h_3) = \cdots = p(h_{2d-2}, h_{2d-1}) = 1$, and all other $p(h_i, h_j) = 0$. For vertices of degree two, $p(h_1, h_2) = 2$. Let $\Omega_m(\mathbb{H})$ denote the resulting weight system.

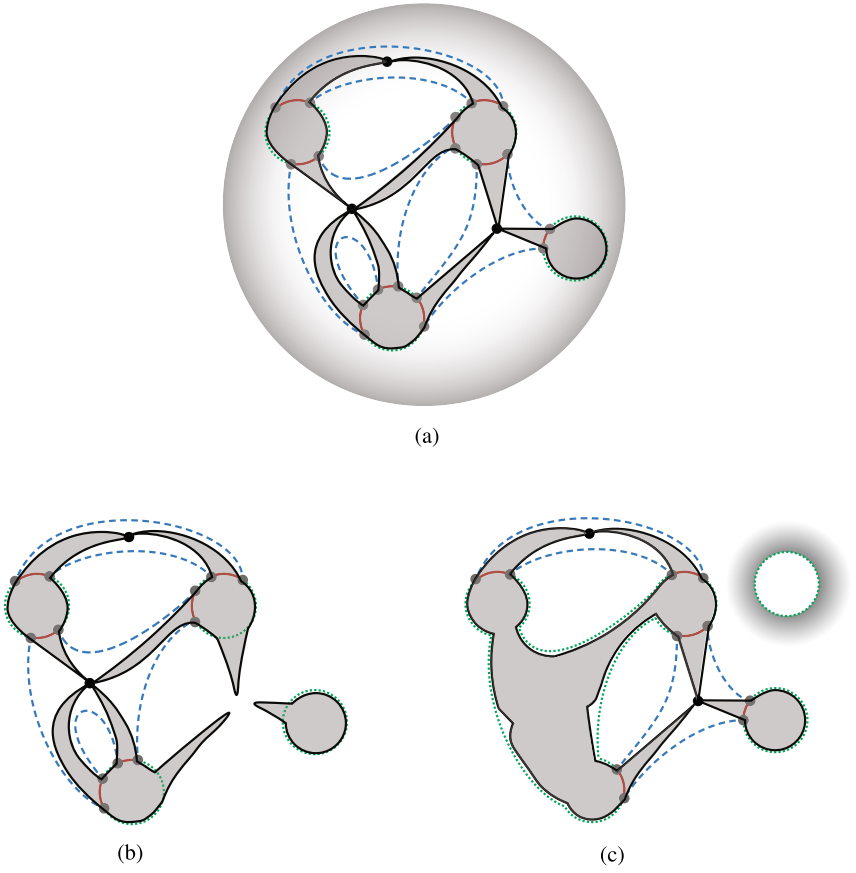


Fig. 9 d- and c-smoothings in a medial map corresponding to deleting and contracting a hyperedge in a gehm. In the bottom two figures, the isolate is in one sphere, while the rest of the diagram lies in a separate sphere. (**a**) The medial map $\mathbb{H}_m$ superimposed on the natural embedding of $\mathbb{H}$ in the sphere. (**b**) A d-smoothing in $\mathbb{H}_m$ corresponding to $\mathbb{H} \setminus e$ where e is the degree three hyperedge in $\mathbb{H}$. (**c**) A c-smoothing in $\mathbb{H}_m$ corresponding to $\mathbb{H} / f$ where f is the degree four hyperedge in $\mathbb{H}$

Theorem 2 *Let $\mathbb{H}$ be a gehm, $\mathbb{H}_m$ be its medial map, and $\Omega_m(\mathbb{H})$ its medial weight system. Then.*

$$\Phi(\mathbb{H}, \Omega_m(\mathbb{H}), v) = Z(\mathbb{H}; u, v).$$

Proof First, notice that $\omega(s) = 0$ unless all vertex states are c-states or d-states. Write $\mathcal{S}'(\mathbb{H}_m)$ for the set of states consisting only of c-states and d-states. Using w_e to denote the vertex of $\mathbb{H}_m$ corresponding to the hyperedge e of $\mathbb{H}$, let $C(s) = \{e \in E(\mathbb{H}) : w_e \in V(\mathbb{H}_m)$ has a c-state in $s\}$. Thus, the states of $\mathcal{S}'(\mathbb{H}_m)$ are in one-to-one correspondence with subsets $A \subseteq E(\mathbb{H})$ by $A_s = C(s)$. Second, notice

that the free loops arising from the smoothing of s correspond to the hyperfaces of $\mathbb{H} \setminus (E(\mathbb{H}) - A_s) \,/\, A_s$ which in turn correspond to the faces of $\mathbb{H}_{|A_s}$. This gives:

$$\begin{aligned}\Phi(\mathbb{H}, \Omega_m(\mathbb{H}), v) &= \sum_{s \in \mathcal{S}(\mathbb{H}_m)} \omega(s) v^{k(s)} \\ &= \sum_{s \in \mathcal{S}'(\mathbb{H}_m)} \Big(\prod_{e \in C(s)} u^{d(e)-1} \Big) v^{k(s)} \\ &= \sum_{A \in E(\mathbb{H})} u^{d(A)-|A|} v^{f(A)} \\ &= Z(\mathbb{H}; \boldsymbol{u}, v).\end{aligned}$$

□

We note that it is straightforward to extend Theorem 2 to recover the multivariate dichromatic polynomial from the transition polynomial.

To recover the topological transition polynomial, if $\mathbb{H}$ is a gem then every vertex in $\mathbb{H}_m$ has degree 4. Following our previous notation we set $p(h_1, h_2) = p(h_3, h_0) = \alpha$, $p(h_0, h_1) = p(h_2, h_3) = \beta$, $p(h_0, h_2) = p(h_1, h_3) = \gamma$. Let $\Omega_t(\mathbb{H})$ denote the resulting weight system. It is then immediate that $\Phi(\mathbb{H}, \Omega_t(\mathbb{H}), t)$ is the topological transition polynomial of [14].

4.3 *Cori and Hetyei's Whitney Polynomial for Hypermaps*

In [11], Cori and Hetyei define a Whitney polynomial for hypermaps, $R(\mathbb{H})$, using a more refined definition of edge deletion than we use here for $Z(\mathbb{H})$. The two polynomials appear to be distinct in that it does not seem possible to recover one from the other.

We first give the edge deletion of [10, 11], which is based on hyperedge refinements, in terms of our gehm constructions, and then compare deletions and the resulting polynomials. A refinement of a hyperedge, in terms of the gehm representation, is the result of viewing the b–r-cycle of the hyperedge as lying on a circle and replacing a subset of its b-edges by the same number of non-crossing chords. These new edges are also b-edges and form smaller alternating b–r-cycles with the remaining b-edges and r-edges of the original hyperedge. See Fig. 10. Any such refinement is a form of hyperedge deletion. A *total* refinement removes all the original b-edges from the b–r-cycle, and replaces them with chords parallel to each of the r-edges. These total refinements give the closest correspondence with our definition of edge deletion. Again, see Fig. 10.

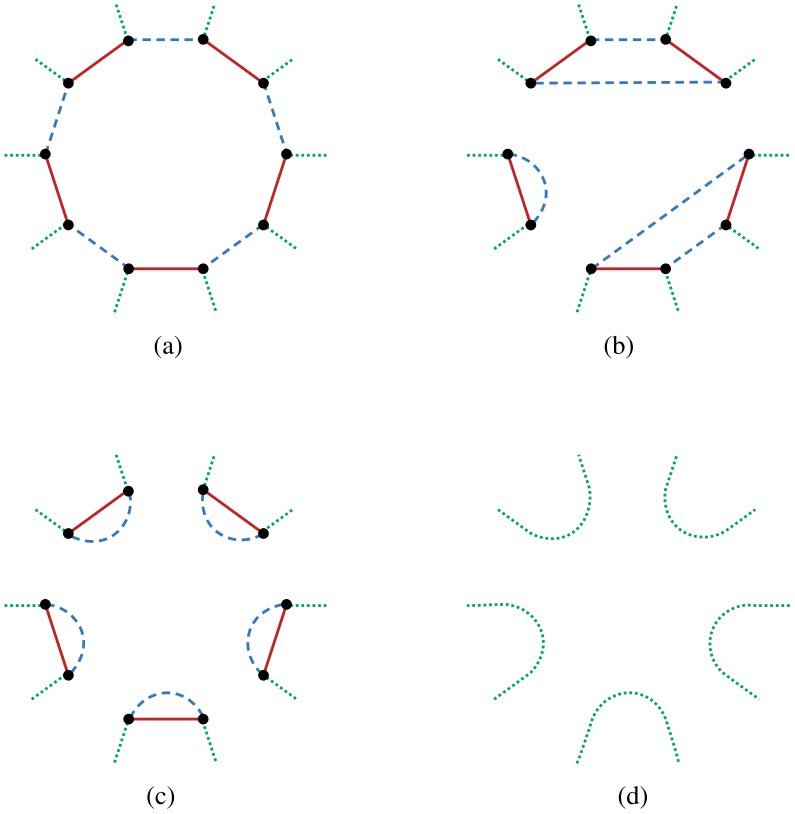


Fig. 10 Comparing edge refinements and deletion. (**a**) A hyperedge in a gehm. (**b**) A refinement of the hyperedge. (**c**) A total refinement of the hyperedge. (**d**) Deleting the hyperedge

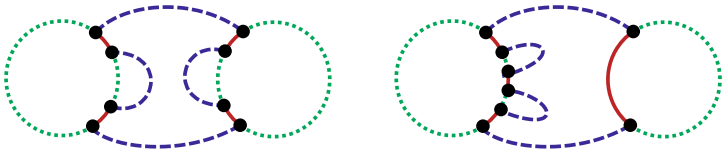
The Whitney polynomial for hypergraphs given by Cori and Hetyei, rewritten in our framework, is

$$R(\mathbb{H}; u, v) = u^{-k(\mathbb{H})} v^{d(\mathbb{H})-v(\mathbb{H})} \sum_{\beta} (uv)^{k(\mathbb{H}_\beta)} v^{-e(\mathbb{H}_\beta)}.$$

By rephrasing the definition in this way $R(\mathbb{H}; u, v)$ extends immediately to nonorientable hypermaps. Here, the sum is over all possible hypermap refinements β, where β gives a choice of refinement for each of the hyperedges, and $\mathbb{H}_\beta$ is the resulting gehm.

Due to the edges of degree one retained in the total refinements, it also does not seem possible to recover the polynomial Z from the polynomial R by restricting the sum in R to the total refinements.

The difference between R and Z is also apparent even in the case of gems where there is a one-to-one correspondence between the refinements defining R and the



(a) (b)

Fig. 11 Two gehms with the same Z but different R. (**a**) $\mathbb{H}_1$. (**b**) $\mathbb{H}_2$

deletions defining Z. For gems, R coincides with the classical Whitney polynomial of the underlying graph, as noted in [11], and thus it does not retain topological information for maps. However, Z coincides with the Tutte polynomial of maps, and thus does encode the topological information. For example, R does not distinguish between two loops on a sphere and on a torus, while Z does. On the other hand, the two gehms shown in Fig. 11 satisfy $Z(\mathbb{H}_1; x, y) = Z(\mathbb{H}_2; x, y) = v^2 + u^3v^3$, but

$$R(\mathbb{H}_1; u, v) = u(1 + 2v + v^2) + (4 + 5v + v^2)$$

and

$$R(\mathbb{H}_2; u, v) = u(1 + 3v + v^2) + (3 + 5v + v^2).$$

5 Concluding Remarks

There are now of course many possible directions for exploring and applying these analogues of the dichromatic, Tutte, and Whitney polynomials for hypermaps. Among them is the question of computational complexity, and we close with a brief discussion of some complexity issues.

In Sect. 4.1 we observed that if $\mathbb{H}$ represents a graph G embedded in the plane, then $T(\mathbb{H}; x, y) = T(G; x, y)$. Vertigan proved in [31] that for a fixed rational point (x, y) it is #P-hard to evaluate the Tutte polynomial $T(G; x, y)$ of a planar graph G except when $(x - 1)(y - 1) \in \{1, 2\}$ or $(x, y) \in \{(-1, -1), (1, 1)\}$. It follows immediately that for a fixed rational point (x, y) it is #P-hard to evaluate the Tutte polynomial $T(\mathbb{H}; x, y)$ of a gehm $\mathbb{H}$ except possibly when $(x-1)(y-1) \in \{1, 2\}$ or $(x, y) \in \{(-1, -1), (1, 1)\}$. The complexity of evaluating $T(\mathbb{H}; x, y)$ at the exceptional points is unclear. For example, in Proposition 7, we observed that if $\mathbb{H}$ is connected and has genus 0, then $T(\mathbb{H}; 1, 1) = t(\mathbb{H})$. Using a straightforward reduction from Exact Cover by 3-Sets [18], it is not difficult to show that deciding whether $t(\mathbb{H}) > 0$ for arbitrary hypermaps is NP-complete, but without specializing this result to hypermaps with genus zero, which seems far

from straightforward, it is not possible to derive any hardness result concerning $T(\mathbb{H}; 1, 1)$, even for arbitrary hypermaps. Therefore we pose the following question.

Question 1 Determine the complexity of computing $T(\mathbb{H}; 1, 1)$.

Given that most evaluations of $T(\mathbb{H}; x, y)$ are #P-hard it is natural to look for appropriate parameters around which one may construct a fixed parameter tractable algorithm. Ultimately one might hope to extend Makowsky's very general approach [25] for graph polynomials expressible in Monadic Second Order Logic to the hypermap setting.

Acknowledgments Part of this work was undertaken while the authors were at the *MATRIX Workshop on Uniqueness and Discernment in Graph Polynomials*. We would like to thank MATRIX for providing a productive and inspiring environment.

Data No underlying data is associated with this article.

Competing Interests There are no conflicts of interest.

References

1. Battiston, F., Amico, E., Barrat, A., et al.: The physics of higher-order interactions in complex systems. Nat. Phys. **17**, 1093–1098 (2021)
2. Battiston, F., Cencetti, G., Iacopini, I., Latora, V., Lucas, M., Patania, A., Young, J.-G., Petri, G.: Networks beyond pairwise interactions: structure and dynamics. Phys. Rep. **874**, 1–92 (2020)
3. Bernardi, O., Kálmán, T., Postnikov, A.: Universal Tutte polynomial. Adv. Math. **402**, 74 (2022). Id/No 108355
4. Bollobás, B., Riordan, O.: A polynomial of graphs on surfaces. Math. Ann. **323**(1), 81–96 (2002)
5. Bondy, J.A., Murty, U.S.R.: Graph Theory. Graduate Texts in Mathematics, vol. 244. Springer, Berlin (2008)
6. Bonnington, C.P., Little, C.H.C.: The Foundations of Topological Graph Theory. Springer, New York (1995)
7. Bradde, S., Bianconi, G.: The percolation transition in correlated hypergraphs. J. Stat. Mech. Theory Exp. **2009**(7), 9 (2009). Id/No p07028
8. Chmutov, S.: Topological extensions of the Tutte polynomial. In: Handbook of the Tutte Polynomial and Related Topics, pp. 497–513. CRC Press, Boca Raton (2022)
9. Chmutov, S., Vignes-Tourneret, F. (2022). Partial duality of hypermaps. Arnold Math. J. **8**(3–4), 445–468 (2022)
10. Cori, R., Hetyei, G.: Spanning hypertrees, vertex tours and meanders (2022). arXiv:2110.00176
11. Cori, R., Hetyei, G.: A Whitney polynomial for hypermaps (2023). arXiv:2311.06662
12. Ellis-Monaghan, J., Noble, S., Moffatt, I.: Tutte polynomials of hypermaps and Farr's invariants of alternating dimaps (in preparation, 2025)
13. Ellis-Monaghan, J.A., Goodall, A.J., Moffatt, I., Noble, S.D., Vena, L.: Irreducibility of the Tutte polynomial of an embedded graph. Algebra Combin. **5**(6), 1337–1351 (2022)

14. Ellis-Monaghan, J.A., Moffatt, I.: Twisted duality for embedded graphs. Trans. Am. Math. Soc. **364**(3), 1529–1569 (2012)
15. Ellis-Monaghan, J.A., Moffatt, I.: Graphs on Surfaces. Dualities, Polynomials, and Knots. SpringerBriefs in Mathematics. Springer, New York (2013)
16. Ellis-Monaghan, J.A., Moffatt, I.: The Tutte polynomial for graphs. In: Handbook of the Tutte Polynomial and Related Topics, pp. 14–26. CRC Press, Boca Raton (2022)
17. Ellis-Monaghan, J.A., Sarmiento, I.: Generalized transition polynomials. Congr. Numerantium **155**, 57–69 (2002)
18. Garey, M.R., Johnson, D.S.: Computers and intractability. A guide to the theory of NP-completeness, W. H. Freeman & Co Ltd (1979)
19. Grimmett, G.: Potts models and random-cluster processes with many-body interactions. J. Stat. Phys. **75**(1–2), 67–121 (1994)
20. Huggett, S., Moffatt, I.: Types of embedded graphs and their Tutte polynomials. Math. Proc. Cambridge Philos. Soc. **169**(2), 255–297 (2020)
21. Jaeger, F.: On transition polynomials of 4-regular graphs. Cycles and rays: basic structures in finite and infinite graphs. In: Proceedings of NATO Advanced Research Workshop, Montréal/Canada 1987. NATO ASI Series, Series C, vol. 301, pp. 123–150 (1990)
22. Kálmán, T.: A version of Tutte's polynomial for hypergraphs. Adv. Math. **244**, 823–873 (2013)
23. Krajewski, T., Moffatt, I., Tanasa, A.: Hopf algebras and Tutte polynomials. Adv. Appl. Math. **95**, 271–330 (2018)
24. Lins, S.: Graph-encoded maps. J. Comb. Theory Ser. B **32**, 171–181 (1982)
25. Makowsky, J.A.: Coloured Tutte polynomials and Kauffman brackets for graphs of bounded tree width. Discrete Appl. Math. **145**(2), 276–290 (2005)
26. Makowsky, J.A.: From a zoo to a zoology: Towards a general theory of graph polynomials. Theory Comput. Syst. **43**(3–4), 542–562 (2008)
27. Makowsky, J.A., Ravve, E.V., Kotek, T.: A logician's view of graph polynomials. Ann. Pure Appl. Logic **170**(9), 1030–1069 (2019)
28. Moffatt, I.: The Tutte polynomial of a graph embedded in a surface. 2023 MATRIX Annals, Springer (2025)
29. Smith, B.: Matroids, Eulerian Graphs and Topological Analogues of the Tutte Polynomial. Ph.D. Thesis, Royal Holloway, University of London (2018)
30. Tutte, W.T.: Duality and trinity. In: Infinite and Finite Sets (Colloquium, Keszthely, 1973; Dedicated to P. Erdős on his 60th birthday), Vols. I, II, III. Colloquia Mathematica Societatis János Bolyai, vol. 10, pp. 1459–1472. North-Holland, Amsterdam (1975)
31. Vertigan, D.: The computational complexity of Tutte invariants for planar graphs. SIAM J. Comput. **35**(3), 690–712 (2006)

Graph Polynomials: Some Questions on the Edge

Graham Farr and Kerri Morgan

Abstract We raise some questions about graph polynomials, highlighting concepts and phenomena that may merit consideration in the development of a general theory. Our questions are mainly of three types: When do graph polynomials have reduction relations (simple linear recursions based on local operations), perhaps in a wider class of combinatorial objects? How many levels of reduction relations does a graph polynomial need in order for it to be expressed in terms of trivial base cases? For a graph polynomial, how are properties such as equivalence and factorisation reflected in the structure of a graph? We illustrate our discussion with a variety of graph polynomials and other invariants. This leads us to reflect on the historical origins of graph polynomials. We also introduce some new polynomials based on partial colourings of graphs and establish some of their basic properties.

1 Introduction

János Makowsky started his research career in mathematical logic over half a century ago. For the last two decades, he has brought many concepts and results from that field—along with the perspective it offers—to bear on the study of graph polynomials. This has led to significant new theorems, links between topics, fresh viewpoints, and deeper understanding.

His contributions to graph polynomials, with a wide set of collaborators, include: complexity classifications for specific computational problems [9, 70, 72, 78] (see also [64]); refining the models of computation used to study the computational complexity of problems concerning graph polynomials [65, 79]; introducing formal logical definitions of graph polynomials, especially using second-order logic (SOL)

G. Farr (✉)
Department of Data Science and AI, Faculty of IT, Monash University, Clayton, VIC, Australia
e-mail: Graham.Farr@monash.edu

K. Morgan
School of Science (Mathematical Sciences), RMIT University, Melbourne, VIC, Australia
e-mail: kerri.morgan@rmit.edu.au

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_18

and variants thereof, and using concepts and tools from logic to develop their theory at a very general level [18, 48, 67, 68, 71, 73]; developing a general framework for studying the distinguishing power of graph polynomials [66, 80, 81]; and showing that the locations of zeros of a graph polynomial is "not a semantic property", in that it derives more from the algebraic form of the polynomial than from the way it partitions the set of all graphs into equivalence classes [81, 82]. Some of his own reflections on these topics may be found in [76, 77, 82].

A pervasive characteristic of his work has been to put specific graph polynomials in a wider mathematical context—to see the forest as well as the trees, to quote his own quotation of Einstein [75]. One manifestation of this has been his advocacy for the development of a general theory of graph polynomials and his own work in that direction [73, 74, 82]. He was a co-organiser of the Dagstuhl Seminar on "Graph Polynomials: Towards a Comparative Theory" in 2016 [24]. At the Dagstuhl Seminar on "Comparative Theory for Graph Polynomials" in 2019 [25], he helped lead an informal group working on the distinguishing power of graph polynomials.

In this paper we propose some polynomials, topics and questions that may merit further consideration in the development of a general theory of graph polynomials. To do this, we journey to the edge of the territory covered by the current theory, meeting some polynomials that seem to lie near, or beyond, that frontier, as well as some that are very familiar and are well covered by the theory.

We begin with some general definitions and notation in Sect. 2 and set the scene in Sect. 3 by defining all the polynomials we will discuss. This includes the introduction of some new graph polynomials related to partial colourings. In Sect. 4 we reflect on the origins of graph polynomials. Some of the polynomials we introduce are then used to illustrate the questions raised in the next four sections. In Sect. 5 we consider the widespread phenomenon of graph polynomials having reduction relations (i.e., simple recursive relations based on local modifications), pointing out that even graph polynomials that do not seem to have such a relation will often be found to have one within a wider class of objects. In Sect. 6 we discuss a notion of "levels" in these reduction relations, where a graph polynomial can be reduced to a large sum of polynomials of reduced objects of some kind, which in turn may each be reduced to another large sum using another relation, and so on. In Sect. 7 we discuss the relationship between the algebraic properties of a graph polynomial and the structure of the graph. In Sect. 8 we pose questions about certificates, a tool for studying equivalence and factorisation of graph polynomials.

Some of the material in this paper was presented by the first author in a talk of the same title at the Dagstuhl Seminar on "Graph Polynomials: Towards a Comparative Theory" in June 2016 [24].

2 Definitions and Notation

Throughout, $G = (V, E)$ is a graph with n vertices and m edges. The number of components of G is denoted by $k(G)$. If $X \subseteq E$ then $V(X)$ denotes the set of

vertices of G that are incident with at least one edge in X (overloading the $V(\_)$ notation slightly). We write $\nu(X)$ for the number of vertices of G that meet an edge in X (succinctly: $\nu(X) = |V(X)|$). The number of components of (V, X) is denoted by $k(X)$, while the number of components of $(V(X), X)$ is denoted by $\kappa(X)$. The former count includes isolated vertices, while the latter count excludes them: $k(X) = \kappa(X) + n - \nu(X)$. We write $\rho(X)$ for the *rank* of X, given by $\rho(X) = \nu(X) - \kappa(X) = |V| - k(X)$, and $\rho(G) := \rho(E(G))$. For any function $r : 2^E \to \mathbb{R}$, its *dual* r^* is defined by $r^*(X) = |X| + r(E \setminus X) - r(E) + r(\emptyset)$. When ρ is the rank function of G (and therefore the rank function of its cycle matroid), ρ^* is the rank function of the dual of the cycle matroid of G.

If $U \subseteq V$ then $G[U]$ is the subgraph of G induced by U. If $G[U]$ has no edges (i.e., no two vertices in U are adjacent) then U is *stable* or *independent*.

The disjoint union of two graphs G and H is denoted by $G \sqcup H$.

If $e \in E$ then $G \setminus e = (V, E \setminus \{e\})$ is the graph obtained from G by deleting edge e and G/e is the graph obtained by contracting edge e, i.e., deleting e and then identifying its former endpoints. If $u, v \in V$ with $uv \notin E$ then $G + uv = (V, E \cup \{uv\})$ is the graph obtained by adding an edge between u and v in G and G/uv is obtained from G by identifying vertices u and v.

A *coloop* of G is an edge e such that $k(G \setminus e) > k(G)$. This is often called a *bridge* and sometimes an *isthmus*.

The maximum degree of G is denoted by $\Delta(G)$.

A *null graph* is a graph with no edges.

A *graph invariant* is a function defined on all graphs that depends only on the isomorphism class of the graph.

We write $[k] = \{1, 2, \ldots, k\}$.

The falling factorial $x(x-1)\cdots(x-k+1)$ is denoted by $(x)_k$.

Let Λ be a set whose members we will call *colours*, and let $\lambda \in \mathbb{N}$. A *Λ-assignment* of G is a function $f : V \to \Lambda$. A *λ-assignment* is a $[\lambda]$-assignment. A *partial Λ-assignment* is a function $f : W \to \Lambda$ where $W \subseteq V$. The vertices of W and $V \setminus W$ are *coloured* and *uncoloured*, respectively, by f. A *partial λ-assignment* is a partial $[\lambda]$-assignment. For every $k \in \Lambda$, its *colour class* $C(k) = C_f(k)$ under a partial Λ-assignment f is given by $C(k) = f^{-1}(k) = \{v \in V : f(v) = k\}$. Every colour class $C(k)$ induces a subgraph $G[C(k)]$ of G. A *chromon*, or *monochromatic component*, of (G, f) is a component of $G[C(k)]$ for some $k \in [\lambda]$.

Every partial λ-assignment f is determined by its λ-tuple $(C_f(i))_{i=1}^{\lambda}$ of colour classes: given a λ-tuple $(C_i)_{i=1}^{\lambda}$ of mutually disjoint subsets of V, we can define a partial λ-assignment $f : \bigcup_{i=1}^{\lambda} C_i \to [\lambda]$ by $f(v) = i$ for each $v \in C_i$ and $i \in [\lambda]$; this then satisfies $C_f(i) = C_i$ for all $i \in [\lambda]$.

A colour class is *proper* if it is a stable set in G. A partial λ-assignment is a *partial λ-colouring* if every colour class is proper; this is regardless of whether or not the partial λ-assignment can be extended to a λ-colouring of G.

An *extension* of a partial λ-assignment f is a partial λ-assignment g such that $f \subseteq g$, which is equivalent to requiring that $\operatorname{dom} f \subseteq \operatorname{dom} g$ and $f(v) = g(v)$ for all $v \in \operatorname{dom} f$.

3 Some Graph Polynomials

Papers by János often include observations about collections of specific graph polynomials as motivation for studying more general phenomena. In a similar spirit, we now discuss an eclectic collection of graph invariants—some old, some new; mostly polynomials, some not—to help motivate some of the questions we ask. The reader can skip those sections treating graph polynomials with which they are familiar.

More comprehensive collections of graph polynomials may be found in [27, 108].

3.1 Chromatic Polynomials and Some Generalisations

The *chromatic polynomial* $P(G; q)$ of a graph G was introduced by Birkhoff [8]. It gives, for each $q \in \mathbb{N}$, the number of q-colourings of G, and is easily shown to be a polynomial, so it extends to all $q \in \mathbb{C}$. It satisfies the *deletion-contraction relation*: for any $e \in E(G)$,

$$P(G; q) = P(G \setminus e; q) - P(G/e; q). \tag{1}$$

If e is a loop then $P(G; q) \equiv 0$. Otherwise, if the endpoints of e have a common neighbour then contraction creates parallel edges. But these extra edges can be removed (leaving just one edge representing each set of parallel edges) when using (1) to compute the chromatic polynomial. (Note, though, that in other contexts, such as for Tutte-Whtiney polynomials (Sect. 3.2), this removal of parallel edges cannot in general be done when doing contraction.)

Two graphs are *chromatically equivalent* if they have the same chromatic polynomial.

For further information on the chromatic polynomial, see [23].

The chromatic polynomial has been generalised in many different ways. A generalisation by Harary [54] takes any class $\mathscr{P}$ of graphs and defines $P_{\mathscr{P}}(G; q)$ to be the number of q-assignments of G such that each colour class induces a member of $\mathscr{P}$. Again, this is a polynomial in q. The chromatic polynomial is the special case when $\mathscr{P}$ is the set of graphs with no edges.

The class of these *Harary polynomials* is very general and has been prominent in work by Makowsky and collaborators in developing the theory of graph polynomials [58, 67, 68].

We consider another—and much older—generalisation of the chromatic polynomial next.

3.2 Tutte-Whitney Polynomials

The preeminent graph polynomial is arguably the Tutte polynomial, due to W.T. Tutte [111, 112] and closely related to the Whitney rank generating function [120].

The *Whitney rank generating function* $R(G; x, y)$ of a graph G is the bivariate polynomial defined by

$$R(G; x, y) = \sum_{X \subseteq E} x^{\rho(E)-\rho(X)} y^{\rho^*(E)-\rho^*(E \setminus X)} = \sum_{X \subseteq E} x^{\rho(E)-\rho(X)} y^{|X|-\rho(X)}$$

The *Tutte polynomial* $T(G; x, y)$ may be defined by

$$T(G; x, y) = \begin{cases} 1, & \text{if } G \text{ has no edges;} \\ x\, T(G \setminus e; x, y), & \text{if } e \text{ is a coloop in } G; \\ y\, T(G/e; x, y), & \text{if } e \text{ is a loop in } G; \\ T(G \setminus e; x, y) + T(G/e; x, y), & \text{otherwise,} \end{cases} \tag{2}$$

for any edge $e \in E(G)$. The polynomials of Whitney and Tutte are related by a simple coordinate translation: $T(G; x, y) = R(G; x - 1, y - 1)$ [111, 112].

The recurrence (2) is the most fundamental property of the Tutte polynomial, and one of Tutte's major conceptual contributions in [111] was to base the theory on this relation, a fundamental conceptual advance upon the pioneering work of Whitney [120]. For a history of these polynomials, see [40].

A graph G is *Tutte equivalent* to a graph H if $T(G; x, y) = T(H; x, y)$ [113, 120, 121]. Graphs with isomorphic cycle matroids are Tutte equivalent, since the Tutte polynomial of a graph depends only on its cycle matroid. But, as an equivalence relation on graphs, Tutte equivalence is coarser than cycle matroid isomorphism. Tutte [113] gave two graphs, due to M.C. Gray, which are not isomorphic, and do not even have isomorphic cycle matroids, but which have the same Tutte polynomial: see Fig. 1. Many other such pairs are known.

The most famous specialisation of the Tutte polynomial is the chromatic polynomial $P(G; q) = (-1)^{\rho(G)} q^{k(G)} T(G; -q + 1, 0)$.

Tutte-Whitney polynomials have been generalised from graphs to many other combinatorial objects [27, 34]. For structures on which deletion and contraction operations exist, Krajewski, Moffatt and Tanasa [69] show how to use Hopf algebras

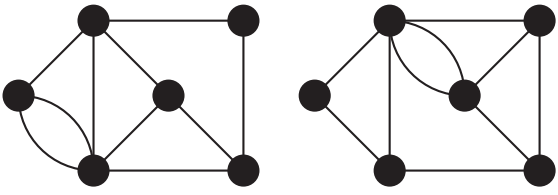


Fig. 1 The two Gray graphs, from [113]

to define a polynomial that may reasonably be called the Tutte polynomial for those structures and which satisfies a deletion-contraction relation.

3.3 Some Partition Functions

We introduce the partition functions of three interaction models on graphs: the Ising, Potts and Ashkin-Teller models. Our main focus later will be on the Ashkin-Teller model.

The set of edges of a graph G whose endpoints receive the same colour under a q-assignment f is denoted by $E^+(f)$; these edges are sometimes called *bad* since they are not properly coloured in the sense of graph colouring. The set of edges whose endpoints are differently coloured under f is denoted by $E^-(f)$; these edges are sometimes called *good*. Note that $E^+(f) \cup E^-(f) = E$. So the positive and negative signs in superscripts here represent "bad" and "good", respectively, which is the opposite of their connotations in ordinary English usage.

With this notation, we may write the chromatic polynomial as

$$P(G;q) = \sum_{f:V\to[q]} 0^{|E^+(f)|}1^{|E^-(f)|}$$

where 0^k is taken to be 1 if $k = 0$ and 0 otherwise, so that only proper colourings contribute to the sum, with all proper colourings counted once.

Suppose now that we do not penalise improper colourings so drastically, but just weight colourings according to an exponential function of the number of good edges they have. This gives the *Potts model*, introduced in [101] and generalising the $q = 4$ case introduced by Ashkin and Teller in [2]. (See [40, §34.12]. The model is called the Ashkin-Teller-Potts model in [118, §4.4].) The *Potts model partition function* is given by[1]

$$Z_{\text{Potts}}(G;K,q) = \sum_{f:V\to[q]} e^{-K\cdot|E^-(f)|}.$$

This is a polynomial in e^{-K} and is known to be a partial evaluation of the Tutte polynomial [28, 46]. The relationship between them can be expressed more simply in terms of the Whitney rank generating function:

$$Z_{\text{Potts}}(G;K,q) = q^{k(G)}\,(e^K-1)^{\rho(G)}e^{-K|E|}R(G;\frac{q}{e^K-1},e^K-1).$$

[1] Sometimes, it is defined instead as $\sum_{f:V\to[q]} e^{K\cdot|E^+(f)|}$, which may be written $e^{K|E|}\sum_{f:V\to[q]} e^{-K\cdot|E^-(f)|}$. So the only change is an extra prefactor $e^{K|E|}$. See, e.g., [5, §2].

The $q = 2$ case of the Potts model partition function is mathematically almost the same as the Ising model partition function [61]. In the Ising model, colours take values in $\{\pm 1\}$, and we sum over all $\{\pm 1\}$-assignments $\sigma : V \to \{\pm 1\}$. For a given $\sigma : V \to \{\pm 1\}$, the edge uv belongs to the set $E^{\sigma(u)\sigma(v)}(f)$. The *Ising model partition function* is

$$\begin{aligned} Z_{\text{Ising}}(G; K) &= \sum_{\sigma: V \to \{\pm 1\}} e^{K \sum_{uv \in E} \sigma(u)\sigma(v)} \\ &= \sum_{\sigma: V \to \{\pm 1\}} e^{K \cdot |E^+(\sigma)| - K \cdot |E^-(\sigma)|} \\ &= e^{K|E|} \sum_{\sigma: V \to \{\pm 1\}} e^{-2K \cdot |E^-(\sigma)|} \\ &= e^{K|E|} \, Z_{\text{Potts}}(G; 2K, 2). \end{aligned}$$

The Ashkin-Teller model [2] extends the Ising model and the $q = 4$ case of the Potts model. Each vertex $v \in V$ has a pair of colours $\sigma(v)$ and $\tau(v)$, each taking one of the two values ± 1, so the available colour pairs $(\sigma(v), \tau(v))$ for each vertex are the four pairs $(\pm 1, \pm 1)$. We think of the $\{\pm 1\}$-assignments $\sigma : V \to \{\pm 1\}$ and $\tau : V \to \{\pm 1\}$ as two Ising spin functions on G. An example is given in Fig. 2, where the spins at upper left and lower right of each vertex are those assigned by σ and τ, respectively. These spins are also shown as colours on the left and right semicircles in each vertex.

These yield a third such function, their product $\sigma\tau : V \to \{\pm 1\}$, given for each $v \in V$ by $(\sigma\tau)(v) = \sigma(v)\tau(v)$. So we have three Ising spin functions, *coupled* in the sense that they are not all independent: we can regard any two of them as independent, but together they determine the third. Each of them gives its

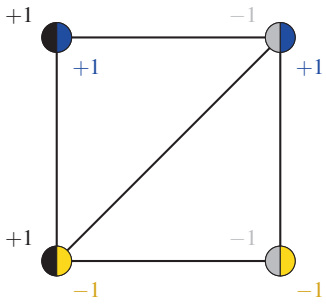


Fig. 2 An Ashkin-Teller model configuration (σ, τ) for a graph

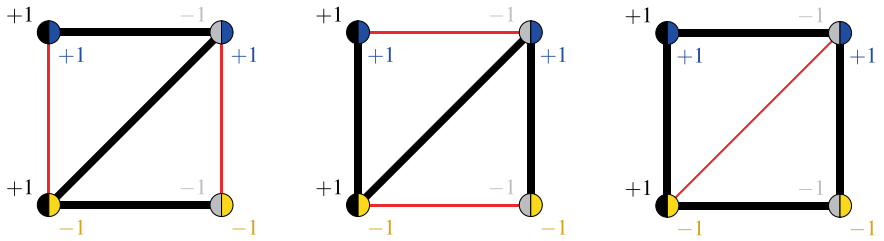


Fig. 3 Good (thick, black) and bad (thin, red) edges for the three Ising-type configurations in the Ashkin-Teller model configuration in Fig. 2

own division of $E(G)$ into good and bad edges, so we have three such divisions altogether:

spin	edge type	
	good	**bad**
σ	$E^-(\sigma)$	$E^+(\sigma)$
τ	$E^-(\tau)$	$E^+(\tau)$
$\sigma\tau$	$E^-(\sigma\tau)$	$E^+(\sigma\tau)$

We illustrate these divisions, for the graph and configuration of Fig. 2, in Fig. 3.

The *partition function of the symmetric Ashkin-Teller model* is given by

$$Z_{\mathrm{SymAT}}(G;K,K') = e^{(2K+K')|E|} \sum_{\sigma,\tau:V\to\{\pm1\}} e^{-2K\cdot|E^-(\sigma)|-2K\cdot|E^-(\tau)|-2K'\cdot|E^-(\sigma\tau)|}.$$

The symmetry is because the coefficients of $|E^-(\sigma)|$ and $|E^-(\tau)|$ in the exponent are the same, namely K. In the general four-state Ashkin-Teller model, these can be different, and the partition function depends on three variables instead of the two (K, K') we have here.

If $K' = 0$ then $Z_{\mathrm{SymAT}}(G;K,K')$ is just the square of an Ising model partition function:

$$Z_{\mathrm{SymAT}}(G;K,0) = Z_{\mathrm{Ising}}(G;K)^2.$$

If $K = K'$ then $Z_{\mathrm{SymAT}}(G;K,K')$ is just a Potts model partition function up to a simple factor:

$$Z_{\mathrm{SymAT}}(G;K,K) = e^{-3K|E|} Z_{\mathrm{Potts}}(G;4K,4).$$

In these two cases, the symmetric Ashkin-Teller model partition function can be obtained from the Tutte polynomial, since the Ising and Potts model partition functions can be so obtained. But this is not possible in general. Direct evaluation shows that the two Tutte equivalent Gray graphs (Fig. 1) have different symmetric Ashkin-Teller model partition functions. From this we see that the symmetric Ashkin-Teller model partition function is not a specialisation of the Tutte polynomial.

3.4 Interpolating Between Contraction and Deletion

We can represent any subset $X \subseteq E$ of the edge set of a graph by its *characteristic vector* $\mathbf{x} = (x_e)_{e \in E} \in \mathrm{GF}(2)^E$ defined by

$$x_e = \begin{cases} 1, \text{ if } e \in E, \\ 0, \text{ if } e \notin E. \end{cases}$$

A *cocircuit* of G is a minimal set of edges whose removal does not disconnect any component of G. These are also the circuits of the dual of the cycle matroid of G. The *cocircuit space* of G is the linear space over GF(2) generated by the characteristic vectors of cocircuits of G. This is also the row space of the binary incidence matrix of G, which has rows indexed by V, columns indexed by E, and each entry is 1 or 0 according as its vertex is, or is not, incident with its edge.

In [30], the operations of contraction and deletion are applied directly to the indicator functions of cocircuit spaces. Let $f : 2^E \to \{0, 1\}$ be the indicator function of the cocircuit space of G. Let $f/\!\!/e : 2^E \to \{0, 1\}$ and $f \setminus\!\!\setminus e : 2^E \to \{0, 1\}$ be the indicator functions of G/e and $G \setminus e$, respectively. Then it is shown in [30] that

$$(f/\!\!/e)(X) = \frac{f(X)}{f(\emptyset)}, \qquad (f \setminus\!\!\setminus e)(X) = \frac{f(X) + f(X \cup \{e\})}{f(\emptyset) + f(\{e\})}.$$

This is used in two related generalisations. Firstly, in [30], contraction and deletion are extended to arbitrary $f : 2^E \to \mathbb{R}$ satisfying $f(\emptyset) = 1$; in later work [35, 36, 38] it was convenient to include all $f : 2^E \to \mathbb{C}$. Such functions f are called *binary functions*, and if f is the indicator function of the cocircuit space of a graph then it is *graphic*. Secondly, in [33], contraction and deletion are considered to be just two specific reductions in a whole family of λ-*reductions* parameterised by $\lambda \in \mathbb{R}$, and later by $\lambda \in \mathbb{C}$ in [35, 36, 38], using the notation and expression in the middle column, where $e \in E$ and $X \subseteq E \setminus \{e\}$:

Contraction ($\lambda = 0$)	λ-reduction	Deletion ($\lambda = 1$)
$(f/\!/e)(X)$	$(f \parallel_\lambda e)(X)$	$(f \setminus\!\setminus e)(X)$
$\dfrac{f(X)}{f(\emptyset)}$	$\dfrac{f(X) + \lambda f(X \cup \{e\})}{f(\emptyset) + \lambda f(\{e\})}$	$\dfrac{f(X) + f(X \cup \{e\})}{f(\emptyset) + f(\{e\})}$

Ordinary contraction and deletion are given by $\lambda = 0$ and $\lambda = 1$, respectively. When $\lambda \in [0, 1]$, one can think of the λ-reduction as interpolating between contraction and deletion. These λ-reductions come in dual pairs, with the dual of λ-reduction being λ^*-reduction where

$$\lambda^* = \frac{1-\lambda}{1+\lambda}.$$

When $\lambda \notin \{0, 1\}$, a λ-reduction of a graphic binary function is not, in general, graphic.

These dual λ-reduction operations are accompanied by parameterised versions of the rank function and Whitney rank generating function. The λ-*rank* function $Q^{(\lambda)} f$ is defined by

$$(Q^{(\lambda)} f)(X) = \log_2 \left(\frac{(1+\lambda^*)^{|V|} \sum_{W \subseteq E} \lambda^{|W|} f(W)}{\sum_{W \subseteq E} \lambda^{|W \cap (E \setminus X)|} (\lambda^*)^{|W \cap X|} f(W)} \right),$$

where $X \subseteq E$. The λ-*Tutte-Whitney function* is defined by

$$R_1^{(\lambda)}(G; x, y) = R_1^{(\lambda)}(f; x, y) = y^{-Q^{(\lambda)} f(E)} \sum_{X \subseteq E} (xy)^{Q^{(\lambda)} f(E) - Q^{(\lambda)} f(X)} y^{|X|}.$$

These functions are shown in [33] to satisfy a generalisation of the contraction-deletion relation of the same form as for Tutte-Whitney polynomials, with λ-reductions and their dual λ^*-reductions taking the place of contraction and deletion. The *loopiness* and *coloopiness* of the element $e \in S$ under the function f are defined by the functions

$$\begin{aligned} \mathrm{loop}^{(\lambda)}(f, e) &= Q^{(\lambda^*)} f(E) - Q^{(\lambda^*)} f(E \setminus e), \\ \mathrm{coloop}^{(\lambda)}(f, e) &= Q^{(\lambda)} f(E) - Q^{(\lambda)} f(E \setminus e) \;=\; \mathrm{loop}^{(\lambda^*)}(f, e). \end{aligned}$$

Theorem 1 ([33]) For any binary function $f : 2^E \to \mathbb{C}$ and any $e \in E$,

$$R^{(\lambda)}(f; x, y) = x^{\mathrm{coloop}^{(\lambda)}(f,e)} R^{(\lambda)}(f \parallel_\lambda e; x, y) + y^{\mathrm{loop}^{(\lambda)}(f,e)} R^{(\lambda)}(f \parallel_{\lambda^*} e; x, y).$$

3.5 Go Polynomials

János is a keen Go player. He and the first author have played several times; as the stronger player, János plays with the white stones and gives a handicap. So it is fitting to mention some surprising points of contact between graph polynomials and the theory of Go.

Here we use the name by which the game is known in Japan and in the West. It is called *Wéiqí* in China and *Baduk* in Korea.

Go is normally played on the square grid graph of 19×19 vertices. But it has long been recognised that Go is an entirely graph-theoretic game and can be played on any graph. It has been played, for example, on a three-dimensional diamond lattice graph [103] and on a map of Milton Keynes [60].

Thorpe and Walden [106, 107] seem to have been among the first to formalise the rules and concepts of Go in order to support mathematical and computational investigation. Their work uses some graph-theoretic concepts but is still embedded in rectangular grid graphs. Benson [6] uses graph-theoretic language and concepts to characterise unconditional life of groups of stones. Tromp and Taylor [104, 110] defined "logical rules" for Go, based on rules from the New Zealand Go Society. These rules are concise, precise, elegant, and purely graph-theoretic. Although they state initially that Go is played on a 19×19 square grid, these rules may be used with only cosmetic modifications to play on any graph.

Let G be a graph and f be a partial λ-assignment of G. A chromon of (G, f) is *free* if it has a vertex that is adjacent to an uncoloured vertex. A partial λ-assignment f is a *legal* λ*-position* in G, or just a *legal position* if λ is clear from the context, if every chromon is free.

Normally, Go is played with just two colours, Black and White, so $\lambda = 2$, although multiplayer versions with $\lambda > 2$ have been proposed and equipment (in the form of coloured Go stones) is available for them.

In Go terminology, chromons correspond to what a Go player might call *chains*. A chromon is free if it is a chain with at least one liberty, and a legal position is one in which every chain has a liberty. When playing the game according to the rules, every position will be a legal position except during the brief intervals after a capture is made and before all the captured stones are removed from the board. We do not discuss the rules of Go further; see, e.g., [62] for more information.

Two *Go polynomials* based on these concepts were introduced in [32], studied further in [37, 39, 42], and mentioned by Makowsky in his first inventory of the Zoo [73, 74]. One of them simply counts legal positions:

$$\mathrm{Go}^{\#}(G; \lambda) = \text{number of legal } \lambda\text{-positions for } G.$$

For example, it can be shown that $\mathrm{Go}^{\#}(C_4; \lambda) = 1 + 14\lambda^2$. The other Go polynomial from [32] is based on a simple probability model. Let $p \leq \frac{1}{2}$ and construct a random partial 2-assignment f as follows. Each $v \in V$ is assigned colour 1 or 2, with

probability p each, or is left uncoloured with probability $r := 1-2p$, with decisions for different vertices being independent. Under this model, define

$$\mathrm{Go}(G;\, p) = \Pr(f \text{ is a legal 2-position for } G).$$

We could also define a bivariate version of the second polynomial by extending the probability model to λ colours, where f is now a partial λ-assignment, the probability p now satisfies $p \leq \lambda^{-1}$, and f is assigned colour $k \in [\lambda]$ with probability p for each colour and is left uncoloured with probability $r := 1 - \lambda p$:

$$\mathrm{Go}(G;\, p, \lambda) = \Pr(f \text{ is a legal } \lambda\text{-position for } G).$$

All three functions can be shown to be polynomials, so they can take as arguments any λ, $p \in \mathbb{C}$ although we only know of combinatorial interpretations for the values used in the definitions above. This suggests the problem of finding combinatorial interpretations at other values of p and λ. We suggest $\lambda = -1$ as one that might be worth exploring, since the chromatic polynomial has an interesting combinatorial interpretation at $\lambda = -1$, namely the number of acyclic orientations [105], and Go polynomials can be expressed naturally as sums of chromatic polynomials [32].

These Go polynomials are all exponential-time computable, and they seem unlikely to be polynomial-time computable because it was shown in [32, §4] that, for any fixed integer $\lambda \geq 2$, computing the value of $\mathrm{Go}^{\#}(G;\, \lambda)$ for an input graph G is #P-hard, using methods from transcendental number theory.

3.6 Polynomials for Partial Colourings

Given a graph G, a nonnegative integer λ representing some number of available colours, and a probability $p \leq \lambda^{-1}$, put $r := 1 - \lambda p$ and define the following random colouring model, noting that the partial λ-assignment it produces is not necessarily a (proper) colouring.

Each vertex $v \in V(G)$ remains uncoloured with probability r and otherwise is given a colour chosen uniformly at random from the λ available colours. So, for any specific colour, the probability that v gets that colour is p. The choices made at different vertices are independent. This process generates a random partial λ-assignment whose domain is a subset of $V(G)$.

A partial λ-assignment f of G is a *partial λ-colouring* if it is a colouring of $G[\mathrm{dom}\, f]$, the subgraph of G induced by the coloured vertices. In a partial λ-colouring, vertices that are adjacent in G cannot both get the same colour; either they receive different colours or at least one of them is uncoloured.

We say f is *λ-extendable* if there is a λ-colouring of G that extends f.

A vertex $v \notin \mathrm{dom}\, f$ is *immediately λ-forced by f* if its neighbours in $\mathrm{dom}\, f$ have exactly $\lambda - 1$ distinct colours among them, and in this case we write $f; v$ for the partial λ-assignment on $(\mathrm{dom}\, f) \cup \{v\}$ which agrees with f on $\mathrm{dom}\, f$ and gives

v the sole colour from $[\lambda]$ that does not appear among its neighbours. The motivation is that, if vertex v is to receive a colour from $[\lambda]$ without creating any bad edges, then it must be given that one colour that is unused by any of its neighbours. If the number of different colours that appear among the neighbours of v is λ, then there is no hope for v: any colouring of it creates one or more bad edges. If the neighbours of v only have $\leq \lambda - 2$ colours, then the colour of v is not determined by the colours of its neighbours, as there are at least two options for it.

If f immediately λ-forces v, then there is only one possible colour for v in all λ-colourings of G that extend f. But the converse does not hold in general: the colour of v may be uniquely determined without being forced in this specific local sense.

The vertex v is *(eventually) λ-forced by f* if there is a sequence of vertices $v_1, \ldots, v_k = v$, all outside $\operatorname{dom} f$, and a sequence of partial λ-assignments $f = f_0, f_1, \ldots, f_k$ such that, for all $i \in [\lambda]$,

- $\operatorname{dom} f_i = (\operatorname{dom} f) \cup \{v_1, \ldots, v_i\}$,
- v_i is immediately forced by f_{i-1},
- $f_i = f_{i-1}; v_i$.

We may drop λ from "λ-forced" when it is clear from the context.

Note that if f is improper then it is not λ-extendable, and that if f is not λ-extendable then (regardless of whether or not it is proper) it cannot force a λ-colouring of G.

For every graph G we define three bivariate functions based on partial colourings.

The *partial chromatic polynomial* $\mathrm{PC}(G;\, p, \lambda)$ is defined by

$$\mathrm{PC}(G;\, p, \lambda) = \Pr(f \text{ is a } \lambda\text{-colouring of } G[\operatorname{dom} f]),$$

where $p \in [0, \lambda^{-1}]$ is the probability that a specific vertex gets a particular colour, and $\lambda \in \mathbb{N}$. Since $\mathrm{PC}(G;\, p, \lambda)$ is a polynomial (as we will shortly see), its domain is $\mathbb{C}^2$. Observe that $\mathrm{PC}(G; \lambda^{-1}, \lambda) = \lambda^{-n} P(G; \lambda)$ and $\mathrm{PC}(G; 0, \lambda) = 1$. The partial chromatic polynomial is a simple algebraic transformation of the *generalised chromatic polynomial* $P(G; x, y)$ introduced by Dohmen et al. [22]. For $x \in \mathbb{N}$ and $y \in [x]$, the value of $P(G; x, y)$ is the number of $[x]$-assignments for which the first y colour classes are proper. If we divide by the total number x^n of all $[x]$-assignments, then this falls within the random partial colouring framework by putting $p = x^{-1}$, $\lambda = y$ and $r = 1 - x^{-1}y$. So we have

$$P(G; x, y) \;=\; \frac{\mathrm{PC}(G; x^{-1}, y)}{x^n}.$$

The change of variables is easily inverted to obtain

$$\mathrm{PC}(G;\, p, \lambda) \;=\; \frac{P(G;\, p^{-1}, \lambda)}{p^n}.$$

By considering all possibilities for the domain of f, we obtain

$$\mathrm{PC}(G;\, p, \lambda) = \sum_{C \subseteq V} P(G[C];\, \lambda)\, p^{|C|} (1 - \lambda p)^{n-|C|}, \tag{3}$$

which is just a rearrangement of [22, Theorem 1] and shows that $\mathrm{PC}(G;\, p, \lambda)$ is indeed a polynomial. When λ is fixed, we denote the resulting polynomial in p by $\mathrm{PC}_\lambda(G;\, p)$.

The partial chromatic polynomial is also a specialisation of some trivariate graph polynomials that count edge-subsets according to $(|X|, \nu(X), \kappa(X))$ or according to simple invertible transformations of them, e.g., $(|X|, \nu(X), \rho(X))$. Historically, the first of these trivariate polynomials seems to have been the *Borzacchini-Pulito polynomial* [11], given by

$$\begin{aligned} \mathrm{BP}(G;\, x, y, z) &= \sum_{X \subseteq E} x^{|X|} y^{k(X)} z^{\nu(X)} \quad = \quad \sum_{X \subseteq E} x^{|X|} y^{\kappa(X)+n-\nu(X)} z^{\nu(X)} \\ &= \sum_{X \subseteq E} x^{|X|} y^{n-\rho(X)} z^{\nu(X)}. \end{aligned}$$

They gave a ternary reduction relation for it in [11, Theorem 2]. Averbouch, Godlin and Makowsky [3, 4] introduced the *edge elimination polynomial* $\xi(G;\, x, y, z)$ as the most general trivariate graph polynomial that satisfies a ternary reduction relation using edge deletion, contraction and extraction (deletion of an edge, its endpoints and their incident edges) and showed how to express it as a sum over pairs of disjoint subsets of edges. Trinks [109] showed that these two trivariate polynomials are equivalent in the sense that each can be transformed to the other by simple transformations of the variables.

The *extendable colouring function* is given by

$$\mathrm{EC}(G;\, p, \lambda) = \Pr(f \text{ is } \lambda\text{-extendable}).$$

Observe that $\mathrm{EC}(G;\, \lambda^{-1}, \lambda) = \lambda^{-n} P(G;\, \lambda)$ and $\mathrm{EC}(G;\, 0, \lambda)$ is 1 if G is 3-colourable and 0 otherwise. We can express $\mathrm{EC}(G;\, p, \lambda)$ as a sum over all possible domains for f,

$$\mathrm{EC}(G;\, p, \lambda) = \sum_{C \subseteq V} \mathrm{EC}(G, C;\, \lambda)\, p^{|C|} (1 - \lambda p)^{n-|C|},$$

where $\mathrm{EC}(G, C;\, \lambda)$ is the number of partial λ-colourings f such that $\mathrm{dom}\, f = C$ and f has an extension which is a λ-colouring of G. But this does not show that $\mathrm{EC}(G;\, p, \lambda)$ is a polynomial. In fact, in general it is not. Consider K_2.

$$\mathrm{EC}(K_2;\, p, \lambda) = \begin{cases} 0, & \text{if } \lambda = 1, \\ 1 - \lambda p^2, & \text{if } \lambda \geq 2. \end{cases} \tag{4}$$

It follows from (4) that $\mathrm{EC}(K_2; p, \lambda)$ is not a polynomial, since when $p = 0$ it is 1 for every positive integer $\lambda \geq 2$ but is 0 for $\lambda = 1$, a property that no polynomial can have. Nonetheless, $\mathrm{EC}(G; p, \lambda)$ is a polynomial in p for every fixed $\lambda \in \mathbb{N}$. When λ is fixed, we denote this polynomial in p by $\mathrm{EC}_\lambda(G; p)$.

For the extendable colouring polynomial $\mathrm{EC}_\lambda(G; p)$, even checking the structures being counted seems hard in general for $\lambda \geq 3$. Testing whether a given partial λ-assignment of a given graph G is extendable to a λ-colouring of G is NP-complete when $\lambda \geq 3$, by polynomial-time reduction from graph λ-colourability: when the partial λ-assignment is the empty λ-assignment, leaving all vertices uncoloured, we just have standard λ-colourability. So this polynomial is likely to be more difficult to work with than many that have been studied.

The *forced colouring function* is given by

$$\mathrm{FC}(G; p, \lambda) = \Pr(f \text{ eventually forces a } \lambda\text{-colouring of } G).$$

Again, we have $\mathrm{FC}(G; \lambda^{-1}, \lambda) = \lambda^{-n} P(G; \lambda)$. We can express $\mathrm{FC}(G; p, \lambda)$ as a sum,

$$\mathrm{FC}(G; p, \lambda) = \sum_{C \subseteq V} \mathrm{FC}(G, C; \lambda) p^{|C|} (1 - \lambda p)^{n - |C|},$$

where $\mathrm{FC}(G, C; \lambda)$ is the number of partial λ-colourings f such that $\mathrm{dom}\, f = C$ and f eventually forces a λ-colouring of G. But, again, we do not have a polynomial, in general. Consider a null graph.

$$\mathrm{FC}(\overline{K_n}; p, \lambda) = \begin{cases} 1, & \text{if } \lambda = 1, \\ (\lambda p)^n, & \text{if } \lambda \geq 2. \end{cases} \tag{5}$$

This is because when there is only one colour, the colour of every initially-uncoloured isolated vertex is forced, while if there are at least two colours, an uncoloured isolated vertex cannot be forced, so it can only get a colour from the initial random partial colouring. It follows from (5) that $\mathrm{FC}(\overline{K_n}; p, \lambda)$ is not a polynomial, since when $p = 0$ it is zero for every positive integer $\lambda \geq 2$ but is 1 for $\lambda = 1$. But $\mathrm{FC}(G; p, \lambda)$ is a polynomial in p for every fixed $\lambda \in \mathbb{N}$, denoted by $\mathrm{FC}_\lambda(G; p)$.

Our three bivariate functions satisfy

$$\mathrm{FC}(G; p, \lambda) \leq \mathrm{EC}(G; p, \lambda) \leq \mathrm{PC}(G; p, \lambda)$$

whenever $\lambda \in \mathbb{N}$ and $0 \leq p \leq \lambda^{-1}$.

For these three functions, the case $\lambda = 3$ is of particular interest:

$$\mathrm{PC}_3(G; p) = \mathrm{PC}(G; p, 3),$$
$$\mathrm{EC}_3(G; p) = \mathrm{EC}(G; p, 3),$$
$$\mathrm{FC}_3(G; p) = \mathrm{FC}(G; p, 3).$$

We are particularly interested in $\mathrm{FC}_3(G;\, p)$ which we call the *forced 3-colouring polynomial* of G. We list some basic examples and observations.

$$\begin{aligned}
\mathrm{FC}_3(\overline{K_n};\, p) &= (3p)^n,\\
\mathrm{FC}_3(K_2;\, p) &= 6p^2,\\
\mathrm{FC}_3(K_3;\, p) &= 6p^2(3-8p),\\
\mathrm{FC}_3(K_{1,2};\, p) &= 6p^2(1-p),\\
\mathrm{FC}_3(G;\, \tfrac{1}{3}) &= 3^{-n}P(G;3),\\
4^n \cdot \mathrm{FC}_3(G;\, \tfrac{1}{4}) &= \#\text{ partial 3-assignments that eventually force a } \lambda\text{-colouring of } G,\\
\mathrm{FC}_3(G \sqcup H;\, p) &= \mathrm{FC}_3(G;\, p)\mathrm{FC}_3(H;\, p).
\end{aligned}$$

In many graph polynomials, the structures being counted can be checked very quickly *in parallel*. Typically, these checking problems belong to the class NC of decision problems solvable by uniform families of logical circuits of polynomial size and polylogarithmic depth, both of these being upper bounds in terms of the input size. (See, e.g., [52] for the theory of NC and P-completeness.) For example, validity of a particular 3-colouring, or an independent set, or a clique, is just a conjunction of local conditions based on the edges of the graph; validity of a flow or a matching or a dominating set is a conjunction of local conditions based on the vertex neighbourhoods.

One motivation for studying the forced 3-colouring polynomial is because of the computational complexity of checking the structures being counted, i.e., checking if a partial 3-assignment forces a 3-colouring of the graph. This can be done in polynomial time, so the situation is not as bad as for extendable colouring polynomials, and may seem more akin to classical graph polynomials based on enumerating structures like colourings, independent sets, matchings and so on. But the forced 3-colouring polynomial has the distinctive feature that its checking problem is, in a precise sense, as hard as any problem in P. Specifically, it is logspace-complete for P [31], which is considered to be evidence that there is no fast parallel algorithm for this test and that it does not belong to NC. Intuitively, this may make these graph polynomials more difficult to compute and to investigate than the many others where the structure-checking can be done in NC. So the study of them might shed light on some less-explored regions of the theory of graph polynomials.

The forced 3-colouring polynomial does fall within the class of SOL-definable graph polynomials, introduced by Makowsky and colleagues [71, 74] and explained in more detail in [48, 63]. To see this, we can express it as

$$\mathrm{FC}_3(G;\, p) \;=\; \sum_{(C_1,C_2,C_3)\in\mathscr{FC}_3(G)} p^{\sum_{i=1}^3 |C_i|}(1-3p)^{n-\sum_{i=1}^3 |C_i|},$$

where $\mathscr{FC}_3(G)$ is the set of mutually disjoint triples (C_1, C_2, C_3) whose corresponding partial 3-assignment f—i.e., the partial 3-assignment f defined by $C_f(i) = C_i$ for $i \in \{1, 2, 3\}$—forces a 3-colouring of G. The condition that $(C_1, C_2, C_3) \in \mathscr{FC}_3(G)$ can be expressed in SOL using the following observation.

Proposition 2 A partial λ-assignment f forces a λ-colouring of G if and only if it has no partial 3-colouring extension g which is not total and forces no vertex outside $\operatorname{dom} g$.

Proof ($\Longrightarrow$) Suppose f forces a λ-colouring of g. Then every partial 3-colouring extension of f forces the same λ-colouring of G. So there is no non-total extension of f that does not force any vertex.

($\Longleftarrow$) If f does not force a λ-colouring of G, then the forcing process must stop with at least one vertex uncoloured. When that happens, the partial λ-assignment that has been forced so far is a partial 3-colouring extension of f that is not total and forces no vertex outside its domain.

□

This observation justifies the following logical formulation of $\mathscr{FC}_3(G)$, which is similar in design to the SOL expression for connectedness in [48, §3.3].

$$
\begin{aligned}
&(C_1, C_2, C_3) \in \mathscr{FC}_3(G) \iff \\
&\neg\exists(D_1, D_2, D_3) : \\
&\left(\bigwedge_{i=1}^{3}(C_i \subseteq D_i)\right) \wedge \left(\bigwedge_{i=1}^{3}\bigwedge_{j=i+1}^{3}(D_i \cap D_j = \emptyset)\right) \wedge (D_1 \cup D_2 \cup D_3 \neq V) \wedge \\
&\left(\forall v, w \in D_1 \cup D_2 \cup D_3 : \left(vw \in E \to \bigwedge_{i=1}^{3}\big((v \notin D_i) \vee (w \notin D_i)\big)\right)\right) \wedge \\
&\forall v \in V \setminus (D_1 \cup D_2 \cup D_3) : \\
&\left(\left(\bigwedge_{i=1}^{3}(\exists w \in D_i : vw \in E)\right)\right. \\
&\left.\vee \bigvee_{i=1}^{3}\bigvee_{j=i+1}^{3}\big(\forall w \in V : vw \in E \to \big(w \notin D_i \cup D_j\big)\big)\right).
\end{aligned}
$$

This can all be expressed using the formalism of SOL, with sets of vertices represented as unary relations, adjacency as a binary relation, and so on.

We study the forced colouring function of a graph further, along with its associated polynomials, in [41].

4 The Diverse Origins of Graph Polynomials

Having discussed a variety of graph polynomials in the previous section, it is a good time to review the origins of graph polynomials in general.

Historically, graph polynomials have been created in several different ways.

(i) Sometimes, sequences of numbers that count structures of interest are used as coefficients to define a generating polynomial for them. For example, polynomials that count independent sets [53, 57], cliques [44, 123, 124], matchings [43, 49, 56] and dominating sets [1] arose in this way.

(ii) Sometimes, a sequence of numbers that count structures of interest in a graph is taken to give the values of a function of the graph and an integer parameter, and the function turns out to be a polynomial in that parameter. This is how the chromatic polynomial arose [8].

(iii) Sometimes, a probability model is defined on graphs, with independent, identically-distributed random choices being made throughout the graph (typically on vertices or edges) according to some probability p, and the probability that this gives a particular type of structure is a polynomial in p. Polynomials defined this way include the all-terminal network reliability polynomial [116] and the stability polynomial [29, 57], as well as the polynomials based on partial colourings that we introduced in Sect. 3.6. Such polynomials are often easily transformed into generating polynomials, as is the case for the relationship between the stability and independent set polynomials.

(iv) Sometimes, a graph invariant of physical interest is defined which turns out to be a polynomial, possibly after an algebraic change of variables, as in the case of the partition functions of the Ising model [61], Ashkin-Teller model [2] and Potts model [101]. There is some overlap between this type and some previous types, e.g., the matching polynomial is framed as a partition function in [56], and the reliability polynomial models the survivability of networks in the presence of local link failures.

(v) Sometimes, certain graph invariants are found to satisfy reduction relations of some kind, and this motivates the development of a common generalisation which turns out to be a polynomial. This is how the Tutte polynomial was created, as Tutte relates in [114, p. 53] and [115].

(vi) Sometimes, a polynomial is created by analogy with existing polynomials and/or by generalising them. See [34, §3.4] for a discussion of the different ways in which Tutte-Whitney polynomials have been generalised; the informal classification given there could apply to analogues and generalisations of any graph polynomial. An early example of this is the Whitney rank generating function itself, which arose in 1932 as a bivariate generalisation of the chromatic polynomial [120] without noting any other combinatorial interpretations of its values and with the deletion-contraction relation only being a "note added in proof" and attributed to R. M. Foster. The Oxley-Whittle polynomial [98, 99] arose as the analogue of Tutte-Whitney polynomials

for 2-polymatroids. Tutte-Whitney polynomials of graphs were also the inspiration for the various topological Tutte polynomials for ribbon graphs and embedded graphs [17, 26]. The chromatic polynomial was generalised in a different direction by Harary [54] and his generalised chromatic polynomials have inspired a fruitful stream of research on *Harary polynomials* by Makowsky and collaborators [50, 58, 67, 68, 83].

(vii) Occasionally, a polynomial is defined by specialisation from an existing polynomial, instead of by generalisation: it is not always the case that the particular motivates the general. The flow polynomial was first discovered (but not named) by Whitney [121] as the dual of the chromatic polynomial and a specialisation of his rank generating function, but without giving its values any combinatorial interpretation; see [40, §34.7]. Only later did Tutte identify the connection with flows [112].

(viii) When counting graphs and other combinatorial objects up to symmetry, cycle index polynomials are used: see [55] for their use in counting various types of graphs up to isomorphism, a field which originated with Redfield [102] and Pólya [100]. Beginning with work by Cameron and collaborators in the early 2000s, cycle index polynomials have been used to define graph polynomials that count colourings and other structures up to symmetries under the automorphism group of the graph [13–15].

(ix) Sometimes, the characteristic polynomial of a square matrix associated with a graph is studied, the main example being the characteristic polynomial of a graph[2] which is obtained from the adjacency matrix.

(x) Sometimes, a polynomial is defined for other mathematical objects, and a natural construction of graphs from those objects leads to a graph polynomial. Some knot polynomials have been translated to graph polynomials and linked to the Tutte polynomial: see, e.g., [59]. The weight enumerator of a linear code over GF(2) may be regarded as a binary matroid polynomial whose coefficients count members of a circuit space according to their size (and similarly for a cocircuit space), thereby yielding a graph polynomial through the special case of graphic matroids (see, e.g., [117, §15.7]).

This classification is just a set of observations of how the field has developed so far and is certainly not any kind of prescription for how it must develop in the future. It may not be exhaustive, and the categories may overlap. Indeed it is common for a graph polynomial to have multiple different formulations, so that it can be *defined* in two or more of the above ways, even if it was historically *created* in just one way. There is no single best route to defining graph polynomials: all the routes listed above have led to new polynomials that have generated a lot of interest and some profound research.

Nonetheless, it is worth reflecting on the various inspirations for the diverse graph polynomials that have been developed. Analogy and generalisation have been very

[2] Not to be confused with the characteristic polynomial of a matroid, which generalises the chromatic polynomial of a graph.

fruitful, but it is arguable that the real worth of a new graph polynomial lies not so much in how well its theory echoes those of other known polynomials, but rather in the information it contains about the graph (including the relationships it reveals between different features of it) and in the accessibility of that information.

In this context, the originality of Tutte's approach (4) is striking. It did not really fall within any earlier approach. It was grounded in important graph invariants. His polynomial emerged naturally as a framework that captured what those invariants had in common. The fact that it is a *polynomial* was a happy outcome, and not surprising since one of the invariants he was abstracting from was the chromatic polynomial, but it does not seem to have been an objective in itself.

It is conceivable that other collections of enumerative (or probabilistic) graph invariants are waiting to be brought into common frameworks, and that these frameworks may sometimes be quite different to Tutte's, and may not always lead to polynomials.

5 Reduction Relations

The recursive relation (2) for the Tutte polynomial has the following characteristics.

- The number of cases is fixed (i.e., independent of the size of the graph).
- In the base case, the expression is a fixed polynomial (in this case, just the constant 1).
- In each other case, the expression is linear (treating the Tutte polynomial symbols $T(G; x, y)$, $T(G \setminus e; x, y)$, $T(G/e; x, y)$ as indeterminates, with coefficients being fixed polynomials in x, y).
- The graphs appearing in the right-hand sides are obtained from the original graph G by simple local operations.
- Certain types of edges must be treated as special cases (namely, loops and coloops). Such edges are "degenerate" in some sense, and the expression on the right-hand side is usually simpler than in the general case, with fewer terms.
- The number of terms in each of these linear expressions is fixed.
- The polynomial is well defined, in that the order in which local operations are used when evaluating the polynomial using (2) does not matter [112].

Many other graph polynomials also satisfy *reduction relations* of this type. Examples include the independent set, stability, clique and matching polynomials, the Oxley-Whittle polynomial for graphic 2-polymatroids [98, 99], the Borzacchini-Pulito polynomial [11], the edge elimination polynomial [3, 4], and the rich class of Tutte polynomials of Hopf algebras [69].[3] One of the earliest studies of graph

[3] Some notable graph polynomials are simple transformations or partial evaluations of the Tutte polynomial, so they satisfy reduction relations of this type because the Tutte polynomial does.

polynomials defined by a variety of reduction relations beyond deletion-contraction is due to Zykov [124]; this has roots in his earliest work in graph theory [123].

Godlin, Katz and Makowsky have given a general definition of reduction relations of roughly the above type, in the context of the logical theory of graph polynomials [48]. Paraphrasing, they showed that every graph polynomial with such a recursive definition in SOL can be expressed as a SOL-definable sum over subsets. This links, in one direction, two of the main ways of defining graph polynomials: reduction relations, and sums over subsets ("subset expansions" in their terminology). They ask whether the link goes the other way too: does every graph polynomial expressible as a SOL-definable sum over subsets satisfy a reduction relation of their general SOL-based form?

Some graph polynomials do not seem to have a natural reduction relation within the class of objects over which they are initially defined. But it often happens that, even in such cases, there is a wider class of objects to which the graph polynomial can be generalised and which supports a reduction relation of this type. We illustrate this with some examples.

5.1 Counting Edge-Colourings

For any graph G, define $P'(G;q)$ to be the number of q-edge-colourings of G (mirroring the standard use of $\chi'(G)$ and $\chi(G)$ to denote the chromatic index[4] and chromatic number, respectively). It is well known that edge-colourings of a graph G correspond to vertex-colourings of the line graph $L(G)$ of G (see, e.g., [45]). So we have

$$P'(G;q) = P(L(G);q), \tag{6}$$

which makes plain that $P'(G;q)$ a polynomial.

The most natural way to get a reduction relation for $P'(G;q)$ is to use (6) and invoke the deletion-contraction relation for chromatic polynomials:[5]

$$P(L(G);q) = \begin{cases} q^m, & \text{if } L(G) \text{ has no edges;} \\ P(L(G)\setminus e;q) - P(L(G)/e;q), & \text{otherwise.} \end{cases}$$

These include the chromatic, flow, reliability, Martin [84] and Jones polynomials, the Ising and Potts model partition functions, the Whitney rank generating function and the coboundary polynomial.

[4] I.e., the minimum q such that G is q-edge-colourable.

[5] We assume that line graphs have no loops, which is the usual practice. If, instead, we assume that loops are created in the line graph corresponding to loops in G, then it is natural to assume that a line graph with a loop has no edge-colouring. If we were to allow colouring of loops in edge-colourings, then our reduction relation would not work.

But this relation holds in the class of *all* graphs since, in general, $L(G) \setminus e$ and $L(G)/e$ are not line graphs.

The base cases for this reduction relation are graphs with no edges. Such a graph is a line graph of a disjoint union of a matching and a set of isolated vertices.

5.2 The Symmetric Ashkin-Teller Model

In Sect. 3.3 we saw that the symmetric Ashkin-Teller model partition function is not a specialisation of the Tutte polynomial. It therefore does not obey the kind of deletion-contraction relation characteristic of evaluations of the Tutte polynomial.

However, it turns out that it does satisfy such a relation in the wider class of binary functions. This follows from the following result in [35] which shows that it is a specialisation of a suitable λ-Tutte-Whitney function.

Theorem 3 ([35]) For each K and $K_{\sigma\tau}$ there exists λ such that the partition function $Z(G; K, K_{\sigma\tau})$ of the symmetric Ashkin-Teller model on a graph G can be obtained from the λ-Tutte-Whitney function.

Details, including expressions for λ in terms of K and $K_{\sigma\tau}$, are given in [35].

Putting Theorems 1 and 3 together gives a reduction relation for the Ashkin-Teller model partition function in the class of binary functions.

5.3 Go Polynomials

None of the Go polynomials introduced in Sect. 3.5 and [32] have an obvious recurrence relation, of the type we have been considering, on graphs. But there are recurrence relations in a wider class of objects that generalise graphs. In [32, §3], it is found that $\mathrm{Go}^{\#}(G; \lambda)$ and $\mathrm{Go}(G; p)$ satisfy a family of linear recurrence relations over a larger class of graphs, there called $\mathscr{L}$-graphs, in which graphs may have extra labels on some of their vertices and edges that modify the conditions that a partial λ-assignment must satisfy in order to be a legal position.

5.4 Polynomials Based on Partial Colourings

We now consider reduction relations for the polynomials we introduced in Sect. 3.6.

The partial chromatic polynomial does not obey a deletion-contraction relation of the usual type, as it is not obtainable from the Tutte polynomial and in fact contains extra information. No reduction relation for it is given explicitly in [22]. But (3) points to a reduction relation based on labelling. A *chromatically labelled graph* is a graph in which each vertex may be labelled C, indicating that it must receive a

colour, or U, indicating that it must be uncoloured; a vertex may have no label, but it cannot have two labels. Each chromatically labelled graph is written as $G^{(C,U)}$ where $C, U \subseteq V$ and $C \cap U = \emptyset$. A *totally chromatically labelled graph* $G^{(C,U)}$ is a chromatically labelled graph in which every vertex is labelled, i.e., $C \cup U = V$.

A *partial* λ*-assignment* of a chromatically labelled graph $G^{(C,U)}$ is a partial λ-assignment f of G such that $C \subseteq \mathrm{dom}\, f \subseteq V \setminus U$. It is a *partial* λ*-colouring* of $G^{(C,U)}$ if it is also a partial λ-colouring of G.

For chromatically labelled graphs, put

$$\begin{aligned}\mathrm{PC}(G^{(C,U)}; p, \lambda) &= \Pr(f \text{ is a partial } \lambda\text{-colouring of } G^{(C,U)}) \\ &= \Pr\Big(\big(f \text{ is a } \lambda\text{-colouring of } G[\mathrm{dom}\, f]\big) \\ &\qquad \wedge (C \subseteq \mathrm{dom}\, f \subseteq V \setminus U)\Big).\end{aligned}$$

This polynomial, in this wider class, has a simple reduction relation.

Theorem 4 For any $v \in V \setminus (C \cup U)$,

$$\mathrm{PC}(G^{(C,U)}; p, \lambda) \;=\; \mathrm{PC}(G^{(C\cup\{v\},U)}; p, \lambda) + \mathrm{PC}(G^{(C,U\cup\{v\})}; p, \lambda).$$

Proof

$$\begin{aligned}&\mathrm{PC}(G^{(C,U)}; p, \lambda) \\ &= \Pr((f \text{ is a } \lambda\text{-colouring of } G[\mathrm{dom}\, f]) \wedge (C \subseteq \mathrm{dom}\, f \subseteq V \setminus U)) \\ &= \Pr((f \text{ is a } \lambda\text{-colouring of } G[\mathrm{dom}\, f]) \wedge (C \subseteq \mathrm{dom}\, f \subseteq V \setminus U) \\ &\qquad \wedge (v \in \mathrm{dom}\, f)) \\ &\quad + \Pr((f \text{ is a } \lambda\text{-colouring of } G[\mathrm{dom}\, f]) \wedge (C \subseteq \mathrm{dom}\, f \subseteq V \setminus U) \\ &\qquad \wedge (v \notin \mathrm{dom}\, f)) \\ &= \Pr((f \text{ is a } \lambda\text{-colouring of } G[\mathrm{dom}\, f]) \wedge (C \cup \{v\} \subseteq \mathrm{dom}\, f \subseteq V \setminus U)) \\ &\quad + \Pr((f \text{ is a } \lambda\text{-colouring of } G[\mathrm{dom}\, f]) \wedge (C \subseteq \mathrm{dom}\, f \subseteq V \setminus (U \cup \{v\}))) \\ &= \mathrm{PC}(G^{(C\cup\{v\},U)}; p, \lambda) + \mathrm{PC}(G^{(C,U\cup\{v\})}; p, \lambda).\end{aligned}$$

□

The reduction relation of Theorem 4 cannot be used on totally chromatically labelled graphs, when $C \cup U = V$. In that case, the partial chromatic polynomial is just a scaled version of the chromatic polynomial of $G - U$:

$$\mathrm{PC}(G^{(V\setminus U,U)}; p, \lambda) = (1 - \lambda p)^{|U|} p^{n-|U|}\, P(G - U; \lambda)\,. \tag{7}$$

We return to this point in Sect. 6.

It is also possible to get a reduction relation for $\mathrm{PC}(G; p, \lambda)$ on certain vertex-weighted graphs using the fact that the partial chromatic polynomial is a specialisation of the U-polynomial of Noble and Welsh [97] (because its equivalent polynomial $\xi(G; x, y, z)$ is) which has a reduction relation on those weighted graphs.

We now consider extendable colouring polynomials and show that $\mathrm{EC}_\lambda(G)$ satisfies a reduction relation in the class of chromatically labelled graphs we introduced above. A partial λ-assignment of $G^{(C,U)}$ is λ-*extendable* in $G^{(C,U)}$ if, as a partial λ-assignment of G, it is λ-extendable. So, although a label U on a vertex specifies that it is uncoloured by our random λ-assignment f, the vertex is allowed to be coloured by an extension of f.

For chromatically labelled graphs, put

$$\begin{aligned}\mathrm{EC}(G^{(C,U)}; p, \lambda) &= \Pr(f \text{ is } \lambda\text{-extendable in } G^{(C,U)}) \\ &= \Pr\left((f \text{ is } \lambda\text{-extendable in } G) \wedge (C \subseteq \operatorname{dom} f \subseteq V \setminus U)\right).\end{aligned}$$

Theorem 5 For any $v \in V \setminus (C \cup U)$,

$$\mathrm{EC}_\lambda(G^{(C,U)}; p) \;=\; \lambda p\, \mathrm{EC}_\lambda(G^{(C\cup\{v\},U)}; p) + (1 - \lambda p)\, \mathrm{EC}_\lambda(G^{(C,U\cup\{v\})}; p).$$

Proof

$$\begin{aligned}\mathrm{EC}_\lambda(G^{(C,U)}; p) &= \Pr(f \text{ is } \lambda\text{-extendable in } G^{(C,U)}) \\ &= \Pr(f \text{ is } \lambda\text{-extendable in } G^{(C,U)} \mid v \in \operatorname{dom} f) \Pr(v \in \operatorname{dom} f) + \\ &\quad \Pr(f \text{ is } \lambda\text{-extendable in } G^{(C,U)} \mid v \notin \operatorname{dom} f) \Pr(v \notin \operatorname{dom} f) \\ &= \lambda p \Pr(f \text{ is } \lambda\text{-extendable in } G^{(C\cup\{v\},U)}) + \\ &\quad (1 - \lambda p) \Pr(f \text{ is } \lambda\text{-extendable in } G^{(C,U\cup\{v\})}) \\ &= \lambda p\, \mathrm{EC}_\lambda(G^{(C\cup\{v\},U)}; p) + (1 - \lambda p)\, \mathrm{EC}_\lambda(G^{(C,U\cup\{v\})}; p).\end{aligned}$$

□

Lastly, we consider forced colouring polynomials. A partial λ-assignment of $G^{(C,U)}$ *forces* a λ-colouring of $G^{(C,U)}$ if, as a partial λ-assignment of G, it forces a λ-colouring of G. Again, a label U on a vertex only prevents it from being coloured by f itself; the label does not stop it from being eventually forced by f. Put

$$\begin{aligned}\mathrm{FC}(G^{(C,U)}; p, \lambda) &= \Pr(f \text{ forces a } \lambda\text{-colouring of } G^{(C,U)}) \\ &= \Pr\left((f \text{ forces a } \lambda\text{-colouring of } G) \wedge (C \subseteq \operatorname{dom} f \subseteq V \setminus U)\right).\end{aligned}$$

A very similar argument to Theorem 5 shows that $\mathrm{FC}_\lambda(G; p)$ satisfies a reduction relation in the class of chromatically labelled graphs.

Theorem 6 For any $v \in V \setminus (C \cup U)$,

$$\mathrm{FC}_\lambda(G^{(C,U)}; p) = \lambda p\, \mathrm{FC}_\lambda(G^{(C\cup\{v\},U)}; p) + (1 - \lambda p)\, \mathrm{FC}_\lambda(G^{(C,U\cup\{v\})}; p).$$

□

5.5 *Questions*

We have now seen six examples of graph polynomials which do not seem to satisfy a local linear reduction relation over the class of graphs but which do satisfy such relations over some larger class.

Other examples exist. The U-polynomial of a graph, introduced by Noble and Welsh, has a reduction relation in the larger class of vertex-weighted graphs in which the weights are positive integers [97] (see also [96]). Krajewski et al. [69] used their Hopf algebra framework to show that various topological Tutte polynomials without full reduction relations[6] can be extended to a larger class of objects (by augmenting the embedded graphs with some extra structure) so that they do have full reduction relations. (See especially [69, Remark 62].)

These examples raise the question of how widespread this phenomenon is. Which graph polynomials exhibit this phenomenon? Can they be characterised in some formal, rigorous way? For the polynomials considered in Sects. 5.3 and 5.4, suitable superclasses of the class of graphs can be found by introducing new labels on vertices and/or edges with specific technical meanings. This is likely to be a wider phenomenon and may be able to be captured using the logical framework of Makowsky and colleagues. But our first two cases, in Sects. 5.1 and 5.2, are not of this type, and it is not clear how to include them in a general characterisation of this phenomenon.

6 Levels of Recursion

For many graph polynomials, repeated application of a reduction relation leaves only trivial graphs. For example, repeated application of deletion-contraction relations for the chromatic or Tutte polynomials leaves null graphs. But sometimes a graph polynomial has a reduction relation in which the base cases are themselves nontrivial graphs and another reduction relation needs to be applied in order to reduce them to simpler base cases; we might say that we have two "levels" of reduction relation.

[6] Because the known relations did not cover all possible edge types.

Partial chromatic polynomials provide an example. The reduction relation we gave in Theorem 4 may be used to reduce the partial chromatic polynomial to a sum involving partial chromatic polynomials of totally chromatically labelled graphs, and each of these polynomials can in turn be expressed in terms of a chromatic polynomial, by (7). So we can apply the deletion-contraction relation to each of the chromatic polynomials, thereby expressing the partial chromatic polynomial as a sum of simple base cases—chromatic polynomials of null graphs—with two levels of reduction.

It can get worse than this! Go polynomials (Sect. 3.5) may be regarded as having three levels of reduction. Firstly, graphs are reduced to $\mathscr{L}$-graphs using [32, Cor. 6 or Theorem 7]. Then $\mathscr{L}$-graphs are reduced to ordinary graphs again using [32, Cor. 10]. Finally, these ordinary graphs are reduced to null graphs using the deletion-contraction relation for the chromatic polynomial. The paper in fact gives a method of expressing a Go polynomial as a large sum of chromatic polynomials.

For some polynomials, the situation is less clear. For extendable colouring and forced colouring polynomials, we gave reduction relations in Theorems 5 and 6 whose base cases require computation of those polynomials for totally chromatically labelled graphs. Those computations do not have obvious analogues of (7).

For extendable colouring polynomials, here is an attempt involving an addition-identification relation on the set of vertices labelled C:

$$\mathrm{EC}_\lambda(G^{(C,U)};\, p) \;=\; \mathrm{EC}_\lambda(G^{(C,U)} + uv;\, p) + \mathrm{EC}_\lambda(G^{(C,U)}/uv;\, p),$$

for any $u, v \in C$ such that $uv \notin E(G)$. This can be applied repeatedly until the vertices labelled C form a clique in G. Whenever this clique has $> \lambda$ vertices, the polynomial is identically 0 since the graph is not λ-colourable. So we end up with a sum of polynomials $\mathrm{EC}_\lambda(H^{(D,U)};\, p)$ of totally chromatically labelled graphs of the form $H^{(D,U)}$ where D is a clique of size $\leq \lambda$ in H. Because D is a clique, a partial λ-assignment of $H^{(D,U)}$ is λ-extendable if and only if H is λ-colourable. So we have

$$\begin{aligned}
&\mathrm{EC}_\lambda(H^{(D,U)};\, p)\\
&= \Pr\big((f \text{ is } \lambda\text{-extendable in } H) \wedge (D \subseteq \operatorname{dom} f \subseteq V \setminus U) \wedge\\
&\qquad (f \text{ is a } \lambda\text{-colouring of } H[D])\big)\\
&= \Pr\left((D \subseteq \operatorname{dom} f \subseteq V \setminus U) \wedge (f \text{ is a } \lambda\text{-colouring of } H[D])\right) \times\\
&\quad \Pr\big((f \text{ is } \lambda\text{-extendable in } H) \mid\\
&\qquad (D \subseteq \operatorname{dom} f \subseteq V \setminus U) \wedge (f \text{ is a } \lambda\text{-colouring of } H[D])\big)\\
&= p^{|D|}(\lambda)_{|D|} \cdot [\![H \text{ is } \lambda\text{-colourable}]\!].
\end{aligned}$$

Here we have used the Iverson bracket:

$$[\![H \text{ is } \lambda\text{-colourable}]\!] = \begin{cases} 1, \text{ if } H \text{ is } \lambda\text{-colourable;} \\ 0, \text{ otherwise.} \end{cases}$$

We seem to have gained something, computationally, by expressing it this way: we "only" have an NP-complete quantity to evaluate, rather than a #P-hard graph polynomial! But it is no longer a natural sum of graph polynomials, so it does not give us another layer of reduction relations of the kind we have been considering.

We can try a similar approach with forced colouring polynomials (Theorem 6). Again, the base cases for the first reduction relation we use are totally chromatically labelled graphs, and again we can use addition-identification repeatedly to get a sum over totally chromatically labelled graphs $H^{(D,U)}$ in which the set D of vertices labelled C induces a clique of size $\leq \lambda$. The summand for $H^{(D,U)}$ includes the factor

$$[\![\text{a } \lambda\text{-colouring of } H[D] \text{ forces a } \lambda\text{-colouring of } H]\!],$$

again using the Iverson bracket. This is polynomial-time computable, so we could reasonably call it a final base case and say that forced λ-colouring polynomials have two levels of recursion. But these base cases are much less simple than the base cases for other recursions we have considered (e.g., null graphs, for chromatic polynomials and (eventually) for Go polynomials). This time, the computational task for each base case is P-complete [31].

It would be interesting to study this phenomenon of levels of recursion for graph polynomials more systematically. Perhaps the notion can be formalised and then related to the logical structure of the definition of the polynomial.

7 Graph *Polynomials*?

We perhaps take it for granted that graph invariants giving counts, or probabilities, of structures of interest are *polynomials*. This is not a *necessary* feature of such invariants. In general, the λ-Whitney function of a graph [33] may have irrational exponents, though in certain forms this can be avoided when evaluating them along hyperbolae $xy = 2^r$ for $r \in \mathbb{N}$. (Some related polynomials in knot theory may have negative or fractional exponents, but this is not a significant exception because in general they can be transformed to polynomials by appropriate changes of variable.)

The choice of parameter is crucial. For example, define $\mathrm{HomCyc}(G; q)$ to be the number of homomorphisms from G onto the cycle C_q. Such homomorphisms may be viewed as q-assignments in which adjacent vertices in G are mapped to "colours" (being vertices in C_q) that are adjacent in C_q; unlike normal graph colouring (when the homomorphism is to K_q), the two distinct colours used on the endpoints of an edge cannot be completely arbitrary but are constrained to be neighbouring colours

(vertices) in C_q. This is not, in general, a polynomial in q. One way to see this is to note that $\mathrm{HomCyc}(K_3; q) = 0$ for all even q but is not identically 0. In other words, in the terminology of de la Harpe and Jaeger [19], the sequence $(C_n : n \in \mathbb{N})$ is not a *strongly polynomial sequence* of graphs. See [51] for a study of the deep question of which sequences of graphs, treated as targets of homomorphisms, give rise to graph polynomials that count homomorphisms.

Graph colouring requires the colour classes to induce null graphs, where the chromons each consist of a single vertex. There has been a lot of work on generalised colourings where the chromons are less severely restricted. One of the simplest relaxations is to bound the sizes of the chromons. Define $\mathrm{mc}(G; s)$ to be the number of 2-assignments of G in which every chromon has size $\leq s$. We have

$$\mathrm{mc}(G; 0) = \begin{cases} 1, \text{ if } n = 0, \\ 0, \text{ if } n \geq 1; \end{cases}$$

$$\mathrm{mc}(G; 1) = \begin{cases} 2^{k(G)}, \text{ if } G \text{ is bipartite}, \\ 0, \quad \text{otherwise}; \end{cases}$$

$$\mathrm{mc}(G; s) = 2^n \quad \forall s \geq n.$$

So $\mathrm{mc}(G; s)$ cannot be a polynomial in s.

Define $\mathrm{emb}(G; g)$ to be the number of orientable 2-cell combinatorial embeddings of G of genus g, where embeddings are given by rotation schemes. This cannot be a polynomial in g because it is positive when g lies between the genus and maximum genus, inclusive, of G, but is zero for all g above the maximum genus.

For graph invariants that *are* polynomials, it is natural to hope that the theory of polynomials may shed light on the graph polynomials and, through them, on graphs themselves. After all, polynomials have a rich mathematical theory that has been built up over centuries.

For example, Birkhoff's work on the chromatic polynomial, beginning with [8], was motivated by the thought that its properties, as a polynomial, might help prove the Four-Colour Conjecture (as it then was). According to Morse [95, p. 386], "Birkhoff hoped that the theory of chromatic polynomials could be so developed that methods of analytic function theory could be applied."

With this motivation, we can ask of any graph polynomial, which aspects of the theory of polynomials correspond to properties of the underlying graph?

It is common for graph polynomials to be multiplicative over components or even over blocks (with multiplicativity over blocks being typical for polynomials that depend only on the cycle matroid of the graph, such as the Tutte polynomial). This is to be expected for polynomials that count things or give probabilities, since the lack of interaction between separate components or blocks typically means we can treat them as contributing independently to counts or probabilities. Does such a relationship work both ways? In other words, does multiplicativity *only* occur over components/blocks? What, in the graph, is represented by the polynomial's

proper factors? How can we characterise the structure of graphs whose polynomial is irreducible?

In the case of the Tutte polynomial, it was shown by Merino, de Mier and Noy that the Tutte polynomial of a matroid is irreducible if and only if the matroid is connected [20, 85]. So the factors of the polynomial correspond exactly to the components of the matroid, which means that, for graphs, they correspond to blocks.

But the situation is not so straightforward for many other graph polynomials, even for those that are specialisations of the Tutte polynomial.

For the chromatic polynomial, it was known to Whitney [119, §14] that the chromatic polynomial factorises when the graph is *clique-separable*[7] in the sense that it has a separating clique, where a *separating clique* is a clique whose removal increases the number of components of the graph. If G is formed from overlap of H_1 and H_2 in a separating r-clique, then Whitney showed that

$$P(G;x) = \frac{P(H_1;x)P(H_2;x)}{P(K_r;x)}.$$

Somewhat surprisingly, such *chromatic factorisations* can occur in other cases, too: in [92, 93], examples are given of chromatic factorisations of graphs that are *strongly non-clique-separable* in that they are not chromatically equivalent to a clique-separable graph (see also [94]). Some studies of this kind have since been done for other polynomials including the reliability polynomial [91] and the stability polynomial [86].

Another fundamental topic in the mathematical theory of polynomials is Galois theory. So it is natural to ask about the relationship between the structure of a graph and the Galois group of a graph polynomial derived from it. An initial investigation of this topic for the chromatic polynomial, including computational results, is reported in [90].

Algebraic aspects of chromatic roots are explored in [10, 16, 21].

The most fundamental property in any mathematical system is *identity*: when are two objects considered the same? For a graph polynomial, this is when two graphs are equivalent in the sense that they have the same polynomial. This leads to the notion of certificates of equivalence, which can also be adapted to chromatic factorisation and which we discuss in the next section.

8 Certificates

The notion of a *certificate* to explain chromatic factorisation and chromatic equivalence was introduced by Morgan and Farr [92, 93] and developed further in [12, 88, 89]. The idea has since been extended to other graph polynomials including

[7] But keep in mind that this term also has a completely different meaning [47].

Fig. 4 Certificate of Tutte equivalence. Graph G is Tutte equivalent to graph H

the stability polynomial [86], reliability polynomial [91] and Tutte polynomial [87]. This use of the term "certificate" is inspired by its use in complexity theory, e.g., in defining NP, but we are using the term in a much more specific sense.

Informally, a *certificate* is a sequence of expressions $E_0, E_1, E_2, \ldots, E_k$ in graphs where each expression E_i, $i > 0$, can be obtained from its predecessor E_{i-1} by applying a relation satisfied by the graph polynomial in question (e.g., a deletion-contraction relation, or multiplicativity for disjoint unions). In these expressions, a graph can be regarded as representing its corresponding polynomial. Replacing each graph by its polynomial, then simplifying the entire expression, gives a polynomial that can be thought of as the graph polynomial for that expression.

An example is given in Fig. 4. In this certificate, we use the deletion-contraction relation for the Tutte polynomial in the form

$$[\mathbf{T1}]: \qquad G \longrightarrow G \setminus e + G/e,$$

with each graph standing for its Tutte polynomial.

A simple rearrangement of $T1$ and renaming of the graphs used in the terms gives us

$$[\mathbf{T2}]: \qquad G \longrightarrow (G + uv) - G/uv.$$

The resulting certificate has the form:

$$\begin{aligned} G &= G \setminus e + G/e \qquad (\text{Applying } \mathbf{T1}) \\ &= (G \setminus e + uv) - G \setminus e/uv + G/e \qquad (\text{Applying } \mathbf{T2}) \\ &= G \setminus e + uv \qquad (\text{Algebraic Step (cancellation)}) \end{aligned} \tag{8}$$

where the graph $G \setminus e + uv \cong H$. If we replace each graph by its Tutte polynomial, each expression in the sequence is equal to the Tutte polynomial of G, and hence we have a certificate that shows that graphs G and H are Tutte equivalent.

Our work so far has focused on certificates of equivalence and factorisation. We expect that certificates could provide a graph-theoretic approach to studying other algebraic properties of graph polynomials.

In [88], Morgan introduced the concept of a *schema* or template for certificates of factorisation and equivalence. A schema specifies the structure of a certificate, including the relation to be applied at each step, but without filling in the actual graphs. So, instead of actual graphs (as in Fig. 4), we just have symbols for them. In fact, as written—with symbols G, $G \setminus e$, etc.—(8) is really a schema for certificates, and the certificate in Fig. 4 is one particular certificate that belongs to this schema.

In [92, 93], it was shown that every graph in a particular infinite family of strongly non-clique-separable graphs has a chromatic factorisation. Each certificate of factorisation in this family used the same sequence of certificate steps, so the entire infinite family of certificates could be described by a single schema.

If two graphs have the same multiset of blocks, then they are Tutte equivalent, since the Tutte polynomial is multiplicative over blocks. We can capture this multiplicativity in certificate steps that allow a graph to be replaced by a formal product of its blocks and vice versa.

[T3]: $G \longrightarrow B_1 B_2 \cdots B_k$ where the B_i are the blocks of G,
[T4]: $B_1 B_2 \cdots B_k \longrightarrow G$ where G is a graph with blocks B_i, $1 \leq i \leq k$,

This enables us to write the following simple schema for certificates of Tutte equivalence for pairs of graphs with the same blocks.

$$\begin{aligned} G &= \prod_{i=1}^{k} B_i \qquad \text{(Applying } \mathbf{T3}\text{)} \\ &= H \qquad \text{(Applying } \mathbf{T4}\text{).} \end{aligned} \tag{9}$$

Effectively, this schema first 'unglues' blocks then 'glues' them back together to produce graph H. This certificate schema works for all pairs of graphs that have the same blocks and may be regarded as a representation of the set of all such pairs.

Schemas 1 and 2 in [12] give two of the shortest certificates for pairs of chromatically equivalent graphs, G and H. In both these schemas, graph H can be obtained by removing an edge from graph G and then adding a different edge.

Schema 2 relates pairs of chromatically equivalent graphs G and H with $G \setminus e \cong H \setminus f$ and $G/e \cong H/f$, where $e \in E(G)$ and $f \in E(H)$. Applying two deletion-contraction steps, we have:

$$\begin{aligned} G &= G \setminus e - G/e \\ &= (G \setminus e) + f \end{aligned} \tag{10}$$

where $f \notin E(G)$ and $G \setminus e + f \cong H$. The second step, (10), is obtained by rearranging the usual deletion-contraction relation.

Schema 1 is similar to Schema 2, but uses the addition-identification relation. Here $(G + e) \setminus f \cong H$. Applying two addition-identification steps, we have:

$$\begin{aligned} G &= G + e + G/e \\ &= (G + e) \setminus f \end{aligned} \tag{11}$$

where $e \notin E(G)$ and $f \in E(G)$.

More sophisticated certificates of equivalence are available. In [12], shortest certificates of chromatic equivalence are given for all pairs of chromatically equivalent graphs of order at most 7. These corresponded to 15 different schemas. It should be noted that a shortest certificate of equivalence may not be unique. In [89], infinitely many pairs of chromatically equivalent non-isomorphic graphs are constructed along with their certificates of equivalence.

The length of certificates has implications for the complexity of testing equivalence with respect to these polynomials (chromatic equivalence, Tutte equivalence, etc.). A short certificate of equivalence, once obtained, gives a means of verifying that two graphs have the same polynomial without computing the polynomial. For example, if the length of certificates of chromatic equivalence is polynomially bounded, then the problem of testing chromatic equivalence belongs to NP [12]. But at present we only have very loose, exponential upper bounds on certificate length. We discuss some implications of certificate length for the computational complexity of chromatic equivalence, factorisation and uniqueness in [12] and Tutte equivalence in [87].

Appropriate versions of certificates of equivalence and factorisation should be applicable to many other graph polynomials. We would like to see a rigorous theory of certificates of (at least) equivalence and factorisation for a broad class of graph polynomials with reduction relations. The commutativity of the operations of deleting/contracting different edges in a graph may be regarded as an instance of the Church-Rosser property from the theory of rewriting systems, as observed by Yetter [122] and Makowsky [74]; see also [7, result 9m, p. 72]. It seems to us that the theory of rewriting systems could shed more light on the kind of certificates we have considered here.

Acknowledgments We thank Andrew Goodall, János Makowsky, Steven Noble and the referee for their comments. Through this *Festschrift* contribution, it is a pleasure to acknowledge János's far-reaching contributions to the study of graph polynomials, through his mathematics and also through his generosity as a colleague, in sharing ideas, organising meetings and supporting the work of others.

References

1. Arocha, J.L., Llano, B.: Mean value for the matching and dominating polynomial. Discuss. Math. Graph Theory **20**, 57–69 (2000)
2. Ashkin, J., Teller, E.: Statistics of two-dimensional lattices with four components. Phys. Rev. **64**, 178–184 (1943)

3. Averbouch, I., Godlin, B., Makowsky, J.A.: A most general edge elimination polynomial. In: Broersma, H., Erlebach, T., Friedetzky, T., Paulusma D. (eds.), Graph-Theoretic Concepts in Computer Science: 34th International Workshop (WG 2008) (Durham, UK, June/July 2008). Lecture Notes in Computer Science, vol. 5344, pp. 31–42. Springer, Berlin (2008)
4. Averbouch, I., Godlin, B., Makowsky, J.A.: An extension of the bivariate chromatic polynomial. Eur. J. Combin. **31**(1), 1–17 (2010)
5. Beaudin, L., Ellis-Monaghan, J., Pangborn, G., Shrock, R.: A little statistical mechanics for the graph theorist. Discrete Math. **310**, 2037–2053 (2010)
6. Benson, D.B.: Life in the game of Go. Inf. Sci. **17**, 17–29 (1976)
7. Biggs, N.L.: Algebraic Graph Theory (2nd edn.). Cambridge University Press, Cambridge (1993)
8. Birkhoff, G.D.: A determinant formula for the number of ways of coloring a map. Ann. Math. **14**, 42–46 (1912–1913)
9. Bläser, M., Dell, H., Makowsky, J.A.: Complexity of the Bollobás-Riordan polynomial. Exceptional points and uniform reductions. Theory Comput. Syst. **46**(4), 690–706 (2010)
10. Bohn, A.: Chromatic roots as algebraic integers. In: 24th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2012) (Nagoya, Japan, 30 July–3 Aug 2012), pp. 539–550 (2012)
11. Borzacchini, L., Pulito, C.: On subgraph enumerating polynomials and Tutte polynomials. Boll. Un. Mat. Ital. B **1**, 589–597 (1982)
12. Bukovac, Z., Farr, G., Morgan, K.: Short certificates for chromatic equivalence. J. Graph Algorithms Appl. **23**(2), 227–269 (2019)
13. Cameron, P.J.: Cycle index, weight enumerator, and Tutte polynomial. Electron. J. Combin. **9**, N2, 10pp. (2002)
14. Cameron, P.J.: Orbit counting and the Tutte polynomial. In: Grimmett G.R., McDiarmid, C.J.H. (eds.), Combinatorics, Complexity and Chance: A Tribute to Dominic Welsh, pp. 1–10. Oxford University Press, Oxford (2007)
15. Cameron, P.J., Jackson, B., Rudd, J.D.: Orbit-counting polynomials for graphs and codes. Discrete Math. **308**(5–6), 920–930 (2008)
16. Cameron P.J., Morgan, K.: Algebraic properties of chromatic roots. Electron. J. Combin. **24**, P1.21 (2017)
17. Chmutov, S.: Topological extensions of the Tutte polynomial. In: Ellis-Monaghan, J., Moffatt, I. (eds.), Handbook on the Tutte Polynomial and Related Topics, chap. 27, pp. 497–513. Chapman and Hall/CRC Press, Boca Raton (2022)
18. Courcelle, B., Makowsky, J.A., Rotics, U.: On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. Discrete Appl. Math. **108**(1–2), 23–52 (2001)
19. de la Harpe, P., Jaeger, F.: Chromatic invariants for finite graphs: theme and polynomial variations. Linear Algebra Appl. **226–228**, 687–722 (1995)
20. de Mier Vinu'e, A.: Graphs and matroids determined by their Tutte polynomials. PhD Thesis, Universitat Politècnica de Catalunya (2003)
21. Delbourgo, D., Morgan, K.: Algebraic invariants arising from chromatic polynomials of theta graphs. Aust. J. Combin. **59**(2), 293–310 (2014)
22. Dohmen, K., Pönitz, A., Tittmann, P.: A new two-variable generalization of the chromatic polynomial. Discrete Math. Theor. Comput. Sci. **6**, 069–090 (2003)
23. Dong, F., Koh, K.M.: Foundations of the chromatic polynomial. In: Ellis-Monaghan, J., Moffatt, I. (eds.), Handbook on the Tutte Polynomial and Related Topics, chap. 11, pp. 213–251. Chapman and Hall/CRC Press, Boca Raton (2022)
24. Ellis-Monaghan, J., Goodall, A., Makowsky, J.A., Moffatt, I.: Graph polynomials: towards a comparative theory (Dagstuhl seminar 16241). Dagstuhl Rep. **6**(6), 26–48 (2016). https://doi.org/10.4230/DagRep.6.6.26. (Schloss Dagstuhl - Leibniz-Zentrum für Informatik)
25. Ellis-Monaghan, J., Goodall, A., Moffatt, I., Morgan, K.: Comparative theory for graph polynomials (Dagstuhl Seminar 19401). Dagstuhl Rep. **9**(9), 135–155 (2019). https://doi.org/10.4230/DagRep.9.9.135. (Schloss Dagstuhl - Leibniz-Zentrum für Informatik)

26. Ellis-Monaghan, J., Moffatt, I.: Graphs on Surfaces: Dualities, Polynomials, and Knots. Springer, Berlin (2013)
27. Ellis-Monaghan, J., Moffatt, I. (eds.): Handbook on the Tutte Polynomial and Related Topics. Chapman and Hall/CRC Press, Boca Raton (2022)
28. Essam, J.W.: Graph theory and statistical physics. Discrete Math. **1**, 83–112 (1971)
29. Farr, G.E.: A correlation inequality involving stable set and chromatic polynomials. J. Combin. Theory (Ser. B) **58**, 14–21 (1993)
30. Farr, G.E.: A generalization of the Whitney rank generating function. Math. Proc. Camb. Phil. Soc. **113**, 267–280 (1993)
31. Farr, G.E.: On problems with short certificates. Acta Inf. **31**, 479–502 (1994)
32. Farr, G.E.: The Go polynomials of a graph. Theor. Comput. Sci. **306**, 1–18 (2003)
33. Farr, G.E.: Some results on generalised Whitney functions. Adv. Appl. Math. **32**, 239–262 (2004)
34. Farr, G.E.: Tutte-Whitney polynomials: some history and generalizations. In: Grimmett, G.R., McDiarmid, C.J.H. (eds.), Combinatorics, Complexity and Chance: A Tribute to Dominic Welsh, pp. 28–52. Oxford University Press, Oxford (2007)
35. Farr, G.E.: On the Ashkin-Teller model and Tutte-Whitney functions. Combin. Probab. Comput. **16**, 251–260 (2007)
36. Farr, G.E.: Transforms and minors for binary functions. Ann. Combin. **17**, 477–493 (2013)
37. Farr, G.E.: The probabilistic method meets Go. J. Korean Math. Soc. **54**, 1121–1148 (2017)
38. Farr, G.E.: Binary functions, degeneracy, and alternating dimaps. Discrete Math. **342**(5), 1510–1519 (2019)
39. Farr, G.E.: Using Go in teaching the theory of computation. SIGACT News **50**(1), 65–78 (2019)
40. Farr, G.E.: The history of Tutte–Whitney polynomials. In: Ellis-Monaghan, J., Moffatt, I. (eds.), Handbook on the Tutte Polynomial and Related Topics, chap. 34, pp. 623–668. Chapman and Hall/CRC Press, Boca Raton (2022)
41. Farr, G.E.: The forced colouring function of a graph (in preparation)
42. Farr, G.E., Schmidt, J.: On the number of Go positions on lattice graphs. Inf. Process. Lett. **105**, 124–130 (2008)
43. Farrell, E.J.: An introduction to matching polynomials. J. Combin. Theory Ser. B **27**, 75–86 (1979)
44. Farrell, E.J.: On a class of polynomials associated with the cliques in a graph and its applications. Int. J. Math. Math. Sci. **12**, 77–84 (1989)
45. Fiorini, S., Wilson, R.J.: Edge-Colourings of Graphs. Research Notes in Mathematics, vol. 16. Pitman, New Jersey (1977)
46. Fortuin, C.M., Kasteleyn, P.W.: On the random cluster model. I. Introduction and relation to other models. Physica **57**, 536–564 (1972)
47. Gavril, F.: Algorithms on clique separable graphs. Discrete Math. **19**, 159–165 (1977)
48. Godlin, B., Katz, E., Makowsky, J.A.: Graph polynomials: from recursive definitions to subset expansion formulas. J. Logic Comput. **22**(2), 237–265 (2012)
49. Godsil, C.D., Gutman, I.: On the theory of the matching polynomial. J. Graph Theory **5**, 137–144 (1981)
50. Goodall, A., Hermann, M., Kotek, T., Makowsky, J.A., Noble, S.D.: On the complexity of generalized chromatic polynomials. Adv. Appl. Math. **94**, 71–102 (2018)
51. Goodall, A.J., Nešetřil, J., Ossona de Mendez, P.: Strongly polynomial sequences as interpretations. J. Appl. Logic **18**, 129–149 (2016)
52. Greenlaw, R., Hoover, H.J., Ruzzo, W.L.: Limits to Parallel Computation: P-Completeness Theory. Oxford University Press, Oxford (1995)
53. Gutman, I., Harary, F.: Generalizations of the matching polynomial. Utilitas Math. **24**, 97–106 (1983)
54. Harary, F.: Conditional colorability in graphs. In Harary, F., Maybee, J.S. (eds.), Graphs and Applications (Boulder, Colo., 1982), pp. 127–136. Wiley, New York (1985)
55. Harary, F., Palmer, E.: Graphical Enumeration. Academic Press, New York (1973)

56. Heilmann, O.J., Lieb, E.H.: Monomers and dimers. Phys. Rev. Lett. **24**(25), 1412–1414 (1970)
57. Helgason, T.: Aspects of the theory of hypermatroids. In: Berge, C., Ray-Chaudhuri, D.K. (eds.) Hypergraph Seminar. Lecture Notes in Mathematics, vol. 411, pp. 191–213. Springer, Berlin (1974)
58. Herscovici, O., Makowsky, J.A., Rakita, V.: Harary polynomials. Enumer. Comb. Appl. **1**(2), Paper No. S2R13, 12pp. (2021)
59. Huggett, S.: The Tutte polynomial and knot theory. In: Ellis-Monaghan, J., Moffatt, I. (eds.), Handbook on the Tutte Polynomial and Related Topics, chap. 18, pp. 352–367. Chapman and Hall/CRC Press, Boca Raton (2022)
60. Hunt, T.: Milton Keynes Go Board. British Go Association (2013). https://www.britgo.org/clubs/mk/mkboard.html
61. Ising, E.: Beitrag zur Theorie des Ferromagnetismus. Z. Phys. **31**, 253–258 (1925)
62. Iwamoto, K.: Go for Beginners. Ishi Press/Penguin Books, Bronx/London (1972/1976)
63. Kotek, T.: Definability of combinatorial functions. Ph.D. Thesis, Computer Science Department, Technion (2012)
64. Kotek, T.A., Makowsky, J.A.: The exact complexity of the Tutte polynomial. In: Ellis-Monaghan, J., Moffatt, I. (eds.), Handbook on the Tutte Polynomial and Related Topics, chap. 9, pp. 175–193. Chapman and Hall/CRC Press, Boca Raton (2022)
65. Kotek, T.A., Makowsky, J.A., Ravve, E.V.: A computational framework for the study of partition functions and graph polynomials. In: Downey, R., Brendle, J., Goldblatt, R., Kim, B. (eds.), Proceedings of 12th Asian Logic Conference (Wellington, 15–20 December 2011), pp. 210–230. World Scientific, Hackensack (2013)
66. Kotek, T., Makowsky, J.A., Ravve, E.V.: On sequences of polynomials arising from graph invariants. Eur. J. Combin. **67**, 181–198 (2018)
67. Kotek, T., Makowsky, J.A., Zilber, B.: On counting generalized colorings. In: Kaminski, M., Martini, S. (eds.), Computer Science Logic: Proceedings of 22nd International Workshop (CSL 2008), 17th Annual Conference of the EACSL (Bertinoro, 16–19 September 2008). Lecture Notes in Computer Science, vol. 5213, pp. 339–353. Springer, Berlin (2008)
68. Kotek, T., Makowsky, J.A., Zilber, B.: On counting generalized colorings. In: Grohe, M., Makowsky, J.A. (eds.) Model Theoretic Methods in Finite Combinatorics. Contemporary Mathematics, vol. 558, pp. 207–241. American Mathematical Society, Providence (2011)
69. Krajewski, T., Moffatt, I., Tanasa, A.: Hopf algebras and Tutte polynomials. Adv. Appl. Math. **95**, 271–330 (2018)
70. Lotz, M., Makowsky, J.A.: On the algebraic complexity of some families of coloured Tutte polynomials. Adv. Appl. Math. **32**(1–2), 327–349 (2004)
71. Makowsky, J.A.: Algorithmic uses of the Feferman-Vaught Theorem. Ann. Pure Appl. Logic **126**, 159–213 (2004)
72. Makowsky, J.A.: Coloured Tutte polynomials and Kauffman brackets for graphs of bounded tree width. Discrete Appl. Math. **145**(2), 276–290 (2005)
73. J.A. Makowsky, From a zoo to a zoology: descriptive complexity for graph polynomials. In: Beckmann, A., Berger, U., Löwe, B., Tucker, J.V. (eds.), Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, CiE 2006 (Swansea, UK, July 2006). Lecture Notes in Computer Science, vol. 3988, pp. 330–341. Springer, Berlin (2006)
74. Makowsky, J.A.: From a zoo to a zoology: towards a general theory of graph polynomials. Theory Comput. Syst. **43**, 542–562 (2008)
75. Makowsky, J.A.: The graph polynomials project in Haifa (2012). https://janos.cs.technion.ac.il/RESEARCH/gp-homepage.html. Accessed 23 Feb. 2024
76. Makowsky, J.A.: How I got to like graph polynomials. (arXiv preprint 2023). https://arxiv.org/abs/2309.02933v1
77. Makowsky, J.A.: Meta-theorems for graph polynomials. In: Wood, D.R., Etheridge, A.M., de Gier, J., Joshi, N. (eds.), 2023 MATRIX Annals. Springer, Berlin (in press)

78. Makowsky, J.A., Mariño, J.P.: Farrell polynomials on graphs of bounded tree width. Adv. Appl. Math. **30**(1–2), 160–176 (2003)
79. Makowsky, J.A., Meer, K.: On the complexity of combinatorial and metafinite generating functions of graph properties in the computational model of Blum, Shub and Smale. In: Clote, P.G., Schwichtenberg, H. (eds.), Computer Science Logic: Proceedings of 14th International Workshop (CSL 2000), Annual Conference of the EACSL (Fischbachau, 21–26 August 2000). Lecture Notes in Computer Science, vol. 1862, pp. 399–410. Springer, Berlin, (2000)
80. Makowsky, J.A., Ravve, E.V.: Semantic equivalence of graph polynomials definable in second order logic. In: Väänänen, J., Hirvonen, Å., de Queiroz, R. (eds.), Logic, Language, Information, and Computation: Proceedings of 23rd International Workshop (WoLLIC 2016) (Benemérita Universidad Autónoma de Puebla, Puebla, 16–19 August 2016). Lecture Notes in Computer Science, vol. 9803, pp. 279–296. Springer, Berlin (2016)
81. Makowsky, J.A., Ravve, E.V., Blanchard, N.K.: On the location of roots of graph polynomials. Eur. J. Combin. **41**, 1–19 (2014)
82. Makowsky, J.A., Ravve, E.V., Kotek, T.: A logician's view of graph polynomials. Ann. Pure Appl. Logic **170**, 1030–1069 (2019)
83. Makowsky, J.A., Zilber, B.: Polynomial invariants of graphs and totally categorical theories. MODNET Preprint No. 21 (2006). https://modnet.imj-prg.fr/Publications/Preprint%20server/papers/21/
84. Martin, P.: Anneau de Tutte-Grothendieck associé aux dénombrements eulériens dans les graphes 4-réguliers planaires, Colloque sur la Théorie des Graphes (Brussels, 1973). Cahiers Centre Études Recherche Opér. **15**, 343–349 (1973)
85. Merino, C., de Mier, A., Noy, M.: Irreducibility of the Tutte polynomial of a connected matroid. J. Combin. Theory **83**, 298–304 (2001)
86. Mo, R., Farr, G., Morgan, K.: Certificates for properties of stability polynomials of graphs. Electron. J. Combin. **21**, P1.66 (25pp.) (2014)
87. Mo, R., Morgan, K., Farr, G.: Certificates for Tutte equivalence (to be submitted)
88. Morgan, K.: Algebraic aspects of the chromatic polynomial. Ph.D. Thesis, Monash University (2010). https://doi.org/10.4225/03/587852a8b487b
89. Morgan, K.: Pairs of chromatically equivalent graphs. Graphs Combin. **27**(4), 547–556 (2010)
90. Morgan, K.: Galois groups of chromatic polynomials. LMS J. Comput. Math. **15**, 281–307 (2012)
91. Morgan, K., Chen, R.: An infinite family of 2-connected graphs that have reliability factorisations. Discrete Appl. Math. **218**, 123–127 (2017)
92. Morgan, K., Farr, G.: Certificates of factorisation for chromatic polynomials. Electron. J. Combin. **16**, R74 (29pp.) (2009)
93. Morgan, K., Farr, G.: Certificates of factorisation for a class of triangle-free graphs. Electron. J. Combin. **16**, R75 (14pp.) (2009)
94. Morgan, K., Farr, G.: Non-bipartite chromatic factors. Discrete Math. **312**, 1166–1170 (2012)
95. Morse, M.: George David Birkhoff and his mathematical work. Bull. Am. Math. Soc. **52**, 357–391 (1946)
96. Noble, S.D.: The U, $\mathbf{V}$, and W polynomials. In: Ellis-Monaghan, J., Moffatt, I. (eds.), Handbook on the Tutte Polynomial and Related Topics, chap. 26, pp. 7733–496. Chapman and Hall/CRC Press, Boca Raton (2022)
97. Noble, S.D., Welsh, D.J.A.: A weighted graph polynomial from chromatic invariants of knots. Ann. Inst. Fourier (Grenoble) **49**(3), 1057–1087 (1999)
98. Oxley, J.G., Whittle, G.P.: Tutte invariants for 2-polymatroids. In: Robertson, N., Seymour, P.D. (eds.) Graph Structure Theory (Seattle, 1991). Contemporary Mathematics, vol. 147, pp. 9–19. American Mathematical Society, Providence (1993)
99. Oxley, J., Whittle, G.: A characterization of Tutte invariants of 2-polymatroids. J. Combin. Theory Ser. B **59**, 210–244 (1993)
100. Pólya, G.: Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen. Acta Math. **68**, 145–254 (1937)

101. Potts, R.B.: Some generalized order-disorder transformations. Proc. Camb. Phil. Soc. **48**, 106–109 (1952)
102. Redfield, H.: The theory of group-reduced distributions. Am. J. Math. **49**, 433–455 (1927)
103. Segerman, H.: Diamond go. https://www.segerman.org/diamond/
104. Sensei's Library: Tromp-Taylor Rules. https://senseis.xmp.net/?TrompTaylorRules. Accessed 19 Feb. 2024
105. Stanley, R.P.: Acyclic orientations of graphs. Discrete Math. **5**, 171–178 (1973)
106. Thorp, E., Walden, W.E.: A partial analysis of Go. Computer J. **7**(3), 203–207 (1964)
107. Thorp, E., Walden, W.E.: A computer assisted study of Go on $M \times N$ boards. Inf. Sci. **4**(1), 1–33 (1972)
108. Tittmann, P.: Graph Polynomials: The Eternal Book. Chapman and Hall/CRC, Boca Raton (2024). https://www.researchgate.net/publication/377572474_Graph_Polynomials_The_Eternal_Book
109. Trinks, M.: The covered components polynomial: a new representation of the edge elimination polynomial. Electron. J. Combin. **19**, P50 (31pp.) (2012)
110. Tromp, J., Taylor, W.: The game of Go. https://tromp.github.io/go.html. Accessed 19 Feb. (2024)
111. Tutte, W.T.: A ring in graph theory. Proc. Camb. Phil. Soc. **43**, 26–40 (1947)
112. Tutte, W.T.: A contribution to the theory of chromatic polynomials. Can. J. Math. **6**, 80–91 (1954)
113. Tutte, W.T.: Codichromatic graphs. J. Combin. Theory Ser. B **16**, 168–174 (1974)
114. Tutte, W.T.: Graph Theory as I Have Known It. Oxford University Press, Oxford (1998)
115. Tutte, W.T.: Graph-polynomials. Adv. Appl. Math. **32**, 5–9 (2004)
116. Van Slyke, R., Frank, H.: Network reliability analysis. I. Networks **1**, 279–290 (1971/72)
117. Welsh, D.J.A.: Matroid Theory. London Mathematical Society Monograph, vol. 8. Academic Press, London (1976)
118. Welsh, D.J.A.: Complexity: Knots, Colourings and Counting. London Mathematical Society Lecture Note Series, vol. 186. Cambridge University Press, Cambridge (1993)
119. Whitney, H.: The Coloring of Graphs. Ph.D. Thesis, Harvard University (1932)
120. Whitney, H.: The coloring of graphs. Ann. Math. (2) **33**, 688–718 (1932)
121. Whitney, H.: A set of topological invariants for graphs. Am. J. Math. **55**, 231–235 (1933)
122. Yetter, D.N.: On graph invariants given by linear recurrence relations. J. Combin. Theory (Ser. B) **48**, 6–18 (1990)
123. Zykov, A.A.: On some properties of linear complexes (in Russian). Math. Sbornik **24**, 163–188 (1949). English translation: Amer. Math. Soc. Transl. **79** (1952)
124. Zykov, A.A.: Recursively calculable functions of graphs. In: Theory of Graphs and its Applications (Proceedings of the Symposium Smolenice, 1963), pp. 99–105. Publishing House of the Czechoslovak Academy of Sciences, Prague (1964)

Pixelating Relations and Functions Without Adding Substructures

Eldar Fischer

To Janos for his 75th birthday, and more than two decades of collaboration between us.

Abstract We investigate models of relations over a bounded continuous segment of real numbers, along with the natural linear order over the reals being provided as a "hard-coded" relation. This paper presents a generalization of a lemma from Ben-Eliezer et al. (Ordered graph limits and their applications. In: Lee, J.R. (ed.) 12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6–8, 2021, Virtual Conference. LIPIcs, vol. 185, pp. 42:1–42:20. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2021)), showing that with a small amount of modification (measured in terms of the Lebesgue measure) we can replace such a model with a "pixelated" one that has a finite description, in a way that preserves all universally quantified statements over the relations, or in other words, without adding any new substructures.

1 Introduction

Consider a vocabulary with one relation of arity d (later we discuss generalizations to more relations), along with a binary relation "$\leq$" which is promised to be a linear order. Moreover, the order relation should be thought of as "hard-coded", that is, determined in advance along with the universe U. We will be concerned with two specific examples. The first one is the case where $U = [n]$, with $\leq$ being the natural order among the integers. The second case is where $U = \{x \in \mathbb{R} : 0 < x \leq 1\}$ is a finite segment of real numbers, with $\leq$ being the natural order among them (for the specific investigation here it will not matter much whether 0 and/or 1 themselves are contained in U).

E. Fischer (✉)
Technion – Israel Institute of Technology, Haifa, Israel
e-mail: eldar@cs.technion.ac.il

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_19

We investigate the question of approximating continuous models over a segment of real numbers by discrete (finite universe) models. The notion of approximation here is tied with the Lebesgue measure, or equivalently the probability of finding a difference when uniformly drawing a tuple $(x_1, \ldots, x_d) \in U$. To be able to compare a continuous model with a discrete model over $[n]$, one would think of "stretching" the latter so that $i \in [n]$ will be identified with the set $\{x \in \mathbb{R} : (i-1)/n < x \leq i/n\}$.

However, the above idea has a problem, exemplified best by the result of drawing x and y from U. In the continuous case, there is zero probability for $x = y$, while for $U = [n]$ this probability is positive. For this reason we will use a slightly more intricate definition of a discrete approximation, one that still has a finite description, while allowing for the additional variations according to the order between x and y along the "diagonal". This is formalized in Definition 5 below.

We must also define what must be preserved when moving to the approximation. Here we will preserve universally quantified sentences. This means that when moving from a model M to a new model M', there will be no sub-model of M' that does not appear already in M. In fact, there will be no sub-models of M' apart from those that appear with *positive probability* in M, when uniformly drawing the elements $x_1 < \cdots < x_n$ from U that compose the universe of the sub-model.

This approximation is what we call the "pixelated version" of the model for the relation. We present here a generalization of a lemma from [1], which was originally used there for dealing with limit objects of vertex-ordered simple graphs. We show here that for any fixed parameter ϵ, a relation can be ϵ-approximated by a pixelated version, namely an l-part homogeneous relation (as per Definition 5 below). The "fineness parameter" l can depend on the original relation itself, which is to be expected, especially that the result holds for sub-models of all finite orders n at once.

For logical concepts, we refer the reader to [2]. However, we will use surprisingly little logic in the following. Essentially we will use an "encoding" of the logical models using functions. For example, a model for a vocabulary that includes r relations of arity d over the universe $[n]$, along with the hard-coded natural order $\leq$ over $[n]$, can be represented as r subsets $R_1, \ldots, R_r \subseteq [n]^d$. However, it can also be encoded using a single function $R : [n]^d \to [k]$ for $k = 2^r$, where $R(i_1, \ldots, i_d)$ provides all information about which of the R_i contain the tuple $(i_1, \ldots, i_d)$. We will work primarily with the later representation.

The proof here follows the general structure of the proof of the special-case lemma from [1], with multiple adaptations for the generalization. Some basic concepts, in particular from measure theory, are presented in Sect. 2. For completeness we supply proofs for the very basic lemmas that we use from measure theory. After some additional necessary definitions, we present the main result, Theorem 1, in Sect. 3. We also demonstrate that the bulk of the work is mandated by the "no new substructures" requirement, by proving a very simple lemma that omits this requirement.

Section 4 contains more definitions that are required for the proof mechanism. In Sect. 5 we provide a proof of a Ramsey-type lemma that refers to a scenario

involving hypergraphs with vertices of multiple "sorts", along with an adaptation for the main proof. This Ramsey-type lemma is much more intricate than the lemma that is used in [1]. This is caused by the combination of the move from graphs to higher arity relations with the requirement of selecting sufficiently many vertices of every sort (which is much easier to satisfy for the binary relation case).

Section 6 finally provides the proof of Theorem 1. It is derived from the simple lemma from Sect. 3, using a key lemma about the positive probability appearance "discrete versions" of pixelated functions in the function to be approximated.

The final Sect. 7 contains a discussion of Theorem 1 and some closely related (and easy to derive) variants, touching also on the role that its special case plays in [1] for proving a removal lemma for vertex-ordered graphs using limit objects.

2 The Basics

Let $\mathbb{I}$ denote the interval of real numbers $\{x \in \mathbb{R} : 0 < x \leq 1\}$, and for any natural number k let $[k]$ denote the set of integers $\{1, \ldots, k\}$. It will make no essential difference to exclude 0 from $\mathbb{I}$, since we are concerned with the natural linear order along with the Lebesgue measure (with $\{0\}$ being a set of measure zero), and we will deal only with universally quantified sentences (and having a minimal element is not thus expressible). Not including 0 in $\mathbb{I}$ will simplify notation later on. We also define for any $l \in \mathbb{N}$ the disjoint subintervals $\mathbb{I}_{i,l} = \{x \in \mathbb{I} : (i-1)/l < x \leq i/l\}$ where $i \in [l]$.

The notion of Lebesgue-measurable sets and functions over $\mathbb{I}^d$ plays a major role here. We refer the reader to [4] for a primer on measure theory, although we will only use here the most basic definitions, and provide self-contained proofs for all else.

We use $\lambda(A)$ to denote the Lebesgue measure of a measurable set A. For two sets A and B we denote by $A \Delta B$ their symmetric difference. In particular we have the following well known and almost immediate consequence of the definition of measurability.

Lemma 1 *For every measurable set $A \subseteq \mathbb{I}^d$ and ϵ there exists l, so that for every $l' \geq l$ there is a set B consisting of a (disjoint) union of sets of the type $\prod_{j=1}^{d} \mathbb{I}_{i_j,l'}$, so that $\lambda(B \Delta A) \leq \epsilon$.*

Proof By definition of the measurability of A, there exists a set B_0 such that $A \subset B_0$, where B_0 is the union of countably many (not necessarily disjoint) "boxes" $C_1, C_2, \ldots$, with every C_j being of the form $\{(x_1, \ldots, x_d) \in \mathbb{I}^d : \bigwedge_{i=1}^{d} \alpha_{j,i} \leq x_i \leq \beta_{j,i}\}$, and such that the sum of measures $\sum_j \prod_{i=1}^{d} (\beta_{j,i} - \alpha_{j,i})$ is at most $\lambda(A) + \epsilon/3$. Now we set t so that $\sum_{j>t} \prod_{i=1}^{d} (\beta_{j,i} - \alpha_{j,i}) \leq \epsilon/3$ (such a t exists because the whole sum is finite), and set $B_1 = \bigcup_{j \leq t} C_j$. Clearly $\lambda(A \setminus B_1) \leq \epsilon/3$ (since B_0 contains A and $B_0 \setminus B_1 \subseteq \bigcup_{j>t} C_j$), and also $\lambda(B_1 \setminus A) \leq \epsilon/3$ (since $\lambda(B_0 \setminus A) \leq \epsilon/3$), hence $\lambda(A \Delta B_1) \leq 2\epsilon/3$.

Now we set $l = \lceil 6dt/\epsilon \rceil$. For any $l' \geq l$, we define B to be the (disjoint) union of all sets $\prod_{j=1}^{d} \mathbb{I}_{i_j,l'}$ that have a non-empty intersection with B_1. Note that $B_1 \subseteq B$. Also, for any set C of the form $\{(x_1, \ldots, x_d) \in \mathbb{I}^d : \bigwedge_{i=1}^{d} \alpha_i \leq x_i \leq \beta_i\}$, the measure of the union of all sets of the type $\prod_{j=1}^{d} \mathbb{I}_{i_j,l'}$ that are not contained in C and yet have a non-empty intersection with it is at most $2d/l'$. Hence $\lambda(B \setminus B_1) \leq 2dt/l' \leq \epsilon/3$, and so $\lambda(A \Delta B) \leq \epsilon$. □

We make the natural identification of the Lebesgue measure over $\mathbb{I}^d$ with the uniform probability space over this set, so in particular the legitimate events over this probability space are exactly the measurable sets, and for any such set A we identify $\Pr[A]$ with $\lambda(A)$. Another almost immediate consequence is the following.

Lemma 2 *If A is a positive probability event over $\mathbb{I}^d$, then for every $\epsilon > 0$ there exists a "box" set $C = \{(x_1, \ldots, x_d) \in \mathbb{I}^d : \bigwedge_{i=1}^{d} \alpha_i \leq x_i \leq \beta_i\}$ (which in particular is also a legitimate event) for which $\Pr[A|C] \geq 1 - \epsilon$.*

Proof Similarly to the proof of Lemma 1, we start with a set B such that $A \subset B$, where B is the union of countably many (not necessarily disjoint) "boxes" $C_1, C_2, \ldots$ with $C_j = \{(x_1, \ldots, x_d) \in \mathbb{I}^d : \bigwedge_{i=1}^{d} \alpha_{j,i} \leq x_i \leq \beta_{j,i}\}$, satisfying $\sum_j \lambda(C_j) \leq (1 + \epsilon)\lambda(A)$ (we use the assumption that $\lambda(A) > 0$ when we pick "$\epsilon\lambda(A)$" as the additive term). This in particular means that $\sum_j \lambda(C_j) \leq (1 + \epsilon) \sum_j \lambda(A \cap C_j)$, so there exists a particular j for which $\lambda(C_j) \leq (1+\epsilon)\lambda(A \cap C_j)$. Hence $\Pr[A|C_j] = \lambda(A \cap C_j)/\lambda(C_j) \geq 1 - \epsilon$ as required. □

We will work with "combinatorial models" of relations. For example, a model of an arity d relation over the universe $[n]$ would be a set $R \subseteq [n]^d$, or alternatively a function $R : [n]^d \to \{0, 1\}$. A relation of arity $k < d$ can still be represented by a function over $[n]^d$, by making it invariant of the last $n - k$ coordinates.

However, for our purposes (and convenience) we would like to deal with a single function, rather than a function for every relation. To achieve this we allow a larger range, so our main object would be a function $R : [n]^d \to [k]$ for some constant k, where for $(i_1, \ldots, i_d) \in [n]^d$ the value $R(i_1, \ldots, i_d)$ provides all information about the existence of this tuple in all relations. For example, for a vocabulary having exactly r relations of arity d, then we would correspondingly have $k = 2^r$, and we would use a correspondence between the values in $[k]$ and the vectors in $\{0, 1\}^r$. Thus the single value $R(i_1, \ldots, i_d)$ would tell us which of the relations contain the tuple $(i_1, \ldots, i_d)$.

Another important point is that we maintain the natural order over $[n]$ as a fixed order of the universe. The bulk of this work is about transforming a model in a way that maintains sentences with universal quantifiers, which translates to transforming a model in a way that does not add any new order-preserving "substructures" that did not exist in the original model.

We will deal with the scenario where the universe is not finite, and is not even countable. Specifically, we deal with functions $F : \mathbb{I}^d \to [k]$. The main theme of this work is to show that these can be approximated by "essentially finite" models, but first we need more clarifications.

We will only deal with measurable functions, that is, functions $F : \mathbb{I}^d \to [k]$ for which the preimage set $F^{-1}(i)$ is Lebesgue-measurable for every $i \in [k]$. In particular, for the uniform probability distribution over $\mathbb{I}^d$, every set $F^{-1}(i)$ corresponds to a legitimate probabilistic event "$F(x_1, \ldots, x_d) = i$". We will use the Hamming distance between functions.

Definition 1 For two measurable functions $F, G : \mathbb{I}^d \to [k]$, the *distance* between them is $d(F, G) = \Pr[F(x_1, \ldots, x_d) \neq G(x_1, \ldots, x_d)]$, where $(x_1, \ldots, x_d)$ is drawn uniformly from $\mathbb{I}^d$. Alternatively we can write $d(F, G) = \lambda(A)$, where $A = \{(x_1, \ldots, x_d) \in \mathbb{I}^d : F(x_1, \ldots, x_d) \neq G(x_1, \ldots, x_d)\}$ is the set of differences.

Given a function $F : \mathbb{I}^d \to [k]$, we can go back and pick functions $R : [n]^d \to [k]$ by picking values $0 < x_1 < \cdots < x_n \leq 1$ for the coordinates, and then defining R by the appropriate restriction. By using a uniformly random choice for $0 < x_1 < \cdots < x_n \leq 1$ we obtain a probability distribution over these functions.

Definition 2 Given a measurable function $F : \mathbb{I}^d \to [k]$, the $[n]^d$*-statistic distribution* is the (finite) probability distribution $\mu_{F,n}$ over functions $R : [n]^d \to [k]$ that results from picking $0 < x_1 < \cdots < x_n \leq 1$ uniformly from the set of all such sequences, and then setting $R(i_1, \ldots, i_d) = F(x_{i_1}, \ldots, x_{i_d})$ for every $(i_1, \ldots, i_d) \in [n]^d$.

Note that in the above definition, a uniform choice of $0 < x_1 < \cdots < x_n \leq 1$ can be obtained by picking $(y_1, \ldots, y_n) \in \mathbb{I}^n$ uniformly (and independently) at random, and then setting x_j to be the j'th smallest value among $y_1, \ldots, y_n$ for every $j \in [n]$ (here and in other places we ignore the probability zero event that $y_j = y_{j'}$ for some $1 \leq j < j' \leq n$).

We also define what it means for a certain R to appear in F, with or without positive probability.

Definition 3 For a measurable function $F : \mathbb{I}^d \to [k]$ and a function $R : [n]^d \to [k]$, we say that R *appears in* F if there exist $0 < x_1 < \cdots < x_n \leq 1$ so that $F(x_{i_1}, \ldots, x_{i_d}) = R(i_1, \ldots, i_d)$ for every $(i_1, \ldots, i_d) \in [n]^d$. We say that R *appears in* F *with positive probability* if we additionally have $\mu_{F,n}(R) > 0$, where $\mu_{F,n}$ is the $[n]^d$-statistic probability distribution.

For the analysis, we will also define an appearance of R in a discrete $S : [m]^d \to [k]$ (in this setting one should think of all appearances as being with positive probability).

Definition 4 For a (discrete) function $S : [m]^d \to [k]$ and a function $R : [n]^d \to [k]$, we say that R *appears in* S if there exist $1 \leq j_1 < \cdots < j_n \leq m$ so that $S(j_{i_1}, \ldots, j_{i_d}) = R(i_1, \ldots, i_d)$ for every $(i_1, \ldots, i_d) \in [n]^d$.

3 Presentation of the Main Result

We will be particularly interested in functions $F : \mathbb{I}^d \to [k]$ that "do not use the infiniteness of $\mathbb{I}$". The definition follows.

Definition 5 A function $F : \mathbb{I}^d \to [k]$ is called *l-part homogeneous* if $F(x_1, \ldots, x_d)$ depends only on $\lceil lx_1 \rceil, \ldots, \lceil lx_d \rceil$ and on the order of $x_1, \ldots, x_d$ (i.e. whether $x_j \leq x_{j'}$ and/or $x_j \geq x_{j'}$ for every $1 \leq j < j' \leq d$).

In other words, given $(i_1, \ldots, i_d) \in [l]^d$, the values of $F(x_1, \ldots, x_d)$ inside $\prod_{j=1}^d \mathbb{I}_{i_j,l}$ depend only on the order of $x_1, \ldots, x_d$. Note also that whenever $i_j < i_{j'}$ we have $x_j < x_{j'}$ unconditionally, so the order of $x_1, \ldots, x_d$ is non-determined only when we have $j \neq j'$ for which $i_j = i_{j'}$. The following two observations are quite immediate.

Observation 1 *If $F : \mathbb{I}^d \to [k]$ is l-part homogeneous, then every $R : [n]^d \to [k]$ that appears in F, appears in it with positive probability, and in particular $\mu_{F,n}(R) \geq 1/l^n n!$.* □

Observation 2 *For every l, d and k there is only a finite number of possible l-part homogeneous functions $F : \mathbb{I}^d \to [k]$.* □

It is almost immediate from Lemma 1 that measurable functions can be approximated by l-part homogeneous functions for l large enough (that may depend on the function itself).

Lemma 3 *For every measurable function $F : \mathbb{I}^d \to [k]$ and $\epsilon > 0$ there exists t (that may depend on F and ϵ), so that for every $l \geq t$ there exists an l-part homogeneous function $G : \mathbb{I}^d \to [k]$ for which $d(F, G) \leq \epsilon$.*

Proof In fact we will prove something stronger, and approximate F by a function G that is completely constant over any set of the type $\prod_{j=1}^d \mathbb{I}_{i_j,l}$. Denoting the preimage sets by $A_i = F^{-1}(i)$ for every $1 \leq i \leq k$, we set $t = \max\{l_1, \ldots, l_k\}$ where l_i is provided by Lemma 1 given A_i (as A) with ϵ/k instead of ϵ. Then we use Lemma 1 for every A_i with $l \geq t \geq l_i$ to obtain B_i for which the measure of $B_i \Delta A_i$ is at most ϵ/k.

To define G, we define the function $R : [l]^d \to [k]$ for which $G(x_1, \ldots, x_d) = R(i_1, \ldots, i_d)$ whenever $x_j \in \mathbb{I}_{i_j,l}$ for every $j \in [d]$. For every $(i_1, \ldots, i_d) \in [l]^d$, we look at the set $\{i : \prod_{j=1}^d \mathbb{I}_{i_j,l} \cap B_i \neq \emptyset\}$. If this is a set of size one, we define $R(i_1, \ldots, i_d)$ to be equal to its sole member, and otherwise we take an arbitrary value for $R(i_1, \ldots, i_d)$. Finally, we note that if $F(x_1, \ldots, x_d) \neq G(x_1, \ldots, x_d)$, then there exists some i for which $(x_1, \ldots, x_d) \in A_i \Delta B_i$ (otherwise, since the sets $A_i \cap B_i$ are all disjoint, $(x_1, \ldots, x_d)$ would have belonged to exactly one of them). This implies that $\lambda(\{(x_1, \ldots, x_d) : F(x_1, \ldots, x_d) \neq G(x_1, \ldots, x_d)\}) \leq \sum_{i=1}^k \lambda(A_i \Delta B_i) \leq \epsilon$, as required. □

The main result here states that a measurable function can not only be approximated by a homogeneous one, but that this can be done in a way that does not

introduce any "new artifacts" into this function, in terms of which $R : [n]^d \to [k]$ appear in it (for all n at once).

Theorem 1 *For every measurable function $F : \mathbb{I}^d \to [k]$ and $\epsilon > 0$ there exists an l-part homogeneous function $G : \mathbb{I}^d \to [k]$ (for some l that may depend on F and ϵ) for which $d(F, G) \leq \epsilon$, and furthermore satisfying that every function $R : [n]^d \to [k]$ (for any natural number n) that appears in G already appears with positive probability in F.*

It would have been nice to show this without the dependency on the order of $x_1, \ldots, x_n$ (see Definition 5), but this cannot be done. Consider for example $F : \mathbb{I}^2 \to [3]$ that is defined by $F(x_1, x_2) = 1$ if $x_1 < x_2$, $F(x_1, x_2) = 2$ if $x_1 = x_2$, and $F(x_1, x_2) = 3$ if $x_1 > x_2$. Any l-part homogeneous function G that has no dependency on the order of x_1 and x_2 would have a positive probability appearance of some $R : [2]^2 \to [3]$ which is completely constant, while for F such an R does not exist (the all-2 function "appears" in F only if we allow equality of the coordinates, $x_1 = x_2$, and even then it appears with zero probability).

4 Homogeneous Functions and Inlays

For the proof of Theorem 1 we will analyze discrete versions of homogeneous functions, and their appearance with positive probability in the original function. The following is their definition.

Definition 6 For l and $s \geq d$, a function $R : [sl]^d \to [k]$ is called *l-part homogeneous* if $R(i_1, \ldots, i_d)$ depends only on $\lceil i_1/l \rceil, \ldots, \lceil i_d/l \rceil$ and on the order of $i_1, \ldots, i_d$ (i.e. whether $i_j \leq i_{j'}$ and/or $i_j \geq i_{j'}$ for every $1 \leq j < j' \leq d$).

The reason for requiring that $s \geq d$ is so that all orders will be "expressed" in every interval. For example, even if $i_1, \ldots, i_d$ all satisfy $\lceil i_j/s \rceil = 1$, requiring that $s \geq d$ makes it still possible to have any order between them (and in particular they can be all unequal). This provides soundness to the following definition of compatibility, which means that two homogeneous functions with different domains (both being discrete, or one of them being continuous) have the same "big picture".

Definition 7 Two l-part homogeneous functions $R : [sl]^d \to [k]$ and $S : [tl]^d \to [k]$ are called *compatible* if for every $(i_1, \ldots, i_d) \in [sl]^d$ and $(i'_1, \ldots, i'_d) \in [tl]^d$, that satisfy $\lceil i_j/s \rceil = \lceil i'_j/t \rceil$ for all $j \in [d]$ as well as that $i_j \leq i_{j'}$ if and only if $i'_j \leq i'_{j'}$ for every $j \neq j'$, the equality $R(i_1, \ldots, i_d) = S(i'_1, \ldots, i'_d)$ holds.

An l-part homogeneous function $R : [sl]^d \to [k]$ and an l-part homogeneous function $F : \mathbb{I}^d \to [k]$ are called *compatible* if for every $(i_1, \ldots, i_d) \in [sl]^d$ and $(x_1, \ldots, x_d) \in \mathbb{I}^d$, that satisfy $\lceil i_j/s \rceil = \lceil l x_j \rceil$ for all $j \in [d]$, as well as that $i_j \leq i_{j'}$ if and only if $x_j \leq x_{j'}$ for every $j \neq j'$, the equality $R(i_1, \ldots, i_d) = F(x_1, \ldots, x_d)$ holds.

The following observation in particular would not have been true without the requirement that $s \geq d$ in Definition 6.

Observation 3 *If $R : [sl]^d \to [k]$ is l-part homogeneous (and $s \geq d$), then for every $t \geq d$ there is exactly one l-part homogeneous function $S : [tl]^d \to [k]$ that is compatible with R. Additionally, there is exactly one l-part homogeneous $F : \mathbb{I}^d \to [k]$ that is compatible with R, and for any l-part homogeneous $F : \mathbb{I}^d \to [k]$ there is exactly one l-part homogeneous $S : [tl]^d \to [k]$ that is compatible with F.*

Proof Sketch As an example, here is how the first statement is proved. Given $R : [sl]^d \to [k]$ that is l-part homogeneous and $S : [tl]^d \to [k]$ that is compatible with R, consider $(i'_1, \ldots, i'_d) \in [tl]^d$. Since $s \geq d$, it is possible to find $(i_1, \ldots, i_d) \in [sl]^d$ that satisfy $\lceil i_j/s \rceil = \lceil i'_j/t \rceil$ for all $j \in [d]$ as well as that $i_j \leq i_{j'}$ if and only if $i'_j \leq i'_{j'}$ for every $j \neq j'$.

To find $(i_1, \ldots, i_d)$, we set $a_j = \lceil i'_j/t \rceil$ and $b'_j = i'_j - (a_j - 1)t$ for $j \in [d]$. We now set $i_j = (a_j - 1)s + b_j$, where $b_j = |\{j' \in [d] : b'_{j'} < b'_j\}| + 1$. This ensures that $b_1, \ldots, b_d$ have the same order relations between them as $b'_1, \ldots, b'_d$ do. Now $\lceil i_j/s \rceil = \lceil i'_j/t \rceil$ for all $j \in [d]$ since $b_j \in [d]$ and $t \geq d$. Finally this implies that $i_1, \ldots, i_d$ and $i'_1, \ldots, i'_d$ have the same order between them by the above guarantee for $b_1, \ldots, b_d$ and $b'_1, \ldots, b'_d$.

Thus $S(i'_1, \ldots, i'_d) = R(i_1, \ldots, i_d)$ by Definition 7, and in particular there is only one possible value for S in this location. Since we had no prior restrictions on $(i'_1, \ldots, i'_d)$ apart from belonging to $[tl]^d$, this determines the entirety of S. □

A crucial part of the proof of the main result requires proving the existence of substructures that also respect certain "boundaries", as per the following definition.

Definition 8 For a function $R : [sl]^d \to [k]$ and a function $F : \mathbb{I}^d \to [k]$, we say that R is an *s over $[l]^d$ inlay of F* if for every $a \in [l]$ there exist $0 < x_{a,1} < \cdots < x_{a,s} \leq 1$, so that for every $(i_1, \ldots, i_d) \in [sl]^d$, denoting $a_j = \lceil i_j/s \rceil$ and $b_j = i_j - (a_j - 1)s$, we have $R(i_1, \ldots, i_d) = F((a_1 - 1 + x_{a_1,b_1})/l, \ldots, (a_d - 1 + x_{a_d,b_d})/l)$.

In other words, an s over $[l]^d$ inlay of F is its "restriction to a subgrid" that results from considering the partition $\mathbb{I}_{1,l}, \ldots, \mathbb{I}_{l,l}$ of $\mathbb{I}$, and selecting a set of size s from every interval.

We also define inlays of discrete functions $S : [tl]^d \to [k]$ for $t \geq s$, this time by considering the partitioning of $[tl]$ to l "intervals" of size $[t]$.

Definition 9 For a function $R : [sl]^d \to [k]$ and a function $S : [tl]^d \to [k]$, where $s \leq t$, we say that R is an *s from t over $[l]^d$ inlay of S* if for every $a \in [l]$ there exist $1 \leq h_{a,1} < \cdots < h_{a,s} \leq t$, so that for every $(i_1, \ldots, i_d) \in [sl]^d$, denoting $a_j = \lceil i_j/t \rceil$ and $b_j = i_j - (a_j - 1)t$, we have $R(i_1, \ldots, i_d) = S((a_1 - 1)t + h_{a_1,b_1}, \ldots, (a_d - 1)t + h_{a_d,b_d})$.

Note the following simple observation.

Observation 4 *If a function $F : \mathbb{I}^d \to [k]$ is l-part homogeneous and $s \geq d$, then every s over $[l]^d$ inlay of F is l-part homogeneous and compatible with F.*

If a function $S : [tl]^d \to [k]$ is l-part homogeneous and $t \geq s \geq d$, then every s from t over $[l]^d$ inlay of S is l-part homogeneous and compatible with S. □

5 A Ramsey-Type Lemma

To prove the existence of homogeneous discrete functions, that appear (as inlays) with positive probability in $F : \mathbb{I}^d \to [k]$, we need a form of Ramsey's theorem. The following is Ramsey's theorem for edge-colored (but otherwise simple and non-oriented) hypergraphs, which is well known.

Lemma 4 (Ramsey's Theorem for Edge-Colored Hypergraphs, See [5]) *For a set V let $V^{=d}$ denote the set of all subsets of size d of V, and let $f : V^{=d} \to A$ be any function whose range is a finite set A. There exists a global function $\mathcal{R}_1 : \mathbb{N}^3 \to \mathbb{N}$ so that if $|V| \geq \mathcal{R}_1(d, |A|, s)$, then there exists $U \subset V$ with $|U| = s$ for which the restriction $f|_{U^{=d}}$ is the constant function with the value a for some $a \in A$.*

It is not hard to generalize this to the setting where the domain of the function f is over all nonempty subsets of V of size *at most* d.

Lemma 5 *For a set V let $V^{\leq d}$ denote the set of all nonempty subsets of size at most d of V, and let $f : V^{\leq d} \to A$ be any function whose range is a finite set A. There exists a global function $\mathcal{R}_2 : \mathbb{N}^3 \to \mathbb{N}$ so that if $|V| \geq \mathcal{R}_2(d, |A|, s)$, then there exists $U \subset V$ with $|U| = s$ for which the restriction $f|_{U^{\leq d}}$ is a function that depends only on the size of the set. That is, there exists $f' : [d] \to A$ so that $f(C) = f'(|C|)$ for all $C \in U^{\leq d}$.*

Proof We set $\mathcal{R}_2(d, k, s) = \mathcal{R}_1(1, k, \mathcal{R}_1(2, k, \cdots \mathcal{R}_1(d, k, s) \cdots))$. Given V, we employ Lemma 4 over V with parameters 1, $|A|$ and $\mathcal{R}_1(2, k, \cdots \mathcal{R}_1(d, k, s) \cdots)$ to obtain U_1 (for the function $f|_{V^{=1}}$), then employ Lemma 4 with parameters 2, $|A|$ and $\mathcal{R}_1(3, k, \cdots \mathcal{R}_1(d, k, s) \cdots)$ over U_1 (for $f|_{U_1^{=2}}$) to obtain U_2, and so on, until we employ Lemma 4 over a set U_{d-1} with parameters d, $|A|$ and s (for $f|_{U_{d-1}^{=d}}$) to finally obtain our required set $U = U_d$. □

We will need an even more general version, that holds when there are several "types" of vertices, and we need to choose a given number of vertices of every type.

Lemma 6 *For a set V that is a disjoint union of l sets $V_1, \ldots, V_l$, let $V^{\leq d}$ denote the set of all nonempty subsets of size at most d of V, and let $f : V^{\leq d} \to A$ be any function whose range is a finite set A. There exists a global function $\mathcal{R} : \mathbb{N}^4 \to \mathbb{N}$ so that if $|V_i| \geq \mathcal{R}(l, d, |A|, s)$ for all $i \in [l]$, then there exist $U_1 \subset V_1, \ldots, U_l \subset V_l$ with $|U_i| = s$ for $i \in [l]$, for which the restriction $f|_{(U_1 \cup \cdots \cup U_l)^{\leq d}}$ is a function that depends only on the sizes of the intersections with $U_1, \ldots, U_l$. That is, there exists $f' : P(d, l) \setminus \{(0, \ldots, 0)\} \to A$ so that $f(C) = f'(|C \cap U_1|, \ldots, |C \cap U_l|)$ for all $C \in (U_1 \cup \cdots \cup U_l)^{\leq d}$, where $P(d, l)$ is the set of all non-negative integer sequences $(d_1, \ldots, d_l)$ that sum up to at most d.*

Proof The proof is by induction over l, where clearly we can set $\mathcal{R}(1,d,k,s) = \mathcal{R}_2(d,k,s)$. To set $\mathcal{R}(l,d,k,s)$, we consider a set V that is the disjoint union of $V_1,\ldots,V_l$ and a function $f : V^{\leq d} \to A$ with $|A| = k$. We will only use a subset W of V_l of size $k' = \mathcal{R}_2(d,k^{(d+1)^l},s)$ that we choose arbitrarily (the eventual value for $\mathcal{R}(l,d,k,s)$ will be much larger than k').

Next we define a function $f' : (V \setminus V_l)^{\leq d} \to A'$ for a corresponding (rather large) range A' that will become clear from the following definitions. For every $C \in (V \setminus V_l)^{=d}$ we will just set $f'(C) = f(C)$. For every $C \in (V \setminus V_l)^{\leq d-1}$, the value $f'(C)$ will be a member of $A \times A^{W^{\leq d-|C|}}$. Specifically, we define $h_C : W^{\leq d-|C|} \to A$ by $h_C(D) = f(C \cup D)$ for every $D \in W^{\leq d-|C|}$, and then define $f'(C) = (f(C), h_C)$.

We now set by induction $\mathcal{R}(l,d,k,s) = \mathcal{R}(l-1,d,k^{k'^d},s)$, and use the induction hypothesis to obtain $U_1 \subset V_1,\ldots,U_{l-1} \subset V_{l-1}$, all of size s, so that $f'(C)$ for any $C \in (U_1 \cup \cdots \cup U_{l-1})^{\leq d}$ depends only on the intersection sizes $|C \cap U_1|,\ldots,|C \cap U_{l-1}|$. Note that this in particular implies the same for $f(C)$, because this value was used for one of the "coordinates" of $f'(C)$.

We now work on obtaining $U_l \subset W$. We define $f'' : W^{\leq d} \to A''$ by the following. For $C \in W^{=d}$ we just set $f''(C) = f(C)$. For $C \in W^{\leq d-1}$, the value $f''(C)$ will be a member of $A^{P(d-|C|,l-1)}$. We first define $h'_C : P(d-|C|, l-1) \to A$ as follows. By the choice of $U_1,\ldots,U_{l-1}$ with respect to f', for every $(d_1,\ldots,d_{l-1}) \in P(d-|C|,l-1) \setminus \{(0,\ldots,0)\}$, if $D, D' \subset U_1 \cup \cdots \cup U_{l-1}$ satisfy $|D \cup U_i| = |D' \cup U_i| = d_i$ for all $i \in [l-1]$ then $f(C \cup D) = f(C \cup D')$. We set $h'_C(d_1,\ldots,d_{l-1})$ to this common value of f. Additionally (and naturally) we set $h'_C(0,\ldots,0) = f(C)$. Having thus fully defined h'_C, we then set $f''(C) = h'_C$.

We now employ Lemma 5 to obtain $U_l \subset W \subset V_l$ of size s, so that $f''(C)$ depends only on $|C|$ for every $C \in W^{\leq d}$. By this guarantee, along with what we already know about $f(C)$ for $C \in (U_1 \cup \cdots \cup U_{l-1})^{\leq d}$, we obtain that $U_1,\ldots,U_l$ are the required sets for the assertion of the lemma. □

We will use a form of Ramsey's theorem that follows from Lemma 6, stated in terms of homogeneous inlays of discrete functions.

Lemma 7 *For every l, $s \geq d$ and k there exists $r(l,s,d,k)$, so that if $t \geq r$, then every function $S : [tl]^d \to [k]$ contains an s from t over $[l]^d$ inlay that is l-part homogeneous.*

Proof This will be by a direct application of Lemma 6. Given l, s, d and k, we first define for every $c \in [d]$ the set B_c of functions from $[d]$ onto $[c]$. We then define $A = \bigcup_{c=1}^{d} A_c$, where A_c is the set of functions from B_c to $[k]$. We finally set $r(l,s,d,k) = \mathcal{R}(l,d,|A|,s)$ where $\mathcal{R}$ is the function of Lemma 6.

Given a function $S : [tl]^d \to [k]$ with $t \geq r(l,s,d,k)$, we define $V_i = \{(i-1)t+1,\ldots,it\}$ for $i \in [l]$, and define $f : (\bigcup_{i=1}^{l} V_i)^{\leq d} \to A$ as follows. Given $C \in (\bigcup_{i=1}^{l} V_i)^{\leq d}$, we sort the set to obtain $C = \{i_1,\ldots,i_{d'}\}$ with $i_1 < \cdots < i_{d'}$ and $d' = |C| \leq d$. We then define $f(C) = g_C$, where $g_C : B_{d'} \to [k]$ is a member of $A_{d'}$. To define $g_C(h) \in [k]$ for an onto function $h : [d] \to [d']$, we set it to the value $S(i_{h(1)},\ldots,i_{h(d)})$.

We now use Lemma 6 to find sets $U_1 \subset V_1, \ldots, U_l \subset V_l$, all of size s, so that for any $C \in (U_1 \cup \cdots \cup U_l)^{\leq d}$ the value $f(C)$ (which is in fact a function from $B_{|C|}$ to $[k]$) depends only on the intersection sizes $|C \cap U_1|, \ldots, |C \cap U_l|$. To conclude, for every $i \in [l]$ we sort U_i to obtain $(i-1)t + h_{i,1} < \cdots < (i-1)t + h_{i,s}$ (noting that $U_i \subset \{(i-1)t+1, \ldots, it\}$ we obtain $h_{i,j} \in [t]$ for every $i \in [l]$ and $j \in [s]$). Considering the function $R : [sl]^d \to [k]$ which is the s from t over $[l]^d$ inlay of S corresponding (as per Definition 9) to the resulting $h_{i,j}$, we obtain an l-part homogeneous function.

To prove that R it is indeed l-part homogeneous, consider $(i_1, \ldots, i_d) \in [sl]^d$ and $(i'_1, \ldots, i'_d) \in [sl]^d$ that satisfy $\lceil i_j/s \rceil = \lceil i'_j/s \rceil$ and $i_j \leq i_{j'}$ if and only if $i'_j \leq i'_{j'}$ for every $j, j' \in [d]$. First consider the result of sorting the sets to obtain $c_1 < \cdots < c_{d'}$ with $|\{c_1, \ldots, c_{d'}\}| = |\{i_1, \ldots, i_d\}|$ and $c'_1 < \cdots < c'_{d'}$ with $|\{c'_1, \ldots, c'_{d'}\}| = |\{i'_1, \ldots, i'_d\}|$, noting that d' is the same for both c_j and c'_j due to the order condition. Due to the same order condition we also have a single onto function $h : [d] \to [d']$ so that $i_j = c_{h(j)}$ and $i'_j = c'_{h(j)}$ for all $j \in [d]$. Finally, noting that this also implies that $\lceil c_j/s \rceil = \lceil c'_j/s \rceil$ for all $j \in [d']$, we have (by the choice of $U_1, \ldots, U_l$ above using Lemma 6) a function $g : B_{d'} \to [k]$ for which $f(\{c_1, \ldots, c_{d'}\}) = f(\{c'_1, \ldots, c'_{d'}\}) = g(h)$. Hence, $R(i_1, \ldots, i_d) = R(i'_1, \ldots, i'_d) = g(h)$ as required. □

There is a way that Ramsey type statements can be converted to probabilistic versions, referring to an event relating to objects (in our case inlays) chosen using an underlying probability distribution. We will use the following version. The reason for using the extra parameters α_j and β_j will become clear later on (it will be the result of invoking Lemma 2 in the proof of Lemma 9 below, on the way towards proving Theorem 1).

Lemma 8 *For every l, $s \geq d$ and k there exists $\delta(l, s, d, k) > 0$ with the following property. Let $F : \mathbb{I}^d \to [k]$ be a measurable function, let $0 < \alpha_a < \beta_a \leq 1$ be parameters for $a \in [l]$ (relating to the intervals $\mathbb{I}_{1,l}, \ldots, \mathbb{I}_{l,l}$ respectively), and let R be a random s over $[l]^d$ inlay of f chosen in the following manner: For every $a \in [l]$, $\alpha_a \leq x_{a,1} < \cdots < x_{a,s} \leq \beta_a$ are chosen uniformly at random, and R is defined (as per Definition 8) by $R(i_1, \ldots, i_d) = F((a_1 - 1 + x_{a_1,b_1})/l, \ldots, (a_d - 1 + x_{a_d,b_d})/l)$ where $a_j = \lceil i_j/s \rceil$ and $b_j = i_j - (a_j - 1)s$. With probability at least δ the inlay R will be l-part homogeneous.*

Proof We consider an alternative way to choose $\alpha_a \leq x_{a,1} < \cdots < x_{a,s} \leq \beta_a$ for $a \in [l]$: For $t = r(l, s, d, k)$ (using the function of Lemma 7), we first uniformly choose $\alpha_a \leq y_{a,1} < \cdots < y_{a,t} < \beta_a$. Then for every $a \in [l]$ we choose uniformly (from the possible $\binom{t}{s}$ choices) $1 \leq j_{a,1} < \cdots < j_{a,s} \leq t$, and set $x_{a,i} = y_{a,j_{a,i}}$ for all $i \in [s]$ and $a \in [l]$.

Now invoking Lemma 7, we know that for every choice of $\alpha_a \leq y_{a,1} < \cdots < y_{a,t} < \beta_a$, there exist for every $a \in [l]$ some $1 \leq h_{a,1} < \cdots < h_{a,s} \leq t$, so that the inlay R' defined by $R'(i_1, \ldots, i_d) = F((a_1 - 1 + y_{a_1,h_{a_1,b_1}})/l, \ldots, (a_d - 1 + y_{a_d,h_{a_d,b_d}})/l)$ with $a_j = \lceil i_j/s \rceil$ and $b_j = i_j - (a_j - 1)s$ is l-part homogeneous.

Namely, this would be the l-part homogeneous s from t over $[l]^d$ inlay of S guaranteed by Lemma 7, where S is the t over $[l]^d$ inlay of F defined by $S(i_1, \ldots, i_d) = F((a_1 - 1 + y_{a_1,b_1})/l, \ldots, (a_d - 1 + y_{a_d,b_d})/l)$ with $a_j = \lceil i_j/t \rceil$ and $b_j = i_j - (a_j - 1)t$.

We finally set $\delta = 1/\binom{t}{s}^l$, the probability that every $j_{a,i}$ is identical to its respective $h_{a,i}$. □

6 Proof of the Main Result

Using Lemma 8 in conjunction with Lemma 2, we can use an l-part homogeneous $G : \mathbb{I}^d \to [k]$ that is close to a given $F : \mathbb{I}^d \to [k]$ (but with no other guarantees, such as the one given by Lemma 3), to find an arbitrarily large l-part homogeneous discrete function $R : [sl]^d \to [k]$ that appears with positive probability in F, and is compatible with some $G' : \mathbb{I}^d \to [k]$ that is not extremely further from F as compared to G. The following lemma formalizes this. Note that there is a dependency of G' on s in the formulation below, but it will be mitigated later on.

Lemma 9 *Suppose that $F : \mathbb{I}^d \to [k]$ is any measurable function and that $G : \mathbb{I}^d \to [k]$ is an l-part homogeneous function. For every $s \geq d$, there exists an l-part homogeneous function $R : [sl]^d \to [k]$ that appears with positive probability in F, so that the l-part homogeneous function $G' : \mathbb{I}^d \to [k]$ that is compatible with R (which is unique by Observation 3) satisfies $d(F, G') \leq 2d(F, G) + \binom{d}{2}/l$.*

Proof We first note that it is enough to show that an s over $[l]^d$ inlay of F, chosen at random as per the distribution in Lemma 8 for some $\alpha_1, \ldots, \alpha_l$ and $\beta_1, \ldots, \beta_l$, satisfies the assertions of this lemma with some positive probability δ'. Once we prove this, we note that the above implies that there exists a single function $R : [sl]^d \to [k]$ satisfying the assertion of the lemma that appears with some positive probability δ'', since there is only a finite number of l-homogeneous functions $R : [sl]^d \to [k]$. But then, by the chain rule, this means that $\mu_{F,sl}(R) \geq \delta'' l^{-sl} \prod_{j=1}^{l} (\beta_j - \alpha_j)^s > 0$.

For the rest of the proof we will show the existence of $\alpha_1, \ldots, \alpha_l$ and $\beta_1, \ldots, \beta_l$ that ensure that the above event happens with $\delta' \geq \delta(l, s, d, k)/2$, where δ is the function of Lemma 8. We first analyze a different probability space $\tilde{\mu}_{F,l}$ over functions $T : [l]^d \to [k]$. This space is defined as the result of choosing uniformly and independently $0 < x_i \leq 1$ for every $i \in [l]$, and defining T by $T(i_1, \ldots, i_d) = F((i_1 - 1 + x_{i_1})/l, \ldots, (i_d - 1 + x_{i_d})/l)$ for $(i_1, \ldots, i_d) \in [l]^d$ (note that this can be viewed as a random "1 over $[l]^d$ inlay"of F).

Consider now the set of tuples without repetitions, $I = \{(i_1, \ldots, i_d) \in [l]^d : |\{i_1, \ldots, i_d\}| = d\}$. We analyze probabilistic bounds for the number of $(i_1, \ldots, i_d) \in I$ for which $T(i_1, \ldots, i_d) \neq G((i_1 - 1 + x_{i_1})/l, \ldots, (i_d - 1 + x_{i_d})/l)$. For this we define $A = \{(x_1, \ldots, x_d) \in \mathbb{I}^d : F(x_1, \ldots, x_d) \neq G(x_1, \ldots, x_d)\}$. For

a tuple $(i_1, \ldots, i_d) \in I$, the probability for having $T(i_1, \ldots, i_d) \neq G((i_1 - 1 + x_{i_1})/l, \ldots, (i_d - 1 + x_{i_d})/l)$ is exactly $l^d \lambda(A \cap \prod_{j=1}^{d} \mathbb{I}_{i_d,l})$.

Hence, the expected size of the "set of differences" $I' = \{(i_1, \ldots, i_d) \in I : T(i_1, \ldots, i_d) \neq G((i_1 - 1 + x_{i_1})/l, \ldots, (i_d - 1 + x_{i_d})/l)\}$ is at most $\lambda(A)l^d = d(F, G)l^d$. This means that with positive probability we have $|I'| \leq d(F, G)l^d$. We use Lemma 2 over the choice of $x_1, \ldots, x_l$ with l as the dimension, to obtain $\alpha_1, \ldots, \alpha_l$ and $\beta_1, \ldots, \beta_l$, so that when we condition on $\bigwedge_{j=1}^{l} \alpha_j \leq x_j \leq \beta_j$, we obtain $|I'| \leq d(F, G)l^d$ with probability at least $1 - \delta(l, s, d, k)/2s^d$.

We now consider the probability space of choosing $\alpha_i \leq x_{a,1} < \cdots < x_{a,s} \leq \beta_i$ uniformly for every $a \in [l]$, and defining R by $R(i_1, \ldots, i_d) = F((a_1 - 1 + x_{a_1,b_1})/l, \ldots, (a_d - 1 + x_{a_d,b_d})/l)$ where $a_j = \lceil i_j/s \rceil$ and $b_j = i_j - (a_j - 1)s$. To analyze further, consider the alternative process of choosing $\alpha_a \leq y_{a,j} \leq \beta_a$ uniformly and independently for $a \in [l]$ and $j \in [s]$, and then sorting $y_{a,1}, \ldots, y_{a,s}$ to obtain $x_{a,1} < \cdots < x_{a,s}$ for every $a \in [l]$.

By the choices of $\alpha_1, \ldots, \alpha_l$ and $\beta_1, \ldots, \beta_l$, for every fixed $(j_1, \ldots, j_d) \in [s]^d$, with probability at least $1 - \delta(l, s, d, k)/2s^d$ we have $|\{(i_1, \ldots, i_d) \in I : R((i_1 - 1)s + j_1, \ldots, (i_d - 1)s + j_d) \neq G((i_1 - 1 + y_{i_1,j_1})/l, \ldots, (i_d - 1 + y_{i_d,j_d})/l)\}| \leq d(F, G)l^d$. By a union bound, with probability at least $1 - \delta(l, s, d, k)/2$ all these events happen at once, and then by summing over all $(j_1, \ldots, j_d) \in [s]^d$ we get $|\{(i_1, \ldots, i_d, j_1, \ldots, j_d) \in I \times [s]^d : R((i_1 - 1)s + j_1, \ldots, (i_d - 1)s + j_d) \neq G((i_1 - 1 + x_{i_1,j_1})/l, \ldots, (i_d - 1 + x_{i_d,j_d})/l)\}| \leq d(F, G)(sl)^d$ (we used here the fact that $x_{i,j}$ are obtained from respective permutations of $y_{i,j}$). By another union bound, with probability at least $\delta(l, s, d, k)/2$ the above event happens concurrently with R being l-part homogeneous.

Now consider the l-part homogeneous function $G' : \mathbb{I}^d \to [k]$ that is compatible with R. Noting the above bound, together with $|[l]^d \setminus I| \leq \binom{d}{2} l^{d-1}$, when the above events all happen we obtain $d(G, G') \leq d(F, G) + \binom{d}{2}/l$, and hence by the triangle inequality $d(F, G') \leq 2d(F, G) + \binom{d}{2}/l$. □

We are finally ready for the proof of the main result.

Proof of Theorem 1 Given $F : \mathbb{I}^d \to [k]$ and ϵ, we first use Lemma 3 to obtain some $G : \mathbb{I}^d \to [k]$ that is l-part homogeneous for $l \geq 3\binom{d}{2}/\epsilon$ and satisfies $d(F, G) \leq \epsilon/3$.

We next use Lemma 9 to obtain $G' : \mathbb{I}^d \to [k]$ that is compatible with an l-part homogeneous $R' : [sl]^d \to [k]$ for which $\mu_{F,sl}(R') > 0$, and satisfies $d(F, G') \leq 2d(F, G) + \binom{d}{2}/l \leq \epsilon$. We do this for every $s \geq d$, and for every s we may obtain a different G'. However, since by Observation 2 there is a finite number of possible G', we can pick a single G' for which this holds for an infinite sequence of possible s.

To complete the proof, it remains to show that no $S : [n]^d \to [k]$ for which $\mu_{F,n}(S) = 0$ appears in G'. Assuming on the contrary that there exists such an S, we note that in particular S also appears in the l-part homogeneous $R' : [n'l]^d \to [k]$ that is compatible with G' for any $n' \geq n$, and by the above choice of G' we know that there exists such an n' for which $\mu_{F,n'l}(R') > 0$. Now recall that we can

view the probability space $\mu_{F,n}$ as the result of first choosing $T : [n'l]^d \to [k]$ according to $\mu_{F,n'l}$, then choosing $1 \leq j_1 < \cdots < j_n \leq n'l$ uniformly (from the $\binom{n'l}{n}$ possible choices), and finally setting $R : [n] \to [k]$ by defining $R(i_1, \ldots, i_d) = T(j_{i_1}, \ldots, j_{i_d})$. But this implies that $\mu_{F,n}(S) \geq \mu_{F,n'l}(R')/\binom{n'l}{n} > 0$, a contradiction. □

7 Discussion and Variants

7.1 Relation to the Original Pixelation Lemma

The pixelation lemma in [1] was stated specifically for functions $F : \mathbb{U}^4 \to \mathbb{U}$, where $\mathbb{U}$ denotes the closed interval $\{x \in \mathbb{R} : 0 \leq x \leq 1\}$. Also, the definition of appearance is different (there are more ways for a structure to appear in F).

Specifically, the domain is interpreted as $(\mathbb{U} \times \mathbb{U})^2$, and F is considered as a *binary* relation with "fractional elements" (so a single "vertex" corresponds to the set $\{a\} \times \mathbb{U}$ for some $0 \leq a \leq 1$). Thus, in the definition of an appearance of a structure (here a vertex-ordered graph), the order relation between the first and the third coordinates is the only one taken into consideration, while for the second and fourth coordinate we are only concerned about whether they can be chosen from a positive probability set (for a positive probability appearance).

Additionally, the range is interpreted as corresponding to a single relation (essentially a vertex-ordered simple graph), corresponding to a function $E : [n] \to \{0, 1\}$. The definition of the probability for appearance involves a final step, where for example the value $F((x_1, a), (x_2, b))$ provides the probability that $E(1, 2) = 1$ (as opposed to $E(1, 2) = 0$), following the random choice of $0 \leq x_1 < x_2 \leq 1$ and $a, b \in \mathbb{U}$ (independently). Accordingly, the approximation guarantee is phrased in terms of the L_1 distance, $d(F, G) = \int_{(x,a,y,b)\in\mathbb{U}^4} |F((x, a), (y, b)) - G((x, a), (y, b))| d(x, a, y, b)$.

This can be converted back to functions with discrete ranges by standard "quantization", approximating a real value by the closest multiple of $\frac{1}{k}$ for some large enough k (and making sure that the "edge values" 0 and 1 are preserved). Also note that recently in [7] it was observed that for the purpose of representing limit objects (including vertex-ordered graphs, and also plain graphs as in the large body of work presented in [6]), there is a way of altogether doing away with values outside 0 and 1 (in the ordered regime the additional unordered coordinates are still necessary).

Finally, the lemma in [1] is restricted to symmetric relations, in the sense that $F((x, a), (y, b)) = F((y, b), (x, a))$ for every $x, a, y, b \in \mathbb{U}$. This is not an essential difference, since a non-symmetric relation can be represented by two symmetric relations as long as we have an underlying vertex order. The way Lemma 6 is converted to Lemma 7 in fact demonstrates how such representations can be constructed for relations of any arity. However, after such a conversion we

must still distinguish which of the coordinates are equal to each other, and so must explicitly handle relations of lower arities. Specifically in [1] the role of equality is also very diminished, since there the values $F((x,a),(y,a))$ for $x = y$ are ignored by sticking to the notion of graphs without loops.

The special case pixelation lemma from [1] is used there in conjunction with a compactness theorem for the above described limit objects (under a suitable topology, weaker than that of the L_1 distance) to derive a removal lemma for vertex-ordered graphs. While the pixelation lemma is generalized to higher arities here, the original concept of limit objects does not generalize that easily. Investigations of limit objects for hypergraphs are present in [3] and [8], and with the addition of a vertex order in [7]. There, a replacement for the pixelation lemma is used that still guarantees its most useful property in that context, namely the assurance that all appearances in the converted function are of positive probability and come from positive probability appearances in the original function.

7.2 Dealing with Diagonals and Lower Arity Relations

Recall that for $k = 2^r$, a function $F : \mathbb{I}^d \to [k]$ can give the information about a model (over $\mathbb{I}$) of r arity d relations. Recall also the earlier comment that lower arity relations can be represented by making them invariant over the last coordinates. Thus an arity d' relation for $d' < d$ can be replaced by an arity d relation, if we stipulate that for every $x_1, \ldots, x_d$ and $x'_{d'+1}, \ldots, x'_d$ we have the equivalence $R(x_1, \ldots, x_d) \leftrightarrow R(x_1, \ldots, x_{d'}, x'_{d'+1}, \ldots, x'_d)$.

When we move to modeling with a single function F, this condition can be converted to stipulating that certain structures do not appear. If for example the relation in question is the first relation in the vocabulary, meaning that it is represented by $F(x_1, \ldots, x_d) \pmod 2$, then the additional "forbidden structures" are all $S : [2]^d \to [k]$ for which $S(1, \ldots, 1) \neq S(1, \ldots, 1, i_{d'+1}, \ldots, i_d) \pmod 2$ for any $i_{d'+1}, \ldots, i_d$ that take values in $\{1, 2\}$.

Thus, the l-part homogeneous G that results from Theorem 1 will also satisfy the condition that makes it conform to a relation of arity d'. Additionally, the way this relation is modeled (being invariant of the last $d - d'$ coordinates), a distance bound between F and G in terms of the Lebesgue measure over $\mathbb{I}^d$ translates to a corresponding distance bound that applies to the original relation in terms of the measure over $\mathbb{I}^{d'}$.

It may at times be useful to also ensure for an arity d relation that measures are preserved for the restrictions to "diagonals", i.e., when there are equality constraints between coordinates. The logical equivalent is when a relation is used with the same variable appearing in more than one place. For example, when we are dealing with a binary relation, one might want to ensure a small distance also with respect to the measure of the set $\{x \in \mathbb{I} : F(x,x) \neq G(x,x)\}$, where F is the function referring to R and other relations.

The way to ensure such a small distance bound is by constructing a unary relation $U(x)$, and adding the condition $U(x) \leftrightarrow R(x,x)$ for all x. This can be converted to a condition about certain substructures not appearing in F. The relation U can then be converted back to a binary relation as explained above and added to the encoding by F, to make sure that the small distance bound from G applies to it.

7.3 Removing the Order Dependency at a Cost

Recall that by the comment following the statement of Theorem 1, to ensure the exclusion of structures that do not appear in the original function F it is necessary that the l-part homogeneous function $G : \mathbb{I}^d \to [k]$ has a dependency on the order between the coordinate values. This is relevant in sets of type $\prod_{j=1}^{l} \mathbb{I}_{i_j,l}$ whenever $i_1, \ldots, i_l$ are not all different. If we insist that we want a "completely pixelated" G, which is constant over all sets of the type $\prod_{j=1}^{l} \mathbb{I}_{i_j,l}$, we have to alter the exclusion requirement.

For example, suppose that we have a single relation R of arity 2 and we are interested in substructures with 2 elements. Then we would look at the quartet $(R(x,x), R(x,y), R(y,x), R(y,y))$ for any $x < y$. But if we count "homomorphisms" as structures to be excluded as well, we would also look at $(R(x,x), R(x,x), R(x,x), R(x,x))$, corresponding to the case $x = y$, and consider the measure of the set of $x \in \mathbb{I}$ that provide a certain value.

If "substructures with equalities" are also considered as substructures that can appear with positive probability in F, then we can have a completely pixelated H with the following procedure: We start with the G provided by Theorem 1, but then for every $(i_1, \ldots, i_d) \in [l]^d$ for which some indexes are equal, we replace the values of G over $\prod_{j=1}^{l} \mathbb{I}_{i_j,l}$ with the constant equal to $G((i_1 + \frac{1}{2})/l, \ldots, (i_d + \frac{1}{2})/l)$. This is the same as "stipulating" that whenever x_j and x'_j *can* be equal (since $i_j = i'_j$), they *must* be equal.

The new structures appearing in H might not appear according to the original definition of appearance in F, but they must all appear in F when we allow equalities among $x_1, \ldots, x_n$ in the definition of appearance. In fact they appear with positive probability when we condition the $[n]^d$-statistic distribution $\mu_{F,n}$ on the event of the respective equalities occurring among $x_1, \ldots, x_n$.

The original lemma in [1] does not have an order dependency, but this is a benefit of dealing only with symmetric binary structures without loops (which correspond to equalities). The version of Ramsey's theorem used in its proof there is also much lighter than the one developed here. If we only ignore orders, for example allowing only the hard-coded relations " $=$" and " $\neq$" instead of " $\leq$", then we would obtain a lemma where the value of $F : (x_1, \ldots, x_d)$ depends only on the partition of $x_1, \ldots, x_d$ into parts with equal values. For arities larger than 2, its proof would still require the version of Ramsey's theorem developed here.

7.3.1 Containment in the Other Direction

Considering the function $F : \mathbb{I}^d \to [k]$, Theorem 1 ensures the existence of an l-part homogeneous $G : \mathbb{I}^d \to [k]$ (for some l) within distance ϵ of F, so that all structures that appear in G already appear in F. One can ask whether this can be made bidirectional, so that G will also be guaranteed to contain every structure that appears with positive probability in F.

However, this does not hold even for the (rather non-interesting) case of $d = 1$. For every natural number r, define $\mathbb{J}_r = \mathbb{I}_{2^r,2}$. Note that $\mathbb{J}_1, \mathbb{J}_2, \ldots$ are all disjoint and their union equals $\mathbb{I}$. Then define $F : \mathbb{I} \to \{1, 2\}$ by setting $F(x) = 1$ if $x \in \mathbb{J}_r$ for an even r, and setting $F(x) = 2$ if $x \in \mathbb{J}_r$ for an odd r. In this construction, for every r there exists $x_1 < \cdots < x_r$ so that $F(x_i) = 1$ if and only if i is odd. However, for every l, an l-part homogeneous function $G : \mathbb{I} \to \{1, 2\}$ will not contain such a sequence for any $r > l$.

On the other hand, for every fixed r, one can still ensure that the l-part homogeneous $G : \mathbb{I}^d \to [k]$ resulting from Theorem 1 contains all structures of size r that appear with positive probability in F. To this end, let δ be the minimum of $\mu_{F,r}(R)$ over all $R : [r]^d \to [k]$ for which $\mu_{F,r}(R) > 0$. Then, deploy Theorem 1, replacing the original parameter ϵ with $\min\{\epsilon, \delta/2r^d\}$.

Acknowledgments Research supported by an Israel Science Foundation grant number 879/22.

References

1. Ben-Eliezer, O., Fischer, E., Levi, A., Yoshida, Y.: Ordered graph limits and their applications. In: Lee, J.R. (ed.) 12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6–8, 2021, Virtual Conference. LIPIcs, vol. 185, pp. 42:1–42:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021)
2. Ebbinghaus, H.D., Flum, J.: Finite Model Theory. Perspectives in Mathematical Logic. Springer, Berlin (2014)
3. Elek, G., Szegedy, B.: A measure-theoretic approach to the theory of dense hypergraphs. Adv. Math. **231**(3), 1731–1772 (2012)
4. Franks, J.M.: A (Terse) Introduction to Lebesgue Integration. Student Mathematical Library. American Mathematical Society, Providence (2009)
5. Katz, M., Reimann, J.: An Introduction to Ramsey Theory. Student Mathematical Library. American Mathematical Society, Providence (2018)
6. Lovász, L.: Large Networks and Graph Limits, vol. 60. American Mathematical Society, Providence (2012)
7. Towsner, H.: A removal lemma for ordered hypergraphs. Proceedings of the London Mathematical Society. **130**(1), e70015 (2025). https://doi.org/10.1112/plms.70015
8. Zhao, Y.: Hypergraph limits: a regularity approach. Random Struct. Algorithms **47**(2), 205–226 (2015)

Reflection and Recurrence

Sakaé Fuchino

Abstract We examine the Zermelo Fraenkel set theory with Choice (ZFC) enhanced by one of the (structural) reflection principles down to a small cardinal and/or Recurrence Axioms defined below. The strongest forms of reflection principles spotlight the three scenarios in which the size of the continuum is either $\aleph_1$, or $\aleph_2$, or very large, while the maximal setting of Recurrence Axioms points to the set-theoretic universe with the continuum of size $\aleph_2$.

We discuss that both the Reflection Principles and Recurrence Axioms can be construed as preferable candidates of the extension of ZFC in terms of the criteria of Gödel's Program. From this view point, the maximal possible (consistent) combination of these principles and axioms, or even some natural strengthening of the combination (which we want to call "Laver-generic Maximum" (LGM)) may be considered as the ultimate extension of ZFC (of course "ultimate" only for now—because of the Incompleteness Theorems): LGM resolves the size of the continuum to be $\aleph_2$ and integrates practically all known statements consistent with ZFC in itself either as its consequences (as it is the case with Martin's Maximum^{++}) or as theorems holding in many grounds of the universe (as it is the case with Cichoń's Maximum).

1 Introduction

In the following, we examine the Zermelo Fraenkel set theory with the Axiom of Choice (ZFC) enhanced by one of the (structural) reflection principles down to a small cardinal and/or Recurrence Axioms defined below. The strongest forms of

An extended postprint version of the chapter is downloadable as: https://fuchino.ddo.jp/papers/reflection_and_recurrence-Janos-Festschrift-x.pdf.

S. Fuchino (✉)
Graduate School of System Informatics, Kobe University, Nada, Kobe, Japan
e-mail: fuchino@diamond.kobe-u.ac.jp

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_20

reflection principles (existence of a/the $\mathcal{P}$-Laver-generic large cardinal—see Sect. 2 below) spotlight the three scenarios in which the size of the continuum is either $\aleph_1$, or $\aleph_2$, or very large (see Theorems 8–10), while the maximal setting of Recurrence Axioms points to the set-theoretic universe with the continuum of size $\aleph_2$ (see the end of Sect. 3).

As we are going to discuss in Sects. 2 and 3, both of the Reflection Principles and Recurrence Axioms can be construed as preferable candidates of the extension of ZFC in terms of Gödel's Program (Gödel [23], see also Bagaria [1]). From this point of view the maximal possible (consistent) combination of these principles and axioms, or even some natural strengthening of the combination, that is, either the principle LGM proposed in Sect. 6 or some further extension of it in the future (which we want to call "Laver-generic Maximum" (LGM), see p. 346) may be considered as the ultimate extension of ZFC (of course "ultimate" only for now—because of the Incompleteness Theorems): LGM resolves the size of the continuum to be $\aleph_2$ and integrates practically all known statements consistent with ZFC in itself either as its consequences (as it is the case with Martin's Maximum^{++}, see Theorem 5) or as theorems holding in many grounds of the universe (as it is the case with Cichoń's Maximum, Goldstern, Kellner and Shelah[24], Goldstern, Kellner, Mejía and Shelah [25], see around p. 346)—for more detailed discussions about the significance of these axioms, see also the end of both of the Sects. 2 and 6.

2 Reflection Down to a Small Cardinal

The small cardinal mentioned in the title of this section may be considered as not very small by non-set-theoretic mathematicians: It is known that many "mathematical" reflection statements with reflection number less than or equal to $\kappa_{\mathrm{refl}} := \max\{\aleph_2, 2^{\aleph_0}\}$ hold (often in some extension of ZFC). Some of them are even theorems in ZFC. For example,

Theorem 1

(1) (Dow [8]) *If X is a countably compact Hausdorff non-metrizable space then there is a subspace Y of X of cardinality $< \aleph_2$ such that Y is also non-metrizable.*

(2) *Let $L(Q)$ be a logic with new (first-order) quantifier such that "$Qx \ldots$" is interpreted as "there are uncountably many x such that $\ldots$". For any structure $\mathfrak{A}$ of countable signature, there is $\mathfrak{B} \prec_{L(Q)} \mathfrak{A}$ of size $< \aleph_2$.* □

From very early on, it was known that, starting from a very large cardinal, we can construct models of set theory in which various strong statements on (structural) reflection down to $< \kappa_{\mathrm{refl}}$ hold.

For example, Ben-David [5] in 1978 mentions a theorem by Shelah which states:

Theorem 2 (S. Shelah, [5]) *Suppose that κ is supercompact and $\mathbb{P} = \mathrm{Col}(\aleph_1, \kappa)$. Then, for $(\mathsf{V}, \mathbb{P})$-generic $\mathbb{G}$, we have*

$$\mathsf{V}[\mathbb{G}] \models \textit{for any structure } \mathfrak{A} \textit{ of countable signature, there is } \mathfrak{B} \prec_{L_{stat}} \mathfrak{A} \textit{ of cardinality } < \aleph_2.$$

Here, L_{stat} denotes the stationary logic with monadic second-order variables X which run over countable subsets of the underlying set of respective structures and with the second-order quantifier $stat\, X$ which is to be interpreted as "there are stationarily many countable sets X".

The elementary submodel relation $\mathfrak{B} \prec_{L_{stat}} \mathfrak{A}$ is defined by $\mathfrak{B} \models \varphi(b_0, ...) \Leftrightarrow \mathfrak{A} \models \varphi(b_0, ...)$ for all L_{stat}-formula $\varphi = \varphi(x, ...)$ without free second-order variables, and for all $b_0 \cdots \in |\mathfrak{B}|$.

Today, we can understand Shelah's theorem above as a special case of the following theorem. For a class $\mathcal{P}$ of posets, a cardinal κ is said to be *$\mathcal{P}$-generically supercompact* if, for any $\lambda > \kappa$, there is a poset $\mathbb{P} \in \mathcal{P}$ such that, for $(\mathsf{V}, \mathbb{P})$-generic $\mathbb{G}$, there are $j, M \subseteq \mathsf{V}[\mathbb{G}]$ such that $j : \mathsf{V} \overset{\prec}{\to}_\kappa M$,[1] $j(\kappa) > \lambda$, and $j''\lambda \in M$.

Theorem 3 *Suppose that $\aleph_2$ is $\mathcal{P}$-generically supercompact where $\mathcal{P}$ is the class of all $< \aleph_1$-closed posets. Then*

(*1) *for any structure $\mathfrak{A}$ of countable signature, there is $\mathfrak{B} \prec_{L_{stat}} \mathfrak{A}$ of cardinality $< \aleph_2$.*

Proof The condition "$\aleph_2$ is $\mathcal{P}$-generic supercompact for $\mathcal{P}$ = the class of all $< \aleph_1$-closed posets." is equivalent to the Game Reflection Principle (GRP) (Theorem 8 in König [31]—see Fuchino et al. [19] for a generalization of König's theorem in [31]—note that what we call GRP here and [19] is called "the global Game Reflection Principle" in [31]). By Theorem 4.7 in [19], GRP implies (*1). □ (Theorem 3)

The downward Löwenheim-Skolem Theorem (*1) for L_{stat} is actually a strong reflection property. For example the reflection of uncountable coloring number of graphs down to $< \aleph_2$ (the following (*2)) is a consequence of (*1):

(*2) For any graph G of uncountable coloring number, there is a subgraph H of G of size $\aleph_1$ with uncountable coloring number.

This implication can be proved directly but we can also see this using the terminology introduced in [19] as follows: The downward Löwenheim-Skolem Theorem (*1) for L_{stat} is equivalent to the Diagonal Reflection Principle $\mathsf{DRP}(\mathsf{IC}_{\aleph_0})$ down to an internally club set (Corollary 3.6 in [19]). This implies the reflection principle $\mathsf{RP}_{\mathsf{IU}_{\aleph_0}}$ down to an internally unbounded set of size $< \aleph_2$. This reflection

[1] With "$j : \mathsf{V} \overset{\prec}{\to}_\kappa M$" we denote the situation that M is transitive, j is an elementary embedding of V into M, and κ is the critical point of j.

principle is equivalent to Axiom R of Fleissner (Lemma 2.6 in [16]). From Axiom R, the Fodor-type Reflection Principle (FRP) follows (Corollary 2.6 in [18]). (*2) is a consequence of (actually equivalent to FRP over ZFC [21]).

As it is mentioned in the proof of Theorem 3, the condition "$\aleph_2$ is $\mathcal{P}$-generic supercompact for $\mathcal{P}$ = the class of all $<\aleph_1$-closed posets" is equivalent to Game Reflection Principle (GRP). As the name suggests, GRP is actually a reflection principle which claims the reflection of the non-existence of winning strategy of certain games of length ω_1 down to subgames of size $<\aleph_2$. A remarkable feature of this principle is that it implies CH (Theorem 8 in König [31]).

The notion of Laver-generic large cardinals was introduced in Fuchino et al. [20] in search for reflection principles which generalize GRP. The following definition of Laver-generic large cardinals is a streamlined version adopted in later papers Fuchino and Ottenbreit Maschio Rodrigues [13], Fuchino [12] etc. and slightly different from the one given in [20].

We call a non-empty class $\mathcal{P}$ of posets *iterable* if it satisfies: $\{\mathbb{1}\} \in \mathcal{P}$, $\mathcal{P}$ is closed with respect to forcing equivalence (i.e. if $\mathbb{P} \in \mathcal{P}$ and $\mathbb{P} \sim \mathbb{P}'$ then $\mathbb{P}' \in \mathcal{P}$), closed with respect to restriction (i.e. if $\mathbb{P} \in \mathcal{P}$ then $\mathbb{P} \restriction \mathbb{p} \in \mathcal{P}$ for any $\mathbb{p} \in \mathbb{P}$), and, for any $\mathbb{P} \in \mathcal{P}$ and $\mathbb{P}$-name $\underset{\sim}{\mathbb{Q}}$, $\Vdash_{\mathbb{P}}$ "$\underset{\sim}{\mathbb{Q}} \in \mathcal{P}$" implies $\mathbb{P} * \underset{\sim}{\mathbb{Q}} \in \mathcal{P}$.

For an iterable class $\mathcal{P}$ of posets, a cardinal κ is said to be *$\mathcal{P}$-Laver-generically supercompact* if, for any $\lambda > \kappa$ and $\mathbb{P} \in \mathcal{P}$ there is a $\mathbb{P}$-name $\underset{\sim}{\mathbb{Q}}$ with $\Vdash_{\mathbb{P}}$ "$\underset{\sim}{\mathbb{Q}} \in \mathcal{P}$", such that for $(\mathsf{V}, \mathbb{P} * \underset{\sim}{\mathbb{Q}})$-generic $\mathbb{H}$, there are $j, M \subseteq \mathsf{V}[\mathbb{H}]$ such that $j : \mathsf{V} \overset{\prec}{\to}_{\kappa} M$, $j(\kappa) > \lambda$, $\mathbb{P}, \mathbb{H}, j''\lambda \in M$.

κ is *tightly $\mathcal{P}$- Laver-generically supercompact* if it is $\mathcal{P}$-Laver-generically supercompact and $\underset{\sim}{\mathbb{Q}}$, j and M for each $\mathbb{P} \in \mathcal{P}$ in the definition of $\mathcal{P}$-Laver-generic supercompactness additionally satisfy that $\mathbb{P} * \underset{\sim}{\mathbb{Q}}$ is forcing equivalent to a poset of size $\leq j(\kappa)$.[2]

Laver-generic large cardinals corresponding to other notions of large large cardinals can be defined similarly. In particular we consider in the following (tightly) $\mathcal{P}$-generic superhuge/ultrahuge/hyperhuge cardinals which correspond to superhuge/ultrahuge/hyperhuge cardinals respectively. For the precise definition of these cardinals see [12, 20] and/or [17] or the extended postprint version of the present article mentioned in the footnote on p. 321.

[2] In the following, we shall denote this condition simply by "$|\mathbb{P} * \underset{\sim}{\mathbb{Q}}| \leq \lambda$". More generally, we shall simply write "$|\mathbb{P}| \leq \lambda$" for a poset $\mathbb{P}$ to say that "the poset $\mathbb{P}$ is forcing equivalent to a poset of size $\leq \lambda$".

The following implications follow from the definitions:

$$\kappa \text{ is (tightly) } \mathcal{P}\text{-Laver-generically ultrahuge}$$
$$\Downarrow$$
$$\kappa \text{ is (tightly) } \mathcal{P}\text{-Laver-generically superhuge}$$
$$\Downarrow$$
$$\kappa \text{ is (tightly) } \mathcal{P}\text{-Laver-generically supercompact}$$

The relationship between Laver-generic hyperhugeness and Laver-generic ultrahugeness is slightly more subtle and at the moment, we need the tightness to obtain the expected implication:

Lemma 4 (Fuchino and Usuba [17]) *For any class $\mathcal{P}$ of posets, if κ is tightly $\mathcal{P}$-Laver-generically hyperhuge then κ is tightly $\mathcal{P}$-Laver-generically ultrahuge.* □

The consistency strength of Laver-generic hyperhugeness can be separated from that of other notions of Laver-generic large cardinals. This is because we know that the existence of $\mathcal{P}$-Laver-generic hyperhuge cardinal for any $\mathcal{P}$ (if it ever exists) is equiconsistent with that of a hyperhuge cardinal (see Theorem 33).

At first glance, it is not immediately clear if the notion of Laver-generic large cardinal is definable in the language $\mathcal{L}_\in$ of ZFC. In [15] an abstract generic version of extender is introduced to show the definability of Laver-generic large cardinals.

Laver-generic supercompactness implies double plus versions of forcing axioms. For a class $\mathcal{P}$ of posets and cardinals κ, μ, we denote with $\mathsf{MA}^{+\mu}(\mathcal{P}, <\kappa)$ and $\mathsf{MA}^{++\leq\mu}(\mathcal{P}, <\kappa)$ the following versions of Martin's Axiom:

$\mathsf{MA}^{+\mu}(\mathcal{P}, <\kappa)$: For any $\mathbb{P} \in \mathcal{P}$, any family $\mathcal{D}$ of dense subsets of $\mathbb{P}$ with $|\mathcal{D}| < \kappa$ and any family $\mathcal{S}$ of $\mathbb{P}$-names such that $|\mathcal{S}| \leq \mu$ and $\Vdash_{\mathbb{P}}$ "$\underset{\sim}{S}$ is a stationary subset of ω_1" for all $\underset{\sim}{S} \in \mathcal{S}$, there is a $\mathcal{D}$-generic filter $\mathbb{G}$ over $\mathbb{P}$ such that $\underset{\sim}{S}[\mathbb{G}]$ is a stationary subset of ω_1 for all $\underset{\sim}{S} \in \mathcal{S}$.

$\mathsf{MA}^{++\leq\mu}(\mathcal{P}, <\kappa)$: For any $\mathbb{P} \in \mathcal{P}$, any family $\mathcal{D}$ of dense subsets of $\mathbb{P}$ with $|\mathcal{D}| < \kappa$ and any family $\mathcal{S}$ of $\mathbb{P}$-names such that $|\mathcal{S}| \leq \mu$ and $\Vdash_{\mathbb{P}}$ "$\underset{\sim}{S}$ is a stationary subset of $\mathcal{P}_{\eta_{\underset{\sim}{S}}}(\theta_{\underset{\sim}{S}})$" for some $\omega < \eta_{\underset{\sim}{S}} \leq \theta_{\underset{\sim}{S}} \leq \mu$ with $\eta_{\underset{\sim}{S}}$ regular, for all $\underset{\sim}{S} \in \mathcal{S}$, there is a $\mathcal{D}$-generic filter $\mathbb{G}$ over $\mathbb{P}$ such that $\underset{\sim}{S}[\mathbb{G}]$ is stationary in $\mathcal{P}_{\eta_{\underset{\sim}{S}}}(\theta_{\underset{\sim}{S}})$ for all $\underset{\sim}{S} \in \mathcal{S}$.

Clearly $\mathsf{MA}^{++<\omega_2}(\mathcal{P}, <\kappa)$ is equivalent to $\mathsf{MA}^{+\omega_1}(\mathcal{P}, <\kappa)$. MM^{++} is $\mathsf{MA}^{+\omega_1}$(stationary preserving posets, $<\aleph_2$).

Theorem 5 (Theorem 5.7 in Fuchino et al. [20], see also Fuchino [12]) *For an iterable class $\mathcal{P}$ of posets such that*

(*3) *the elements of $\mathcal{P}$ preserve stationarity of subsets of $\mathcal{P}_\mu(\theta)$ for all $\mu \leq \theta < \kappa$,*

if $\kappa > \aleph_1$ is $\mathcal{P}$-Laver-generically supercompact then $\mathsf{MA}^{++\leq\mu}(\mathcal{P}, <\kappa)$ holds for all $\mu < \kappa$. □

In contrast to usual generic large cardinals, a Laver-generic large cardinal if it exists, is unique and it is κ_{refl} in many cases.

Lemma 6 ([12, 17], See Also Proposition 4, in [11])

(1) *If κ is $\mathcal{P}$-generically measurable for an ω_1 preserving iterable $\mathcal{P}$, then $\omega_1 < \kappa$.*
(2) *If κ is $\mathcal{P}$-Laver-generically supercompact for an ω_1-preserving iterable $\mathcal{P}$ with $\mathrm{Col}(\omega_1, \{\omega_2\}) \in \mathcal{P}$ then $\kappa = \omega_2$.*
(3) *If κ is $\mathcal{P}$-Laver-generically supercompact for an iterable $\mathcal{P}$ which contains a poset adding a new real, then $\kappa \leq 2^{\aleph_0}$.*
(4) *If κ is $\mathcal{P}$-generically supercompact for an iterable $\mathcal{P}$ such that all posets in $\mathcal{P}$ do not add any reals then $2^{\aleph_0} < \kappa$.*
(5) *If κ is $\mathcal{P}$-Laver-generically supercompact for an iterable $\mathcal{P}$ which contains a poset which collapses $\aleph_1$ then $\kappa = \aleph_1$.*

Proof We only prove (5) since it is not explicitly given in [17]. Suppose that κ is $\mathcal{P}$-Laver-generically supercompact and $\mathbb{P} \in \mathcal{P}$ is such that $\Vdash_{\mathbb{P}}$ "$\aleph_1{}^{\mathsf{V}}$ is countable". If $\kappa \neq \aleph_1$, then we have $\aleph_1 < \kappa$. Let $\underset{\sim}{\mathbb{Q}}$ be a $\mathbb{P}$-name of a poset such that, for $(\mathsf{V}, \mathbb{P} * \underset{\sim}{\mathbb{Q}})$-generic $\mathbb{H}$, there are j, $M \subseteq \mathsf{V}[\mathbb{H}]$ such that $j : \mathsf{V} \overset{\prec}{\to}_\kappa M$ and $\mathbb{P}$, $\mathbb{H} \in M$. By $\mathbb{H} \cap \mathbb{P} \in M$ and since $\aleph_1{}^{\mathsf{V}} < crit(j)$, we have $M \models$ "$\aleph_1{}^{\mathsf{V}} = j(\aleph_1{}^{\mathsf{V}})$ is countable". This is a contradiction to the elementarity of j. □ (Lemma 6)

A cardinal κ is called *greatly weakly Mahlo* if κ is weakly inaccessible and there exists a non-trivial $<\kappa$-complete normal filter $\mathcal{F}$ over κ such that $\{\mu < \kappa : \mu \text{is a regular cardinal}\} \in \mathcal{F}$, and $\mathcal{F}$ is closed with respect to the Mahlo operation $M\ell$[3] where

$$S \mapsto M\ell(S) := \{\alpha \in S : \alpha \text{ has uncountable cofinality and } S \cap \alpha \text{ is stationary in } \alpha\} \quad [14].$$

Note that the Mahlo operation given above is slightly different from the one in [4].

If κ is greatly weakly Mahlo then it is hyper-weakly Mahlo (Proposition 3.4 in [14]).

The tightness of the Laver-genericity can be still strengthened as follows: a cardinal κ is *tightly*$^+$ *$\mathcal{P}$-Laver-generically x-large*, for a notion "x-large" of large cardinal (e.g. "supercompact", "superhuge" etc.) if it satisfies the definition of

[3] Closedness here means that for any $S \in \mathcal{F}$, we have $M\ell(S) \in \mathcal{F}$.

tightly $\mathcal{P}$-Laver-generically x-large cardinal with the condition "$|\mathbb{P}*\underset{\sim}{\mathbb{Q}}| \leq j(\kappa)$" in the definition being replaced by the condition "there is a complete Boolean algebra $\mathbb{B}$ of size $j(\kappa)$ such that $\mathbb{B}^+$ is forcing equivalent to $\mathbb{P} * \underset{\sim}{\mathbb{Q}}$".

Theorem 7

(1) (Theorem 3.5 in [14]) *If κ is a $\{\mathbb{P}\}$-generically measurable for a poset $\mathbb{P}$ with the μ-cc for some $\mu < \kappa$, then κ is greatly weakly Mahlo.*

(2) (Theorem 5.8 in Fuchino et al. [20]) *If κ is tightly $\mathcal{P}$-Laver-generically superhuge for a class $\mathcal{P}$ of ccc posets such that at least one element of $\mathcal{P}$ adds a real, then $\kappa = 2^{\aleph_0}$.*

(3) *For an iterable class $\mathcal{P}$ of posets, if κ is tightly$^+$ $\mathcal{P}$-Laver-generically hyperhuge, then $2^{\aleph_0} \leq \kappa$.* □

We give a sketch of the proof of Theorem 7, (3) after Corollary 38

In Fuchino and Usuba [17], it is proved as a Corollary of Theorem 32 and Theorem 33 that for $\kappa = \aleph_1$, the '+' in "tightly$^+$" in Theorem 7, (3) above can be dropped.

Theorem 8 (Fuchino et al. [20], Fuchino and Usuba [17] for (Δ) and (Δ'))

(A) *If $\mathcal{P}$ is the class of all $<\aleph_1$-closed posets, and κ is $\mathcal{P}$-Laver-generically supercompact, then $\kappa = \aleph_2$ and CH holds.*

(B) *If $\mathcal{P}$ is either the class of all proper posets or the class of all semi-proper posets, and κ is $\mathcal{P}$-Laver-generically supercompact, then $\kappa = 2^{\aleph_0} = \aleph_2$.*

(Γ) *If $\mathcal{P}$ is the class of all ccc posets, and κ is $\mathcal{P}$-Laver-generically supercompact, then κ is very large and $\kappa \leq 2^{\aleph_0}$.*

(Γ') *If $\mathcal{P}$ is the class of all ccc posets, and κ is tightly $\mathcal{P}$-Laver-generically superhuge, then κ is very large and $\kappa = 2^{\aleph_0}$.*

(Δ) *If $\mathcal{P}$ is the class of all posets, and κ is $\mathcal{P}$-Laver-generically supercompact, then $\kappa = \aleph_1$.*

(Δ') *If $\mathcal{P}$ is the class of all posets, and κ is tightly$^+$ $\mathcal{P}$-Laver-generically supercompact, then $\kappa = \aleph_1$.*

Proof (A): By Lemma 6, (2), (4). (B): By Lemma 6, (2), (3) and Theorem 5. (Γ): By Lemma 6, (3), and Theorem 7, (1). (Γ'): By (Γ) and Theorem 7, (2). (Δ): By Lemma 6, (5). (Δ'): By (Δ) and Theorem 7, (3). □ (Theorem 8)

The consistency of the existence of a $\mathcal{P}$-Laver-generic large cardinal can be proved under the existence of corresponding genuine large cardinal.

Theorem 9 (Theorem 5.2, [20])

(A) *Suppose that κ is supercompact and $\mathbb{P} = \mathrm{Col}(\aleph_1, \kappa)$, then, in $\mathsf{V}[\mathbb{G}]$, for any $(\mathsf{V}, \mathbb{P})$-generic $\mathbb{G}$, ${\aleph_2}^{\mathsf{V}[\mathbb{G}]}$ $(= \kappa)$ is tightly $\mathcal{P}$-closed-Laver-generically supercompact for the class $\mathcal{P}$ of all σ-closed posets (and CH holds).*

(B) *Suppose that κ is superhuge with a Laver function $f : \kappa \to V_\kappa$ for superhugeness. If $\mathbb{P}$ is the CS-iteration for forcing PFA along with f, then,*

in $\mathsf{V}[\mathbb{G}]$ *for any* $(\mathsf{V}, \mathbb{P})$*-generic* $\mathbb{G}$, $\aleph_2^{\mathsf{V}[\mathbb{G}]}$ $(= \kappa)$ *is tightly*$^+$ $\mathbb{P}$*-Laver-generically superhuge for the class* $\mathcal{P}$ *of all proper posets (and* $2^{\aleph_0} = \aleph_2$ *holds).*

(B') *Suppose that* κ *is superhuge with a Laver function* $f : \kappa \to V_\kappa$ *for superhugeness. If* $\mathbb{P}$ *is the RCS-iteration for forcing* MM *along with* f*, then, in* $\mathsf{V}[\mathbb{G}]$ *for any* $(\mathsf{V}, \mathbb{P})$*-generic* $\mathbb{G}$, $\aleph_2^{\mathsf{V}[\mathbb{G}]}$ $(= \kappa)$ *is tightly*$^+$ $\mathcal{P}$*-Laver-generically superhuge for the class* $\mathcal{P}$ *of all semi-proper posets (and* $2^{\aleph_0} = \aleph_2$ *holds).*

(Γ) *Suppose that* κ *is supercompact with a Laver function* $f : \kappa \to V_\kappa$ *for supercompactness. If* $\mathbb{P}$ *is a FS-iteration for forcing* MA *along with* f*, then, in* $\mathsf{V}[\mathbb{G}]$ *for any* $(\mathsf{V}, \mathbb{P})$*-generic* $\mathbb{G}$, $2^{\aleph_0}$ $(= \kappa)$ *is tightly*$^+$ $\mathcal{P}$*-Laver-generically supercompact for the class* $\mathcal{P}$ *of all ccc posets (and* $\kappa = 2^{\aleph_0}$ *while* κ *still is very large).*

(Δ) *Suppose that* κ *is supercompact with a Laver function* $f : \kappa \to V_\kappa$ *for supercompactness. If* $\mathbb{P}$ *is a FS-iteration for forcing* f *where* f *is used to book-keep through all posets in* V_κ*, then in* $\mathsf{V}[\mathbb{G}]$ *for any* $(\mathsf{V}, \mathbb{P})$*-generic* $\mathbb{G}$, $2^{\aleph_0}$ $(= \kappa)$ *is tightly*$^+$ $\mathcal{P}$*-Laver-generically supercompact for the class* $\mathcal{P}$ *of all posets (and* CH *holds).* □

Theorem 9 above also holds for all other notions of large cardinal and corresponding Laver-generic version of generic large cardinal except (B) and (B') in which the supercompactness does not seem to be strong enough to show that the resulting generic extension in the proof satisfies the expected Laver-genericity.

In a sense, the cases treated in Theorem 9 are (almost) exhaustive. This can be seen in the following:

Theorem 10 *Suppose that* $\mathcal{P}$ *is an iterable class of posets such that all* $\mathbb{P} \in \mathcal{P}$ *are* ω_1*-preserving and* $\mathcal{P}$ *contains a poset* $\mathbb{P}^*$ *whose generic filter destroys a stationary subset of* ω_1*.*[4] *Then there is no* $\mathcal{P}$*-Laver-generically supercompact cardinal.*

Proof Suppose, toward a contradiction, that $\mathcal{P}$ is as above and there is $\mathcal{P}$-Laver-generically supercompact cardinal κ.

Let $S \subseteq \omega_1$ be stationary such that there is a poset $\mathbb{P}^* \in \mathcal{P}$ shooting a club in $\omega_1 \setminus S$. Let $\lambda > |\mathbb{P}^*|$ be large enough. By assumption, there is a $\mathbb{P}^*$-name $\underset{\sim}{\mathbb{Q}}$ of a poset such that $\Vdash_{\mathbb{P}^*}$ "$\underset{\sim}{\mathbb{Q}} \in \mathcal{P}$" and, for $(\mathsf{V}, \mathbb{P}^* * \underset{\sim}{\mathbb{Q}})$-generic $\mathbb{H}$, there are j, $M \subseteq \mathsf{V}[\mathbb{H}]$ such that

$$j : \mathsf{V} \overset{\prec}{\to}_\kappa M, \tag{*4}$$

[4] "$\mathbb{P}^*$ destroys a stationary subset of ω_1" means here that a $\mathbb{P}^*$-generic set codes a club subset of $\omega_1 \setminus S$ in some absolute way.

Note that, for stationary and co-stationary subset S of ω_1, various posets are known which preserve ω_1 while shooting a club in $\omega_1 \setminus S$.

$$j(\kappa) > \lambda, \quad \text{and} \tag{*5}$$

$$j''\lambda, \mathbb{P}, \ \mathbb{H} \in M. \tag{*6}$$

By the choice of $\mathbb{P}^*$ ($\lessdot \mathbb{P}^* * \underset{\sim}{\mathbb{Q}}$), there is a nice $\mathcal{P}^*$-name $\underset{\sim}{C} \in \mathsf{V}$ of a club set $\subseteq {\omega_1}^{\mathsf{V}} \setminus S$. By (*5), (*6) and by the choice of λ, we have $\underset{\sim}{C} \in M$.

Thus $M \models$ "S is a non-stationary subset of ω_1" by (*6). Since $crit(j) = \kappa > \omega_1$ by Lemma 6, (1), we have $S = j(S)$. By $\mathsf{V} \models$ "$S$is stationary subset of ω_1", this is a contradiction to the elementarity (*4) of j. □ (Theorem 10)

$\mathcal{P}$-Laver genericity for stationary preserving $\mathcal{P}$, in particular those $\mathcal{P}$ containing all σ-closed posets can be regarded as a strong reflection principle.

Theorem 11 *Suppose that $\mathcal{P}$ is an iterable class of posets which are ω_1-preserving and include all σ-closed posets. If κ is $\mathcal{P}$-Laver-generically supercompact then*

(*1) *for any structure $\mathfrak{A}$ of countable signature, there is $\mathfrak{B} \prec_{L_{stat}} \mathfrak{A}$ of cardinality $< \aleph_2$.*

Proof By Theorem 5, $\mathsf{MA}^{++}(\sigma\text{-closed})$ holds. Cox [7] proved that $\mathsf{MA}^{++}(\sigma\text{-closed})$ implies $\mathsf{DRP}(< \aleph_2, \mathsf{IC}_{\aleph_0})$ (in the notation of [19]). By Lemma 3.5 in [19], this principle is equivalent to (*1). □ (Theorem 11)

The existence of $\mathcal{P}$-Laver generic large cardinal for the class of all ccc-posets also implies a reflection statement similar to (*1) (see Theorem 5.9, (3) in [20]). It is not known if the existence of $\mathcal{P}$-Laver generic large cardinal for the class of all ccc-posets imply FRP but it easy to see that we can force FRP and the existence of the Laver generic large cardinal starting from a model with a supercompact with the relevant large cardinal above it (Fuchino et al. [20]).

In summary, Laver-generic large cardinal axiom (i.e. the axiom asserting the existence of a $\mathcal{P}$-Laver-generic large cardinal for some iterable class of posets $\mathcal{P}$) is an assertion about the existence of a $\mathcal{P}$-generic large cardinal (without "Laver-" !) with the feature of resurrection property similar to the one in Resurrection Axioms studied by Hamkins and Johnstone [27, 28], and Tsaprounis [37]: for any generic extension by some $\mathbb{P} \in \mathcal{P}$ we find a further generic extension by a poset of the form $\mathbb{P} * \underset{\sim}{\mathbb{Q}} \in \mathcal{P}$ in which a previously chosen instance of generic elementary embedding corresponding to the generic largeness of the cardinal resurrects.

In spite of this similarity, it is only recent that we found there is a connection between these two similar types of axioms. First remember that, for each reasonable class $\mathcal{P}$ consisting of proper or stationary preserving posets $\mathcal{P}$, $\mathcal{P}$-Laver generic large cardinal axiom implies the $\mathcal{P}$-Laver generic large cardinal is $\kappa_{\mathfrak{refl}}$ (Theorem 9). In Fuchino [12] it is then proved that the tightly $\mathcal{P}$-Laver-generic superhuge axiom implies the boldface version Resurrection Axiom (in the sense of Hamkins and Johnstone) for parameters $\mathcal{P}$ and $\mathcal{H}(\kappa_{\mathfrak{refl}})$ (Fuchino [12]). The tightly $\mathcal{P}$-Laver generic ultrahuge axiom implies the tight version of the Unbounded Resurrection Axiom of Tsaprounis for $\mathcal{P}$ (Fuchino [12]).

$\mathcal{P}$-Laver generic large cardinal axioms for reasonable classes $\mathcal{P}$ consisting of proper or stationary preserving posets imply double plus versions of Forcing Axiom (Theorem 5). By Theorem 3 (and its proof), Theorem 11 and the remarks after that most of the known structural reflection principles down to $< \kappa_{\mathrm{refl}}$ are covered by these axioms.

$\mathcal{P}$-Laver generic large cardinal axioms also support the intuition shared by many set-theorists that the continuum is either $\aleph_1$ or $\aleph_2$ or very large (Theorems 8–10). Perhaps we can also formulate this circumstance as: if we are to support one of the reasonable instances of Laver-generic large cardinal axiom then the continuum must be either $\aleph_1$ or $\aleph_2$ or extremely large ($\aleph_{2^{\aleph_0}}$ in particular, and much more).

Laver generic large cardinal axiom can be considered as a strong reflection principle with the feature of resurrection (à la Hamkins and Johnstone). Certain amount of absoluteness is also inherent in Laver generic large cardinal axiom in that it implies strong versions of forcing axioms (Theorem 5). In Sect. 5 we shall see that more generic absoluteness in terms of recurrence axioms (see the next section for these axioms) can be integrated into the Laver genericity by moving to the notions of large cardinals with the additional properties (in plural since they can be formulated only in an axiom scheme—or in some cases even not first-order formalizable) which we call "super $C^{(\infty)}$" (Theorem 28) and that this is still realizable below the consistency strength of 2-huge (Theorems 29 and 30).

We can also feel more confident (or insecure in case you do not believe in large large cardinals[5]) with these stronger versions of Laver-generic large cardinal axiom since their consistency strength can be decided (see Corollaries 34–37).

3 Recurrence, Maximality, and the Solution(s) of the Continuum Problem

The Recurrence Axioms we are going to introduce below can be considered as reflection statements in set-generic multiverse down to set-generic geology. Their strengthenings can be also considered as a sort of absoluteness statements (see Proposition 16).

Adopting the terminology of set-theoretic geology (see e.g. Fuchs et al. [22]), an inner model W of a universe U (in most of the cases U is the real universe V but sometimes it is some other universe obtained from V) is called a *ground of* U, if there is a poset $\mathbb{P} \in \mathsf{W}$ and $(\mathsf{W}, \mathbb{P})$-generic $\mathbb{G} \in \mathsf{U}$ such that $\mathsf{U} = \mathsf{W}[\mathbb{G}]$.

For a class $\mathcal{P}$ of posets and a set A (of parameters), the *Recurrence Axiom for* $\mathcal{P}$ *and* A ($(\mathcal{P}, A)$-RcA, for short[6]) is the following assertion formulated as an axiom scheme in the language $\mathcal{L}_{\in}$ of set theory:

[5] Note (not only) for the editor: "large large cardinals" is not a typo.

[6] The notation "RcA" is chosen to avoid the collision with "RCA" which is used in Reverse Mathematics to denote "recursive comprehension axiom".

$(\mathcal{P}, A)$-RcA: For any $\mathcal{L}_\in$-formula $\varphi = \varphi(\overline{x})$ and $\overline{a} \in A$, if $\Vdash_{\mathbb{P}}$ "$\varphi(\overline{a}^{\check{}})$" for a $\mathbb{P} \in \mathcal{P}$, then there is a ground W of the universe V such that $\overline{a} \in \mathsf{W}$ and $\mathsf{W} \models \varphi(\overline{a})$.[7]

The term "Recurrence Axiom" is chosen in allusion to, but not necessarily in (full) agreement with, Nietzsche's „ ewige Wiederkehr des Gleichen" (eternal recurrence of the same), or perhaps rather to Poincaré recurrence theorem: if we understand the relation "N is (set) generic extension of M" as the timeline in the set generic multiverse, we can interpret $(\mathcal{P}, A)$-RcA as saying that

> *if something (formulatable with parameters in A) happens in one of the near future universe (in terms of $\mathcal{P}$) then is already happened in a not very far past universe (not very far, in the sense that the "present" is attainable from there by a set forcing).*

The following is a natural strengthening of the Recurrence Axiom:

$(\mathcal{P}, A)$-RcA$^+$: For any $\mathcal{L}_\in$-formula $\varphi = \varphi(\overline{x})$ and $\overline{a} \in A$, if $\Vdash_{\mathbb{P}}$ "$\varphi(\overline{a}^{\check{}})$" for a $\mathbb{P} \in \mathcal{P}$, then there is a $\mathcal{P}$-ground W of the universe V such that $\overline{a} \in \mathsf{W}$ and $\mathsf{W} \models \varphi(\overline{a})$.

Here an inner model W of V is called a $\mathcal{P}$- *ground* if there is a poset $\mathbb{P} \in \mathsf{W}$ with $\mathsf{W} \models \mathbb{P} \in \mathcal{P}$ and $(\mathsf{W}, \mathbb{P})$-generic $\mathbb{G} \in \mathsf{V}$ such that $\mathsf{V} = \mathsf{W}[\mathbb{G}]$.

When we regard reflection principles including Laver-generic large cardinal axioms as natural axioms, the idea behind this should be that the continuum must be rich enough so that many aspects of the situations with sets of high cardinality should be reflected down to cardinality $< \kappa_{\mathrm{refl}}$.

Similarly we can consider Recurrence Axioms and their strengthenings as natural requirements: since our set-theoretic universe V must be highly saturated, as many aspects as possible of the situations in set generic multiverse should be reflected down to the set generic geology at which we can look down form our universe V.

We shall use the following version of Laver-Woodin Theorem often without mention. It implies in particular that $(\mathcal{P}, A)$-RcA and $(\mathcal{P}, A)$-RcA$^+$ are actually formalizable as axiom schemes in $\mathcal{L}_\in$.

Theorem 12 (Reitz [36], Fuchs-Hamkins-Reitz [22]) *There is an $\mathcal{L}_\in$-formula $\Phi(x, r)$ (without any parameters) such that the following is provable in ZFC:*

(*7) *for all r, $\Phi(\cdot, r) := \{x : \Phi(s, r)\}$ is a ground in V,*
(*8) *for any ground W (of V), there is r such that $\mathsf{W} = \Phi(\cdot, r)$, and*
(*9) *if W is a ground of V and $\mathsf{V} = \mathsf{W}[\mathbb{G}]$ where $\mathbb{G}$ is a $(\mathsf{W}, \mathbb{P})$-generic for $\mathbb{P} \in \mathsf{W}$, then r such that $\mathsf{W} = \Phi(\cdot, r)$ can be chosen as an element of $\mathcal{P}((|\mathbb{P}|^+)^{\mathsf{V}})$.*

We put together here some other basic facts which will be used in the following. The next lemma was actually already used in the proof of Lemma 4.

[7] For sets $a_0, \ldots, a_{n-1}$ and $\overline{a} := \langle a_0, \ldots, a_{n-1}\rangle$, we simply write "$\overline{a} \in A$" for "$a_0, \ldots, a_{n-1} \in A$". For $\mathbb{P}$-check names (standard $\mathbb{P}$-names of the elements $a_0, \ldots, a_{n-1}$ in V) $\check{a}_0, \ldots, \check{a}_{n-1}$ we write $\overline{a}^{\check{}} := \langle \check{a}_0, \ldots, \check{a}_{n-1}\rangle$.

Lemma 13 (See Fuchino and Usuba [17] for a Proof) *If α is a limit ordinal and V_α satisfies a large enough fragment of ZFC, then for any $\mathbb{P} \in V_\alpha$ and $(\mathsf{V}, \mathbb{P})$-generic $\mathbb{G}$ we have $V_\alpha[\mathbb{G}] = V_\alpha^{\mathsf{V}[\mathbb{G}]}$.* □

Lemma 14 *For any $n \in \mathbb{N} \setminus 1$, there is a Σ_n-formula $\varphi_n = \varphi_n(\overline{x}, y)$ (for each finite sequence $\overline{x}$ of variables[8]) such that, if U is a transitive models of large enough fragment of ZFC, then for any Σ_n formula $\psi = \psi(\overline{x})$ there is $p \in \mathcal{H}(\omega)$ such that $\mathsf{U} \models \psi(\overline{a})$ if and only if $\mathsf{U} \models \varphi_n(\overline{a}, p)$ for all $a \in \mathsf{U}$.* □

Lemma 15 *For any $n^* \in \mathbb{N}$ there is $n > n^*$ such that, if $V_\alpha \prec_{\Sigma_n} \mathsf{V}$, then $V_\alpha[\mathbb{G}] \prec_{\Sigma_{n^*}} \mathsf{V}[\mathbb{G}]$ for any $\mathbb{P} \in V_\alpha$ and $(V, \mathbb{P})$-generic $\mathbb{G}$.* □

RcA and RcA^+ are actually (almost) identical with (certain variations of) already well-known axioms and principles.

For a class $\mathcal{P}$ of posets, an $\mathcal{L}_\in$-formula $\varphi(\overline{a})$ with parameters $\overline{a}$ ($\in \mathsf{V}$) is said to be a *$\mathcal{P}$-button* if there is $\mathbb{P} \in \mathcal{P}$ such that, for any $\mathbb{P}$-name $\underset{\sim}{\mathbb{Q}}$ of poset with $\Vdash_{\mathbb{P}}$ "$\underset{\sim}{\mathbb{Q}} \in \mathcal{P}$", we have $\Vdash_{\mathbb{P}*\underset{\sim}{\mathbb{Q}}}$ "$\varphi(\overline{a}^{\check{}})$".

If $\varphi(\overline{a})$ is a $\mathcal{P}$-button then we call $\mathbb{P}$ as above a *push of the button $\varphi(\overline{a})$*.

For a class $\mathbb{P}$ of posets and a set A (of parameters), the *Maximality Principle for $\mathcal{P}$ and A* ($\mathsf{MP}(\mathcal{P}, A)$, for short) introduced in Hamkins [26] is the following assertion formulated in an axioms scheme in $\mathcal{L}_\in$:

$\mathsf{MP}(\mathcal{P}, A)$: For any $\mathcal{L}_\in$-formula $\varphi(\overline{x})$ and $\overline{a} \in A$, if $\varphi(\overline{a})$ is a $\mathcal{P}$-button then $\varphi(\overline{a})$ holds.

Proposition 16 (Barton et al. [3]) *Suppose that $\mathcal{P}$ is an iterable class of posets and A a set (of parameters).* (1) *$(\mathcal{P}, A)$-RcA^+ is equivalent to $\mathsf{MP}(\mathcal{P}, A)$.*
(2) *$(\mathcal{P}, A)$-RcA is equivalent to the following assertion:*

(*10) *For any $\mathcal{L}_\in$-formula $\varphi(\overline{x})$ and $\overline{a} \in A$, if $\varphi(\overline{a})$ is a $\mathcal{P}$-button then $\varphi(\overline{a})$ holds in a ground of V.*

Proof (1) Suppose first that $(\mathcal{P}, A)$-RcA^+ holds. We show that $\mathsf{MP}(\mathcal{P}, A)$ holds. Suppose that $\mathbb{P} \in \mathcal{P}$ is a push of the $\mathcal{P}$-button $\varphi(\overline{a})$. Let $\varphi'(\overline{x})$ be the formula expressing

$$\text{for any } \mathbb{Q} \in \mathcal{P},\ \Vdash_{\mathbb{Q}} \text{"}\varphi(\overline{x}^{\check{}})\text{" holds.} \tag{*11}$$

Then we have $\Vdash_{\mathbb{P}}$ "$\varphi'(\overline{a}^{\check{}})$". By $(\mathcal{P}, A)$-RcA^+, there is a $\mathcal{P}$-ground M of V such that $\overline{a} \in M$ and $M \models \varphi'(\overline{a})$ holds. By the definition (*11) of φ', it follows that $\mathsf{V} \models \varphi(\overline{a})$ holds.

Now suppose that $\mathsf{MP}(\mathcal{P}, A)$ holds and $\mathbb{P} \in \mathcal{P}$ is such that $\Vdash_{\mathbb{P}}$ "$\varphi(\overline{a})$" for $\overline{a} \in A$.

[8] Similarly to the convention of some computer languages we consider here that we have distinct $\varphi_n(\overline{x}, y)$ for each length of sequence $\overline{x}$ of variables.

Let φ' be a formula claiming that

$$\text{there is a } \mathcal{P}\text{-ground } N \text{ such that } \overline{x} \in N \text{ and } N \models \varphi(\overline{x}). \tag{*12}$$

Then $\varphi'(\overline{a})$ is a $\mathcal{P}$-button and $\mathbb{P}$ is its push.

By $\mathsf{MP}(\mathcal{P}, A)$, $\varphi'(\overline{a})$ holds in V and hence there is a $\mathcal{P}$-ground M of V such that $\overline{a} \in M$ and $M \models \varphi(\overline{a})$. This shows that $(\mathcal{P}, A)$-RcA^+ holds.

(2) Can be proved similarly to (1). Suppose first that $(\mathcal{P}, A)$-RcA holds. We show that (*10) holds. Suppose that $\mathbb{P} \in \mathcal{P}$ is a push of the $\mathcal{P}$-button $\varphi(\overline{a})$. Let $\varphi'(\overline{x})$ be the formula expressing

$$\text{for any } \mathbb{Q} \in \mathcal{P},\ \Vdash_{\mathbb{Q}} \text{“}\varphi(\overline{x}^{\check{}})\text{” holds.} \tag{*13}$$

Then we have $\Vdash_{\mathbb{P}} \text{“}\varphi'(\overline{a}^{\check{}})\text{”}$. By $(\mathcal{P}, A)$-RcA, there is a ground M of V such that $\overline{a} \in M$ and $M \models \varphi'(\overline{a})$ holds. Since $\mathcal{P} \ni \{\mathbb{1}\}$, it follows that $M \models \varphi(\overline{a})$.

Now suppose that (*10) holds and $\mathbb{P} \in \mathcal{P}$ is such that $\Vdash_{\mathbb{P}} \text{“}\varphi(\overline{a}^{\check{}})\text{”}$ for $\overline{a} \in A$.

Let φ'' be a formula asserting that

$$\text{there is a } \mathcal{P}\text{-ground } N \text{ such that } \overline{x} \in N \text{ and } N \models \varphi(\overline{x}). \tag{*14}$$

Then $\varphi''(\overline{a})$ is a $\mathcal{P}$-button and $\mathbb{P}$ is its push. Thus, By (*10), $\varphi''(\overline{a})$ holds in a ground M of V with $\overline{a} \in M$. By the definition *14 of φ'', there is a $\mathcal{P}$-ground N of M such that $\overline{a} \in N$ and $N \models \varphi(\overline{a})$. Since N is also a ground of V, this shows that $(\mathcal{P}, A)$-RcA holds. □ (Theorem 16)

Recurrence Axioms are also related to the Inner Model Hypothesis introduced by Sy Friedman in [9]. *The Inner Model Hypothesis* (*IMH*) is the following assertion formulated in the language of second-order set theory (e.g. in the context of von Neumann-Bernays-Gödel set theory):

IMH : For any statement φ without parameters, if φ holds in an inner model of an inner extension of V then φ holds in an inner model of V.

Here we say a (not necessarily first-order definable) transitive class M an *inner model* of V if M is a model of ZF and $\mathsf{On}^M = \mathsf{On}^{\mathsf{V}}$. In the perspective from such M, we call V an *inner extension* of M.

We call a set-forcing version of this principle *Inner Ground Hypothesis* (IGH): For a (definable) class $\mathcal{P}$ of posets and a set A (of parameters),

IGH$(\mathcal{P}, A)$: For any $\mathcal{L}_\in$-formula $\varphi = \varphi(\overline{x})$ and $\overline{a} \in A$, if $\mathbb{P} \in \mathcal{P}$ forces “there is a ground M with $\overline{a} \in M$ satisfying $\varphi(\overline{a})$”, then there is a ground W of V such that $\overline{a} \in \mathsf{W}$ and $\mathsf{W} \models \varphi(\overline{a})$.

Proposition 17 (Barton et al. [3]) *For a class $\mathcal{P}$ of posets with $\{\mathbb{1}\} \in \mathcal{P}$ and a set A (of parameters), $(\mathcal{P}, A)$-RcA holds if and only if* IGH$(\mathcal{P}, A)$ *holds.*

Proof Suppose that $(\mathcal{P}, A)$-RcA holds. Let $\varphi = \varphi(\overline{x})$ be an $\mathcal{L}_\in$-formula, $\overline{a} \in A$, and $\mathbb{P} \in \mathcal{P}$ be such that $\Vdash_{\mathbb{P}} \text{“}\varphi(\overline{a}^{\check{}})$holds in a ground”.

Let $\varphi'(\overline{x})$ be the $\mathcal{L}_\in$-formula asserting that $\varphi(\overline{x})$ holds in a ground. Then $\Vdash_{\mathbb{P}}$ "$\varphi'(\overline{a}^{\checkmark})$". By $(\mathcal{P}, A)$-RcA, it follows that there is a ground W of V such that $\mathsf{W} \models \varphi'(\overline{a}^{\checkmark})$. Since a ground of a ground is a ground, we conclude that there is a ground W_0 of V such that $\overline{a} \in M_0$ and $\mathsf{W}_0 \models \varphi(\overline{a})$. This shows that $\mathsf{IGH}(\mathcal{P}, A)$ holds.

Suppose now that $\mathsf{IGH}(\mathcal{P}, A)$ holds. Assume that $\Vdash_{\mathbb{P}}$ "$\varphi(\overline{a}^{\checkmark})$" for an $\mathcal{L}_\in$-formula $\varphi = \varphi(\overline{x})$, $\overline{a} \in A$, and $\mathbb{P} \in \mathcal{P}$. Then $\Vdash_{\mathbb{P}}$ "$\varphi(\overline{a}^{\checkmark})$holds in a $\mathcal{P}$-ground (of the universe)" since $\Vdash_{\mathbb{P}}$ "$\{\mathbb{1}\} \in \mathcal{P}$". Thus, by $\mathsf{IGH}(\mathcal{P}, A)$, there is a ground W of V such that $\mathsf{W} \models \varphi(\overline{a})$. □ (Proposition 17)

$(\mathcal{P}, A)$-RcA$^+$ ($\Leftrightarrow$ $\mathsf{MP}(\mathcal{P}, A)$ for an iterable $\mathcal{P}$) can be also characterized in terms of a strengthening of Inner Ground Hypothesis: For a (definable) class $\mathcal{P}$ of posets and a set A (of parameters),

$\mathsf{IGH}^+(\mathcal{P}, A)$: For any $\mathcal{L}_\in$-formula $\varphi = \varphi(\overline{x})$ and $\overline{a} \in A$ if $\mathbb{P} \in \mathcal{P}$ forces "there is a $\mathcal{P}$-ground M with $\overline{a} \in M$ satisfying $\varphi(\overline{a})$", then there is a $\mathcal{P}$-ground W of V such that $\overline{a} \in \mathsf{W}$ and $\mathsf{W} \models \varphi(\overline{a})$.

The following proposition can be proved similarly to Proposition 17.

Proposition 18 *For an iterable class $\mathcal{P}$ of posets and a set A (of parameters), $(\mathcal{P}, A)$-RcA$^+$ holds if and only if $\mathsf{IGH}^+(\mathcal{P}, A)$ holds.* □

In spite of these characterizations and near characterizations, we want to keep the Recurrence Axioms as autarchic axioms. On of the reasons is that we want to retain the narration that is a reflection principle with reflection from the set-theoretic multiverse to the geology; another reason is that we have the following monotonicity which does not hold e.g. for Maximality Principles.

Lemma 19 (Monotonicity of Recurrence Axioms) *For classes of posets $\mathcal{P}$, $\mathcal{P}'$ and sets A, A' of parameters, if $\mathcal{P} \subseteq \mathcal{P}'$ and $A \subseteq A'$, then we have*

$$(\mathcal{P}', A')\text{-RcA} \;\Rightarrow\; (\mathcal{P}, A)\text{-RcA}.$$ □

If we decide that the Recurrence Axioms are desirable extensions of the axioms of ZFC, then we should adopt the maximal instance of these axioms (i.e. the one with maximal strength among the instances consistent with ZFC) By Lemma 19, this means we should try to take the instance of Recurrence Axioms with the maximal $\mathcal{P}$ and A (with respect to inclusion) among the consistent ones.

Theorem 20 in the next section suggests that the following two as candidates of such maximal instances:

(E) ZFC + $(\mathcal{P}, \mathcal{H}(\kappa_{\mathfrak{refl}}))$-RcA for the class $\mathcal{P}$ of all stationary preserving posets.

(Z) ZFC + $(\mathcal{Q}, \mathcal{H}(2^{\aleph_0}))$-RcA for the class $\mathcal{Q}$ of all posets.

The consistency of (Z) follows from the consistency of ZFC + "there are stationarily many inaccessible cardinals" [26]. The consistency of (E) follows from Lemma 29, Theorem 30, (B'), and Theorem 28.

The maximality of (E) and (Z) follows from Lemma 20, (2') and (5') respectively.

By Lemma 20, (4) and (5), (E) implies $2^{\aleph_0} = \aleph_2$, and (Z) implies CH. In particular, these two extensions of ZFC are not compatible. However, as we are going to discuss in Sect. 7, we can combine (E) with a reasonable weakening of (Z).

(Z^+) $\mathsf{ZFC} + (\mathcal{P}, \mathcal{H}(\kappa_{\mathrm{refl}}))\text{-}\mathsf{RcA}^+ + (\mathcal{Q}, \mathcal{H}(\omega_1)^{\overline{\mathsf{W}}})\text{-}\mathsf{RcA}^+$ where $\mathcal{P}$ is the class of all proper posets, $\mathcal{Q}$ the class of all posets, and $\overline{\mathsf{W}}$ the bedrock[9] which is also assumed here to exist.

(Z^+) implies $2^{\aleph_0} = \aleph_2$ (see Lemma 20, (4) in the next section). This is what I meant when I wrote "the maximal setting of Recurrence Axioms points to the universe with the continuum of size $\aleph_2$" in the introduction. We shall further discuss about (Z^+) in Sect. 6.

4 Restricted Recurrence Axioms

The following restricted forms of Recurrence Axioms are enough to decide many interesting aspects including the cardinal arithmetic around the continuum.

For an iterable class $\mathcal{P}$ of posets, a set A (of parameters), and a set Γ of $\mathcal{L}_\in$-formulas, *$\mathcal{P}$-Recurrence Axiom for formulas in Γ with parameters from A* ($(\mathcal{P}, A)_\Gamma$-RcA, for short) is the following assertion expressed as an axiom scheme in $\mathcal{L}_\in$:

$(\mathcal{P}, A)_\Gamma$-RcA: For any $\varphi(\overline{x}) \in \Gamma$ and $\overline{a} \in A$, if $\Vdash_{\mathbb{P}}$ "$\varphi(\overline{a}^{\check{}})$", then there is a ground W of V such that $\overline{a} \in \mathsf{W}$ and $\mathsf{W} \models \varphi(\overline{a})$.

$(\mathcal{P}, A)_\Gamma$-RcA^+ corresponding to $(\mathcal{P}, A)$-RcA^+ is defined similarly.

Lemma 20 (Fuchino and Usuba [17]) *Assume that $\mathcal{P}$ is an iterable class of posets.*

(1) *If $\mathcal{P}$ contains a poset which adds a real (over the universe), then $(\mathcal{P}, \mathcal{H}(\kappa_{\mathrm{refl}}))_{\Sigma_1}$-$\mathsf{RcA}$ implies $\neg\mathsf{CH}$.*

(2) *Suppose that $\mathcal{P}$ contains a poset which forces ${\aleph_2}^{\mathsf{V}}$ to be equinumerous with ${\aleph_1}^{\mathsf{V}}$. Then $(\mathcal{P}, \mathcal{H}(2^{\aleph_0}))_{\Sigma_1}$-$\mathsf{RcA}$ implies $2^{\aleph_0} \leq \aleph_2$.*

(2') *If $\mathcal{P}$ contains a posets which forces ${\aleph_2}^{\mathsf{V}}$ to be equinumerous with ${\aleph_1}^{\mathsf{V}}$, then $(\mathcal{P}, \mathcal{H}((\aleph_2)^+))_{\Sigma_1}$-$\mathsf{RcA}$ does not hold.*

(3) *If $(\mathcal{P}, \mathcal{H}(\kappa_{\mathrm{refl}}))_{\Sigma_1}$-$\mathsf{RcA}$ holds then all $\mathbb{P} \in \mathcal{P}$ preserve $\aleph_1$ and they are also stationary preserving.*

(4) *If $\mathcal{P}$ contains a poset which adds a real as well as a poset which collapses ${\aleph_2}^{\mathsf{V}}$, then $(\mathcal{P}, \mathcal{H}(\kappa_{\mathrm{refl}}))_{\Sigma_1}$-$\mathsf{RcA}$ implies $2^{\aleph_0} = \aleph_2$.*

[9] For the definition of the bedrock see Sect. 5.

(5) *If $\mathcal{P}$ contains a poset which collapses $\aleph_1^{\mathsf{V}}$, then $(\mathcal{P}, \mathcal{H}(2^{\aleph_0}))_{\Sigma_1}$-RcA implies CH.*

(5') *If $\mathcal{P}$ contains a poset which collapses $\aleph_1^{\mathsf{V}}$ then $(\mathcal{P}, \mathcal{H}((2^{\aleph_0})^+))_{\Sigma_1}$- RcA does not hold.*

Proof All assertions are proved similarly. We see here only the proof of (1): Assume that $\mathcal{P}$ is an iterable class of posets containing a poset $\mathbb{P}$ adding a real and $(\mathcal{P}, \mathcal{H}(\kappa_{\mathfrak{refl}}))_{\Sigma_1}$- RcA holds. If CH holds, then $\mathcal{P}(\omega)^{\mathsf{V}} \in \mathcal{H}(\kappa_{\mathfrak{refl}})$. Hence

$$\text{“}\exists x\, (x \subseteq \omega \wedge x \notin \mathcal{P}(\omega)^{\mathsf{V}})\text{”} \tag{*15}$$

is a Σ_1-formula with parameters from $\mathcal{H}(\kappa_{\mathfrak{refl}})$ and $\mathbb{P}$ forces the formula in the forcing language corresponding to this formula: “$\exists x\, (s \subseteq \check{\omega} \wedge x \notin (\mathcal{P}(\omega)^{\mathsf{V}})^{\check{}})$”.

By $(\mathcal{P}, \mathcal{H}(\kappa_{\mathfrak{refl}}))_{\Sigma_1}$- RcA, the formula (*15) must hold in a ground. This is a contradiction. □ (Lemma 20)

Laver-genericity implies the plus version of Recurrence Axiom ($\Leftrightarrow$ Maximality Principle) restricted to Σ_2.

Theorem 21 *Suppose that κ is tightly $\mathcal{P}$-Laver-generically ultrahuge for an iterable class $\mathcal{P}$ of posets. Then $(\mathcal{P}, \mathcal{H}(\kappa))_{\Sigma_2}$-RcA$^+$ holds.*

Proof Assume that κ is tightly $\mathcal{P}$-Laver generically ultrahuge for an iterable class $\mathcal{P}$ of posets.

Suppose that $\varphi = \varphi(\overline{x})$ is Σ_2 formula (in $\mathcal{L}_\in$), $\overline{a} \in \mathcal{H}(\kappa)$, and $\mathbb{P} \in \mathcal{P}$ is such that

$$\mathsf{V} \models \Vdash_{\mathbb{P}} \text{“}\varphi(\overline{a}^{\check{}})\text{”}. \tag{*16}$$

Let $\lambda > \kappa$ be such that $\mathbb{P} \in \mathsf{V}_\lambda$ and

$$V_\lambda \prec_{\Sigma_n} \mathsf{V} \text{ for a sufficiently large } n. \tag{*17}$$

In particular, we may assume that we have chosen the n above so that a sufficiently large fragment of ZFC holds in V_λ (“sufficiently large” means here, in particular, in terms of Lemma 13 and that the argument at the end of this proof is possible).

Let $\underset{\sim}{\mathbb{Q}}$ be a $\mathbb{P}$-name such that $\Vdash_{\mathbb{P}}$ “$\underset{\sim}{\mathbb{Q}} \in \mathcal{P}$”, and for $(\mathsf{V}, \mathbb{P} * \underset{\sim}{\mathbb{Q}})$-generic $\mathbb{H}$, there are j, $M \subseteq \mathsf{V}[\mathbb{H}]$ with

$$j : \mathsf{V} \overset{\prec}{\to}_\kappa M, \tag{*18}$$

$$j(\kappa) > \lambda, \tag{*19}$$

$$\mathbb{P} * \underset{\sim}{\mathbb{Q}},\ \mathbb{P},\ \mathbb{H},\ V_{j(\lambda)}{}^{\mathsf{V}[\mathbb{H}]} \in M, \text{ and} \tag{*20}$$

$$|\,\mathbb{P} * \underset{\sim}{\mathbb{Q}}\,| \leq j(\kappa). \tag{*21}$$

By (*21), we may assume that the underlying set of $\mathbb{P} * \underset{\sim}{\mathbb{Q}}$ is $j(\kappa)$ and $\mathbb{P} * \underset{\sim}{\mathbb{Q}} \in V_{j(\lambda)}{}^{\mathsf{V}}$.

Let $\mathbb{G} := \mathbb{H} \cap \mathbb{P}$. Note that $\mathbb{G} \in M$ by (*20) and we have

$$V_{j(\lambda)}{}^{M} \underbrace{=}_{\text{by (*20)}} V_{j(\lambda)}{}^{\mathsf{V}[\mathbb{H}]} \overbrace{=}^{\substack{\text{Since } V_{j(\lambda)}{}^{M} (= V_{j(\lambda)}^{\mathsf{V}[\mathbb{H}]}) \text{ satisfies a sufficiently large fragment of } \mathsf{ZFC} \\ \text{by elementarity of } j, \text{ and hence the equality follows by Lemma 13}}} V_{j(\lambda)}{}^{\mathsf{V}}[\mathbb{H}]. \tag{*22}$$

Thus, by (*20) and by the definability of grounds, we have $V_{j(\lambda)}{}^{\mathsf{V}} \in M$ and $V_{j(\lambda)}{}^{\mathsf{V}}[\mathbb{G}] \in M$.

Claim 1 $V_{j(\lambda)}{}^{\mathsf{V}}[\mathbb{G}] \models \varphi(\overline{a})$.

$\vdash$ By Lemma 13, $V_{\lambda}{}^{\mathsf{V}}[\mathbb{G}] = V_{\lambda}{}^{\mathsf{V}[\mathbb{G}]}$, and $V_{j(\lambda)}{}^{\mathsf{V}}[\mathbb{G}] = V_{j(\lambda)}{}^{\mathsf{V}[\mathbb{G}]}$. By (*17), both $V_{\lambda}{}^{\mathsf{V}}[\mathbb{G}]$ and $V_{j(\lambda)}^{\mathsf{V}}[\mathbb{G}]$ satisfy large enough fragment of ZFC. Thus

$$V_{\lambda}{}^{\mathsf{V}}[\mathbb{G}] \prec_{\Sigma_1} V_{j(\lambda)}{}^{\mathsf{V}}[\mathbb{G}]. \tag{*23}$$

By (*16) and (*17), we have $V_{\lambda}{}^{\mathsf{V}}[\mathbb{G}] \models \varphi(\overline{a})$. By (*23) and since φ is Σ_2, it follows that $V_{j(\lambda)}{}^{\mathsf{V}}[\mathbb{G}] \models \varphi(\overline{a})$. $\dashv$ (Claim 1)

Thus we have

$$M \models \text{“there is a } \mathcal{P}\text{-ground } N \text{ of } V_{j(\lambda)} \text{ with } N \models \varphi(\overline{a})\text{”}. \tag{*24}$$

By the elementarity (*18), it follows that

$$\mathsf{V} \models \text{“there is a } \mathcal{P}\text{-ground } N \text{ of } V_{\lambda} \text{ with } N \models \varphi(\overline{a})\text{”}. \tag{*25}$$

Now by (*17), it follows that there is a $\mathcal{P}$-ground W of V such that $\mathsf{W} \models \varphi(\overline{a})$. $\square$ (Theorem 21)

5 Recurrence, Laver-Generic Large Cardinal, and Beyond

Laver-genericity does not imply full Recurrence Axiom (Theorem 26 and Corollary 27 below).

For an $\mathcal{L}_\in$-formula $\psi = \psi(\overline{x})$, a (large) cardinal κ is ψ- *absolute* if the formula ψ is absolute between V_κ and V (i.e. $(\forall \overline{x} \in V_y)(\psi^{V_y}(\overline{x}) \leftrightarrow \psi(\overline{x}))$ holds for $y = \kappa$).

Lemma 22 *For any $n \in \mathbb{N}$, there is an $\mathcal{L}_\in$-formula ψ_n^* such that, for any inaccessible κ, κ is ψ_n^*-absolute if and only if*

(*26) *for any ground W of V such that $\mathsf{V} = \mathsf{W}[\mathbb{G}]$ for a poset $\mathbb{P} \in {V_\kappa}^{\mathsf{W}}$ and $(\mathsf{W}, \mathbb{P})$-generic $\mathbb{G}$,*[10] *we have that all Σ_n-formulas are absolute between ${V_\kappa}^{\mathsf{W}}$ and W.*

Proof By Theorem 12 and Lemma 14. □ (Lemma 22)

Lemma 23 *Let ψ_n^* be as in Lemma 22. ψ_n^*-absolute inaccessible cardinals are not resurrectable. That is, if a cardinal λ satisfies*

$$\Vdash_{\mathbb{P}} \text{“}\check{\lambda} \text{ is } \psi_n^*\text{-absolute inaccessible”} \tag{*27}$$

for some poset $\mathbb{P}$, then λ is really ψ_n^*-absolute inaccessible.

Proof (*27) implies that λ is inaccessible. By the definition (*26) of ψ_n^*, if ψ_n^* is absolute between ${V_\lambda}^{\mathsf{V}[\mathbb{G}]}$ and $\mathsf{V}[\mathbb{G}]$ for some $(\mathsf{V}, \mathbb{P})$-generic $\mathbb{G}$ then, it is absolute between ${V_\lambda}^{\mathsf{V}}$ and V. □ (Lemma 23)

Lemma 24 *Suppose that there are stationarily many inaccessible cardinals.*[11] *Then, for each $n \in \mathbb{N}$, there are stationarily many ϕ_n^*-absolute inaccessible cardinals.*

Proof For $n \in \mathbb{N}$ let $n^+ \geq n$ be such that ψ_n^* is Σ_{n^+}. For any club $C \subseteq \mathsf{On}$, $C \cap C^{(n^+)} = \{\alpha \in C : V_\alpha \prec_{\Sigma_{n^+}} \mathsf{V}\}$ is a club in On (Lévy-Montague Reflection Theorem), there is an inaccessible cardinal $\mu \in C \cap C^{(n^+)}$. By the choice of n^+, such μ is a ψ_n^*-absolute inaccessible cardinal. □ (Lemma 24)

Theorem 25 *Suppose that $(\mathcal{P}, \emptyset)$-RcA holds, where $\mathcal{P}$ is a class of posets such that either* (a) *$\mathcal{P}$ contains posets collapsing arbitrary large cardinals to a small cardinality (less than the first inaccessible if there are inaccessibles at all), or* (b) *$\mathcal{P}$ contains posets adding arbitrarily many reals.*

If there is a ψ_n^*-absolute inaccessible cardinal for some $n \in \mathbb{N}$, then there are cofinally many ψ_n^*-absolute inaccessible cardinals.

Proof Assume that $(\mathcal{P}, \emptyset)$-RcA holds for $\mathcal{P}$ as above and there is a cardinal λ such that there are some ψ_n^*-absolute inaccessible cardinals but all of them are below λ.

[10] Note that this includes the case that $\mathbb{P} = \{\mathbb{1}\}$ and $\mathsf{V} = \mathsf{W}$.

[11] “There are stationarily many inaccessible cardinals” is the statement formalizable in an axiom scheme claiming, for each $\mathcal{L}_\in$-formula $\varphi = \varphi(x)$, that “if $\varphi(x)$ defines a club subclass of On then there is an inaccessible μ with $\varphi(\mu)$”.

Let $\mathbb{P}$ be a poset which either collapses λ to small cardinality or add at least λ many reals. Then, by Lemma 23, we have $\Vdash_{\mathbb{P}}$ "there is no ψ_n^*-absolute inaccessible cardinal".

By $(\mathcal{P}, \emptyset)$-RcA, it follows that there is a ground W of V such that W $\models$ "there is no ψ_n^*-absolute inaccessible cardinal". Again by Lemma 23, this is a contradiction. □ (Theorem 25)

Theorem 26 *Suppose that λ is an inaccessible cardinal, $\kappa < \lambda$ is such that $V_\lambda \models$ "$\kappa$is x-large cardinal", where "x-large cardinal" is a notion of large cardinal, for which a Laver function exists. Assume also that $\{\mu < \lambda : \mu$ is inaccessible$\}$ is stationary in λ.*

Then, for each of the classes $\mathcal{P}$ of posets considered in Theorem 9, there are λ_0 with $\lambda > \lambda_0 > \kappa$, and $\mathbb{P} \in \mathcal{P}$ with $\mathbb{P} \subseteq V_\kappa$ such that, for a $(V_\lambda, \mathbb{P})$-generic $\mathbb{G}$, we have

$$V_{\lambda_0}[\mathbb{G}] \models \text{"}\kappa \text{ is a tightly}^+\mathcal{P}\text{-Laver-generically x-large cardinal and } \neg(\mathcal{P}, \emptyset)\text{-RcA"}.$$

Proof Suppose that $\mathcal{P}$ is one of the classes of posets considered in Theorem 9. Note that then, (a) or (b) of Theorem 25 holds. Let $n \in \mathbb{N}$ be such that the formula "κ is an x-large cardinal" is Σ_n. By the assumption, there is an inaccessible cardinal μ with $\lambda > \mu > \kappa$ such that $V_\lambda \succ V_\mu$. Let $\lambda > \lambda_0 > \mu$ be the minimal cardinal such that $V_\lambda \models \lambda_0$ is ψ_n^*-absolute inaccessible cardinal—such λ_0 exists by Lemma 24. Then we have

$$V_{\lambda_0} \models \text{"}\kappa \text{ is an x-large cardinal"} \tag{*28}$$

In V_{λ_0}, let $\mathbb{P}$ be the limit of κ-iteration with appropriate support as described in Theorem 9 which forces that κ is tightly$^+$ $\mathcal{P}$-Laver generically x-large cardinal in the generic extension of V_{λ_0}. Let $\mathbb{G}$ be $(V_\lambda, \mathbb{P})$-generic filter. Then we have $V_{\lambda_0}[\mathbb{G}] \models \kappa$ is a tightly$^+$ $\mathcal{P}$-Laver generically x-large cardinal and $V_\lambda[\mathbb{G}] \succ V_\mu[\mathbb{G}]$ by Lemma 15. In particular, by Lemma 23, we have $V_{\lambda_0}[\mathbb{G}] \models \mu$ is the largest ψ_n^*-absolute inaccessible cardinal. By Theorem 25, it follows that $V_{\lambda_0}[\mathbb{G}] \models \neg(\mathcal{P}, \emptyset)-$RcA. □ (Theorem 26)

The conditions of Theorem 26 are satisfied by practically all large cardinal notions. For example, under the consistency of the existence of a 2-huge cardinal, the conditions of Theorem 26 are satisfied by x-large cardinal = hyperhuge cardinal (see Lemma 29 below). Thus we obtain the following:

Corollary 27 *Under the assumption of the consistency of the existence of a 2-huge cardinal, the existence of a tightly$^+$ $\mathcal{P}$-Laver generically hyperhuge cardinal does not imply $(\mathcal{P}, \emptyset)$-RcA for any class $\mathcal{P}$ of posets as in Theorem 9.* □

A natural strengthening of Laver-genericity does imply the full Maximality Principle (hence also the full Recurrence Axiom). As the proof of Theorem 25

suggests, such property must be formulated not in a single formula but as an axiom scheme.

For a natural number n, we call a cardinal κ *super $C^{(n)}$-hyperhuge* if for any $\lambda_0 > \kappa$ there are $\lambda \geq \lambda_0$ with $V_\lambda \prec_{\Sigma_n} \mathsf{V}$, and $j, M \subseteq \mathsf{V}$ such that $j : \mathsf{V} \overset{\prec}{\to}_\kappa M$, $j(\kappa) > \lambda$, ${}^{j(\lambda)}M \subseteq M$ and $V_{j(\lambda)} \prec_{\Sigma_n} \mathsf{V}$.

κ is *super $C^{(n)}$-ultrahuge* if the condition above holds with "${}^{j(\lambda)}M \subseteq M$" replaced by "${}^{j(\kappa)}M \subseteq M$ and $V_{j(\lambda)} \subseteq M$".

If κ is super $C^{(n)}$-hyperhuge then it is super $C^{(n)}$-ultrahuge. This can be shown similarly to Lemma 4.

We shall also say that κ is *super $C^{(\infty)}$-hyperhuge* (*super $C^{(\infty)}$-ultrahuge*, resp.) if it is super $C^{(n)}$-hyperhuge (super $C^{(n)}$-ultrahuge, resp.) for all natural number n.

A similar kind of strengthening of the notions of large cardinals which we call here "super $C^{(n)}$" appears also in Boney [6]. It is called in [6] "$C^{(n)+}$" and the notion is considered there in connection with extendibility.

For a natural number n and an iterable class $\mathcal{P}$ of posets, a cardinal κ is *super $C^{(n)}$-$\mathcal{P}$-Laver-generically ultrahuge* if, for any $\lambda_0 > \kappa$ and for any $\mathbb{P} \in \mathcal{P}$, there are a $\lambda \geq \lambda_0$ with $V_\lambda \prec_{\Sigma_n} \mathsf{V}$, a $\mathcal{P}$-name $\underset{\sim}{\mathbb{Q}}$ with $\Vdash_{\mathbb{P}}$ "$\underset{\sim}{\mathbb{Q}} \in \mathcal{P}$", such that, for $(\mathsf{V}, \mathbb{P} * \underset{\sim}{\mathbb{Q}})$-generic $\mathbb{H}$, there are $j, M \subseteq \mathsf{V}[\mathbb{H}]$ with $j : \mathsf{V} \overset{\prec}{\to}_\kappa M$, $j(\kappa) > \lambda$, $\mathbb{P}, \mathbb{H}$, ${V_{j(\lambda)}}^{\mathsf{V}[\mathbb{H}]} \in M$ and ${V_{j(\lambda)}}^{\mathsf{V}[\mathbb{H}]} \prec_{\Sigma_n} \mathsf{V}[\mathbb{H}]$.

A super $C^{(n)}$- $\mathcal{P}$-Laver-generically ultrahuge cardinal κ is *tightly super $C^{(n)}$-$\mathcal{P}$-Laver-generically ultrahuge*, if additionally $|\mathbb{P} * \underset{\sim}{\mathbb{Q}}| \leq j(\kappa)$ (see footnote (2)) holds in the definition above.

Super $C^{(\infty)}$-$\mathcal{P}$-Laver-generically hyperhugeness and *tightly super $C^{(\infty)}$-$\mathcal{P}$-Laver generically hyperhugeness* are defined similarly to super $C^{(\infty)}$-ultrahugeness.

Note that, in general, super $C^{(\infty)}$-hyperhugeness and super $C^{(\infty)}$-ultrahugeness are notions not formalizable in the language of ZFC without introducing a new constant symbol for κ since we need infinitely many $\mathcal{L}_\in$-formulas to formulate them. Exceptions are when we are talking about a cardinal in a set model being with one of these properties like in Lemma 29 below (and in such a case "natural number n" actually refers to "$n \in \omega$"), or when we are talking about a cardinal definable in V having these properties in an inner model like in Corollary 36 or 37. In the latter case, the situation is formalizable with infinitely may $\mathcal{L}_\in$-sentences.

In contrast, the super $C^{(\infty)}$-$\mathcal{P}$-Laver generically ultrahugeness of κ is expressible in infinitely many $\mathcal{L}_\in$-sentences. This is because a $\mathcal{P}$-Laver generic large cardinal κ for relevant classes $\mathcal{P}$ of posets is uniquely determined as κ_{ref} or $2^{\aleph_0}$ (see e.g. Theorems 7 and 8).

A modification of the proof of Theorem 21 shows the following:

Theorem 28 (Fuchino and Usuba [17]) *Suppose that $\mathcal{P}$ is an iterable class of posets and κ is super $C^{(\infty)}$-$\mathcal{P}$-Laver-generically ultrahuge. Then $(\mathcal{P}, \mathcal{H}(\kappa))$-$\mathsf{RcA}^+$ tightly holds.* □

The following Lemma can be proved similarly to Theorem 5c in Barbanel-DiPrisco-Tan [2] (see also Theorem 24.13 in Kanamori [30]).

Lemma 29 (Fuchino and Usuba [17]) *If κ is 2-huge with the 2-huge elementary embedding j, that is, there is $M \subseteq \mathsf{V}$ such that $j : \mathsf{V} \prec_{\kappa} M \subseteq \mathsf{V}$, and*

$$j^{2}(\kappa) M \subseteq M, \tag{29}$$

then $V_{j(\kappa)} \models$ "κ is super $C^{(\infty)}$-hyperhuge cardinal", and for each $n \in \omega$, $V_{j(\kappa)} \models$ "there are stationarily many super $C^{(n)}$-hyperhuge cardinals". □

The proof of the existence of Laver-function for a supercompact cardinal can be modified to show that super $C^{(\infty)}$-hyperhuge cardinal in V_μ has a Laver function for super $C^{(\infty)}$-hyperhugeness [17]. Similarly to Theorem 9 we obtain the following:

Theorem 30 (Fuchino and Usuba [17])

(*A*) *Suppose that μ is inaccessible and $\kappa < \mu$ is super $C^{(\infty)}$-hyperhuge in V_μ. Let $\mathbb{P} = \mathrm{Col}(\aleph_1, \kappa)$. Then, in $V_\mu[\mathbb{G}]$, for any V_μ, $\mathbb{P}$-generic $\mathbb{G}$, $\aleph_2^{V_\mu[\mathbb{G}]}$ $(= \kappa)$ is tightly super $C^{(\infty)}$-σ-closed-Laver-generically hyperhuge and* CH *holds.*

(*B*) *Suppose that μ is inaccessible and $\kappa < \mu$ is super $C^{(\infty)}$-hyperhuge with a Laver function $f : \kappa \to V_\kappa$ for super $C^{(\infty)}$-hyperhugeness in V_μ. If $\mathbb{P}$ is the CS-iteration of length κ for forcing* PFA *along with f, then, in $V_\mu[\mathbb{G}]$ for any $(V_\mu, \mathbb{P})$-generic $\mathbb{G}$, $\aleph_2^{V_\mu[\mathbb{G}]}$ $(= \kappa)$ is tightly super $C^{(\infty)}$-proper-Laver-generically hyperhuge and $2^{\aleph_0} = \aleph_2$ holds.*

(*B′*) *Suppose that μ is inaccessible and $\kappa < \mu$ is super $C^{(\infty)}$-hyperhuge with a Laver function $f : \kappa \to V_\kappa$ for super $C^{(\infty)}$-hyperhugeness in V_μ. If $\mathbb{P}$ is the RCS-iteration of length κ for forcing* MM *along with f, then, in $V_\mu[\mathbb{G}]$ for any $(V_\mu, \mathbb{P})$-generic $\mathbb{G}$, $\aleph_2^{V_\mu[\mathbb{G}]}$ $(= \kappa)$ is tightly super $C^{(\infty)}$-stationary_preserving-Laver-generically hyperhuge and $2^{\aleph_0} = \aleph_2$ holds.*

(*Γ*) *Suppose that μ is inaccessible and κ is super $C^{(\infty)}$-hyperhuge with a Laver function $f : \kappa \to V_\kappa$ for super $C^{(\infty)}$-hyperhugeness in V_μ. If $\mathbb{P}$ is a FS-iteration of length κ for forcing* MA *along with f, then, in $V_\mu[\mathbb{G}]$ for any $(V_\mu, \mathbb{P})$-generic $\mathbb{G}$, $2^{\aleph_0}$ $(= \kappa)$ is tightly super $C^{(\infty)}$-c.c.c.-Laver-generically hyperhuge, and $2^{\aleph_0}$ is very large in $V_\mu[\mathbb{G}]$.*

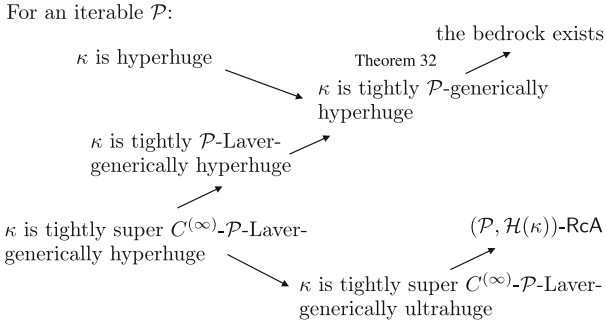
(*Δ*) *Suppose that μ is inaccessible and κ is super $C^{(\infty)}$-hyperhuge with a Laver function $f : \kappa \to V_\kappa$ for super $C^{(\infty)}$-hyperhugeness in V_μ. If $\mathbb{P}$ is a FS-iteration of length κ along with f enumerating "all" posets, then, in $V_\mu[\mathbb{G}]$ for any $(V_\mu, \mathbb{P})$-generic $\mathbb{G}$, $2^{\aleph_0}$ $(= \aleph_1)$ is tightly super $C^{(\infty)}$-all posets-Laver-generically hyperhuge, and* CH *holds.* □

6 Toward the Laver-Generic Maximum

Besides Theorems 21 and 28, we also have some other advantages of assuming the existence of $\mathcal{P}$-Laver-generically hyperhuge cardinal or even its "super $C^{(\infty)}$" version. One of them is that they imply the existence of the (set-generic) bedrock (see below for definition); another is that we know the exact consistency strength of these principles.

For a class $\mathcal{P}$ of posets, a cardinal κ is *tightly $\mathcal{P}$- generic hyperhuge* if for any $\lambda > \kappa$, there is $\mathbb{Q} \in \mathcal{P}$ such that for a $(\mathsf{V}, \mathbb{Q})$-generic $\mathbb{H}$, there are $j, M \subseteq \mathsf{V}[\mathbb{H}]$ such that $j : \mathsf{V} \overset{\prec}{\to}_\kappa M$, $\lambda < j(\kappa)$, $|\mathbb{Q}| \leq j(\kappa)$, and $j''j(\lambda), \mathbb{H} \in M$.

Note that, for any class $\mathcal{P}$ of posets with $\{\mathbb{1}\} \in \mathcal{P}$, the hyperhugeness of a cardinal κ implies its tightly $\mathcal{P}$-generically hyperhugeness. Likewise, if $\mathcal{P}$ is iterable then, the tightly $\mathcal{P}$-Laver-generic hyperhugeness of κ implies its tightly $\mathcal{P}$-generically hyperhugeness.



Usuba [38] proved that the grounds of V are downward directed (with respect to subclass relation) for class many grounds (this is formalizable by virtue of Theorem 12). More concretely

Theorem 31 (Theorem 1.3 in Usuba [38]) *For any collection of grounds of V, indexed by a set of parameters (in the sense of Theorem 12), there is a ground which is included in all grounds in the collection.* □

From this theorem, it follows that the *mantle*, i.e., the intersection of all grounds is a model of ZFC. In [38], it is proved that the mantle is a ground and hence it is the *bedrock*, i.e., the smallest ground of V provided that there exists a hyperhuge cardinal(Theorem 1.6 in [38]). Later the assumption of the existence of a hyperhuge cardinal in this theorem is weakened to the existence of an extendible cardinal (Theorem 1.3 in Usuba [39]).

In [17], we obtained the following generalization of Theorem 1.6 in [38]. In the following theorems, tightness of a $\mathcal{P}$-generic Large cardinal is defined similarly to the tightness of $\mathcal{P}$-Laver generic large cardinal. Note that the (in many cases unique (Theorem 8)) tightly $\mathcal{P}$-Laver generic large cardinal as well as corresponding genuine large cardinals are tightly $\mathcal{P}$-generic large cardinals by definition.

Theorem 32 (Fuchino and Usuba [17]) *If there is a tightly $\mathcal{P}$-generically hyperhuge cardinal κ, then the mantle is a ground of* V. *In particular it is the bedrock.*

A Sketch of the Proof The overall structure of the structure of the proof is just the same as that of Theorem 1.6 in [38] or Theorem 1.3 in [39].

We call a ground W of V a $\leq\kappa$-ground if there is $\mathbb{P} \in W$ with $|\mathbb{P}|^{\mathsf{V}} \leq \kappa$ and a $(\mathsf{W}, \mathbb{P})$-generic $\mathbb{G}$ such that $\mathsf{V} = \mathsf{W}[\mathbb{G}]$. Let

$$\overline{W} = \bigcap\{W \ : \ W \text{ is a } \leq\kappa\text{-ground}\}. \tag{*30}$$

By Theorem 31, there is a ground $\mathsf{W} \subseteq \overline{\mathsf{W}}$. For such W it is enough to show that actually $\overline{\mathsf{W}} \subseteq \mathsf{W}$ holds.

Let $\mathbb{S} \in \mathsf{W}$ be a poset with cardinality μ (in V) such that there is a $(\mathsf{W}, \mathbb{S})$-generic $\mathbb{F} \in \mathsf{V}$ with $\mathsf{V} = \mathsf{W}[\mathbb{F}]$. Without loss of generality, $\mu \geq \kappa$.

By Theorem 12, there is $r \in \mathsf{V}$ such that $\mathsf{W} = \Phi(\cdot, r)^{\mathsf{V}}$.

Let $\theta \geq \mu$ be such that $r \in V_\theta$, and for a sufficiently large natural number n, we have ${V_\theta}^{\mathsf{V}} \prec_{\Sigma_n} \mathsf{V}$. By the choice of θ, $\Phi(\cdot, r)^{{V_\theta}^{\mathsf{V}}} = \Phi(\cdot, r)^{\mathsf{V}} \cap {V_\theta}^{\mathsf{V}} = \mathsf{W} \cap {V_\theta}^{\mathsf{V}} = {V_\theta}^{\mathsf{W}}$. Let $\mathbb{Q} \in \mathcal{P}$ such that for $(\mathsf{V}, \mathbb{Q})$-generic $\mathbb{H}$, there are $j, M \subseteq \mathsf{V}[\mathbb{H}]$ with $j : \mathsf{V} \overset{\prec}{\to}_\kappa M, \theta < j(\kappa), |\mathbb{Q}| \leq j(\kappa), {V_{j(\theta)}}^{\mathsf{V}[\mathbb{H}]} \subseteq M$, and $\mathbb{H}, j''j(\theta) \in M$.

Using this j we can show that ${V_\theta}^{\overline{\mathsf{W}}} \subseteq {V_\theta}^{\mathsf{W}}$ holds (this part of the proof is quite involved, for the details, the reader is referred to [17]). Since θ can be arbitrary large, It follows that $\overline{\mathsf{W}} \subseteq \mathsf{W}$. □ (Theorem 32)

Analyzing the details of the proof of Theorem 32 which we omitted from our present exposition, we also obtain the following result with many surprising consequences:

Theorem 33 (Fuchino and Usuba [17]) *Suppose that $\mathcal{P}$ is any class of posets. If κ is a tightly $\mathcal{P}$-generically hyperhuge cardinal, then κ is a hyperhuge cardinal in the bedrock $\overline{W}$ of* V. □

The following equiconsistency results are immediate consequences of the theorem above:

Corollary 34 *Suppose that $\mathcal{P}$ is the class of all posets. Then the following theories are equiconsistent:*

(a) ZFC + *"there is a hyperhuge cardinal".*
(b) ZFC + *"there is a tightly $\mathcal{P}$-Laver generically hyperhuge cardinal".*
(c) ZFC + *"there is a tightly $\mathcal{P}$-generically hyperhuge cardinal".*
(d) ZFC + *"the bedrock $\overline{\mathsf{W}}$ exists and ω_1 is a hyperhuge cardinal in $\overline{\mathsf{W}}$".* □

Corollary 35 *Suppose that $\mathcal{P}$ is one of the following classes of posets: all semi-proper posets; all proper posets; all ccc posets; all σ-closed posets. Then the following theories are equiconsistent:*

(a) ZFC + *"there is a hyperhuge cardinal".*
(b) ZFC + *"there is a tightly $\mathcal{P}$-Laver generically hyperhuge cardinal".*

(c) ZFC + *"there is a tightly $\mathcal{P}$-generically hyperhuge cardinal"*.
(d) ZFC + *"the bedrock $\overline{\mathsf{W}}$ exists and κ_{refl} is a hyperhuge cardinal in $\overline{\mathsf{W}}$"*. □

These equiconsistency results are quite remarkable when we remember that equiconsistency of axioms like PFA, MM, MM^{++} etc. are unknown at the moment.

The tightness of generic large cardinals was originally thought as a technical condition when it was introduced to overcome the difficulty in the proof of Theorem 7, (2). The (proofs of) Theorems 32 and 33 and their consequences, some of which we are presenting here as their corollaries, suggest that this notion is much more intrinsic than merely a technicality.

A slight modification of the proofs of the theorems above also show the following. Note that as we already noticed, super-$C^{(\infty)}$-large cardinal is not formalizable in the language of ZFC. However, the assertions (a) and (b) in the following Corollaries 36 and 36 can be formulated as schemes of sentences in $\mathcal{L}_\in$.

Corollary 36 *Suppose that $\mathcal{P}$ is the class of all posets. Then the following theories are equiconsistent:*

(a) ZFC + *"c is a super $C^{(\infty)}$ hyperhuge cardinal" where c is a new constant symbol but "... is super $C^{(\infty)}$ hyperhuge ..." is formulated in an infinite collection of formulas in $\mathcal{L}_\in$.*
(b) ZFC + *"there is a tightly super $C^{(\infty)}$-$\mathcal{P}$-Laver generically hyperhuge cardinal"*.
(c) ZFC + *"the bedrock $\overline{\mathsf{W}}$ exists and ω_1^{V} is a super $C^{(\infty)}$-hyperhuge cardinal in $\overline{\mathsf{W}}$"*. □

Corollary 37 *Suppose that $\mathcal{P}$ is one of the following classes of posets: all semi-proper posets; all proper posets; all ccc posets; all σ-closed posets. Then the following theories are equiconsistent:*

(a) ZFC + *"c is a super $C^{(\infty)}$ hyperhuge cardinal" where c is a new constant symbol but "... is super $C^{(\infty)}$ hyperhuge ..." is formulated in an infinite collection of formulas in $\mathcal{L}_\in$.*
(b) ZFC + *"there is a tightly super $C^{(\infty)}$-$\mathcal{P}$-Laver generically hyperhuge cardinal"*.
(c) ZFC + *"the bedrock $\overline{\mathsf{W}}$ exists and $\kappa_{\mathrm{refl}}{}^{\mathsf{V}}$ is a super $C^{(\infty)}$-hyperhuge cardinal in $\overline{\mathsf{W}}$"*. □

Finally, we move to the promised proof of Theorem 7, (3).

Corollary 38 *Suppose that $\mathcal{P}$ is an arbitrary class of posets and κ is a tightly $\mathcal{P}$-generically hyperhuge cardinal. Then*

(1) *there are cofinally many huge cardinals.*
(2) SCH *holds above some cardinal.*

Proof Suppose that κ is a tightly $\mathcal{P}$-generically hyperhuge cardinal. By Theorem 32, there is the bedrock $\overline{\mathsf{W}}$ and κ is hyperhuge cardinal in $\overline{\mathsf{W}}$.

(1) Since the existence of a hyperhuge cardinal implies the existence of cofinally many huge cardinals (it is easy to show that the target $j(\kappa)$ of hyperhuge embedding for a sufficiently large inaccessible λ is a huge cardinal), there are cofinally many huge cardinals in $\overline{\mathsf{W}}$. Since V is attained by a set forcing starting from $\overline{\mathsf{W}}$, a final segment of these huge cardinals survive in V.
(2) By Theorem 20.8 in [29], SCH holds above κ in $\overline{\mathsf{W}}$. Since V is a set generic extension of $\overline{\mathsf{W}}$. SCH should hold above some cardinal $\mu \geq \kappa$. □ (Corollary 38)

For iterable stationary preserving $\mathcal{P}$ containing all proper posets, Corollary 38, (2) holds already under the $\mathcal{P}$-Laver-generic supercompactness of κ. The reason is that in such case PFA holds by Theorem 5, and by Viale [40], SCH follows from it.

Proof of Theorem 7, (3) Let λ and $\mathbb{Q}$ be such that

(*31) $\lambda > 2^{\aleph_0}$, κ and λ is large enough such that SCH holds above some $\mu < \lambda$ (this is possible by Corollary 38 , (2), and it is here that we need a strong property like the Laver generic hyperhugeness of κ),
(*32) $\mathbb{Q}$ is positive elements of a complete Boolean algebra, and,
(*33) for $(\mathsf{V}, \mathbb{Q})$-generic $\mathbb{H}$, there are j, $M \subseteq \mathsf{V}[\mathbb{H}]$ such that (1) $j : \mathsf{V} \prec_\kappa M$, (2) $j(\kappa) > \lambda$, (3) $|\mathbb{Q}| \leq j(\kappa)$, and (4) ${V_{j(\lambda)}}^{\mathsf{V}[\mathbb{H}]} \subseteq M$.

By (*32), each $\mathbb{Q}$-name $\dot{x}$ of a real corresponds to a mapping $f : \omega \to \mathbb{Q}$. By (*31) and by (*33), (3), there are at most $j(\kappa)$ many such mappings. Thus we have $\mathsf{V}[\mathbb{H}] \models$ "$2^{\aleph_0} \leq j(\kappa)$", By (*33), (4), it follows $M \models$ "$2^{\aleph_0} \leq j(\kappa)$". By elementarity, it follows that $\mathsf{V} \models$ "$2^{\aleph_0} \leq \kappa$". □ (Theorem 7, (3))

Returning to (E) and (Z) at the end of Sect. 3, we now know that the existence of a super $C^{(\infty)}$-stationary preserving-Laver-generically hyperhuge cardinal implies (E) (actually it even implies (stationary preserving posets, $\mathcal{H}(\kappa_{\mathfrak{refl}})$)-$\mathsf{RcA}^+$), and that the existence of a super $C^{(\infty)}$-all posets-Laver-generically hyperhuge cardinals implies (Z) (actually it even implies (all posets, $\mathcal{H}(2^{\aleph_0})$)-$\mathsf{RcA}^+$). These two scenarios are not compatible since the former implies $2^{\aleph_0} = \aleph_2$ while the latter implies CH.

However, with the following axiom, (E) is reconciled with a fragment of (Z):

LGM: the continuum is tightly super $C^{(\infty)}$-stationary preserving-Laver generically hyperhuge and there is a ground W of V such that the continuum is tightly super $C^{(\infty)}$-all posets-Laver generically hyperhuge in W.

This combination of Laver-genericity and "ground"[12] Laver-genericity above implies (Z^+) mentioned at the end of Sect. 3 (by Theorem 28). As (Z^+) represents in a sense the maximal amount of available strengthening of recurrence, I would like to choose the name Laver-Generic Maximum (LGM) for it.

[12] I am using the word "ground" here as an adjective contrasting with the word "generic" in "generic large cardinal".

If we admit that Recurrence Axioms, Maximal Principles and Laver-genericity are natural requirements, we should be also ready to accept $\mathsf{ZFC} + \mathsf{LGM}$ as a natural candidate of the extension of ZFC.

By Theorem 5, LGM implies the double plus version of Martin's Maximum (MM^{++}) and hence all the consequences of it including $2^{\aleph_0} = \aleph_2$.

By Theorem 32, LGM implies that there is the bedrock. So by Theorem 28, LGM implies (Z^+) on p. 335. (Z^+) implies that if some statement φ is forceable by a stationary preserving poset, then for any $A \in \mathcal{H}(\aleph_2)$, there is a semi-proper-ground W of V such that $A \in \mathsf{W}$ and $\mathsf{W} \models \varphi$. In particular, Cichoń's Maximum [24, 25] is a phenomena in many semi-proper-grounds in this sense. Note that, by Corollary 38, (1) the forcing argument for φ may even utilize class many huge cardinals (e.g. the proof in [24] uses four strongly compact cardinals).[13]

Even in the case that the forcing to show the consistency of φ is not stationary preserving, we can still find some ground W of V which satisfies φ.

Thus $\mathsf{ZFC} + \mathsf{LGM}$ is a very strong axiom system which integrates practically all statements into itself as far as these statements can be proved to be consistent by way of forcing and/or methods of inner models or some combination of them. Against this backdrop, we want to call the axiom system LGM (or possibly some further extension of it in the future) the *Laver Generic Maximum*.

The consistency and equiconsistency of LGM is easily established: we start from a model with two super $C^{(\infty)}$ hyperhuge cardinals $\kappa_0 < \kappa_1$. We force κ_0 to be tightly super $C^{(\infty)}$-all posets-Laver generically hyperhuge (Theorem 30, (Δ)). We then force make κ_1 to be tightly super $C^{(\infty)}$-stationary preserving-Laver generically hyperhuge (Theorem 30, (B')).

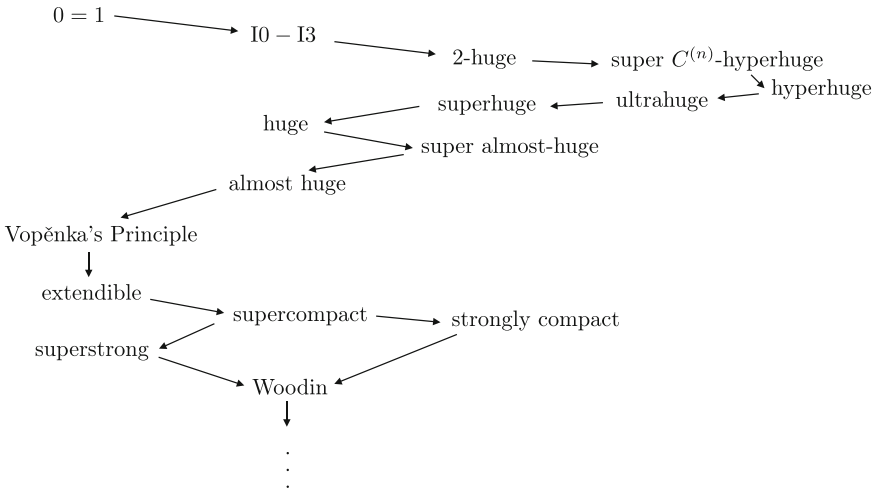
By Theorem 33 the consistency strength of LGM can be proved to be equivalent with that of two super $C^{(\infty)}$ hyperhuge cardinals (which may be formulated by using two new constant symbols).

7 More About Consistency Strength

When I was a student of Janos Makowsky in the early 1980s at the Free University of Berlin, one of the papers he was preparing then was [32] in which the effects of Vopěnka Principle on properties of model theoretic logics is studied. I remember that Janos gave a talk on this subject in the (West) Berliner Logic Colloquium. Back then, I was still living in a set theory of consistency strength way below a measurable cardinal, and could not begin with the material of his paper at all because of the vertiginous consistency strength of the Vopěnka Principle.

[13] In [17], we even show that LGM implies that there are stationarily many super-$C^{(\infty)}$-hyperhuge cardinals.

Janos left Berlin before I wrote up my diploma thesis on abstract elementary classes which was the subject Janos gave me; all the assertions I proved in the thesis remained in the consistency strength of ZFC.



The consistency strength of my set theoretic world view reached the realm of one supercompact cardinal when I wrote [10] in the early 1990s in which some consequences of $\mathsf{MA}^+(\sigma\text{-closed})$ were discussed. However, it is only quite recently that I caught up with Janos definitively (at least in terms of consistency strength) when I considered in [17] the super $C^{(\infty)}$-$\mathcal{P}$-Laver-generically hyperhuge cardinals whose consistency strength is (demonstrably—see Sect. 6) strictly between hyperhuge and 2-huge.

In the meantime, active research on abstract model classes is resumed and Janos's [32] begins to attract the attention of young logicians. For example, Janos's [32] was recently cited in Boney [6] which was already mentioned in Sect. 6. Boney cites the main theorem of [32] as Fact 3.12 in his paper and comments in allusion to Aki Kanamori's comment on Kunen's inconsistency proof in [30] that Vopěnka's Principle " 'rallies at least to force a veritable Götterdömmerung' for compactness cardinals for logics." The gap between Vopěnka's Principle and a huge cardinal Boney mentions in connection with this „götterdämmerigen" statement seems to have some resemblance to the discrepancy between usual Laver-generic large cardinal axioms and the super $C^{(\infty)}$-Laver-generic large cardinals.

Now one of the urgent items in my to-do-list is to check Janos's [32] as well as [33–35] more carefully to find out further possible connections of his results to the context I described above.

Is this also an instance of (the eternal?) recurrence?

Acknowledgments A part of the material of the chapter was presented at the RIMS set theory workshop 2023 in Kyoto. The author would like to thank Gunter Fuchs and Joan Bagaria for valuable comments at the talk. The author also would like to thank Andrés Villaveces and the

anonymous referee for their critical remarks on the first draft of the chapter. The draft was largely rewritten according to their suggestions. The author hopes very much that the present revised version of the chapter is now readable for a broader audience.

One of the suggestions of the referee was that the terminology of Laver-genericity should be simplified. We decided however that, at least for the present article, the cumbersome and surely annoying expressions like "super $C^{(\infty)}$ tightly $\mathcal{P}$-Laver generically hyperhuge" will be kept as they are. The main reason for the decision is that this clumsy way of saying does express subtle differences which might be crucial for the further development of the theory of Laver-genericity.

The theory of Laver-genericity is indeed still under transition. When the tightness of Laver genericity was introduced in [20], the author thought that it is only a technical condition. Later, in [17], we realized that the tightness is *the* central notion in connection with the existence of bedrock (see Theorem 32). Thus it is proved to be a right decision that we did not change the definition of Laver genericity to make the condition of tightness a part of its definition.

An attempt is currently being made to integrate the theory of Laver-genericity into the larger context of the set theory of Hugh Woodin (or possibly other way round). The author hopes that, in a future expository article, when this project has been accomplished, the whole theory can be represented in a nice and simplified choice of terminology.

Competing Interests The author's research was supported by Kakenhi Grant-in-Aid for Scientific Research (C) 20K03717.

References

1. Bagaria, J.: Natural axioms of set theory and the continuum problem. In: Proceedings of the 12-th International Congress of Logic, Methodology, and Philosophy of Science. King's College London, pp. 43–64 (2005)
2. Barbanel, J.B., DiPrisco, C.A., Tan, I.B.: Many-times huge and superhuge cardinals. J. Symb. Logic **49**(1), 112–122 (1984)
3. Barton, N., Caicedo, A.E., Fuchs, G., Hamkins, J.D., Reitz, J., Schindler, R.: Inner-model reflection principles. Stud. Log. **108**, 573–595 (2020)
4. Baumgartner, J.E., Taylor, A.D., Wagon, S.: On splitting stationary subsets of large cardinals. J. Symb. Logic **42**(2), 203–214 (1977)
5. Ben-David, S.: On Shelah's compactness of cardinals. Israel J. Math. **31**(1), 34–56 (1978)
6. Boney, W.: Model theoretic characterizations of large cardinals. Israel J. Math. **236**, 133–181 (2020)
7. Cox, S.: The diagonal reflection principle. Proc. Am. Math. Soc. **140**(8), 2893–2902 (2012)
8. Dow, A.: An introduction to applications of elementary submodels to topology. Topol. Proc. **13**(1), 17–72 (1988)
9. Friedman, S.-D.: Internal consistency and the inner model hypothesis. Bull. Symb. Logic **12**(4), 591–600 (2006)
10. Fuchino, S.: Some remarks on openly generated Boolean algebras. J. Symb. Logic **59**(1), 302–310 (1994)
11. Fuchino, S.: Digital images of a presentation of a seminar talk given at Nagoya Set Theory Seminar on May 31 (2019). https://fuchino.ddo.jp/talks/talk-nagoya-2019-05-31.pdf
12. Fuchino, S.: Resurrection and maximality in light of Laver-generic large cardinal (in preparation). Pre-preprint: https://fuchino.ddo.jp/papers/RIMS2022-RA-MP-x.pdf
13. Fuchino, S., Rodrigues, A.O.M.: Reflection principles, generic large cardinals, and the continuum problem. In: The Proceedings of the Symposium on Advances in Mathematical Logic 2018, pp. 1–26. Springer, Singapore (2022)

14. Fuchino, S., Sakai, H.: Generically supercompact cardinals by forcing with chain conditions. RIMS Kôkyûroku **2213**. In: Kada, M. (ed.) Recent Developments in Set Theory of the Reals, pp. 94–111 (2022)
15. Fuchino, S., Sakai, H.: The first-order definability of generic large cardinals (submitted). Extended version of the paper: https://fuchino.ddo.jp/papers/definability-of-glc-x.pdf
16. Fuchino, S., Usuba, T.: A reflection principle formulated in terms of games. RIMS Kôkyûroku **1895**, 37–47 (2014)
17. Fuchino, S., Usuba, T.: On recurrence axioms. Preprint. https://arxiv.org/abs/2402.02693. Extended version of the preprint: https://fuchino.ddo.jp/papers/recurrence-axioms-x.pdf
18. Fuchino, S., Juhász, I., Soukup, L., Szentmiklóssy, Z., Usuba, T.: Fodor-type Reflection Principle and reflection of metrizability and meta-Lindelöfness. Topol. Appl. **157**(8), 1415–1429 (2010)
19. Fuchino, S., Rodrigues, A.O.M., Sakai, H.: Strong downward Löwenheim-Skolem theorems for stationary logics, I. Arch. Math. Logic **60**(1–2), 17–47 (2021). Extended postprint: https://fuchino.ddo.jp/papers/SDLS-x.pdf
20. Fuchino, S., Rodrigues, A.O.M., Sakai, H.: Strong downward Löwenheim-Skolem theorems for stationary logics, II: reflection down to the continuum. Arch. Math. Logic **60**(1–2), 17–47 (2021). Extended postprint: https://fuchino.ddo.jp/papers/SDLS-II-x.pdf
21. Fuchino, S., Soukup, L., Sakai, H., Usuba, T.: More about Fodor-type Reflection Principle (submitted). Extended version: https://fuchino.ddo.jp/papers/moreFRP-x.pdf
22. Fuchs, G., Hamkins, J.D., Reitz, J.: Set-theoretic geology. Ann. Pure Appl. Logic **166**(4), 464–501 (2015)
23. Gödel, K.: What is Cantor's continuum problem?. Am. Math. Mon. **54**(9), 515–525 (1947). Errata: ibid., **55**, 151 (1948). Revised and expanded version In: Benacerraf, P., Putnam, H. (eds.) Philosophy of Mathematics. Selected Readings, pp. 258–273. Englewood Cliffs, Prentice Hall (1964)
24. Goldstern, M., Kellner, J., Shelah, S.: Cichoń's maximum. Ann. Math. **190**, 113–143 (2019)
25. Goldstern, M., Kellner, J., Mejía, D.A., Shelah, S.: Cichoń's maximum without large cardinals. Preprint. J. Eur. Math. Soc. **24**, 3951–3967 (2022)
26. Hamkins, J.: A simple maximality principle. J. Symb. Logic **68**(7), 527–550 (2003)
27. Hamkins, J.D., Johnstone, T.A.: Resurrection axioms and uplifting cardinals. Arch. Math. Logic **53**(3–4), 463–485 (2014)
28. Hamkins, J.D., Johnstone, T.A.: Strongly uplifting cardinals and the boldface resurrection axioms. Arch. Math. Logic **56**, 1115–1133 (2017)
29. Jech, T.: Set Theory. The Third Millennium Edition. Springer, Berlin (2001/2006)
30. Kanamori, A.: The Higher Infinite. Springer, Berlin (1994/2003)
31. König, B.: Generic compactness reformulated. Arch. Math. Logic **43**, 311–326 (2004)
32. Makowsky, J.A.: Vopěnka's principle and compact logics. J. Symb. Logic **50**(1), 42–48 (1985)
33. Makowsky, J.A.: Compactness, embeddings and definability, Chapter XVIII. In: Barwise, J., Feferman, S. (eds.) Model-Theoretic Logics, pp. 645–716. Springer, New York (1985)
34. Makowsky, J.A.: Abstract embedding relations, Chapter XX. In: Barwise, J., Feferman, S. (eds.) Model-Theoretic Logics, pp. 747–792. Springer, New York (1985)
35. Makowsky, J.A., Mundici, D.: Abstract equivalence relations, Chapter XIX. In: Barwise, J., Feferman, S. (eds.) Model-Theoretic Logics, pp. 717–746. Springer, New York (1985)
36. Reitz, J.: The ground axiom. J. Symb. Logic **72**(4), 1299–1317 (2007)
37. Tsaprounis, K.: On resurrection axioms. J. Symb. Logic **80**(2), 587–608 (2015)
38. Usuba, T.: The downward directed grounds hypothesis and very large cardinals. J. Math. Logic **17**(2), 1750009 (24 pp.) (2017)
39. Usuba, T.: Extendible cardinals and the mantle. Arch. Math. Logic **58**, 71–75 (2019)
40. Viale, M.: The proper forcing axiom and the singular cardinal hypothesis. J. Symb. Logic **71**(2), 473–479 (2006)

Provenance Analysis and Semiring Semantics for First-Order Logic

Erich Grädel and Val Tannen

Abstract A provenance analysis for a query evaluation or a model checking computation extracts information on how its result depends on the atomic facts of the model or database. Traditional work on data provenance was, to a large extent, restricted to positive query languages or the negation-free fragment of first-order logic and showed how provenance abstractions can be usefully described as elements of commutative semirings—most generally as multivariate polynomials with positive integer coefficients. We describe and evaluate here a provenance approach for dealing with negation, based on quotient semirings of polynomials with dual indeterminates. This not only provides a semiring provenance analysis for full first-order logic (and other logics and query languages with negation) but also permits a reverse provenance analysis, i.e., finding models that satisfy various properties under given provenance tracking assumptions. We describe the potential for applications to explaining missing query answers or failures of integrity constraints, and to using these explanations for computing repairs. This approach also is the basis of a systematic study of semiring semantics in a broad logical context.

1 Introduction

Semiring provenance was originally developed for positive database query languages by Green, Karvounarakis, and Tannen [32]. It is based on the idea to annotate the atomic facts in a database by values in some commutative semiring, and to propagate these annotations through a query, keeping track whether information is used alternatively (as in disjunctions or existential quantifications) or jointly (as

E. Grädel (✉)
RWTH Aachen University, Aachen, Germany
e-mail: graedel@logic.rwth-aachen.de; graedel@informatik.rwth-aachen.de

V. Tannen
University of Pennsylvania, Philadelphia, PA, USA
e-mail: val@cis.upenn.edu

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_21

in conjunctions or universal quantifications). From this baseline, we have started in [25] to investigate a new approach to the provenance analysis of model checking for languages with negation, and in particular full first-order logic (FO). This approach is based on transformations to negation normal form, quotient semirings of polynomials with dual indeterminates, and a close relationship to semiring valuations of games [26]. Since then, semiring provenance has been extended to a systematic investigation of *semiring semantics* for many logical systems, including first-order logic, modal logic, description logics, guarded logic and fixed-point logic [9, 13–15] and also to a general method for strategy analysis in games [26, 28]. The present paper is a thoroughly revised and considerably expanded version of our paper [25] from 2017 which has previously only appeared as a preprint in arXiv—although it has been the basis of much of the subsequent work on provenance analysis and semiring semantics. We shall also give a brief overview on the work that has been done since 2017, and discuss some questions for future research.

Data provenance is extremely useful in many computational disciplines. Suppose that a computational process is applied to a complex input consisting of multiple items. Provenance analysis allows us to understand how these different input items affect the output of the process. It can be used to answer questions of the following type:

- Which ones of input items are actually used in the computation of the output?
- Can the same output be obtained from different combinations of input items?
- In how many different ways can the same output be computed?

As a consequence, provenance can be further applied to issues such as deciding how much to trust the output, assuming that we may trust some input items more than others, deciding what clearance level is required for accessing the output, assuming that we know access restrictions for the input items, or, assuming that one has to pay for the input items, how to minimize the cost of obtaining the output. More generally, *reverse* provenance analysis allows us to find input data (here first-order structures) that satisfy various properties under given provenance tracking assumptions.

It turns out that the questions listed above, as well as several other questions of interest, can be answered for database transformations (queries and views) via interpretations in commutative semirings. In past work, the semiring provenance approach has been applied to query and view languages such as the positive relational algebra [30, 32], nested relations/complex values (objects) [19, 41], Datalog [16, 32], XQuery (for unordered XML) [19], relational algebra on $\mathbb{Z}$-annotated relations [33], SQL aggregates [4], workflows with map-reduce modules [2], and languages for data-centric (data-dependent) processes [17]. Moreover, the semiring approach has been successfully implemented in the software systems Orchestra [31, 36, 37] and Propolis [17]. For a survey, see also [22].

However, for a long time, semiring provenance has essentially been restricted to negation-free query languages. There have been algebraically interesting attempts to cover difference of relations [3, 20, 21, 33] but they had not resulted in systematic tracking of *absent* or *negative information*, and for quite some time, this has been an

obstacle for extending semiring provenance to other branches of logic in computer science. Indeed, while there are many applications in databases where one can get quite far with considering only positive information, logical applications in most other areas are based on formalisms that use negation in an essential way. A main objective of our new approach, proposed in [25], was the extension of semiring provenance to more general logical formalisms, and in particular to full first-order logic.

Provenance Semantics We consider a non-standard semantics for first-order logic (FO) that will help us to understand how a sentence φ ends up being true in a finite structure $\mathfrak{A}$, i.e., whether $\mathfrak{A} \models \varphi$ holds or not (we call this *provenance in model checking*). The non-standard semantics that we champion involves various *commutative semirings*. Here we strive to justify this choice.

First of all, the standard semantics for first-order logic maps formulae to truth values in $\mathbb{B} = \{\bot, \top\}$, which form a commutative semiring with respect to the operations of disjunction and conjunction, with units $\bot$ and $\top$. Second, in a provenance semantics we want to understand the connections between the facts (positive or negative) that are embodied in a model $\mathfrak{A}$ and their use in a justification that $\mathfrak{A} \models \varphi$. We can think of such a justification as a disjunction-conjunction *proof tree* (an example appears in Sect. 4.2) or, equivalently, as a winning strategy in the model checking game for $\mathfrak{A}$ and φ. If we had a provenance semantics for model checking, it would, in particular, help us to count such proof trees or evaluation strategies. This particular case suffices to suggest the semiring structure as well as some ways in which such non-standard semantics can be quite different from the standard one.

Notice that a semiring semantics refines the classical Boolean semantics, and formulae that are classically equivalent may become non-equivalent under a semantics that counts proof trees. Indeed, already a sentence $\varphi \vee \varphi$ has in general more proof trees than φ. We further illustrate this with the failure of some of the usual logical equivalences invoked in transforming sentences to *prenex form*.

Let $\rho := (\forall x\, \varphi) \wedge \psi$ and $\sigma := \forall x\, (\varphi \wedge \psi)$. Every proof tree of ρ can be transformed into a proof tree of σ by making copies of the subtree rooted at ψ. However, when ψ has two or more distinct proof trees we see that σ can have strictly more proof trees than ρ. Similarly we can argue that $\forall x\, (\varphi \vee \psi)$ can have strictly more proof trees than $(\forall x\, \varphi) \vee \psi$. Now consider $\rho := (\exists x\, \varphi) \vee \psi$ and $\sigma := \exists x\, (\varphi \vee \psi)$. Let us write $\varphi(x)$ to show occurrences of x in φ. For simplicity suppose that the model has exactly two elements, a and b, and that each of $\varphi(a)$, $\varphi(b)$, and ψ has exactly one proof tree. Then, ρ will have three proof trees but σ will have four. Finally, we note that $(\exists x\, \varphi) \wedge \psi$ and $\exists x\, (\varphi \wedge \psi)$ have exactly the same number of proof trees and this reflects the fact that multiplication distributes over addition.

For other sentences, we can see that the number-of-proof-trees constitutes a non-standard semantics for first-order sentences constructed using disjunction, conjunction, existentials and universals, because, moreover, addition and multiplication are associative and commutative.

This discussion provides some partial justification for considering commutative semirings as semantic domains. The rest of the justification will follow from the subsequent development.

2 Commutative Semirings

Definition 1 (Semiring) A commutative semiring is an algebraic structure $\mathcal{S} = (S, +, \cdot, 0, 1)$ with $0 \neq 1$, such that $(S, +, 0)$ and $(S, \cdot, 1)$ are commutative monoids, $\cdot$ distributes over $+$, and $0 \cdot s = s \cdot 0 = 0$.

A commutative semiring is *naturally ordered* (by addition) if $s \leq t :\Leftrightarrow \exists r(s + r = t)$ defines a partial order. Notice that $\leq$ is always reflexive and transitive, so a semiring is naturally ordered if, and only if, $\leq$ is antisymmetric, i.e. $r \leq s$ and $s \leq r$ only hold for $s = r$. In particular, this excludes rings. In this paper, we only consider commutative and naturally ordered semirings and simply refer to them as *semirings*. A semiring $\mathcal{S}$ is *idempotent* if $s + s = s$ for each $s \in S$ and *multiplicatively idempotent* if $s \cdot s = s$ for all $s \in S$. If both properties are satisfied, we say that $\mathcal{S}$ is fully idempotent. Finally, $\mathcal{S}$ is *absorptive* if $s + st = s$ for all $s, t \in S$ or, equivalently, if multiplication is decreasing in $\mathcal{S}$, i.e. $st \leq s$ for $s, t \in S$. Every absorptive semiring is idempotent, and every idempotent semiring is naturally ordered.

Application Semirings
There are many applications which can be modelled by semirings and provide useful practical information about the evaluation of a formula.

- The Boolean semiring $\mathbb{B} = (\mathbb{B}, \vee, \wedge, \bot, \top)$ is the standard habitat of logical truth.
- A totally ordered set $(S, \leq)$ with least element s and greatest element t induces the *min-max semiring* $(S, \max, \min, s, t)$. Specific important examples are the Boolean semiring, the fuzzy semiring $\mathbb{F} = ([0, 1], \max, \min, 0, 1)$, and the *access control semiring*, also called the *security semiring* [19], which is a min-max semiring with elements $0 < \mathsf{T} < \mathsf{S} < \mathsf{C} < \mathsf{P} = 1$ where 0 stands for "inaccessible" (or "false"), T is "top secret", S is "secret", C is "confidential", and P is "public". It is used for reasoning about access restrictions to atomic facts, and the clearance levels that are necessary to verify the truth of a sentence under such restrictions.
- A more general class (than min-max semirings) is the class of *lattice semirings* $(S, \sqcap, \sqcup, s, t)$ induced by a bounded distributive lattice $(S, \leq)$. Clearly, lattice semirings are fully idempotent.
- The *tropical semiring* $\mathbb{T} = (\mathbb{R}_+^\infty, \min, +, \infty, 0)$ is used to annotate atomic facts with a cost for accessing them and to compute minimal costs for verifying a logical statement. It is not fully idempotent but absorptive.

- The *Viterbi semiring* $\mathbb{V} = ([0,1]_{\mathbb{R}}, \max, \cdot, 0, 1)$, which is in fact isomorphic to $\mathbb{T}$ via $y \mapsto -\ln y$ can be used for reasoning about confidence.
- An alternative semiring for reasoning about confidence scores is the *Lukasiewicz semiring* $\mathbb{L} = ([0,1]_{\mathbb{R}}, \max, \odot, 0, 1)$, where multiplication is given by $s \odot t = \max(s + t - 1, 0)$. It is isomorphic to the semiring of doubt $\mathbb{D} = ([0,1]_{\mathbb{R}}, \min, \oplus, 1, 0)$ with $s \oplus t = \min(s + t, 1)$. Also $\mathbb{L}$ and $\mathbb{D}$ are absorptive semirings.
- The *natural semiring* $\mathbb{N} = (\mathbb{N}, +, \cdot, 0, 1)$ is used to count the number of proof trees or evaluation strategies that establish the truth of a sentence. It is also important for bag semantics in databases.

Provenance Semirings

Provenance semirings of polynomials provide information on which combinations of literals imply the truth of a formula. The universal provenance semiring is the semiring $\mathbb{N}[X]$ of multivariate polynomials with indeterminates from X and coefficients from $\mathbb{N}$. Other provenance semirings are obtained, for example, as quotient semirings of $\mathbb{N}[X]$ induced by congruences for idempotence and absorption. The resulting provenance values are less informative but their computation is more efficient.

- By dropping coefficients from $\mathbb{N}[X]$, we get the free idempotent semiring $\mathbb{B}[X]$ whose elements are (in one-to-one correspondence with) finite sets of monomials with coefficient 1. It is the quotient induced by $x + x \sim x$.
- If, in addition, exponents are dropped, we obtain the Why-semiring $\mathbb{W}(X)$ of finite sums of monomials with coefficient 1 that are linear in each indeterminate.
- The free absorptive semiring $\mathbb{S}(X)$ consists of 0, 1 and all antichains of monomials with respect to the absorption order $\succcurlyeq$. A monomial m_1 absorbs m_2, denoted $m_1 \succcurlyeq m_2$, if it has smaller exponents, i.e. $m_2 = m \cdot m_1$ for some monomial m.
- Finally, $(\text{PosBool}(X), \vee, \wedge, \bot, \top)$ is the semiring whose elements are classes of equivalent (in the usual sense) positive boolean expressions with boolean variables from X. Its elements are in bijection with the positive boolean expressions in irredundant disjunctive normal form. This is the lattice semiring freely generated by the set X. It arises from $\mathbb{S}(X)$ by dropping exponents.

3 First-Order Logic Interpreted in Commutative Semirings

We are interested in the provenance analysis of the model checking computation of first-order sentences. Such a computation is nicely and *declaratively* driven by the structure of the sentence, and thus amounts to a non-standard semantics for FO. In its simplest form, model checking takes as input a finite structure and the input items are the various facts (positive or negative) which hold in the model. We have found however that it pays to take a more general approach and specify not a structure but just its (finite) universe. This way we can track the use of positive and negative facts

in checking a sentence under multiple possible models on that universe. This allows a certain amount of *reverse analysis*: finding models that satisfy useful constraints.

3.1 Semiring Interpretations

Consider a finite relational vocabulary $\tau = \{R, S, \ldots\}$. From this vocabulary and a finite, non-empty universe A of *ground values* we construct the set $\mathsf{Facts}_A(\tau)$ of all ground relational atoms (facts) $R\overline{a}$, the set $\mathsf{NegFacts}_A(\tau)$ of all negated facts $\neg R\overline{a}$ and thus the set $\mathsf{Lit}_A(\tau) = \mathsf{Facts}_A(\tau) \cup \mathsf{NegFacts}_A(\tau)$ of all *literals*, positive and negative facts, over τ and A. By convention we will identify $\neg\neg R\overline{a}$ with $R\overline{a}$ so the negation of a literal is again a literal.

Any finite structure $\mathfrak{A} = (A, R^{\mathfrak{A}}, S^{\mathfrak{A}}, \ldots)$ with universe A makes some of these literals true and the remaining ones false. Note, however, that much of the development does not assume a specific model, and this can be usefully exploited. Let $\mathcal{S} = (S, +, \cdot, 0, 1)$ be a commutative semiring. Very roughly speaking, $0 \in S$ is intended to interpret false assertions, while an element $s \neq 0$ in $\mathcal{S}$ provides a "nuanced" or "annotated" interpretation for true assertions.

Next, $\mathcal{S}$-interpretations will map literals to elements of $\mathcal{S}$ and are then extended to all formulae. Disjunction and existential quantification are interpreted by the addition operation of $\mathcal{S}$. Conjunction and universal quantification are interpreted by the multiplication operation of $\mathcal{S}$. For quantifiers, the finiteness of the universe A of ground values will be essential. Extensions to infinite universes are possible for semirings with appropriate infinitary addition and multiplication operations, see [11], but they will not be considered in this paper. For negation we use the well-known syntactic transformation to *negation normal form (NNF)*, denoted $\psi \mapsto \mathrm{nnf}(\psi)$. Note that $\mathrm{nnf}(\psi)$ is a formula constructed from literals (positive and negative facts) and equality/inequality atoms using just $\wedge, \vee, \exists, \forall$.

Definition 2 An **$\mathcal{S}$-interpretation** is a mapping $\pi : \mathsf{Lit}_A(\tau) \to S$. This extends to valuations $\pi[\![\varphi(\overline{a})]\!]$ of any instantiation of a formula $\varphi(\overline{x}) \in \mathrm{FO}(\tau)$, by a tuple $\overline{a} \subseteq A$. We first extend π by mapping equalities and inequalities to their truth values, by setting $\pi[\![a = a]\!] := 1$ and $\pi[\![a = b]\!] := 0$ for $a \neq b$ (and analogously for inequalities). Further

$$\pi[\![\psi \vee \varphi]\!] := \pi[\![\psi]\!] + \pi[\![\varphi]\!] \qquad \pi[\![\psi \wedge \varphi]\!] := \pi[\![\psi]\!] \cdot \pi[\![\varphi]\!]$$

$$\pi[\![\exists x\, \psi(x)]\!] := \sum\nolimits_{a \in A} \pi[\![\psi(a)]\!] \qquad \pi[\![\forall x\, \psi(x)]\!] := \prod\nolimits_{a \in A} \pi[\![\psi(a)]\!]$$

$$\pi[\![\neg\varphi]\!] := \pi[\![\mathrm{nnf}(\neg\varphi)]\!]$$

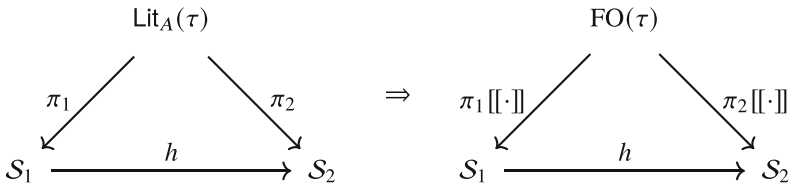
Notice that equality and inequality atoms are interpreted in $\mathcal{S}$ as 0 or 1, i.e., their provenance is not tracked. One could give a similar treatment to other such relations with "fixed" meaning, e.g., assuming an ordering on A, however, we omit this here.

By a trivial induction it follows that, as intended, it suffices to consider formulae in negation normal form.

Proposition 1 $\pi[\![\varphi]\!] = \pi[\![\mathrm{nnf}(\varphi)]\!]$.

A useful consequence of Proposition 1 is that we can prove further results by induction on formulas in NNF, and hence avoid the non-atomic negation altogether.

Proposition 2 (Fundamental Property) *Let* $h : \mathcal{S}_1 \to \mathcal{S}_2$ *be a semiring homomorphism and let* $\pi_1 : \mathsf{Lit}_A(\tau) \to \mathcal{S}_1$ *and* $\pi_2 : \mathsf{Lit}_A(\tau) \to \mathcal{S}_2$ *be interpretations such that* $h \circ \pi_1 = \pi_2$. *Then, for every sentence* $\varphi \in \mathrm{FO}(\tau)$ *we have* $h(\pi_1[\![\varphi]\!]) = \pi_2[\![\varphi]\!]$. *As diagrams*



Proof By Proposition 1 the proof can proceed by induction on formulae in NNF. For example $h(\pi_1[\![\varphi \wedge \psi]\!]) = h(\pi_1[\![\varphi]\!] \cdot_1 \pi_1[\![\psi]\!]) = h(\pi_1[\![\varphi]\!]) \cdot_2 h(\pi_1[\![\psi]\!]) = \pi_2[\![\varphi]\!] \cdot_2 \pi_2[\![\psi]\!] = \pi_2[\![\varphi \wedge \psi]\!]$. □

The somewhat bombastic name "fundamental property" is motivated by two observations. First, the property checks that the definition of our semantics is nicely compositional. Second, the property plays a central role in a strategy that we have widely applied with query languages in databases: compute provenance as generally as (computationally) feasible, then specialize via homomorphisms to coarser-grained provenance, or to specific domains, e.g., count, trust, cost or access control.

3.2 *Intermezzo: Positive Semirings*

We say that a semiring $\mathcal{S}$ has *divisors of 0* if there exist non-zero elements $s, t \in S$ such that $st = 0$. Among the semirings described in Sect. 2, only the Lukasiewicz semiring $\mathbb{L}$ and its isomorphic variant $\mathbb{D}$ have divisors of 0. Indeed in $\mathbb{L}$, we have that $s \odot t = 0$ if, and only if $s + t \leq 1$. We shall discuss in Sect. 4 further interesting semirings with divisors of 0.

A semiring $\mathcal{S}$ is $+$-*positive* if $s + t = 0$ implies $s = 0$ and $t = 0$. All semirings described in Sect. 2 are +-positive, but rings are not. Finally, a semiring is *positive* if it is $+$-positive and has no divisors of 0.

Proposition 3 *A semiring $\mathcal{S}$ is positive if, and only if, $\dagger_{\mathcal{S}} : S \to \mathbb{B}$ defined by*

$$\dagger_{\mathcal{S}}(s) = \begin{cases} \top & \text{if } s \neq 0 \\ \bot & \text{if } s = 0 \end{cases} \qquad \textit{is a homomorphism.}$$

3.3 Sanity Checks

Let $\mathfrak{A} = (A, R^{\mathfrak{A}}, S^{\mathfrak{A}}, \ldots)$ be a (finite) τ-model. The **canonical truth interpretation** for $\mathfrak{A}$ is, of course, $\pi_{\mathfrak{A}} : \mathsf{Lit}_A(\tau) \to \mathbb{B}$ where

$$\pi_{\mathfrak{A}}(L) = \begin{cases} \top & \text{if } \mathfrak{A} \models L \\ \bot & \text{otherwise} \end{cases}$$

Earlier we have discussed the "number of proof trees" as a non-standard semantics for FO-model checking. This is also captured by interpretations in a semiring. The **canonical counting interpretation** for $\mathfrak{A}$ is $\pi_{\#\mathfrak{A}} : \mathsf{Lit}_A \to \mathbb{N}$ where

$$\pi_{\#\mathfrak{A}}(L) = \begin{cases} 1 & \text{if } \mathfrak{A} \models L \\ 0 & \text{otherwise} \end{cases}$$

Proposition 4 (Sanity Checks) *For any first-order sentence φ we have $\mathfrak{A} \models \varphi$ if, and only if, $\pi_{\mathfrak{A}}[\![\varphi]\!] = \top$. Moreover, $\pi_{\#\mathfrak{A}}[\![\varphi]\!]$ is the number of proof trees that witness $\mathfrak{A} \models \varphi$.*

Now, let $\mathcal{S}$ be a commutative semiring, and let $\pi : \mathsf{Lit}_A(\tau) \to S$ be a $\mathcal{S}$-interpretation. As we have indicated, for a sentence φ we intend to interpret $\pi[\![\varphi]\!] = 0$ as "φ is false for π", while $\pi[\![\varphi]\!] = s \neq 0$ is interpreted as "π makes φ true with annotation s". We examine how this meshes with standard logical truth in a model.

Definition 3 A $\mathcal{S}$-interpretation $\pi : \mathsf{Lit}_A(\tau) \to S$ is **model-defining** when, for each fact $R\overline{a}$, precisely one of the values $\pi(R\overline{a})$ and $\pi(\neg R\overline{a})$ is 0. Indeed, every model-defining interpretation π uniquely defines a model $\mathfrak{A}_\pi$ with universe A such that for any literal L we have $\mathfrak{A}_\pi \models L$ if, and only if, $\pi(L) \neq 0$.

Both $\pi_{\mathfrak{A}}$ and $\pi_{\#\mathfrak{A}}$ shown above are model-defining and the model they define is $\mathfrak{A}$. If $\mathcal{S}$ is not $\mathbb{B}$ then several model-defining interpretations may define the same model. It is also clear that any finite model can be defined by such an interpretation, for any $\mathcal{S}$.

Proposition 5 (Another Sanity Check) *Let $\mathcal{S}$ be* positive*, and let π be a model-defining $\mathcal{S}$-interpretation. Then for any* FO*-sentence φ*

$$\mathfrak{A}_\pi \models \varphi \quad \Leftrightarrow \quad \pi[\![\varphi]\!] \neq 0$$

Proof By Proposition 3, since $\mathcal{S}$ is positive, $\dagger_{\mathcal{S}}$ is a homomorphism. Since π is model-defining let $\mathfrak{A}$ be the model defined by π. Clearly, $\dagger_{\mathcal{S}} \circ \pi$ is the canonical truth interpretation $\pi_{\mathfrak{A}}$. Applying Proposition 2 we get $\dagger_{\mathcal{S}}(\pi[\![\varphi]\!]) = \pi_{\mathfrak{A}}[\![\varphi]\!]$. Now the result follows from Proposition 4. □

In fact, we can refine the previous proposition as follows.

Proposition 6 (Refinement of Proposition 5) *For any semiring $\mathcal{S}$ (positive or not!), for any model-defining $\mathcal{S}$-interpretation π, and for any* FO*-sentence φ we have*

$$\pi[\![\varphi]\!] \neq 0 \quad \Rightarrow \quad \mathfrak{A}_\pi \models \varphi.$$

Moreover, a semiring $\mathcal{S}$ is positive if, and only if, for any model-defining $\mathcal{S}$-interpretation π and any FO*-sentence φ we have*

$$\mathfrak{A}_\pi \models \varphi \quad \Rightarrow \quad \pi[\![\varphi]\!] \neq 0.$$

Proof The first part of the proposition is a simple induction on φ. For the second implication we first prove that $\mathcal{S}$ has no divisors of 0. Suppose that $s, t \in \mathcal{S}$ are such that $s \neq 0$, $t \neq 0$ but $st = 0$. Consider $A = \{a_1, a_2\}$ and the model-defining interpretation defined by $\pi(\neg Ra_1) = \pi(\neg Ra_2) = 0$, $\pi(Ra_1) = s$, $\pi(Ra_2) = t$ as well as the sentence $\varphi = Ra_1 \wedge Ra_2$. We have $\mathfrak{A}_\pi \models \varphi$ hence $\pi[\![\varphi]\!] \neq 0$, contradiction.

Next we prove that $\mathcal{S}$ is $+$-positive. Let $s, t \in \mathcal{S}$ be such that $s \neq 0$ or $t \neq 0$. Consider the same interpretation π as above, with the sentence $\psi = Ra_1 \vee Ra_2$. We have $\mathfrak{A}_\pi \models \psi$ hence $0 \neq \pi[\![\psi]\!] = s + t$. □

3.4 *"Consistency" and "Completeness" for Semiring Interpretations*

In the study of provenance we shall also have occasion to consider interpretations that do not correspond to a single specific model (as formalized in Definition 3). Additional issues arise for such interpretations.

An interpretation in which both $\pi[\![\varphi]\!] \neq 0$ and $\pi[\![\neg\varphi]\!] \neq 0$ for some sentence φ is seemingly "inconsistent". On the other hand, an interpretation in which both $\pi[\![\varphi]\!] = 0$ and $\pi[\![\neg\varphi]\!] = 0$ for some sentence φ seems to be "incomplete". Of course, neither of these situations arises for a model-defining $\mathcal{S}$-interpretation when

$\mathcal{S}$ is positive (by Proposition 5). We analyze each of these issues in turn for general interpretations.

Proposition 7 *Let $\pi : \mathsf{Lit}_A(\tau) \to \mathcal{S}$ be a $\mathcal{S}$-interpretation. If for every $L \in \mathsf{Lit}_A(\tau)$ at least one of $\pi(L)$ and $\pi(\neg L)$ is 0 then there exists no sentence ψ for which both $\pi[\![\psi]\!] \neq 0$ and $\pi[\![\neg\psi]\!] \neq 0$.*

Proof Suppose that both $\pi[\![\psi]\!]$ and $\pi[\![\neg\psi]\!]$ are non-zero. Since we assume A to be finite, there exist finitely many sentences $\varphi_1, \ldots, \varphi_k$ such that one of the values $\pi[\![\psi]\!]$ and $\pi[\![\neg\psi]\!]$ is the sum of $\pi[\![\varphi_1]\!], \ldots, \pi[\![\varphi_k]\!]$, and the other is the product of $\pi[\![\neg\varphi_1]\!], \ldots \pi[\![\neg\varphi_k]\!]$. It follows that all values $\pi[\![\neg\varphi_i]\!]$ are non-zero. But by induction hypothesis, this implies that all values $\pi[\![\varphi_i]\!]$, and hence also their sum, must be 0, so we have a contradiction. □

Observe that if at least one of $\pi[\![\varphi]\!]$ or $\pi[\![\neg\varphi]\!]$ is 0 then $\pi[\![\varphi]\!] \cdot \pi[\![\neg\varphi]\!] = 0$. If $\mathcal{S}$ has no divisors of 0 the converse holds as well. Although most of the examples described in Sect. 2 are positive semirings, we are about to introduce, in Sect. 4.1, a semiring for FO-provenance that *does* have divisors of 0. For this reason we note also the following:

Proposition 8 *Let $\pi : \mathsf{Lit}_A(\tau) \to \mathcal{S}$ be an $\mathcal{S}$-interpretation. If for every $L \in \mathsf{Lit}_A$ we have $\pi(L) \cdot \pi(\neg L) = 0$ then for any sentence ψ we have $\pi[\![\psi]\!] \cdot \pi[\![\neg\psi]\!] = 0$.*

Proof If ψ is not a literal, then there exists sentences $\varphi_1, \ldots, \varphi_k$ such that one of the values $\pi[\![\psi]\!]$ and $\pi[\![\neg\psi]\!]$ is the sum of $\pi[\![\varphi_1]\!], \ldots, \pi[\![\varphi_k]\!]$, and the other is the product of $\pi[\![\neg\varphi_1]\!], \ldots \pi[\![\neg\varphi_k]\!]$. By induction hypothesis we have that $\pi[\![\varphi_i]\!] \cdot \pi[\![\neg\varphi_i]\!] = 0$ for all $i \leq k$. It follows that

$$\pi[\![\psi]\!] \cdot \pi[\![\neg\psi]\!] = \sum_{i \leq k} \pi[\![\varphi_i]\!] \cdot \prod_{j \leq k} \pi[\![\neg\varphi_j]\!] = \sum_{i \leq k} \Big(\pi[\![\varphi_i]\!] \cdot \prod_{j \leq k} \pi[\![\neg\varphi_j]\!] \Big)$$
$$= \sum_{i \leq k} \Big(\pi[\![\varphi_i]\!] \cdot \pi[\![\neg\varphi_i]\!] \cdot \prod_{j \neq i} \pi[\![\neg\varphi_j]\!] \Big) = 0.$$

□

Propositions 7 and 8 hold in arbitrary semirings and each supports a kind of "consistency", with the two kinds coinciding when the semiring has no divisors of 0.

Turning to "completeness", note that if both $\pi[\![\varphi]\!]$ and $\pi[\![\neg\varphi]\!]$ are 0 then $\pi[\![\varphi]\!] + \pi[\![\neg\varphi]\!] = 0$. If $\mathcal{S}$ is +-positive then the converse holds as well. However, for arbitrary $\mathcal{S}$, neither an analog of Proposition 7 nor one of Proposition 8 holds. Indeed, let $\mathcal{S} = \mathbb{Z}_4$. Consider the vocabulary consisting of one unary relation symbol R and let $A = \{a_1, a_2\}$. For the interpretation given by $\pi(\neg Ra_1) = \pi(\neg Ra_2) = \pi(Ra_1) = \pi(Ra_2) = 2$ and the sentence $\varphi = Ra_1 \wedge Ra_2$ we have $\pi[\![\varphi]\!] = \pi[\![\neg\varphi]\!] = 0$.

Instead, we have the following for positive semirings.

Proposition 9 *Let $\pi : \mathsf{Lit}_A(\tau) \to \mathcal{S}$ be an $\mathcal{S}$-interpretation into a positive semiring. If for every $L \in \mathsf{Lit}_A(\tau)$ we have $\pi(L) \neq 0$ or $\pi(\neg L) \neq 0$ (equivalently, $\pi(L) + \pi(\neg L) \neq 0$) then for any sentence ψ we have $\pi[\![\psi]\!] \neq 0$ or $\pi[\![\neg\psi]\!] \neq 0$ (equivalently, $\pi[\![\psi]\!] + \pi[\![\neg\psi]\!] \neq 0$).*

Proof Towards a contradiction, suppose that $\pi[\![\psi]\!] = \pi[\![\neg\psi]\!] = 0.$, As in the two previous proofs, take $\varphi_1, \ldots, \varphi_k$ such that one of the values $\pi[\![\psi]\!]$ and $\pi[\![\neg\psi]\!]$ is the sum of $\pi[\![\varphi_1]\!], \ldots, \pi[\![\varphi_k]\!]$, and the other is the product of $\pi[\![\neg\varphi_1]\!], \ldots \pi[\![\neg\varphi_k]\!]$. Since $\mathcal{S}$ has no divisors of 0, it follows that $\pi[\![\neg\varphi_i]\!] = 0$ for at least one $i \leq k$. By induction hypothesis, $\pi[\![\varphi_i]\!] \neq 0$, which, by +-positivity, contradicts the assumption that $\pi[\![\psi]\!] = 0$. □

3.5 Proof Trees

Reasoning about the proof trees that a particular semiring interpretation admits for a given first-order sentence is an important aspect of provenance analysis. We shall prove that the provenance value of every sentence is the same as the sum of the valuations of its proof trees. To establish this result, we have to provide a precise definition of proof trees and their valuations.

An *evaluation tree* for a sentence $\psi \in$ FO and a semiring interpretation $\pi : \mathsf{Lit}_A(\tau) \to \mathcal{S}$ (into an arbitrary semiring $\mathcal{S}$) is a directed tree $\mathcal{T}$, whose nodes are (labelled by) occcurrences[1] of formulae $\varphi(\overline{a})$, where $\varphi(\overline{x})$ is a subformula of ψ whose free variables $\overline{x}$ are instantiated by a tuple $\overline{a}$ of elements from A, such that the following conditions hold.

- The root of $\mathcal{T}$ is ψ.
- A node $\varphi \vee \vartheta$ has one child which is either φ or ϑ.
- A node $\varphi \wedge \vartheta$ has two children φ and ϑ.
- A node $\exists y \varphi(\overline{a}, y)$ has one child $\varphi(\overline{a}, b)$ for some $b \in A$.
- A node $\forall y \varphi(\overline{a}, y)$ has the children $\varphi(\overline{a}, b)$ for all $b \in A$.
- The leaves of $\mathcal{T}$ are literals $L \in \mathsf{Lit}_A(\tau)$.

For any literal L, let $\#_L(\mathcal{T})$ be the number of occurrences of L in $\mathcal{T}$. The valuation $\mathcal{T}$ is

$$\pi(\mathcal{T}) := \sum_{L \in \mathsf{Lit}_A(\tau)} \pi(L)^{\#_L(\mathcal{T})}.$$

A *proof tree* for π and $\psi \in$ FO is an evaluation tree $\mathcal{T}$ with $\pi(\mathcal{T}) \neq 0$. If π is clear from the context, we write $T(\psi)$ for the set of all proof trees for π and ψ.

[1] Notice that we consider different occurrences of the same subformula as separate objects. In particular, a sentence $\varphi \vee \varphi$ has twice as many evaluation trees as φ.

Theorem 1 *For every semiring interpretation $\pi : \mathsf{Lit}_A(\tau) \to \mathcal{S}$ and every sentence $\psi \in \mathrm{FO}(\tau)$, we have that*

$$\pi[\![\psi]\!] = \sum_{\mathcal{T} \in T(\psi)} \pi(\mathcal{T}).$$

Proof We proceed by induction on ψ.

- Let ψ be a literal. If $\pi(\psi) = 0$ then ψ has no proof tree, so the sum over the valuations of its proof trees is 0. Otherwise ψ has precisely one proof tree which is the literal itself. In both cases, the desired equality holds trivially.
- Let $\psi = \varphi \vee \vartheta$. A proof tree $\mathcal{T}$ for ψ has the root ψ followed by a proof tree $\mathcal{T}'$ for either φ or for ϑ; clearly $\pi(\mathcal{T}) = \pi(\mathcal{T}')$. Thus

$$\pi[\![\psi]\!] = \pi[\![\varphi]\!] + \pi[\![\vartheta]\!] = \sum_{\mathcal{T}' \in T(\varphi)} \pi(\mathcal{T}') + \sum_{\mathcal{T}' \in T(\vartheta)} \pi(\mathcal{T}') = \sum_{\mathcal{T} \in T(\psi)} \pi(\mathcal{T}).$$

- Let $\psi = \varphi \wedge \vartheta$. A proof tree $\mathcal{T}$ for ψ has the root ψ, attached to which are a proof tree $\mathcal{T}'$ for φ and a proof tree $\mathcal{T}''$ for ϑ. We can thus identify every $\mathcal{T} \in T(\psi)$ with a pair $(\mathcal{T}', \mathcal{T}'') \in T(\varphi) \times T(\vartheta)$, and since $\#_L(\mathcal{T}) = \#_L(\mathcal{T}') + \#_L(\mathcal{T}'')$ for every literal L we have that, $\pi(\mathcal{T}) = \pi(\mathcal{T}')\pi(\mathcal{T}'')$. It follows that

$$\begin{aligned} \pi[\![\psi]\!] = \pi[\![\varphi]\!] \cdot \pi[\![\vartheta]\!] &= \sum_{\mathcal{T}' \in T(\varphi)} \pi(\mathcal{T}') \cdot \sum_{\mathcal{T}'' \in T(\vartheta)} \pi(\mathcal{T}'') \\ &= \sum_{(\mathcal{T}', \mathcal{T}'') \in T(\psi)} \pi(\mathcal{T}')\pi(\mathcal{T}'') = \sum_{\mathcal{T} \in T(\psi)} \pi(\mathcal{T}). \end{aligned}$$

- If $\psi = \exists y \varphi(y)$, then a proof tree $\mathcal{T}$ for ψ consists of the root ψ, attached to which is a proof tree $\mathcal{T}_a$ for $\varphi(a)$, for some $a \in A$. Clearly $\pi(\mathcal{T}) = \pi(\mathcal{T}_a)$. It follows that

$$\pi[\![\psi]\!] = \sum_{a \in A} \pi[\![\varphi(a)]\!] = \sum_{a \in A} \sum_{\mathcal{T}_a \in T(\varphi(a))} \pi(\mathcal{T}_a) = \sum_{\mathcal{T} \in T(\psi)} \pi(\mathcal{T}).$$

- Let finally $\psi = \forall y \varphi(y)$. A proof tree $\mathcal{T}$ for ψ consists of the root ψ attached to which are proof trees $\mathcal{T}_a$ for $\varphi(a)$, for all $a \in A$, so we can identify every proof tree $\mathcal{T} \in T(\psi)$ with the tuple $(\mathcal{T}_a)_{a \in A}$. Further, for every literal L, we have that $\#_L(\mathcal{T}) = \sum_{a \in A} \#_L(\mathcal{T}_a)$ and therefore $\pi(\mathcal{T}) = \prod_{a \in A} \pi(\mathcal{T}_a)$. It follows, by the

distributive law for tuples, that

$$\pi[\![\psi]\!] = \prod_{a\in A} \pi[\![\varphi(a)]\!] = \prod_{a\in A} \sum_{\mathcal{T}_a\in T(\varphi(a))} \pi(\mathcal{T}_a)$$
$$= \sum_{(\mathcal{T}_a)_{a\in A}\in T(\psi)} \prod_{a\in A} \pi(\mathcal{T}_a) = \sum_{\mathcal{T}\in T(\psi)} \pi(\mathcal{T}).$$

□

4 A Provenance Semiring for First-Order Logic

We have claimed in Sect. 2 that $\mathbb{N}[Y]$, the commutative semiring freely generated by a set Y is used for provenance tracking. The elements of Y label the information whose propagation we wish to capture in provenance. This works fine for *positive* database query languages [32] but difference or negation cause problems. Here we shall use a variation on the idea of polynomials in order to deal with negated facts in provenance analysis.

We construct a semiring whose elements can be identified with certain polynomials that describe the provenance of first-order model checking. The main insight is the use of indeterminates in "positive-negative pairs". We show that the resulting polynomials provide a nicely dual interpretation for provenance that captures model-checking proofs. We illustrate this with a running example.

4.1 Dual-Indeterminate Polynomials

Let $X, \overline{X}$ be two disjoint sets together with a one-to-one correspondence between X and nnX. We denote by $p \in X$ and $\overline{p} \in \overline{X}$ two elements that are in this correspondence. We refer to the elements of $X \cup \overline{X}$ as **provenance tokens** as they will be used to label or annotate some of the "data", i.e., literals over some ground values, via the concept of $\mathcal{S}$-interpretation that we defined previously. Indeed, if we fix a finite non-empty set A and consider $\mathsf{Lit}_A(\tau) = \mathsf{Facts}_A(\tau) \cup \mathsf{NegFacts}_A(\tau)$ then we shall use X for Facts_A and $\overline{X}$ for $\mathsf{NegFacts}_A$. By convention, if we annotate $R\overline{a}$ with the "positive" token p then the "negative" token $\overline{p}$ can only be used to annotate $\neg R\overline{a}$, and vice versa. We refer to p and $\overline{p}$ as **complementary tokens**.

Definition 4 We denote by $\mathbb{N}[X, \overline{X}]$ the quotient of the semiring of polynomials $\mathbb{N}[X \cup \overline{X}]$ by the congruence generated by the equalities $p \cdot \overline{p} = 0$ for all $p \in X$.[2] We will call the elements of $\mathbb{N}[X, \overline{X}]$ **dual-indeterminate (provenance) polynomials**.

Observe that two polynomials $\mathfrak{p}, \mathfrak{q} \in \mathbb{N}[X \cup \overline{X}]$ are congruent if, and only if, they become identical after deleting from each of them the monomials that contain complementary tokens. Hence, the congruence classes in $\mathbb{N}[X, \overline{X}]$ are in one-to-one correspondence with the polynomials in $\mathbb{N}[X \cup \overline{X}]$ whose monomials do not contain complementary tokens, so we could have defined these to be dual-indeterminate polynomials. In particular, we can multiply such polynomials as usual, provided that we eliminate the monomials with complementary tokens afterwards. The following is the universality property of the semiring of dual-indeterminate polynomials.

Proposition 10 *For any commutative semiring $\mathcal{S}$ and for any $f : X \cup \overline{X} \to \mathcal{S}$ such that $\forall p \in X\ f(p) \cdot f(\overline{p}) = 0$ there exists a unique semiring homomorphism $h : \mathbb{N}[X, \overline{X}] \to \mathcal{S}$ such that $h(x) = f(x)$ for all $x \in X \cup \overline{X}$.*

The dual-indeterminate provenance polynomials serve to track both positive and negative facts about the model throughout a model-checking computation, as we shall illustrate in Sect. 4.2. We note that $\mathbb{N}[X, \overline{X}]$ is $+$-positive, but not positive, since it has divisors of 0, such as

$$p \cdot \overline{p} \ = \ 0, \quad (p + \overline{q})\overline{p}q \ = \ 0, \quad (p\overline{q} + \overline{p}q)(pq + \overline{pq}) \ = \ 0.$$

However, keeping both p and $\overline{p}$ around and even using them in certain "inconsistent" $\mathbb{N}[X, \overline{X}]$-interpretations can be very useful in provenance analysis, as we shall see in Sect. 5.1, and the subsequent sections.

Finally, we remark that the construction with dual indeterminates can be replicated for the provenance semirings $\mathbb{B}[X]$, $\mathbb{S}(X)$, $\mathbb{W}(X)$, and PosBool(X). For the last one, the result, PosBool$(X, \overline{X})$, corresponds to the usual boolean expressions, in irredundant disjunctive normal form.

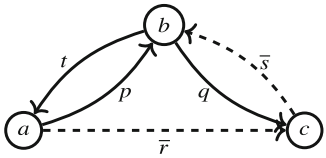
4.2 Provenance Tracking: An Example and a Characterization

We can now consider interpretations into the semiring of dual-indeterminate polynomials.

Definition 5 A **provenance-tracking** interpretation is a $\mathbb{N}[X, \overline{X}]$-interpretation $\pi : \mathsf{Lit}_A \to \mathbb{N}[X, \overline{X}]$ such that $\pi(\mathsf{Facts}_A) \subseteq X \cup \{0, 1\}$ and $\pi(\mathsf{NegFacts}_A) \subseteq \overline{X} \cup \{0, 1\}$.

[2] This is the same as factoring by the ideal generated by the polynomials $p\overline{p}$ for all $p \in X$.

Fig. 1 The model G



To track provenance for a given model $\mathfrak{A}$ we use a provenance-tracking interpretation that is also model-defining, and in fact defines precisely the model $\mathfrak{A}$. The idea is that if the interpretation annotates a positive or negative fact with a token, then we wish to track that fact through the model-checking computation. On the other hand annotating with 0 or 1 is done when we do not track the fact, yet we need to recall whether it holds or not in the model.

For the following example, the vocabulary of directed graphs consists of one binary predicate E denoting directed edges. Consider, over this vocabulary, the following formula and sentence

$$\mathsf{dominant}(x) \;:=\; \forall y \,\big(x = y \vee (Exy \wedge \neg Eyx)\big), \qquad \varphi \;:=\; \forall x\, \neg \mathsf{dominant}(x).$$

Obviously, $\mathsf{dominant}(x)$ says that in a digraph with edge relation E the vertex x is "dominant" while φ says that the digraph does not have a dominant vertex. Consider also, as a model, the digraph G depicted in Fig. 1 with vertices $V = \{a, b, c\}$. The finite set V is the universe of ground values for our interpretations.

We adopt the visual convention of representing the edges of the digraph as solid arrows and of labeling them with positive tokens when we wish to track their presence through model-checking (see p, q, t) and with 1 when we are not interested in tracking them (not occurring in this example). Moreover, we represent with dashed arrows absent edges, but only those whose absence, however, we also wish to track, by labeling them with negative tokens $(\overline{r}, \overline{s})$. We extend this to all literals as follows, with the provenance-tracking $\mathbb{N}[X, \overline{X}]$-interpretation $\beta : \mathsf{Lit}_V \to X \cup \overline{X} \cup \{0, 1\}$ defined by

$$\beta(L) \;=\; \begin{cases} p & \text{if } L = Eab \\ 0 & \text{if } L = \neg Eab \\ q & \text{if } L = Ebc \\ 0 & \text{if } L = \neg Ebc \\ 0 & \text{if } L = Eac \\ \overline{r} & \text{if } L = \neg Eac \end{cases} \quad \text{and} \quad \beta(L) \;=\; \begin{cases} 0 & \text{if } L = Ecb \\ \overline{s} & \text{if } L = \neg Ecb \\ t & \text{if } L = Eba \\ 0 & \text{if } L = \neg Eba \\ 0 & \text{for the other positive facts} \\ 1 & \text{for the other negative facts.} \end{cases}$$

For example, $\beta(Eca) = \beta(Ecc) = \ldots = 0$ and $\beta(\neg Eca) = \beta(\neg Ecc) = \ldots = 1$. Note that β is model-defining in the sense of of Sect. 3 and that the model it defines is precisely G.

The assumptions made in the definition of β indicate that we choose to track positive facts like Ebc and negative facts like $\neg Eab$, etc., as they are used in establishing the truth of some sentence in G. They also indicate that we accept, and thus do not track, the absence of the other potential edges such as Eca. We think of data annotated with 0 as being "forget-about-it" absent and of data annotated with 1 as "available for free" present.

Clearly, $G \models \varphi := \forall x\, \neg\mathsf{dominant}(x)$. But how can we justify this in terms of the facts, negative or positive, that hold in the model? By computing the semantics of φ under the interpretation β we will obtain *provenance information* for the result that $G \models \varphi$. Clearly

$$\mathrm{nnf}(\varphi) \;=\; \forall x\, \exists y\, \big(x \neq y \wedge (\neg Exy \vee Eyx)\big)$$

and therefore

$$\beta[\![\varphi]\!] \;=\; \beta[\![\mathrm{nnf}(\varphi)]\!] \;=\; (\overline{r}+t)\cdot p\cdot(1+q+\overline{s}) \;=\; p\overline{r}+pt+pq\overline{r}+pqt+p\overline{rs}+p\overline{s}t.$$

Each of the monomials of the polynomial $\beta[\![\varphi]\!]$ has coefficient 1 and all the exponents are 1. This is certainly not the case in general. For instance, we could have changed the provenance assumptions β to have 1 instead of r (and therefore 0 instead of $\overline{r}$). In addition suppose that the potential presence/absence of all three edges Eab, Ecb and Eba has the *same* provenance (e.g., same data source) so we choose to track their presence/absence with the same tokens $p/\overline{p}$. This results in $\beta[\![\varphi]\!] = (\overline{p}+p)(2p+\overline{q}+\overline{p})(2+q+\overline{p}) = \ldots + 4p^2 + \ldots + 2\overline{pq} + \ldots + \overline{p}^3 + \ldots$. In fact, it is possible to show that any polynomial can be computed as some provenance, with suitable choices of sentence, model, and interpretation.

Each of the monomials obtained with a provenance-tracking interpretation corresponds to a different (model-checking) proof tree of φ from the literals described by the monomial. We illustrate this with the proof tree corresponding to another monomial, $p\overline{rs}$, using the following formula abbreviations:

$$\begin{aligned}
\mathsf{denydom}(x,y) &:= \big(x \neq y \wedge (\neg Exy \vee Eyx)\big) && y \text{ denies dominance of } x\\
\mathsf{notdom}(x) &:= \exists y\, \big(x \neq y \wedge (\neg Exy \vee Eyx)\big) && x \text{ is not dominant}\\
\mathsf{noVdom} &:= \forall x\, \exists y\, \big(x \neq y \wedge (\neg Exy \vee Eyx)\big) && \text{no vertex is dominant}
\end{aligned}$$

Further, we abbreviate $\neg Exy \vee Eyx$ by $\vartheta(x,y)$. With these, the proof tree corresponding to $p\overline{rs}$ is:

$$\dfrac{\dfrac{\dfrac{a \neq c \qquad \dfrac{\neg Eac\ [\overline{r}]}{\vartheta(a,c)}}{\mathsf{denydom}(a,c)}}{\mathsf{notdom}(a)} \qquad \dfrac{\dfrac{b \neq a \qquad \dfrac{Eab\ [p]}{\vartheta(b,a)}}{\mathsf{denydom}(b,a)}}{\mathsf{notdom}(b)} \qquad \dfrac{\dfrac{c \neq b \qquad \dfrac{\neg Ecb\ [\overline{s}]}{\vartheta(c,b)}}{\mathsf{denydom}(c,b)}}{\mathsf{notdom}(c)}}{\mathsf{noVdom}}$$

The formulae on the leaves of the tree are accompanied by the tokens that β annotates them with.

The following proposition, which is an immediate consequence of Theorem 1, summarizes the situation.

Proposition 11 *Let* $\beta : \mathsf{Lit}_A \to \mathbb{N}[X, \overline{X}]$ *be a provenance-tracking model-defining interpretation, and let* φ *be an* FO*-sentence. Then, the dual-indeterminate polynomial* $\beta[\![\varphi]\!]$ *describes* all *the proof trees that verify* φ *using premises from among the literals that* β *maps to provenance tokens or to 1 (i.e., from the literals that hold in* $\mathfrak{A}_\beta$*). Specifically, each monomial* $m\, x_1^{m_1} \cdots x_k^{m_k}$ *corresponds to* m *distinct proof trees that use* m_1 *times a literal that* β *annotates by* x_1*, . . . , and* m_k *times a literal annotated by* x_k*, as well as any number of the literals annotated with 1. In particular,* $\beta[\![\varphi]\!] \neq 0$ *if, and only if, some proof tree exists, and if, and only if,* $\mathfrak{A}_\beta \models \varphi$.

Note that since $\mathbb{N}[X, \overline{X}]$ is not positive this proposition does not follow from Proposition 5. (Nor does this contradict Proposition 6 because provenance-tracking interpretations have a special form.) Nonetheless, albeit not positive, $\mathbb{N}[X, \overline{X}]$ has many remarkable properties and this proposition is a corollary of a more general one that we shall state in Sect. 5.2.

4.3 The Size of Provenance

In this section we explore the size of provenance as a complexity measure. In Sect. 4.1 we have introduced the dual-indeterminate polynomials as formal embodiments of the provenance of model checking assertions $\mathfrak{A} \models \varphi$. However, syntactically, these polynomials are normal forms of expressions built from tokens with the operations $+, \cdot, 0, 1$. Many such expressions are equationally equivalent under the laws of semirings. Thus, we can think of *representing* provenance as polynomials (normal forms), or, more economically, parenthesized expressions, or even as circuits in which common subexpressions are shared. A simple induction on the structure of formulas yields:

Proposition 12 *Provenance of* $\mathfrak{A} \models \varphi$ *computed as parenthesized expression has size polynomial in* $|\mathfrak{A}|$ *and exponential in* $|\varphi|$.

On the other hand, converting provenance to polynomial normal form can involve a combinatorial explosion in the number of monomials (although the size of each monomial remains polynomial because there are only polynomially many facts). Indeed, consider the universe $\{a_1, \ldots, a_n\}$, the sentence (a trivial tautology) $\forall x(Rx \vee \neg Rx)$ and the interpretation $\pi(Ra_i) = p_i, \pi(\neg Ra_i) = \overline{p}_i$. As a parenthesized expression, the model checking provenance is $(p_1 + \overline{p}_1) \cdots (p_n + \overline{p}_n)$ which has size $O(n)$ while in polynomial normal form it has size $\Omega(2^n)$.

This is due to the presence of the universal quantifiers. Indeed, an examination of the proof of the previous proposition yields:

Proposition 13 *If φ is an existential sentence in NNF then the dual-indeterminate polynomial corresponding to checking it in a model $\mathfrak{A}$ has only polynomially (in $|\mathfrak{A}|$) many monomials.*

4.4 Explanations Obtained from Provenance

In Sect. 4.2 we have calculated the provenance of the sentence φ that asserts "there is no dominant vertex",

$$\varphi := \forall x\, \neg \mathsf{dominant}(x) \quad \text{where} \quad \mathsf{dominant}(x) := \forall y\big(x = y \vee (Exy \wedge \neg Eyx)\big),$$

being checked in the model and interpretation in Fig. 1, resulting in the polynomial $p\bar{r} + pt + pq\bar{r} + pqt + p\bar{r}\bar{s} + p\bar{s}t$. As we have explained, each monomial of this polynomial corresponds to a proof tree for model-checking.

As a practical matter, we can think of each of the monomials of this polynomial as an *alternative explanation*. For example, the monomial $p\bar{r}\bar{s}$ (shown to correspond to a proof tree illustrated above) gives an explanation that can be stated informally as follows: there is no dominant vertex because of the presence of an edge from a to b together with the absence of edges from a to c and from c to b.

Note that if we examine the proof tree we can obtain more information: the presence of Eab is used to show that b is not dominant, the absence of Eac shows that a is not dominant, and the absence of Ecb shows that c is not dominant; therefore none of the vertices is dominant.

Another example that helps underlining the difference between monomials and proof trees is given by $p\bar{r}$. As an explanation, this monomial is a strict part of the explanation supplied by $p\bar{r}\bar{s}$. However, $p\bar{r}$ corresponds to a proof tree of φ in which the presence of Eab and the absence of Eac are used as in the proof tree for $p\bar{r}\bar{s}$ above, yet, this tree uses the absence of Eca, which is *accepted without tracking*—it has provenance 1—to show that c is not dominant. Overall for φ we could stick for practical purposes to just minimal explanations, which here are just $p\bar{r}$ and pt.

4.5 From Provenance to Confidence

Recall from Sect. 2 the Viterbi semiring $\mathbb{V}$. We think of the elements of $\mathbb{V}$ as confidence scores. Going back to the example in Sect.4.2, and assuming specific confidence scores for the literals that G makes true, and that we track, we wish to compute a confidence score for $G \models \varphi$.

Specifically, consider the $\mathbb{V}$-interpretation $\gamma : \mathsf{Lit}_V \to [0, 1]$ defined by

- $\gamma(Eab) = \gamma(Ebc) = 0.9, \gamma(Eba) = 0.2$, and $\gamma(Euv) = 0$ for any *other* positive fact Euv,
- $\gamma(\neg Euv) = 0$ whenever $\gamma(Euv) \neq 0$,
- $\gamma(\neg Eac) = \gamma(\neg Ecb) = 0.6$, and
- $\gamma(\neg Euv) = 1$ for any *other* negative fact with $\gamma(Euv) = 0$.

With this we could use Definition 2 to compute $\gamma[\![\varphi]\!] \in [0, 1]$, which is the desired confidence score.

However, since we have already computed in Sect. 4.2 the provenance $\beta[\![\varphi]\!]$ we can take advantage of the Fundamental Property (Proposition 2) via a homomorphism whose existence is guaranteed by Proposition 10.

We define $f : X \cup \overline{X} \to [0, 1]$ by

- $f(p) = f(q) = 0.9$, $f(t) = 0.2$, and $f(x) = 0$ for $x \notin \{p, q, t\}$,
- $f(\overline{x}) = 0$ for $x \in \{p, q, t\}$,
- $f(\overline{r}) = f(\overline{s}) = 0.6$, and
- $f(\overline{x}) = 1$ for $\overline{x} \notin \{\overline{p}, \overline{q}, \overline{t}, \overline{r}, \overline{s}\}$.

The condition on f in Proposition 10 is satisfied, hence f can be extended to a homomorphism $h : \mathbb{N}[X, \overline{X}] \to \mathbb{V}$. From the definition of f we have $h \circ \beta = \gamma$. By the Fundamental Property

$$\gamma[\![\varphi]\!] \;=\; h(\beta[\![\varphi]\!]).$$

Hence the score we wish to compute can be obtained by applying the homomorphism h to the dual polynomial $\beta[\![\varphi]\!] = p\overline{r} + pt + pq\overline{r} + pqt + p\overline{r}\overline{s} + p\overline{s}t$. It is easier to use the factored form of $\beta[\![\varphi]\!]$:

$$h(p(\overline{r} + t)(1 + q + \overline{s})) \;=\; 0.9 \cdot \max(0.6,\ 0.2) \cdot \max(1,\ 0.9,\ 0.6) \;=\; 0.54.$$

In general, confidence calculation may be only one of the analyses that we wish to perform. When these analyses are based on semiring calculations we can compute the provenance just once and then evaluate it in multiple semirings and under multiple valuations, by virtue of the Fundamental Property.

4.6 *Detailed Provenance Analysis: Top-Secret Proofs*

We describe here another kind of provenance analysis that we can perform in conjunction with interpretation in various semirings. Recall from Sect. 2 the access control semiring $\mathbb{A}$. Its elements are interpreted as *clearance levels* $0 < \mathsf{T} < \mathsf{S} < \mathsf{C} < \mathsf{P} = 1$. For example, administrators would assign clearance levels to the different items in the input data. The resulting clearance level for the output of a

computation determines which users get to access that output. In the context of this paper there would be an assignment of clearance levels to literals.

Going back to the example in Sect. 4.2, consider the $\mathbb{A}$-interpretation $\alpha : \mathrm{Lit}_V(\tau) \to \mathbb{A}$ defined by setting

$$\alpha(Eab) = \alpha(Ebc) = \alpha(Eba) = \mathsf{P}, \qquad \alpha(\neg Eac) = \alpha(\neg Ecb) = \mathsf{T},$$

and $\alpha(Euv) = 0$ for any *other* positive fact, $\alpha(\neg Euv) = \mathsf{P}$ for the corresponding negative fact.

As in Sect. 4.5 we have $\alpha[\![\varphi]\!] = h(p\overline{r} + pt + pq\overline{r} + pqt + p\overline{r}\overline{s} + p\overline{s}t)$, where h is the unique homomorphism $\mathbb{N}[X, \overline{X}] \to \mathbb{A}$ such that $h(p) = h(q) = h(t) = \mathsf{P}$, $h(\overline{r}) = h(\overline{s}) = \mathsf{T}$, and otherwise equals 0 on the rest of X and equals P on the rest of $\overline{X}$.

We can see that $\alpha[\![\varphi]\!] = \mathsf{P}$ but we can also perform a more detailed analysis in which we can associate clearance levels to individual proof trees. Thus, while it will be publicly known that $G \models \varphi$, those with top-secret clearance can know that also $p\overline{r}$ describes a proof of the assertion $G \models \varphi$. This may become relevant if we have particularly high confidence (as described above in Sect. 4.5) in the literals that p and $\overline{r}$ annotate, that is, in the presence of the edge from a to b and in the absence of an edge from a to c.

5 Reverse Provenance Analysis

There are limitations to what we can do with the provenance of a model-checking assertion $\mathfrak{A} \models \varphi$ under an interpretation that defines the model $\mathfrak{A}$. It is even more interesting to consider provenance-tracking interpretations that allow us to *choose*, from among multiple models, the ones that fulfill various desiderata.

5.1 A Reverse Analysis Example

Let $V = \{a, b, c\}$ be a set of ground values. As before, these will eventually play the role of the vertices of a digraph. However, we do not yet specify a set of edges, i.e., we do not specify a finite model with universe V. Instead, as illustrated by the dotted edges in Fig. 2, we supply a set of provenance tokens $X = \{p, q, r, s, t\}$ that correspond to the *potential presence* of some edges that we wish to track. Therefore, $\overline{X} = \{\overline{p}, \overline{q}, \overline{r}, \overline{s}, \overline{t}\}$ are the provenance tokens allowing us to track the *potential absence* of the same edges. These **provenance tracking assumptions** can be formalized via a provenance-tracking $\mathbb{N}[X, \overline{X}]$-interpretation.

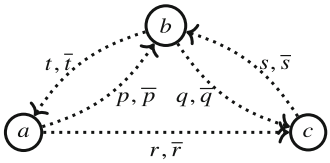


Fig. 2 Provenance tracking assumptions

Define $\pi : \mathrm{Lit}_V(\{E\}) \to X \cup \overline{X} \cup \{0, 1\}$ by

$$\pi(L) = \begin{cases} p & \text{if } L = Eab \\ \overline{p} & \text{if } L = \neg Eab \\ q & \text{if } L = Ebc \\ \overline{q} & \text{if } L = \neg Ebc \\ r & \text{if } L = Eac \\ \overline{r} & \text{if } L = \neg Eac \end{cases} \quad \text{and} \quad \pi(L) = \begin{cases} s & \text{if } L = Ecb \\ \overline{s} & \text{if } L = \neg Ecb \\ t & \text{if } L = Eba \\ \overline{t} & \text{if } L = \neg Eba \\ 0 & \text{for the other positive facts} \\ 1 & \text{for the other negative facts.} \end{cases}$$

For example, $\pi(Eca) = \pi(Ecc) = \ldots = 0$ and $\pi(\neg Eca) = \pi(\neg Ecc) = \ldots = 1$. This particular interpretation does not feature a positive fact annotated with 1 but we could have just as well had $\pi(Eab) = 1$ and $\pi(\neg Eab) = 0$ if we chose to assume that edge without tracking it.

Note that π is not model-defining (in the sense of Definition 3), i.e., it does not correspond to any *single* model. As we shall see, this is not a bug but a feature, as it will allow us to consider, under the given provenance assumptions, multiple models that can satisfy a sentence.

Now we compute the semantics of the sentence $\varphi \equiv \forall x\, \neg\mathsf{dominant}(x)$ from Sect. 4.2, under this interpretation and we obtain

$$\pi[\![\varphi]\!] = (\overline{p} + \overline{r} + t) \cdot (p + \overline{q} + s + \overline{t}) \cdot (1 + q + r + \overline{s}).$$

If we multiply the three sums and apply $p\overline{p} = q\overline{q} = r\overline{r} = s\overline{s} = 0$ we get a polynomial with $48 - 4 - 3 - 3 - 4 = 34$ monomials (the reader shall be spared the pleasure of admiring it).

As in Sect. 4.2, each of these monomials has coefficient 1 and (as shown in Sect. 5.2) each corresponds to a different proof tree of φ from the literals described by the monomial.

For example, the monomial pqt corresponds to a proof tree of φ in which the fact Eba is used to deny the dominance of a, the fact Eab is used to deny the dominance of b, and the fact Ebc is used to deny the dominance of c. Recalling the notations from Sect. 4.2, note that the same monomial is part of the dual polynomial $\beta[\![\varphi]\!]$

and that the same proof tree justifies $G \models \varphi$. Note also that setting $r = s = \overline{p} = \overline{q} = \overline{t} = 0$ in the definition of π gives the definition of β. Doing the same in $\pi[\![\varphi]\!]$ gives

$$(0 + \overline{r} + t) \cdot (p + 0 + 0 + 0) \cdot (1 + q + 0 + \overline{s}) \;=\; (\overline{r} + t) \cdot p \cdot (1 + q + \overline{s}),$$

which is the same as the polynomial $\beta[\![\varphi]\!]$ obtained with the model-defining interpretation β which corresponds to the model G. In this sense, π is a "generalization" of β, or, β can be obtained by *specializing* π. All this will be made precise in full generality in Sect. 5.2 while here we explore two other interesting specializations of π.

One of the monomials in $\pi[\![\varphi]\!]$ is $\overline{pq}$. This means that we can find a specialization of π that is model-defining and that defines, in fact, a model with *no* positive information, namely the digraph with vertices V and no edges. Hence, denoting with $\mathfrak{E}$ this no-edge model, we have $\mathfrak{E} \models \varphi$. How many proof trees verify that $\mathfrak{E} \models \varphi$? The specialization of π that we are after (let's call it β_1) corresponds to setting $p = q = r = s = t = 0$ in π and in $\pi[\![\varphi]\!]$. This gives

$$\beta_1[\![\varphi]\!] \;=\; (\overline{p} + \overline{r}) \cdot (\overline{q} + \overline{t}) \cdot (1 + \overline{s}),$$

which is a polynomial with 8 monomials, each with coefficient 1. It follows that there are 8 distinct proof trees for $\mathfrak{E} \models \varphi$.

One can also figure out that pqt, prt, qst, rst are among the monomials in $\pi[\![\varphi]\!]$. This means that we can find another specialization of π (let's call it β_2) that is also model-defining and that defines a model with *maximum* positive information (allowed by π), namely the digraph with vertices V and edges Eab, Ebc, Eac, Ecb and Eba. Let's denote with $\mathcal{F}$ this all-allowed-edges model (see Fig. 3). How many proof trees verify that $\mathcal{F} \models \varphi$? The specialization β_2 that we look for here corresponds to setting $\overline{p} = \overline{q} = \overline{r} = \overline{s} = \overline{t} = 0$. This gives

$$\beta_2[\![\varphi]\!] \;=\; t \cdot (p + s) \cdot (1 + q + r),$$

which is a polynomial with 6 monomials, each with coefficient 1, hence there are 6 proof trees for this.

Finally, we also wish to consider for this example the provenance of the *negation* of the sentence φ considered above, i.e., the sentence $\neg\varphi$ that says that the digraph *has* a dominant vertex:

$$\neg\varphi \;=\; \neg\forall x\, \neg\mathsf{dominant}(x).$$

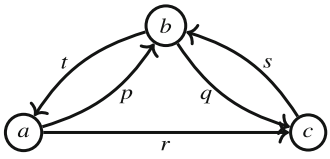


Fig. 3 The model $\mathcal{F}$

Since $\mathsf{dominant}(x) = \forall y \left(x = y \vee (Exy \wedge \neg Eyx)\right)$ is already in NNF, we have $\mathrm{nnf}(\neg\varphi) = \exists x\, \mathsf{dominant}(x)$. We compute the semantics of this sentence under the same interpretation:

$$\pi[\![\neg\varphi]\!] \;=\; pr\bar{t} + \overline{p}q\bar{s}t + 0 \cdot \overline{qr}s \;=\; pr\bar{t} + \overline{p}q\bar{s}t.$$

Thus, under the provenance tracking assumptions we have made, there are only two proof trees for $\neg\varphi$.

The attentive reader may have also noticed a certain "duality" between $\pi[\![\neg\varphi]\!]$ above (specifically, the expression that includes $0 \cdot \overline{qr}s$) and $\pi[\![\varphi]\!]$ given earlier, if we think of $+$ as dual to $\cdot$ and p as dual to $\overline{p}$, etc. Indeed, this duality can be made precise (before dealing with complementary tokens, that is, in $\mathbb{N}[X \cup \overline{X}]$ rather than in $\mathbb{N}[X, \overline{X}]$) and we do that in Sect. 5.2.

5.2 *Properties of Provenance*

The interpretation exhibited in Sect. 5.1 belongs to a class that merits its own definition.

Definition 6 A provenance-tracking interpretation $\pi : \mathsf{Lit}_A(\tau) \to \mathbb{N}[X, \overline{X}]$ is said to be **model-compatible** if for each fact $R\overline{a}$ one of the following (mutually exclusive) three properties holds:

1. $\exists z \in X$ s.t $\pi(R\overline{a}) = z$ and $\pi(\neg R\overline{a}) = \overline{z}$, or
2. $\pi(R\overline{a}) = 0$ and $\pi(\neg R\overline{a}) = 1$, or
3. $\pi(R\overline{a}) = 1$ and $\pi(\neg R\overline{a}) = 0$

As promised, we state a more powerful version of Proposition 11 (which was about provenance-tracking model-defining interpretations).

Theorem 2 *Let $\pi : \mathsf{Lit}_A(\tau) \to \mathbb{N}[X, \overline{X}]$ be a model-compatible interpretation and let ψ be sentence in* FO(τ). *Then, $\pi[\![\psi]\!]$ describes* all *the proof trees that verify φ using premises from among the literals that π maps to provenance tokens or to 1. Specifically, each monomial $m\, x_1^{m_1} \cdots x_k^{m_k}$ corresponds to m distinct proof trees that use m_1 times a literal annotated by x_1, ..., and m_k times a literal annotated by x_k, where $x_1, \ldots, x_k \in X \cup \overline{X}$. In particular, when $\pi[\![\psi]\!] = 0$ no proof tree exists.*

Again, the proof is a direct consequence of Theorem 1. Note that, in this context, an evaluation tree $\mathcal{T}$ for a model-compatible interpretation $\pi : \mathsf{Lit}_A(\tau) \to \mathbb{N}[X, \overline{X}]$ has the valuation

$$\pi(\mathcal{T}) := \prod_{x \in X \cup \overline{X}} x^{\#_x(\mathcal{T})},$$

where $\#_x(\mathcal{T})$ is the number of leaves of $\mathcal{T}$ that π maps to x. Further, $\pi(\mathcal{T}) = 0$ if $\mathcal{T}$ contains leaves mapped to complementary provenance tokens.

Corollary 1 *Let π be a model-compatible interpretation. Then, the sum of the monomial coefficients in $\pi[\![\varphi]\!]$ counts the number of proof trees that verify φ using premises from among the literals that π maps to provenance tokens or to 1. The same count can be obtained from an $\mathbb{N}$-interpretation as $(h \circ \pi)[\![\varphi]\!] \in \mathbb{N}$ where $h : X \cup \overline{X} \cup \{0,1\} \to \mathbb{N}$ is defined by $h(0) = 0$ and $h(p) = h(\overline{p}) = h(1) = 1$.*

A model-compatible interpretation may allow the tracking of both a literal and its negation. Therefore, model-compatible interpretations are not model-defining unless they do not make use of provenance tokens at all (in which case they are essentially canonical truth interpretations). Hence, Proposition 11 is not a simple particular case of Theorem 2. Nonetheless, we shall see how model-defining interpretations can be seen as *specializations* of model-compatible interpretations with respect to models that "agree" (i.e., are *compatible*) with them, as defined below.

Definition 7 Let $\pi : \mathsf{Lit}_A(\tau) \to \mathbb{N}[X, \overline{X}]$ be a model-compatible interpretation and let $\mathfrak{A}$ be a model with universe A. We say that $\mathfrak{A}$ is **compatible** with π if $\mathfrak{A} \models L$ for any literal L such that $\pi(L) = 1$. Further, let $\mathsf{Mod}_\pi := \{\mathfrak{A} \mid \mathfrak{A} \text{ is compatible with } \pi\}$.

For instance, the models shown in Figs. 1 and 3 are compatible with the interpretation defined in Sect. 5.1.

Now we can talk about satisfiability and validity *restricted to the class of models that agree with the provenance tracking assumptions* made by an interpretation.

Corollary 2 (To Theorem 2) *Let $\pi : \mathsf{Lit}_A(\tau) \to \mathbb{N}[X, \overline{X}]$ be a model-compatible interpretation and let φ be a first-order sentence. Then,*

- *φ is Mod_π-satisfiable if, and only if, $\pi[\![\varphi]\!] \neq 0$,*
- *φ is Mod_π-valid if, and only if, $\pi[\![\neg\varphi]\!] = 0$.*

This is not finite satisfiability (shown undecidable by Trakhtenbrot), of course. Even if we map every possible literal to a different provenance token we only decide satisfiability in a model with exactly $|A|$ elements, which is easily in NP (without talking about provenance).

Example 1 With the same (digraph) vocabulary as in Sects. 4.2 and 5.1 consider the sentence

$$\psi := \exists x\, \forall y\, Exy \;\to\; \forall y\, \exists x\, Exy.$$

This is a well-known tautology (holding in all models, not just in finite ones). Obviously, $\mathrm{nnf}(\neg\psi) = \exists x\, \forall y\, Exy \;\wedge\; \exists y\, \forall x\, \neg Exy$. Now consider $V = \{a, b\}$ and a truth-compatible interpretation π that annotates Eab, Eba, Eaa, Ebb with

p, q, r, s respectively, and the corresponding negated facts with $\overline{p}, \overline{q}, \overline{r}, \overline{s}$. Then

$$\pi[\![\neg\psi]\!] \;=\; (pr + qs)(\overline{qr} + \overline{ps}) = 0,$$

verifying that ψ is Mod_π-valid.

From the provenance analysis of (provenance-restricted) validity/satisfiability that is enabled by Corollary 2 we can obtain a provenance analysis of model checking, for each model of a given sentence, as follows.

Definition 8 Let π be model-compatible and let $\mathfrak{A} \in \mathsf{Mod}_\pi$. The **specialization** of π with respect to $\mathfrak{A}$ is the $\mathbb{N}[X, \overline{X}]$-interpretation $\pi|_{\mathfrak{A}} : \mathsf{Lit}_A \to \mathbb{N}[X, \overline{X}]$ defined by

$$\pi|_{\mathfrak{A}}(L) \;=\; \begin{cases} \pi(L) & \text{if } \mathfrak{A} \models L \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\pi|_{\mathfrak{A}}$ is always model-defining and the model it defines is, of course, $\mathfrak{A}$. Note also, that for any sentence φ the dual-indeterminate polynomial $\pi|_{\mathfrak{A}}[\![\varphi]\!]$ is obtained by replacing in $\pi[\![\varphi]\!]$ the tokens $\pi(L)$ with 0 for all literals L such that $\mathfrak{A} \not\models L$.

The model-defining interpretation β in Sect. 4.2 is the specialization with respect to the model G of the model-compatible interpretation π in Sect. 5.1, $\beta = \pi|_G$. Other specializations of π are given in Sect. 5.1. The next corollary implies Proposition 11.

Corollary 3 (To Theorem 2) *Let $\pi : \mathsf{Lit}_A(\tau) \to \mathbb{N}[X, \overline{X}]$ be a model-compatible interpretation, let $\mathfrak{A}$ be a τ-structure that is compatible with π, and let φ be a first-order sentence such that $\mathfrak{A} \models \varphi$ (hence, by Corollary 2, $\pi[\![\varphi]\!] \neq 0$).*

Then, $\pi|_{\mathfrak{A}}[\![\varphi]\!] \neq 0$ and every monomial in $\pi|_{\mathfrak{A}}[\![\varphi]\!]$ also appears in $\pi[\![\varphi]\!]$, with the same coefficient.

Moreover, $\pi|_{\mathfrak{A}}[\![\varphi]\!]$ describes all *the proof trees that verify $\mathfrak{A} \models \varphi$. In particular, the sum of all the monomial coefficients in $\pi|_{\mathfrak{A}}[\![\varphi]\!]$ counts the number of distinct such proof trees (as in Corollary 1, the same count can be obtained from an $\mathbb{N}$-interpretation).*

While $\pi|_{\mathfrak{A}}[\![\varphi]\!]$ analyzes the provenance of checking in a specific model, the more general $\pi[\![\varphi]\!]$ allows for a form *reverse analysis*. Indeed, to each monomial $m\, x_1^{m_1} \cdots x_k^{m_k}$ in $\pi[\![\varphi]\!] \neq 0$ we can associate a model from Mod_π that makes true the literals that are annotated by $x_1, \ldots, x_k$ (and possibly more literals) and, as we have seen, every model $\mathfrak{A} \in \mathsf{Mod}_\pi$ such that $\mathfrak{A} \models \varphi$ can be obtained this way.

Example 2 (Example 1 Cont'd) Let us also compute the provenance of the tautology ψ itself:

$$\pi[\![\psi]\!] = (\overline{p} + \overline{r})(\overline{q} + \overline{s}) + (q + r)(p + s).$$

Here Mod_π consists of all possible structures with universe $\{a, b\}$ and, for any such $\mathfrak{A}$, the model-refinement $\pi|_{\mathfrak{A}}$ sets to 0 exactly one of the two tokens in a complementary pair. No matter how this is done, observe that $\pi|_{\mathfrak{A}}[\![\tau]\!] \neq 0$.

Dualizing Dual-Indeterminate Polynomials This is best defined on *expressions* over the semiring $\mathbb{N}[X \cup \overline{X}]$ and then separately discussing the effect of factoring by $p \cdot \overline{p} = 0$. Dualizing is defined inductively as follows:

$$\begin{array}{rlrl} \mathrm{dual}(e_1 + e_2) &:= e_1 \cdot e_2 & \mathrm{dual}(e_1 \cdot e_2) &:= e_1 + e_2 \\ \mathrm{dual}(p) &:= \overline{p} & \mathrm{dual}(\overline{p}) &:= p \\ \mathrm{dual}(0) &:= 1 & \mathrm{dual}(1) &:= 0 \end{array}$$

Note that dualization is self-inverse.

Proposition 14 *Let $\pi : \mathsf{Lit}_A \to \mathbb{N}[X \cup \overline{X}]$ be a model-compatible interpretation. Then, for any first-order sentence φ we have*

$$\pi[\![\varphi]\!] \;=\; \mathrm{dual}(\pi[\![\neg\varphi]\!])$$

Since $\pi[\![\neg\neg\varphi]\!] = \pi[\![\varphi]\!]$ it follows that also $\pi[\![\neg\varphi]\!] \;=\; \mathrm{dual}(\pi[\![\varphi]\!])$.

5.3 Confidence Maximization

As in Sect. 4.5 we use the Viterbi semiring $\mathbb{V}$ from Sect. 2 interpreting its values as confidence scores. Interestingly, we can reverse analyze the provenance polynomials and use confidence scores to find a model in which confidence is maximized.

In the context of the example in Sect. 5.1, suppose that we have confidence $1/3$ in all the literals that the model-compatible interpretation π maps to a (positive or negative) provenance token. This yields a $\mathbb{V}$-interpretation π' which, by Proposition 2 and Proposition 10, factors as $\pi' = h \circ \pi$ where h is the unique semiring homomorphism $\mathbb{N}[X \cup \overline{X}] \to \mathbb{V}$ that maps all the tokens $p, \ldots, \overline{p}, \ldots$ to $1/3$ (this is perfectly plausible, as confidence is *not* probability).

Now recall from Sect. 5.1 the sentence $\neg\varphi$ (which asserts that there exists a dominant vertex). We have computed $\pi[\![\neg\varphi]\!] \;=\; pr\bar{t} + \overline{p}q\bar{s}t$. Obviously, π' is inconsistent so further applying $h(pr\bar{t} + \overline{p}q\bar{s}t) \;=\; 1/27 + 1/81 \;=\; 4/81$ is not meaningful. However, we know from Corollary 3 that each monomial in $\pi[\![\neg\varphi]\!]$ corresponds to some model of $\neg\varphi$. In this case we have exactly two proof tree choices, corresponding to different models, and they give different confidence to $\neg\varphi$. To maximize confidence we choose the monomial $pr\bar{t}$ therefore a model in which we have an edge Eab, an edge Eac and *no* edge Eba. This will ensure the dominance of vertex a with confidence $1/27$, in other words, $\neg\varphi$ is $1/27$-true in this model. This model is shown in Fig. 4 (the edge Eba is dashed because it is absent but we still wanted to show the confidence 1/3 in this absence). The edges Ebc

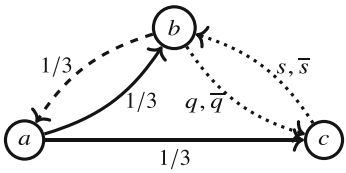


Fig. 4 Maximum confidence model with dominant vertex

and Ecb are dotted because neither their presence nor their absence contradicts the provenance assumptions. We can, in fact, continue with a provenance analysis for these two edges if other properties of the model are of interest.

6 From Model Update to Provenance Update

In this section we indicate a method for updating provenance polynomials corresponding to a model-defining interpretation when the associated model is updated by inserting or deleting facts, without recomputing the provenance polynomial from scratch. One can think of this method as *incremental provenance maintenance*, and relate it to incremental view maintenance [12].

For example, recall from Sect. 4.2 the interpretation β and the structure G that it defines (Fig. 1), and the sentence φ asserting "no dominant vertex". We had computed

$$\beta[\![\varphi]\!] \;=\; (\overline{r} + t) \cdot p \cdot (1 + q + \overline{s}).$$

First suppose that we update G by *deleting* Eab and Ebc. Let us also assume that in the resulting model, we wish to track the presence/absence of the edges unaffected by the update, as well to introduce tokens that track the absence of the two newly deleted edges. This results in the interpretation and the model $\mathcal{H}$ depicted in Fig. 5.

What is the corresponding update on the dual polynomial $\beta[\![\varphi]\!]$? For the provenance polynomials used for positive queries, as in [32], this update is performed by setting $p = q = 0$. However, this would result in the polynomial 0. Of course this cannot be right because $\mathcal{H} \models \varphi$. The deeper reason why it's wrong is that in β we already had the literal $\neg Eab$ interpreted as 0. If we also interpret Eab as $p = 0$ the resulting interpretation does not define any model, much less $\mathcal{H}$, and Proposition 11 does not apply. The same issue arises with setting $q = 0$.

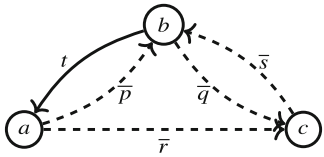


Fig. 5 The model $\mathcal{H}$

Instead, the right way to perform this update takes advantage of the results in Sect. 5.2. We use the model-compatible interpretation π given in Sect. 5.1. Any other model-compatible interpretation that both G and $\mathcal{H}$ are compatible with and that specializes with respect to G to β would do, but π is in an obvious sense the most "economical" such. Recall from Sect. 5.1 that

$$\pi[\![\varphi]\!] \;=\; (\overline{p}+\overline{r}+t)\cdot(p+\overline{q}+s+\overline{t})\cdot(1+q+r+\overline{s}),$$

and therefore

$$\pi|_{\mathcal{H}}[\![\varphi]\!] \;=\; (\overline{p}+\overline{r}+t)\cdot\overline{q}\cdot(1+\overline{s})$$

is the update we desire.

Next, suppose that we update G by *inserting* Eac and Ecb resulting in the model $\mathcal{F}$ in Fig. 3. Then, the update of $\beta[\![\varphi]\!]$ is

$$\pi|_{\mathcal{F}}[\![\varphi]\!] \;=\; t\cdot(p+s)\cdot(1+q+r).$$

To prove that this update method is sound, we describe it more generally. Let $\beta : \mathsf{Lit}_A(\tau) \to \mathbb{N}[X,\overline{X}]$ be a model-defining interpretation which describes the model $\mathfrak{A}_\beta$ and tracks the facts that can potentially be added to or deleted from $\mathfrak{A}_\beta$. More precisely, the set of these facts is $R^+ \cup R^-$ where

$$R^+ = \{\alpha \in \mathsf{Facts}_A(\tau) : \beta(\alpha) = 0 \text{ and } \beta(\neg\alpha) = \overline{x}_\alpha\} \quad \text{and}$$

$$R^- = \{\alpha \in \mathsf{Facts}_A(\tau) : \beta(\alpha) = x_\alpha \text{ and } \beta(\neg\alpha) = 0\}.$$

An update can be any subset $U \subseteq R^+ \cup R^-$ and it updates the model $\mathfrak{A}_\beta$ to a new model $\mathfrak{A}_\beta[U]$, with its associated model defining $\mathbb{N}[X,\overline{X}]$-interpretation β_U.

Given any sentence $\varphi \in$ FO and its provenance polynomial $\beta[\![\varphi]\!]$ our method permits us to compute, for every given update $U \subseteq R^+ \cup R^-$, the updated provenance polynomial $\beta_U[\![\varphi]\!]$, without explicitly computing the entire interpretation β_U.

For this purpose, we consider the general *model-compatible* interpretation π with $\pi(\alpha) = x_\alpha$ and $\pi(\neg\alpha) = \overline{x}_\alpha$ for every $\alpha \in R^+ \cap R^-$ (and which maps all other literals to their truth values). This interpretation is compatible with the class of all updated structures $\mathfrak{A}_\beta[U]$, for $U \subseteq R^+ \cap R^-$. We now compute the provenance polynomial $\pi[\![\varphi]\!]$ for the model-compatible interpretation π and use it to derive the polynomials $\beta_U[\![\varphi]\!]$ for any given U.

Proposition 15 *For every update U the provenance polynomial $\beta_U[\![\varphi]\!]$ can be computed from $\pi[\![\varphi]\!]$ by setting $x_\alpha = 0$ for $\alpha \in ((R^+ \setminus U) \cup (R^- \cap U))$ and $\overline{x}_\alpha = 0$ for $\alpha \in ((R^+ \cap U) \cup (R^- \setminus U))$.*

The proof is an immediate consequence of the fact that the substitutions setting $x_\alpha, \overline{x}_\alpha$ to 0 as described in the proposition are precisely those specializing the model-compatible interpretation π to the model defining interpretation β_U. Indeed they set to 0 precisely the indeterminates associated with the literals that are either already false in $\mathfrak{A}_\beta$ and not changed by U or true in $\mathfrak{A}_\beta$ but changed by U.

7 Explanations and Repairs

In this section we discuss *why-not* provenance questions. These arise as we wish to understand why a certain tuple does *not* appear in the answer to a query, i.e., it is *missing* [34, 35, 38] as well as when we wish to understand why *integrity constraints* fail in a model. A first stab at the material in this section was made in [42]. For a very recent related paper see [7].

In databases, many integrity constraints are first-order expressible and *repairs* for them have been studied in the context of query answering over inconsistent databases (i.e., databases in which integrity constraints fail (see [5] and the many references in there). We will discuss both explanations and repairs for both missing answers and integrity constraint failure as problems that can be approached using dual provenance polynomials.

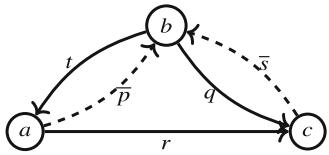
The approach proposed in this paper for dealing with negation in queries allows us to treat such questions in the same way in which the provenance polynomials in [32] were used to answer questions about the presence of tuples in the answer and derive explanations for why-not questions in the same manner as in Sect. 4.4.

We try to explain this method first via examples for missing query answers and failures of integrity constraints, and we will then formulate it in a more general way, as for updates of provenance polynomials in the previous section.

7.1 Missing Query Answers

Consider the model and the model-defining provenance-tracking interpretation in Fig. 6. Let's call this interpretation δ. Following the same visual conventions as

Fig. 6 Model $\mathcal{M}$



with the model-defining, provenance-tracking interpretation β illustrated in Fig. 1 we have:

$$\delta(L) = \begin{cases} 0 & \text{if } L = Eab \\ \overline{p} & \text{if } L = \neg Eab \\ q & \text{if } L = Ebc \\ 0 & \text{if } L = \neg Ebc \\ r & \text{if } L = Eac \\ 0 & \text{if } L = \neg Eac \end{cases} = \begin{cases} 0 & \text{if } L = Ecb \\ \overline{s} & \text{if } L = \neg Ecb \\ t & \text{if } L = Eba \\ 0 & \text{if } L = \neg Eba \\ 0 & \text{for the other positive facts} \\ 1 & \text{for the other negative facts.} \end{cases}$$

We also consider the query $\mathsf{dominant}(x) = \forall y\,(x = y \vee (Exy \wedge \neg Eyx))$. b is an answer for the query. Indeed, the provenance of $\mathsf{dominant}(b)$ under δ is $(0 + t\overline{p})(1 + (0 \cdot 0))(0 + q\overline{s}) = \overline{p}q\overline{s}t$. Now suppose we ask:

Why is a not an answer to this query?

Using Proposition 11 we reason as follows. Since $\mathsf{dominant}(a)$ is not valid in the model $\mathcal{M}$, its provenance under δ must be the 0 dual polynomial. However, its negation, namely (put in negation normal form)

$$\neg\mathsf{dominant}(x) = \exists y(x \neq y \wedge (\neg Exy \vee Eyx)).$$

is valid in $\mathcal{M}$ and, as *explanations* (causes) for the missing answer a, we will take (see Sect. 4.4) the reasons the provenance under δ of $\neg\mathsf{dominant}(a)$ is *not* the 0 polynomial.

Calculating this provenance for the interpretation δ gives

$$0 \cdot (0 + 1) + 1 \cdot (\overline{p} + t) + 1 \cdot (0 + 0) = \overline{p} + t$$

This leads to two alternative explanations (causes) that answer $(*)$:

1. $\overline{p} \neq 0$ (absence of edge Eab)
2. $t \neq 0$ (presence of edge Eba)

In addition to explanations, this methodology can also indicate *repairs*, that is, updates to the model $\mathcal{M}$ that would produce a model that makes $\mathsf{dominant}(a)$ true. This will be achieved by solving the equation

$$\delta[\![\neg\mathsf{dominant}(a)]\!] = 0$$

This means $\overline{p} + t = 0$ and therefore $\overline{p} = t = 0$. We interpret $\overline{p} = 0$ as "insert Eab" and $t = 0$ as "delete Eba".

Consider a model-compatible interpretation that uses the tokens used by δ, namely $\overline{p}, q, r, \overline{s}, t$. As a model-compatible interpretation, it must also use $p, \overline{q}, \overline{r}, s, \overline{t}$. The interpretation π described in Fig. 2 is such an interpretation. We have

$$\pi[\![\mathsf{dominant}(a)]\!] = (1 + 0 \cdot 1)(0 + p \cdot \overline{t})(0 + r \cdot 1) \;=\; p\overline{t}r$$
$$\pi[\![\neg\mathsf{dominant}(a)]\!] = 0 \cdot (1 + 0) + 1 \cdot (\overline{p} + t) + 1 \cdot (\overline{r} + 0) \;=\; \overline{p} + t + \overline{r}$$

We could have started with this model-compatible interpretation and then we would specialize it (as in Definition 8) to the model $\mathcal{M}$ in Fig. 6 obtaining the δ above as $\delta = \pi|_{\mathcal{M}}$.

Now let's denote by $\mathcal{M}^u$ the model $\mathcal{M}$ updated with the repair we obtained: insert Eab and delete Eba. As expected, $\pi|_{\mathcal{M}^u}[\![\neg\mathsf{dominant}(a)]\!] = 0$ and $\pi|_{\mathcal{M}^u}[\![\mathsf{dominant}(a)]\!] = p\overline{t}r$ is the updated provenance of $\mathsf{dominant}(a)$.

7.2 Integrity Constraint Failure

Consider a slightly different model and interpretation.

Consider also the integrity constraint (IC): "AT LEAST ONE VERTEX IS DOMINANT"

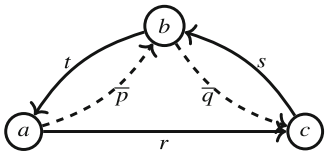
$$\exists x\, \mathsf{dominant}(x) \;=\; \exists x\, \forall y\, (x = y \vee (Exy \wedge \neg Eyx))$$

As before, we ask *"why is the IC failing in C?"*.

Note that the interpretation illustrated in Fig. 7 indicates the facts in C that we are interested in tracking in explanation and willing to involve in repairs. The same facts and the tokens used to label them appear in the model-compatible interpretation π that was used for the missing answers example so we can use it again. We have (note that both these polynomials were calculated in 5.1)

$$\pi[\![\exists x\, \mathsf{dominant}(x)]\!] = pr\overline{t} + \overline{p}q\overline{s}t$$
$$\pi[\![\neg\exists x\, \mathsf{dominant}(x)]\!] = (\overline{p} + \overline{r} + t) \cdot (p + \overline{q} + s + \overline{t}) \cdot (1 + q + r + \overline{s})$$

Fig. 7 Model C



As expected $\pi|_C[\![\exists x\ \mathsf{dominant}(x)]\!] = 0$. We look for explanations in

$$\pi|_C[\![\neg\exists x\ \mathsf{dominant}(x)]\!] = (\overline{p}+t)\cdot(\overline{q}+s)\cdot(1+r)$$
$$= \overline{pq}+\overline{p}s+t\overline{q}+ts+\overline{pq}r+\overline{p}sr+t\overline{q}r+tsr$$

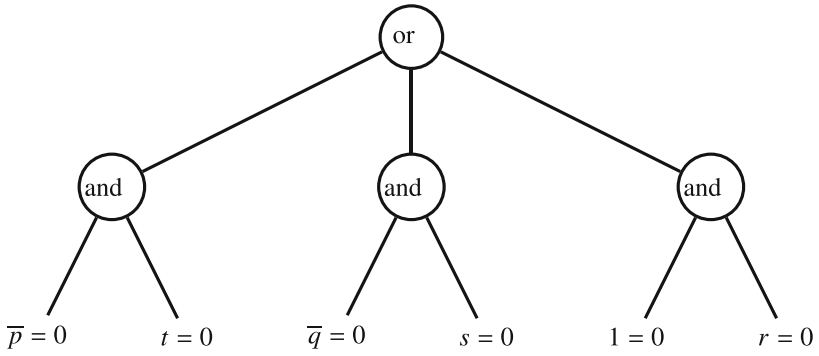
This gives 8 alternative explanations but 4 of them are "redundant". We are left with 4 minimal explanations:

1. $\overline{pq} \neq 0$ (absence of Eab and Ebc)
2. $\overline{p}s \neq 0$ (absence of Eab and presence of Ecb)
3. $t\overline{q} \neq 0$ (presence of Eba and absence of Ebc)
4. $ts \neq 0$ (presence of Eba and presence of Ecb)

In order to determine repairs to the model C that make the IC true we solve the equation

$$(\overline{p}+t)\cdot(\overline{q}+s)\cdot(1+r) \;=\; 0$$

The minimal solutions appear in the following and-or tree:



Each solution corresponds to a different *alternative repair*:

1. $\{\overline{p}=t=0\}$ (insert Eab and delete Eba), or
2. $\{\overline{q}=s=0\}$ (insert Ebc and delete Ecb)

We can now update the provenance of $\exists x\ \mathsf{dominant}(x)$ according to each of the repairs. Let C_1^u be the model obtained by updating C with the first repair. We obtain $\pi|_{C_1^u}[\![\exists x\ \mathsf{dominant}(x)]\!] = pr\bar{t}$. And let C_2^u be the model obtained by updating C with the second repair. We obtain $\pi|_{C_2^u}[\![\exists x \mathsf{dominant}(x)]\!] = \overline{p}q\bar{s}t$.

We can use this updated provenance to choose between the two repairs based on a provenance analysis, for example by cost (using the tropical semiring, $\mathbb{T}$). Toward this we make the following *assumptions:* cost of one insertion: 20, cost of one deletion: 15; cost of pos/neg facts in the model C initially, are $\text{cost}(\overline{p}) = \text{cost}(\overline{q}) = 10$, $\text{cost}(s) = \text{cost}(t) = 5$, $\text{cost}(r) = 10$.

Then, $\text{cost}(pr\bar{t}) = 20+10+15 = 45$ and $\text{cost}(\overline{p}q\bar{s}t) = 10+20+15+5 = 50$. We conclude that the first repair is cheaper.

In general, there can be an exponential number of minimal repairs. When we calculate the dual polynomial associated to a sentence we should leave it in expression form rather than put it in polynomial form because the latter may have exponential size. The expression can stay in polysize. When we solve the equation for repairs, the and-or tree of repairs is also of polynomial size. (All this is data complexity.) Any minimal repair is a subset of a repair represented in the tree.

7.3 Repairs by Provenance Polynomials

In a general sense, the method to compute (minimal) repairs is just a variation of the method computing updated provenance polynomials described in Sect. 6. Assume that $\mathfrak{A}$ is a τ-structure with universe A such that $\mathfrak{A} \not\models \psi$. Assume further that to repair $\mathfrak{A}$ and make it a model of ψ, we have sets R^- and R^+ of facts that we are allowed to remove from or add to $\mathfrak{A}$.

As in Sect. 6 we have a provenance-tracking model-defining interpretation $\beta : \mathsf{Lit}_A(\tau) \to \mathbb{N}[X, \overline{X}]$ which defines the model $\mathfrak{A}_\beta = \mathfrak{A}$ and tracks the facts in $R^+ \cup R^-$ that can potentially be added to or deleted from $\mathfrak{A}$.

A *repair* for $\mathfrak{A}$ and ψ is now an update $R \subseteq R^+ \cup R^-$ such that the modified structure $\mathfrak{A}[R]$ is a model for ψ.

Let X be the set of indeterminates x_α where $\alpha \in R^+ \cup R^-$. We again consider the *model-compatible* interpretation $\pi : \text{Lit}_A(\tau) \to \mathbb{N}[X, \overline{X}]$ such that, for every atom $\alpha \in R^+ \cup R^-$, we have $\pi(\alpha) = x_\alpha$ and $\pi(\neg\alpha) = \overline{x_\alpha}$, and all other literals $\beta = \alpha$ or $\beta = \neg\alpha$ where α is *not* in $R^+ \cup R^-$ (i.e. can not be changed) is simply mapped it to its truth value in $\mathfrak{A}$.

We now consider the provenance polynomial $\pi[\![\psi]\!] \in \mathbb{N}[X, \overline{X}]$ and the subset $X^\pm = \{x_\alpha : \alpha \in R^+\} \cup \{\overline{x_\alpha} : \alpha \in R^-\} \subseteq X \cup \overline{X}$ of the dual indeterminates that are directly associated with the possible changes.

We write $\pi[\![\psi]\!] = m_1 + \cdots + m_k$ as a sum of monomials, and let $v(m)$ be the set of variables from $X \cup \overline{X}$ appearing in m. However, not all these variables are necessarily in $X^\pm$, since $X \cup \overline{X}$ also contains their duals. By examining what combinations of indeterminates from $X^\pm$ occur in the monomials of $\pi[\![\psi]\!]$, we can read off all minimal repairs.

Proposition 16 *For every monomial m of $\pi[\![\psi]\!]$ the set*

$$R_m := \{\alpha \in R^\pm : x_\alpha \in v(m) \cap X^\pm \text{ or } \overline{x}_\alpha \in v(m) \cap X^\pm\}$$

is a repair of $\mathfrak{A}$ for ψ. Conversely, for every repair $R \subseteq R^+ \cup R^-$ of $\mathfrak{A}$ for ψ, there is a monomial m of $\pi[\![\psi]\!]$ such that $R_m \subseteq R$. If R is a minimal repair, then $R_m = R$.

Proof The first claim follows directly from Theorem 2. Indeed, every monomial m in $\pi[\![\psi]\!]$ corresponds to a proof tree for ψ that uses only literals that are true in $\mathfrak{A}[R_m]$ (because such a literal is either already true in $\mathfrak{A}$ or is changed to true by R_m) so $\mathfrak{A}[R_m] \models \psi$. Suppose now that R is any repair of $\mathfrak{A}$ for ψ. As $\mathfrak{A}[R]$ differs from $\mathfrak{A}$ only for literals in $R^{\pm}$ there is a unique assignment $h\colon X \cup \overline{X} \to \mathbb{B}$ such that $h \circ \pi$ is the $\mathbb{B}$-interpretation that describes $\mathfrak{A}[R]$, and therefore $h \circ \pi[\![\psi]\!] = h(\pi[\![\psi]\!]) = 1$. So there must be a monomial $m \in \pi[\![\psi]\!]$ with $h(m) = 1$. For every variable $x_\alpha \in v(m) \cap X^{\pm}$ we have that $h(x_\alpha) = 1$ and hence $\alpha \in R$. For $\overline{x}_\alpha \in v(m) \cap X^{\pm}$ we have $h(\overline{x}_\alpha) = 1$ and thus again $\alpha \in R$ by construction of h. This proves that $R_m \subseteq R$. If R is minimal, we have equality, because otherwise R_m would be a smaller repair. □

We remark that instead of $\mathbb{N}[X, \overline{X}]$ one can use as well simpler semirings such as $\mathbb{S}(X, \overline{X})$ or even PosBool$(X, \overline{X})$, for which the provenance polynomial $\pi[\![\psi]\!]$ is smaller and easier to compute. Indeed, even the polynomial in PosBool$(X, \overline{X})$ suffices to determine the minimal repairs, since these only depend on the sets of variables occurring in some monomial. However, a computation in a more informative semiring may have the advantage that we can compare the different repairs according to costs, clearance levels, confidence scores etc., for which the PosBool-semirings are not sufficient.

An alternative road towards the computation of minimal repairs is based on computing not the provenance polynomial $\pi[\![\psi]\!]$ but, as in the examples discussed above, the provenance polynomial $\pi[\![\neg\psi]\!]$, to explain why $\mathfrak{A} \models \neg\psi$.

Consider the two provenance polynomials $\pi[\![\psi]\!]$ and $\pi[\![\neg\pi]\!]$ and specialise them to $\mathfrak{A}$ by setting x_α to 0, if $\mathfrak{A} \models \neg\alpha$ and $\overline{x}_\alpha = 0$ if $\mathfrak{A} \models \alpha$. For the resulting model-defining interpretation $\pi|_{\mathfrak{A}}$ we obviously have that $\pi|_{\mathfrak{A}}[\![\psi]\!] = 0$ (since $\mathfrak{A} \not\models \psi$).

To compute repairs, we then compute solutions for the equation $\pi|_{\mathfrak{A}}[\![\neg\psi]\!] = 0$. A minimal solution is a minimal set of indeterminates such that setting these to 0 annihilates $\pi|_{\mathfrak{A}}[\![\neg\psi]\!]$. If $\pi[\![\varphi]\!]$, and hence also $\pi|_{\mathfrak{A}}$, are written as a sum of monomials, we can write $\pi|_{\mathfrak{A}}[\![\neg\psi]\!]$ as a product of sums of indeterminates, and compute solutions by setting each of these sums individually to 0. When we have computed such a set Y, we get the repair

$$R_Y := \{\alpha \in R^+ : \overline{x}_\alpha \in Y\} \cup \{\alpha \in R^- : x_\alpha \in Y\}.$$

Indeed, for $\mathfrak{B} = \mathfrak{A}[R_Y]$ the model-defining interpretation $\pi|_{\mathfrak{B}}$ is obtained from π by setting to 0 the indeterminates in Y as well as the variables associated with literals that are false in $\mathfrak{A}$ and are not touched by R_Y. It follows that $\pi|_{\mathfrak{B}}[\![\neg\varphi]\!] = 0$. But since $\pi|_{\mathfrak{B}}$ is model-defining this implies that $\pi|_{\mathfrak{B}}[\![\varphi]\!] \neq 0$ and hence that $\mathfrak{B} \models \varphi$.

8 Other Approaches to Modeling Negation

The semiring semantics for first-order logic outlined in this paper handles negation via transformation to negation normal form and polynomials with dual indeterminates. This has meanwhile emerged as the standard approach for dealing with negation, not just in first-order logic, but also many other logical systems. One of the issues with this approach (but we actually consider it as a feature!) is that negation is not a compositional algebraic operation, contrary to disjunction and conjunction. Indeed, a semiring valuation of $\neg\varphi$ is, in general, not uniquely determined by the valuation of φ, but depends on the *syntax* of φ. Also, equivalences are not always preserved under negation, and the usual logical rules involving negation do not necessarily hold for all semirings. Indeed, consider a semiring $\mathcal{S}$ where addition is idempotent, but multiplication is not, such as the tropical semiring. Then $\varphi \vee \varphi \equiv_{\mathcal{S}} \varphi$, but in general $\pi[\![\neg(\varphi \vee \varphi)]\!] = \pi[\![\mathrm{nnf}(\neg\varphi) \wedge \mathrm{nnf}(\neg\varphi)]\!] = \pi[\![\mathrm{nnf}(\neg\varphi)]\!] \cdot \pi[\![\mathrm{nnf}(\neg\varphi)]\!]$ which is, in general, not the same as $\pi[\![\neg\varphi]\!] = \pi[\![\mathrm{nnf}(\neg\varphi)]\!]$.

However there are several alternative approaches to negation that we want to discuss here.

8.1 Flattening

For any compositional interpretation of (non-atomic) negation in a semiring $\mathcal{S}$, we need a function $f : \mathcal{S} \to \mathcal{S}$ so that, for every $\mathcal{S}$-interpretation π and every (non-atomic) formula φ, we can define the valuation $\pi[\![\neg\varphi]\!] := f(\pi[\![\varphi]\!])$. If we wish to remain consistent with the intuition that 0 stands for *false* and all other values $s \in \mathcal{S}$ correspond to an annotated variant of *true* then f must have the property that $f(s) = 0$ for $s \neq 0$ and $f(0) = t$ for some fixed value $t \in \mathcal{S}$. The most reasonable choice is to take $t = 1$.

Interpreting negation in this way has the consequence that, under the scope of negation, semiring sematics reduces to Boolean semantics. For every non-atomic formula φ and every model-defining $\mathcal{S}$-interpretation π we have that $\pi[\![\neg\varphi]\!] = 1$ if $\mathfrak{A}_\pi \models \neg\varphi$ and $\pi[\![\neg\varphi]\!] = 0$ if $\mathfrak{A}_\pi \models \varphi$. Further although in general $\neg\neg\psi \not\equiv_{\mathcal{S}} \psi$ for atomic formulae ψ, we obviously have that $\neg\neg\neg\psi \equiv_{\mathcal{S}} \neg\psi$ for every formula ψ. We can conclude, that while the use of a flattening function f permits to define a compositional negation operator that avoids the transformation to negation normal form, this does not lead to interesting provenance information once we are under the scope of a negation.

In special cases, for instance for semirings over the real interval [0, 1] such as the fuzzy semiring or the Łukasiewicz semiring, and *if we deviate from the standard intuition*, also other functions can make sense, such as $f(x) := 1 - x$. We shall come back to this in the context of dealing with implication.

A more general approach towards negation is based on the observation that, classically, negation can be obtained from other standard logical operations such

as implication or difference. So we can try to define appropriate semiring semantics for first-order logic with implication and/or difference, and then identify $\neg\psi$ with $\psi \to 0$ or with $1 - \psi$.

8.2 Monus Semirings

Monus is an operation on certain naturally ordered commutative monoids generalising the special cases of the natural numbers, where it is subtraction, truncated to 0, and of the Boolean truth values where it $x \wedge \neg y$. Monus was equationally axiomatized by Bosbach [8] and Amer [1], and it was introduced to semirings for provenance by Geerts and Poggi [20].

We generally restrict attention to naturally ordered semirings $(S, +, \cdot, 0, 1)$, where $s \leq t :\leftrightarrow \exists r(s + r = t)$ is antisymmetric and hence defines a partial order. This condition, and also the monus operation, only depend on the additive monoid $(S, +, 0)$ of the semiring. Natural order does not admit an equational characterization. It is therefore remarkable that with the addition of monus, the resulting structures form an algebraic variety.

Proposition 17 *Let $(S, +, 0)$ be a commutative monoid. For a binary operation $a \dot{-} b$ on S the following characterizations are equivalent*

1. *S is naturally ordered and for any $a, b \in S$, $a \dot{-} b = \min\{c : a \leq b + c\}$.*
2. *S is naturally ordered and for any $a, b, c \in S$, $a \dot{-} b \leq c$ if, and only if, $a \leq b + c$.*
3. *S is naturally ordered and for any $b, c \in S$, $b + c$ is the largest a such that $a \dot{-} b \leq c$.*
4. *The following four equational axioms hold*

   (a) $a \dot{-} a = 0$
   (b) $0 \dot{-} a = 0$
   (c) $a + (b \dot{-} a) = b + (a \dot{-} b)$
   (d) $a \dot{-} (b + c) = (a \dot{-} b) \dot{-} c$

Any idempotent commutative monoid is naturally ordered (in fact, the natural order is the semilattice order).

Proposition 18 *Let $(S, +, 0)$ be an idempotent commutative monoid. For a binary operation $a \dot{-} b$ on S the following characterizations are equivalent*

1. *For any $a, b \in S$, $a \dot{-} b = \min\{c : a \leq b + c\}$.*
2. *For any $a, b, c \in S$, $a \dot{-} b \leq c$ if, and only if, $a \leq b + c$.*
3. *For any $b, c \in S$, $b + c$ is the largest a such that $a \dot{-} b \leq c$.*
4. *The following four equational axioms hold*

   (a) $a \dot{-} a = 0$
   (b) $(a \dot{-} b) + b = a + b$
   (c) $(a \dot{-} b) + a = a$
   (d) $(a + b) \dot{-} c = (a \dot{-} c) + (b \dot{-} c)$

The following naturally ordered commutative monoids have a monus operation (unique by characterization 1):

- $(\mathbb{N}, +, 0)$, where monus is truncated subtraction, i.e. $a \dot{-} b = 0$ if $a \leq b$.
- $(B, \vee, \bot)$ where $(B, \vee, \wedge, \bot, \top, \neg)$ is a Boolean alegebra, and where monus is $a \wedge \neg b$. For the Boolean algebra of subsets of a set, $(2^X, \cup, \cap, \varnothing, X)$, monus is set difference.
- Let $(T, \leq)$ be a totally ordered set with least element (denoted $\bot$). Then $(T, \max, \bot)$ is a commutative monoid that is idempotent, hence naturally ordered, and that has monus:

$$a \dot{-} b = \begin{cases} \bot & \text{when } a \leq b \\ a & \text{when } a > b \end{cases}$$

  So we can define monus in the security semiring, the Viterbi semiring (hence the tropical semiring), the Łukasiewicz semiring, and the fuzzy semiring
- Let $(L, \leq)$ be a complete lattice. L has a least element (notation $\bot$) and binary join (notation $\sqcup$) so that $(L, \sqcup, \bot)$ is an idempotent commutative monoid. When L is distributive this monoid has monus, by the following proposition.

Proposition 19 *Let $(L, \leq)$ be a distributive complete lattice. For any $a, b \in L$ let $z = \inf\{c \mid a \leq b \sqcup c\}$. Then $a \leq b \sqcup z$.*

Indeed, Let $C = \{c \mid a \leq b \sqcup c\}$. Then $b \sqcup z = b \sqcup \inf C = \inf(b \sqcup C)$ and for any $x \in b \sqcup C$ we have $x = b \sqcup c$ for some $c \in C$ so $a \leq x$.

In particular for the distributive complete lattice $(2^X, \subseteq)$ the commutative monoid $(2^X, \cup, \varnothing)$ has monus. It better be the same as the one that emerges from the Boolean algebra structure! Indeed, one can check that for any $A, B \in 2^X$

$$\bigcap_{A \subseteq B \cup C} C \;=\; A \setminus B$$

Moreover, any finite distributive lattice is also a complete distributive lattice, and thus this class of semirings, in particular PosBool(X), has monus.

The monus operation gives a semiring interpretation of the difference operator of relational algebra or, generally for "logical difference": for any two formulae ψ, φ, we can also admit the difference $\psi - \varphi$ (which in Boolean semantics is the same as $\psi \wedge \neg\varphi$) and define $\pi[\![\psi - \varphi]\!] := \pi[\![\psi]\!] \dot{-} \pi[\![\varphi]\!]$. In absorptive semirings, where we have 1 as the maximal value, we can then identify $\neg\psi$ with $1 - \psi$ and define $\pi[\![\neg\psi]\!] := 1 \dot{-} \pi[\![\psi]\!]$. But does this give a "reasonable" interpretation for negation? Recall that on semirings where $a + b = \max(a, b)$, we have that $a \dot{-} b = a$ whenever $a > b$ and $a \dot{-} b = 0$ otherwise. Hence $\pi[\![\neg\psi]\!] = 1$ whenever $\pi[\![\psi]\!] < 1$, and $\pi[\![\neg\psi]\!] = 0$ only in case $\pi[\![\psi]\!] = 1$. Hence this "negation" is purely Boolean, and is just the negation of "ψ has the maximal truth value"; in particular this is not consistent with the standard intuition that only 0 stands for *false*, and all other values for an annotated variant of *true*. Further, the monus operator,

and hence also negation, only depends on the additive monoid of the semiring, and does not take into account the interpretation of multiplication at all. Finally, the correspondence of the difference $\psi - \varphi$ with $\psi \wedge \neg\varphi$ also breaks down on certain semirings. Consider for instance min-cost computations in the tropical semiring $\mathbb{T} = (\mathbb{R}_+^\infty, \min, +, \infty, 0)$ and assume that ψ and φ have the same cost $\pi[\![\psi]\!] = \pi[\![\varphi]\!]$ which is neither 0 nor ∞. Then the cost of $\psi - \varphi$ is ∞ whereas the cost of $\psi \wedge \neg\varphi$ is $\pi[\![\psi]\!] + \pi[\![\neg\varphi]\!] = \psi[\![\psi]\!] + 0 = \pi[\![\psi]\!]$.

For all these reasons we conclude that while monus semirings provide an interpretation of (relational) difference that may be interesting in certain settings (such as for bag semantics on the natural semiring), the monus operation does not lead to a convincing valuation for negation.

8.3 Negation Via Implication

A further possibility to deal with negation is based on the idea to define an appropriate interpretation for implications $\psi \to \varphi$ and to identify then $\neg\psi$ with $\psi \to \bot$. But what is an appropriate interpretation of $\psi \to \varphi$ and what properties of negation does it imply? Implication corresponds to the semantic notion that ψ *entails* φ, and should satisfy classical properties such as modus ponens and the deduction theorem. All this works best in the context of absorptive semirings, preferably with additional properties that permit to define natural infinitary addition and multiplication operations, based on infima and suprema [11], so that we have natural semiring valuations also for infinite sets of formulae.

Definition 9 An *infinitary absorptive semiring* is based on an absorptive semiring $\mathcal{S}$ which satisfies the additional properties that

- the natural order $(\mathcal{S}, \leq)$ is a complete lattice.
- $\mathcal{S}$ is (fully) continuous: for every non-empty chain $C \subseteq S$, the supremum $\bigsqcup C$ and the infimum $\bigsqcap C$ are compatible with addition and multiplication, i.e.

$$s \circ \bigsqcup C = \bigsqcup(s \circ C) \quad \text{and} \quad s \circ \bigsqcap C = \bigsqcap(s \circ C),$$

where $(s \circ C) := \{s \circ c : c \in C\}$ for every $s \in S$ and $\circ \in \{+, \cdot\}$.

As a consequence, we can define natural infinitary addition and multiplication operations in $\mathcal{S}$ (and thus semiring provenance for first-order logic also on infinite universes) by taking suprema of finite subsums and infima of finite subproducts:

$$\sum_{i \in I} s_i := \bigsqcup_{\substack{I_0 \subseteq I \\ I_0 \text{ finite}}} \Big(\sum_{i \in I_0} s_i \Big) \quad \text{and} \quad \prod_{i \in I} s_i := \bigsqcap_{\substack{I_0 \subseteq I \\ I_0 \text{ finite}}} \Big(\prod_{i \in I_0} s_i \Big).$$

Since addition is idempotent in absorptive semirings, the infinitary addition is in fact the same as the supremum: $\sum_{i\in I} s_i = \bigsqcup_{i\in I} s_i$. However, unless multiplication is also idempotent (so that the semiring is a lattice semiring), infinitary products need not coincide with infima.

Let $\pi : \mathrm{Lit}_A(\tau) \to \mathcal{S}$ be a model-defining $\mathcal{S}$-interpretation for a finite or infinite universe A and a relational vocabulary τ. For sets of sentences $\Phi \subseteq \mathrm{FO}$, we put $\pi[\![\Phi]\!] := \prod_{\varphi\in\Phi} \pi[\![\varphi]\!]$.

Definition 10 Let $\Phi \subseteq \mathrm{FO}$ and $\psi \in \mathrm{FO}$, and let $\mathcal{S}$ be an infinitary absorptive semiring. We write

1. $\Phi \equiv_{\mathcal{S}} 0$ if $\pi[\![\Phi]\!] = 0$ for every model-defining $\mathcal{S}$-interpretation π;
2. $\Phi \models_{\mathcal{S}} \psi$ if $\pi[\![\Phi]\!] \leq \pi[\![\psi]\!]$ for every model-defining $\mathcal{S}$-interpretation π.

Notice that once we have committed ourselves to a specific semiring $\mathcal{S}$, semiring semantics over $\mathcal{S}$ is a particular case of a multivalued logic, and we may look for connections to interpretations of implication and negation in that area.

The Gödel Implication We now add to an absorptive semiring $\mathcal{S}$ the new binary operation $\to$ mapping every pair (s, t) to

$$s \to t := \begin{cases} 1 & \text{if } s \leq t \\ t & \text{otherwise.} \end{cases}$$

Further we use the abbreviation $\sim s := s \to 0$ which map 0 to 1 and all other elements of $\mathcal{S}$ to 0.

We can now extend propositional logic PL and first-order logic FO with semiring semantics to $\mathrm{PL}^{\to}$ and $\mathrm{FO}^{\to}$ which permit to build formulae $\psi \to \varphi$ such that, for every $\mathcal{S}$-interpretation π we have that

$$\pi[\![\psi \to \varphi]\!] := \pi[\![\psi]\!] \to \pi[\![\varphi]\!] = \begin{cases} 1 & \text{if } \psi \models \varphi \\ \pi[\![\varphi]\!] & \text{otherwise.} \end{cases}$$

Of course there are other possibilities to interpret the connective $\to$ and thus other ways to extend the semiring semantics of PL and FO by an interpretation for implication. We argue that the one we discuss here, which has originally been proposed by Gödel, has specifically interesting properties. If the underlying semiring $\mathcal{S}$ is a min-max semiring then this interpretation of $\to$ makes $\mathrm{FO}^{\to}$ a Gödel logic.

Definition 11 Let $\mathcal{S}$ be an infinitary absorptive semiring with implication. We say that the (semantic) deduction theorem (DT) holds for $\mathcal{S}$, i.e. if for all $\Phi \subseteq \mathrm{FO}$ and $\psi, \varphi \in \mathrm{FO}$

$$\Phi, \psi \models_{\mathcal{S}} \varphi \quad \Leftrightarrow \quad \Phi \models_{\mathcal{S}} \psi \to \varphi.$$

Note that (DT) implies modus ponens: $\psi, \psi \to \varphi \models_{\mathcal{S}} \varphi$. Informally, the deduction theorem expresses that the semantics of $\to$ captures entailment in a sound and complete way. For $\Phi = \varnothing$ this is trivially true and just says that an implication $\psi \to \varphi$ is a strong tautology in $\mathcal{S}$ (i.e. evaluates to 1 under all $\mathcal{S}$-interpretations) if, and only if, $\psi \models_{\mathcal{S}} \varphi$.

But under a non-empty set Φ of hypothesis, the situation is more complicated. The soundness part of (DT), i.e. $\Phi \models_{\mathcal{S}} \psi \to \varphi \;\Rightarrow\; \Phi, \psi \models_{\mathcal{S}} \varphi$ holds for every infinitary absorptive semiring. Indeed, if $\pi[\![\psi]\!] \leq \pi[\![\varphi]\!]$ then also $\pi[\![\Phi, \psi]\!] \leq \pi[\![\varphi]\!]$ and otherwise $\pi[\![\psi \to \varphi]\!] = \pi[\![\varphi]\!]$ so $\Phi \models_{\mathcal{S}} \psi \to \varphi$ implies that $\pi[\![\Phi, \psi]\!] \leq \pi[\![\Phi]\!] \leq \pi[\![\varphi]\!] = \pi[\![\psi \to \varphi]\!]$. However, the completeness part of (DT) is only true for min-max semirings.

Proposition 20 *The deduction theorem holds for an infinitary absorptive semiring $\mathcal{S}$ if, and only if, $\mathcal{S}$ is a min-max semiring, i.e. if, and only if,* $\mathrm{FO}^{\to}$ *with semiring semantics given by $\mathcal{S}$ is a Gödel logic.*

Proof It is well-known that (DT) holds for Gödel logics. To see this it only remains to show that $\Phi, \psi \models_{\mathcal{S}} \varphi \;\Rightarrow\; \Phi \models_{\mathcal{S}} \psi \to \varphi$. Assume that the left side holds and consider any $\mathcal{S}$-interpretation π. If $\pi[\![\psi]\!] \leq \psi[\![\varphi]\!]$ then $\pi[\![\psi \to \varphi]\!] = 1$ so $\pi[\![\Phi]\!] \leq \pi[\![\psi \to \varphi]\!]$ holds trivially. Otherwise $\pi[\![\psi]\!] > \psi[\![\varphi]\!]$ so $\pi[\![\Phi, \psi]\!] = \min(\pi[\![\Phi]\!], \pi[\![\psi]\!]) \leq \varphi[\![\varphi]\!]$ implies that $\pi[\![\Phi]\!] \leq \pi[\![\varphi]\!] = \pi[\![\psi \to \varphi]\!]$, so $\Phi \models_{\mathcal{S}} \psi \to \varphi$.

To prove the converse it suffices to prove that (DT) fails if $\mathcal{S}$ is not multiplicatively idempotent or if $\mathcal{S}$ is not linearly ordered by the natural order. Indeed an absorptive and multiplicatively idempotent semiring is necessarily a lattice semiring and if the order is linear, it is a min-max semiring. We first show that (DT) fails, even for propositional formulae, if the natural order on $\mathcal{S}$ is not a linear order, i.e. if there exist incomparable element $s, t \in \mathcal{S}$. For propositional variables x, y, let $\Phi := \{x\}$, $\psi := y$ and $\varphi := (x \wedge y)$. Obviously $\Phi, \psi \models_{\mathcal{S}} \varphi$. However, for the $\mathcal{S}$-interpretation $\pi : x \mapsto s, y \mapsto t$ we have that $\pi[\![\Phi]\!] = \pi(x) = s$, but since $\pi[\![\psi]\!] = s \not\leq st = \pi[\![\varphi]\!]$, we have that $\pi[\![\psi \to \varphi]\!] = \pi[\![\varphi]\!] = st$, so $\Phi \not\models_{\mathcal{S}} \psi \to \varphi$. Assume next that $\mathcal{S}$ is not fully idempotent, i.e $s \cdot s < s$ for some $s \in \mathcal{S}$. Again it suffices to argue propositionally, this time with a single variable x. Let $\Phi = \{x\}$, $\psi := x$ and $\varphi = (x \wedge x)$. Obviously $\Phi, \psi \models_{\mathcal{S}} \varphi$, but setting $\pi(x) := s$, we have $\pi[\![\Phi]\!] = \pi[\![\psi]\!] = s$, but since $s > s \cdot s = \pi[\![\varphi]\!]$, we have that $\pi[\![\psi \to \varphi]\!] = \pi[\![\varphi]\!] = s \cdot s$, so $\Phi \not\models_{\mathcal{S}} \psi \to \varphi$. □

A More General Definition for Implication Proposition 20 raises the question, whether there are alternative interpretations for $\to$ that satisfy (DT) on more general classes of semirings. We first observe that the Gödel implication is the only interpretation of $\to$ on min-max semirings that is sound for entailment and satisfies (DT).

Proposition 21 *Let $\mathcal{S}$ be a min-max semiring, and let $s \multimap t$ be an operation on $\mathcal{S}$ such that the induced interpretation of $\psi \multimap \varphi$ satisfies the following properties:*

- *If $\pi[\![\psi]\!] \leq \pi[\![\varphi]\!]$, then $\pi[\![\psi \multimap \varphi]\!] = 1$;*
- *(DT) holds for $\multimap$ on $\mathcal{S}$, i.e. $\Phi, \psi \models_{\mathcal{S}} \varphi \Leftrightarrow \Phi \models_{\mathcal{S}} \psi \multimap \varphi$.*

Then $\multimap$ is the Gödel implication, i.e. $(s \multimap t) = (s \to t)$.

Proof If $\pi[\![\psi]\!] \leq \pi[\![\varphi]\!]$ then $\pi[\![\psi \multimap \varphi]\!] = \pi[\![\psi \to \varphi]\!] = 1$. So assume that $\pi[\![\psi]\!] > \pi[\![\varphi]\!]$. We have to prove that $\pi[\![\psi \multimap \varphi]\!] = \pi[\![\varphi]\!]$. Since $\varphi, \psi \models_{\mathcal{S}} \varphi$ it follows by (DT) that $\varphi \models_{\mathcal{S}} \psi \multimap \varphi$, hence $\pi[\![\varphi]\!] \leq \pi[\![\psi \multimap \varphi]\!]$. From $\psi \multimap \varphi \models_{\mathcal{S}} \psi \multimap \varphi$ we get by (DT) that $\psi \multimap \varphi, \psi \models_{\mathcal{S}} \varphi$. Hence $\min(\pi[\![\psi \multimap \varphi]\!]\pi[\![\psi]\!]) \leq \pi[\![\varphi]\!]$. But since $\pi[\![\psi]\!] > \varphi[\![\varphi]\!]$ it follows that $\pi[\![\psi \multimap \varphi]\!] \leq \pi[\![\varphi]\!]$. □

We thus look for an more general interpretation of implication, which coincides with the Gödel implication on min-max semirings, but not necessarily on other semirings. What could be reasonable values r for $s \to t$? Since from ψ and $\psi \to \varphi$ we want to able to infer φ, we should have that $s \cdot (s \to t) \leq t$. On the other side, the larger we define the value of $s \to t$ the more powerful the reasoning with implications becomes. Thus, the following definition might be reasonable. Set

$$s \to t := \sup\{r : r \cdot s \leq t\}$$

- On min-max semirings this coincides with the Gödel implication.
- On the tropical semiring $\mathbb{T} = (\mathbb{R}_+^\infty, \min, +, \infty, 0)$, we have that $s \to t = \min(0, t - s)$ (recall that the natural order on $\mathbb{T}$ is the inverse of the usual order on the reals). The tropical semiring is used for minimal-cost computations. Suppose that the costs for establishing ψ and φ are s and t, respectively, and that you have already payed s for ψ. Now you want to establish $\psi \to \varphi$ to infer φ. If $s \geq t$ you have to pay nothing (implicitely you have already established φ), otherwise you pay $t - s$.
- On the Lukasiewicz semiring $\mathbb{L} = ([0, 1]_{\mathbb{R}}, \max, \odot, 0, 1)$, where multiplication is given by $s \odot t = \max(s + t - 1, 0)$, we have that $s \to t$ is the Lukasiewicz implication $s \to t := \min(1 - s + t, 1)$.
- In the polynomial semirings PosBool$[X]$ we can view each monomial as a set $Y \subseteq X$, multiplication of monomials corresponds to their union, and the empty set is 1. The natural order on monomials is defined by $Y \leq Z \Leftrightarrow Y \supseteq Z$, in which case we say that Z *absorbs* Y. For monomials Y, Z we thus have $Y \to Z = Z \backslash Y$. Each expression $p \in \text{PosBool}[X]$ is an antichain of monomials, i.e. a collection of subsets $Y \subseteq X$ none of which is a subset of another one. Addition $p + q$ is defined by taking all monomials in p and q and then deleting those that are absorbed by another one, keeping only the minimal subsets $Y \in p \cup q$. The natural order on such expressions is that $p \leq q$ if every $Y \in p$ is a superset of some $Z \in q$; indeed it then follows that $p + q = q$ because every monomial $Y \in p \cup q$ is absorbed by some $Z \in q$. It follows that $p \to q$ is the collection

of the $\subseteq$-minimal sets U such that for each $Y \in p$, $U \supseteq Z \setminus Y$ for some $Z \in q$. How difficult is this to compute?

Proposition 22 *With this implication, the deduction theorem holds for every infinitary absorptive semiring.*

Proof Assume $\Phi \models_{\mathcal{S}} \psi \to \varphi$. Then $\pi[\![\Phi]\!] \leq \sup\{r : r \cdot \pi[\![\psi]\!] \leq \pi[\![\varphi]\!]\}$ for each $\mathcal{S}$-interpretation π, so by monotonicity $\pi[\![\Phi]\!] \cdot \pi[\![\psi]\!] \leq \pi[\![\varphi]\!]$. This proves that $\Phi, \psi \models_{\mathcal{S}} \varphi$.

Conversely assume $\Phi, \psi \models_{\mathcal{S}} \varphi$. For every $\mathcal{S}$-interpretation π we have $\pi[\![\Phi]\!] \cdot \pi[\![\psi]\!] \leq \pi[\![\varphi]\!]$, which implies that $\pi[\![\Phi]\!] \leq \sup\{r : r \cdot \pi[\![\psi]\!] \leq \pi[\![\varphi]\!]\} = \pi[\![\psi \to \varphi]\!]$. This proves that $\Phi \models_{\mathcal{S}} \psi \to \varphi$. □

Negation We can use the more general interpretation of $s \to t$ defined above and put $\sim s := (s \to 0)$. Accordingly we have a non-atomic negation operator on $FO^{\to}$ with

$$\pi[\![\sim \psi]\!] := \pi[\![\psi \to 0]\!] = \sup\{r : r \cdot \pi[\![\psi]\!] = 0\}.$$

- On every semiring $\mathcal{S}$ without divisors of 0 we have

$$\pi[\![\sim \psi]\!] = \begin{cases} 1 & \text{if } \pi[\![\psi]\!] = 0 \\ 0 & \text{otherwise.} \end{cases}$$

- On the Łukasiewicz semiring $\mathbb{L}$ with the Łukasiewicz implication $s \to t := \min(1 - s + t, 1)$, we have that $\pi[\![\sim \psi]\!] = 1 - \pi[\![\psi]\!]$.

8.4 Double Valuations

Given that, under the standard approach, negation is not a compositional algebraic operation, it makes sense to associate which each logical statement not one, but two semiring values, one for the statement itself, and one for its negation. Given a semiring interpretation $\pi : \mathrm{Lit}_A(\tau) \to \mathcal{S}$ we associate with each first-order statement $\psi \in \mathrm{FO}(\tau)$, which is not necessarily in negation normal form, a pair of values $\pi^*[\![\psi]\!] = (\psi^+, \psi^-) \in S \times S$ according to the following inductive rules:

$$\begin{aligned}
\pi^*[\![\alpha]\!] &:= (\pi(\alpha), \pi(\neg\alpha)) \quad \text{if } \alpha \text{ is an atomic formula} \\
\pi^*[\![\psi \vee \varphi]\!] &:= (\psi^+ + \varphi^+, \psi^- \cdot \varphi^-) \\
\pi^*[\![\psi \wedge \varphi]\!] &:= (\psi^+ \cdot \varphi^+, \psi^- + \varphi^-) \\
\pi^*[\![\exists x \psi]\!] &:= \Big(\sum\nolimits_{a \in A} \psi(a)^+, \prod\nolimits_{a \in A} \psi(a)^- \Big)
\end{aligned}$$

$$\pi^*[\![\forall x\psi]\!] := \left(\prod\nolimits_{a\in A} \psi(a)^+, \sum\nolimits_{a\in A} \psi(a)^-\right)$$
$$\pi^*[\![\neg\psi]\!] := (\psi^-, \psi^+)$$

A simple induction shows that for every $\psi \in \mathrm{FO}(\tau)$,

$$\pi^*[\![\psi]\!] = (\pi[\![\mathrm{nnf}(\psi)]\!], \pi[\![\mathrm{nnf}(\neg\psi)]\!]).$$

For model-defining semiring interpretations into a positive semiring this is not particularly interesting since we will always have that $\pi^*[\![\psi]\!] = (s, 0)$ or $\pi^*[\![\psi]\!] = (0, s)$ for some $s \in S \setminus \{0\}$. However, for instance for model-compatible interpretations into $\mathbb{N}[X, \overline{X}]$ a valuation gives insights into both proof trees and refutation trees, or equivalently winning strategies for both players in the associated model-checking game, depending on which literals are set to true.

This approach has so far not been explored any further, but we believe that it deserves investigation.

9 Further Developments in Semiring Semantics

The extension of semiring provenance from positive database queries to a semiring semantics of full first-order logic, as proposed in [25] has motivated further work in several directions, beyond the topics addressed above.

9.1 Semiring Semantics of Fixed-Point Logic

Semiring semantics has meanwhile be defined not only first-order logic, but for many other logics as well. The most interesting challenges arise for fixed-point logics, such as LFP or the modal μ-calculus. For one of the most simple fixed-point formalisms, namely the query language *datalog*, a provenance analysis has already been provided in the original paper [32], and has later been extended in [16]. Due to the need of unbounded least fixed-point iterations in the evaluation of Datalog queries, the underlying semirings have to satisfy the additional property of being ω-continuous, which essentially means that ascending chains (with respect to the natural order) have a supremum that is compatible with the semiring operations. By Kleene's Fixed-Point Theorem, systems of polynomial equations then have least fixed-point solutions that can be computed by induction, reaching the fixed-point after at most ω stages. Most of the common application semirings are ω-continuous, or can easily be extended to one that is so, but the general ω-continuous provenance semiring over X is no longer a semiring of polynomials but the semiring of formal power series over X (consisting of potentially infinite sums of monomials), denoted

$\mathbb{N}^\infty[\![X]\!]$, with coefficients in $\mathbb{N}^\infty := \mathbb{N} \cup \{\infty\}$. As above, provenance valuations $\pi[\![\psi]\!] \in \mathbb{N}^\infty[\![X]\!]$ give precise information about the possible proof trees for a Datalog query. Even though the databases are assumed to be finite there may be infinitely many proof trees, but each of them can use each atomic fact only a finite number of times, so the proof trees are still finite.

Our approach of extending polynomial semiring by dual indeterminates readily extends to semirings of formal power series, which gives us the semirings $\mathbb{N}^\infty[\![X, \overline{X}]\!]$ of dual-indeterminate power series [26]. These are the general provenance semirings for semipositive Datalog (with negated input predicates) and, much more generally, also for posLFP, the fragment of LFP that consists of formulae in negation normal form such that all its fixed-point operators are least fixed-points. This is a powerful fixed-point calculus, that plays an important role in finite model theory and captures all polynomial-time computable properties of ordered finite structures [27].

Nevertheless, for the general objective of a provenance analysis of fixed-point calculi, the restriction to (positively used) least fixed points is not really satisfactory. The transformation from a fixed-point formula with arbitrary interleavings of least and greatest fixed points into one in posLFP is, contrary to transformations into negation normal form, not a simple syntactic translation. It goes through the Stage Comparison Theorem [27, 39] and can make a formula much longer and more complicated. Further, such transformations are not available for important fixed-point formalisms such as the modal μ-calculus, stratified Datalog, transitive closure logics, and even simple temporal languages such as CTL. So the question arises what kind of semirings are adequate for a meaningful and informative provenance analysis of unrestricted fixed-point logics, with arbitrary interleavings of least and greatest fixed points. This has been studied in [15].

Rather than just ω-continuity one needs semirings that are *fully continuous* which means that every chain has not only a supremum, but also an infimum, and these are compatible with the algebraic operations of the semiring. For an informative provenance semantics, there is a second important condition that is connected with the symmetry, or duality, between least and greatest fixed points. Indeed, in the Boolean setting, a greatest fixed point of a monotone operator is the complement of the least fixed of the dual operator (which is also monotone). It is this duality that permits to push negations through to the atoms and work with formulae in negation normal form. To have a similar kind of symmetry in the provenance setting, it is required that the semirings are *absorptive* so that multiplication is decreasing, i.e., $a \cdot b \leq b$ for all a, b. In particular, the powers of an element form a decreasing chain. It has been shown in [15] that absorptive and fully continuous semirings guarantee a well-defined and informative provenance semantics for arbitrary fixed-point formulae.

The most general absorptive and fully continuous semirings are the semirings $\mathbb{S}^\infty[X]$ of *generalized absorptive polynomials*. Informally such a polynomial is a sum of monomials, with possibly infinite exponents, that are maximal with respect to absorption. For instance a monomial $x^2y^\infty z$ occurring in a provenance value $\pi[\![\psi]\!]$ indicates a model-checking proof that uses the atom labelled by x twice, the

atom labelled by y an infinite number of times, and the atom labelled by z once. This monomial absorbs all those that have larger exponents for all variables, such as for instance $x^3y^\infty z^\infty u$, but not, say, $x^\infty y^3$. Absorptive polynomials thus describe shortest model-checking proofs. Absorption has the further pleasant consequence that generalized formal power series collapse to generalized polynomials. Indeed we can view a polynomial or formal power series as an antichain of monomials (wrt. absorption order), and one can show [15, 26] that all these antichains are finite.

As for semirings of polynomials, we can also in this case construct quotient semirings $\mathbb{S}^\infty[X, \overline{X}]$ of dual indeterminate generalized polynomials to treat positive and negative atomic information appropriately, and these semirings do indeed have universality properties that make them the most general semirings for LFP [15]. An algorithmic analysis for computing least and greatest fixed point in absorptive semirings has been given in [40].

9.2 Strategy Analysis of Games

Semiring provenance for logics is intimately related to semiring valuations of two-player games based on a correspondence that goes both ways. On one side, evaluation problems for logical formulae can be cast as the problem whether the verifying player has a winning strategy in an associated model checking game. On the other side the winning region of a player (i.e., the set of positions from which she has a winning strategy) can be defined by a formula in an appropriate logic (often in LFP). By computing valuations of game positions and strategies in an appropriate semiring, one can obtain detailed information about the available strategies of a player, far beyond the fact, that a player will win or lose.

The mathematical basis of a strategy analysis via semiring valuations are *Sum-of-Strategies Theorems*. The ingredients of such theorems are:

- A valuation of game positions and/or moves by elements of a semiring $\mathcal{S}$. In an acyclic game, which admits only finite plays, this is done by a simple backwards induction; in more complicated games such as Büchi or parity games, this can be defined by a valuation of a formula which states that there is a winning strategy from the given position.
- An appropriate class of strategies for the game.
- A valuation of these strategies by elements of the semiring $\mathcal{S}$; normally this is the product of the valuations of the moves or positions that occur in the strategy.

The Sum-of-Strategies Theorem then says that the semiring valuation of any position in the game coincides with the sum of the valuations of the available strategies from that position.

The simplest instance of this concerns the evaluation of a first-order sentence on a finite structure or more generally, on a semiring interpretation. A winning strategy in the associated model checking game is precisely the same as a proof tree, and the Sum-of-Strategies Theorem for first-order model checking games is

just a different way to state the Sum-of-Proof-Trees Theorem for FO (Theorem 1 in Sect. 3.5). More complicated instances concern acyclic reachability games [26] and the model checking games for LFP [15].

A specific case study for evaluating the power of this approach has been done in [28] for Büchi games. These are games with a winning condition requiring that some good position is seen infinitely often during the play. Büchi games have a number of practical applications, but they are also of interest because they are among the simplest games where any formula win(v) saying that v is a winning position requires a genuine nesting of a least fixed point inside a greatest fixed point. The appropriate class of strategies in this case are the *absorption-dominant* strategies (strategies that win with minimal effort), and the Sum-of-Strategies Theorem states that for any position v in a Büchi game, the valuation of the LFP-formula win(v) in an absorptive, fully-continuous semiring coincides with the sum of the valuations of all absorption-dominant winning strategies from v. From such a valuation on can derive not only whether a player wins from v, but also the number and shapes of all absorption-dominant (as also all positional) winning strategies. Further one can determine whether a player still wins if certain moves are forbidden, or must be used only finitely often. Finally, such valuations also can be used to repair a game, i.e. to find minimal changes of a game that cannot be won into one that admits a winning strategy, by techniques that are similar to the ones in Sect. 7.

9.3 The Model Theory of Semiring Semantics

The development of semiring semantics for various logics, and specifically for full first-order logic, raises the question to what extent classical techniques and results of logic extend to semiring semantics, and how this depends on the algebraic properties of the underlying semiring. In a general research programme that explores such questions, the following topics have been investigated so far.

9.3.1 Elementary Equivalence Versus Isomorphism

It is a rather obvious logical fact that every finite structure (with a finite vocabulary) can be axiomatised, up to isomorphism, by a first-order sentence. In particular, two finite τ-structures $\mathfrak{A}$ and $\mathfrak{B}$ are isomorphic if, and only if, they are elementarily equivalent, in short $\mathfrak{A} \equiv \mathfrak{B}$, which means that they cannot be distinguished by any first-order sentence. Is this also the case for semiring interpretations? Notice that standard notions such as isomorphism and elementary equivalence generalise in a natural way from τ-structures to semiring interpretations, which raises, for any given semiring $\mathcal{S}$, the following questions.

1. Are elementary equivalent finite $\mathcal{S}$-interpretations always isomorphic?
2. Is every finite $\mathcal{S}$-interpretation π_A first-order axiomatisable, i.e. is there is a set of axioms $\Phi_A \subseteq \mathrm{FO}$ such that whenever $\pi_B[\![\varphi]\!] = \pi_A[\![\varphi]\!]$ for all $\varphi \in \Phi_A$, then $\pi_B \cong \pi_A$?
3. Does every finite $\mathcal{S}$-interpretation admit an axiomatisation by a *finite* set of axioms?
4. Can every finite $\mathcal{S}$-interpretation be axiomatised by a single first-order sentence?

Clearly, the first two questions are equivalent, and a positive answer to the third question implies also positive ones to the first two. The converse is not necessarily true, because a first-order axiomatisation of a finite semiring interpretation might require an infinite collection of sentences, and, contrary to the Boolean case, it is a priori also not clear that an axiomatisation by a finite set of sentences implies an axiomatisation by a single sentence, because from the value of a conjunction we cannot necessary infer the values of its components.

It has been shown in [24] that the answers to these questions strongly depend on the chosen semiring. There are in fact rather simple semirings, such as min-max semirings with at least three elements, for which one can construct examples of non-isomorphic interpretations which are, however, elementarily equivalent. Since the standard method for establishing elementary equivalence is not generally available in semiring semantics (see Sect. 9.3.4), new methods based on separating sets of semiring homomorphisms had to be developed for this purpose. Elementarily equivalent but non-isomorphic semiring interpretations also exist for provenance semirings, such as $\mathbb{S}(X)$, $\mathbb{B}[X]$ and $\mathrm{Why}(X)$. On the other side, there are semirings, such as the Viterbi semiring $\mathbb{V}$, the tropical semiring $\mathbb{T}$, the natural semiring $\mathbb{N}$ and the universal polynomial semiring $\mathbb{N}[X]$, for which any finite interpretation is first-order axiomatisable, so that elementary equivalence does indeed imply isomorphism. At least for $\mathbb{V}$ and $\mathbb{T}$, finite axiomatisations are always possible, but not axiomatisations by a single sentence, so there exist semirings where the answers to questions (3) and (4) are different.

9.3.2 0-1 Laws

The classical 0-1 law for first-order logic, due to Glebskii et al. [23] and Fagin [18], says that the probabilities that a relational first-order sentence is true in a random finite structure converge exponentially fast to either 0 or 1, as the size of the structures grows to infinity. Informally speaking, on random finite structures, every first-order sentence is either almost surely false or almost surely true. Random semiring interpretations, induced by a probability distribution on the non-zero elements of a semiring, generalise random structures, and the question arises whether also the 0-1 law generalise to semiring semantics. This has been studied in [29] with the following results.

On many different semirings $\mathcal{S}$ there indeed is a 0-1 law, saying that with probabilities converging to 1 exponentially fast, the valuation $\pi[\![\psi]\!]$ of a first-order

sentence ψ almost surely concentrates on one specific value $s \in \mathcal{S}$. This induced a partition of $\mathrm{FO}(\tau)$ into classes $(\Phi_s)_{s\in\mathcal{S}}$ such that sentences in Φ_s evaluate almost surely to s. On finite lattice semirings, this partition collapses to just three classes Φ_0, Φ_1, and Φ_ε, of sentences that respectively, almost surely evaluate to 0, 1, and to the smallest value $\varepsilon \neq 0$. For all other values $s \in S$ we have that $\Phi_s = \varnothing$. The problem of computing the almost sure valuation of a first-order sentence on finite lattice semirings is PSPACE-complete.

The methods to prove such results combine on the one hand techniques that are adapted from traditional studies of logic on random structures, such as extension properties of atomic types, and on the other side specific ideas of semiring semantics, such as a specific variant of provenance tracking polynomials.

A semiring where the analysis is somewhat different is the *natural semiring* $(\mathbb{N}, +, \cdot, 0, 1)$. The 0-1 law still holds for the natural semiring, but the proof relies on more general ∞-expressions instead of polynomials and there are rather trivial constructions showing that every number $j \in \mathbb{N}$ appears as a possible almost sure valuation.

9.3.3 Locality

Locality is a fundamental property of first-order logic and an important limitation of its expressive power. Informally, this means that the truth of a first-order formula $\psi(\overline{x})$ in a given structure only depends on a neighbourhood of bounded radius around $\overline{x}$, and on the existence of a bounded number of local substructures. Consequently, first-order logic cannot express global properties such as connectivity or acyclicity of graphs. Two fundamental theorems that make this precise are *Hanf's locality theorem* and *Gaifman's normal form theorem*. In a nutshell, Hanf's theorem gives a criterion for the m-equivalence (i.e. indistinguishability by sentences of quantifier rank up to m) of two structures based on the number of local substructures of any given isomorphism type, while Gaifman's theorem states that every first-order formula is equivalent to a Boolean combination of local formulae and basic local sentences; this has many model-theoretic and algorithmic consequences. The question whether such locality theorems also hold in semiring semantics has been studied in [6]. It is shown that Hanf's theorem generalises to all semirings where both operations are idempotent, but fails for many other semirings. For formulae with free variables, Gaifman's theorem does not generalise beyond the Boolean semiring, and also for sentences, it fails in some important semirings such as the natural semiring and the tropical semiring. The main result, however, is a constructive proof of the existence of Gaifman normal forms for min-max and lattice semirings. In fact, this proof also implies a stronger version of Gaifman's classical theorem in Boolean semantics, saying that every sentence has a Gaifman normal form which does not add negations.

9.3.4 Ehrenfeucht–Fraïssé Games

To prove elementary equivalence (and equivalence up to a fixed quantifier rank) of relational structures a standard method (in particular in finite model theory) is provided by Ehrenfeucht–Fraïssé games or, equivalently, back-and-forth systems of local isomorphisms. But while Ehrenfeucht–Fraïssé games are sound and complete for logical equivalences in classical semantics, and thus on the Boolean semiring, this is in general not the case for other semirings. A detailed analysis of the soundness and completeness of model comparison games on specific semirings, not just for classical Ehrenfeucht–Fraïssé games but also for other variants based on bijections or counting, is provided in [10]. It turns out that m-move Ehrenfeucht–Fraïssé games are sound (but in general not complete) for m-equivalence on fully idempotent semirings, whereas m-move bijection games are sound on all semirings. Ehrenfeucht–Fraïssé games without a fixed restriction on the number of moves are sound for elementary equivalence on a number of further important semirings, but completeness only holds in rare cases. Based on the results in [24] that there exist certain rather simple semiring interpretations that are locally very different and can be separated even in a one-move game, but which can be proved to be elementarily equivalent via separating sets of homomorphisms, a new kind of games, called *homomorphism games* has been developed in [10], which provide a sound and complete method for logical equivalences on finite lattice semirings.

References

1. Amer, K.: Equationally complete classes of commutative monoids with monus. Alg. Universalis **18**, 129–131 (1984)
2. Amsterdamer, Y., Davidson, S., Deutch, D., Milo, T., Stoyanovich, J., Tannen, V.: Putting lipstick on pig: enabling database-style workflow provenance. Proc. VLDB **5**(4), 346–357 (2011)
3. Amsterdamer, Y., Deutch, D., Tannen, V.: On the limitations of provenance for queries with difference. In: 3rd Workshop on the Theory and Practice of Provenance, TaPP'11 (2011). See also arXiv:1105.2255
4. Amsterdamer, Y., Deutch, D., Tannen, V.: Provenance for aggregate queries. In: Principles of Database Systems, PODS, pp. 153–164 (2011). See also arXiv:1101.1110
5. Bertossi, L.: Database repairs and consistent query answering: origins and further developments. In: Principles of Database Systems, PODS 2019, pp. 48–58. ACM (2019). Available from: https://doi.org/10.1145/3294052.3322190
6. Bizière, C., Grädel, E., Naaf, M.: Locality theorems in semiring semantics. In: Proceedings of MFCS 2023 (2023). Full version: arXiv 2303.12627
7. Bogaerts, B., Jakubowski, M., Van den Bussche, J.: Postulates for provenance: instance-based provenance for first-order logic. In: Companion of the 43rd Symposium on Principles of Database Systems. PODS (to appear, 2024). https://doi.org/10.1145/3651596
8. Bosbach, B.: Komplementäre Halbgruppen. Math. Ann. **161**, 279–295 (1965)
9. Bourgaux, C., Ozaki, A., Peñaloza, R., Predoiu, L.: Provenance for the description logic ELHr. In: Proceedings of IJCAI 2020, pp. 1862–1869 (2020). https://doi.org/10.24963/ijcai.2020/258

10. Brinke, S., Grädel, E., Mrkonjić, L.: Ehrenfeucht–Fraïssé games in semiring semantics. In: Proceedings of CSL 2024 (2024). Full version: arXiv 2308.04910
11. Brinke, S., Grädel, E., Mrkonjić, L., Naaf, M.: Semiring provenance in the infinite. In: Tannen's Festschrift, vol. 119. Open Access Series in Informatics (2024). Available from: http://logic.rwth-aachen.de/pub/BrinkeGraedelMrkonjicNaaf24.pdf
12. Buneman, P., Clemons, E.: Efficiently monitoring relational databases. ACM Trans. Database Syst. **4**(3), 368–382 (1979). Available from: https://doi.org/10.1145/320083.320099
13. Dannert, K., Grädel, E.: Provenance analysis: a perspective for description logics? In: Lutz, C., et al. (ed.) Description Logic, Theory Combination, and All That, vol. 11560. Lecture Notes in Computer Science, pp. 266–285. Springer (2019). https://doi.org/10.1007/978-3-030-22102-7_12
14. Dannert, K., Grädel, E.: Semiring provenance for guarded logics. In: Hajnal Andréka and István Németi on Unity of Science: From Computing to Relativity Theory Through Algebraic Logic. Outstanding Contributions to Logic, pp. 53–79. Springer (2020). https://doi.org/10.1007/978-3-030-64187-0_3
15. Dannert, K., Grädel, E., Naaf, M., Tannen, V.: Semiring provenance for fixed-point logic. In: Baier, C., Goubault-Larrecq, J. (eds.) 29th EACSL Annual Conference on Computer Science Logic (CSL 2021), vol. 183. Leibniz International Proceedings in Informatics (LIPIcs), pp. 17:1–17:22. Dagstuhl (2021). https://doi.org/10.4230/LIPIcs.CSL.2021.17
16. Deutch, D., Milo, T., Roy, S., Tannen, V.: Circuits for datalog provenance. In: Proc. 17th International Conference on Database Theory ICDT, pp. 201–212. OpenProceedings.org (2014). https://doi.org/10.5441/002/icdt.2014.22
17. Deutch, D., Moskovitch, Y., Tannen, V.: Provenance-based analysis of data-centric processes. VLDB J. **24**(4), 583–607 (2015)
18. Fagin, R.: Probabilities on finite models. J. Symb. Logic **41**, 50–58 (1976). https://doi.org/10.1017/S0022481200051756
19. Foster, J., Green, T., Tannen, V.: Annotated XML: queries and provenance. In: Proceedings of PODS 2008, pp. 271–280 (2008). https://doi.org/10.1145/1376916.1376954
20. Geerts, F., Poggi, A.: On database query languages for K-relations. J. Appl. Logic **8**(2), 173–185 (2010). https://doi.org/10.1016/j.jal.2009.09.001
21. Geerts, F., Unger, T., Karvounarakis, G., Fundulaki, I., Christophides, V.: Algebraic structures for capturing the provenance of SPARQL queries. J. ACM **63**(1), 7:1–7:63 (2016). https://doi.org/10.1145/2810037
22. Glavic, B.: Data provenance. Found. Trends Databases **9**(3–4), 209–441 (2021). https://doi.org/10.1561/1900000068
23. Glebskii, Y., Kogan, D., Liogon'kii, M., Talanov, V.: Range and degree of realizability of formulas in the restricted predicate calculus. Kibernetika **2**, 17–28 (1969). https://doi.org/10.1007/BF01071084
24. Grädel, E., Mrkonjić, L.: Elementary equivalence versus isomorphism in semiring semantics. In: 48th International Colloquium on Automata, Languages, and Programming (ICALP 2021), Dagstuhl, vol. 198, pp. 133:1–133:20 (2021). https://doi.org/10.4230/LIPIcs.ICALP.2021.133
25. Grädel, E., Tannen, V.: Semiring provenance for first-order model checking. arXiv:1712.01980 [cs.LO] (2017). Available from: https://arxiv.org/abs/1712.01980, arXiv:1712.01980
26. Grädel, E., Tannen, V.: Provenance analysis for logic and games. Moscow J. Comb. Number Theory **9**(3), 203–228 (2020). Preprint available at https://arxiv.org/abs/1907.08470. https://doi.org/10.2140/moscow.2020.9.203
27. Grädel, E., Kolaitis, P.G., Libkin, L., Marx, M., Spencer, J., Vardi, M.Y., Venema, Y., Weinstein, S.: Finite Model Theory and Its Applications. Texts in Theoretical Computer Science. An EATCS Series. Springer (2007). https://doi.org/10.1007/3-540-68804-8
28. Grädel, E., Lücking, N., Naaf, M.: Semiring provenance for Büchi games: strategy analysis with absorptive polynomials. arXiv:2106.12892 [cs.LO] (2021). Available from: https://arxiv.org/abs/2106.12892

29. Grädel, E., Helal, H., Naaf, M., Wilke, R.: Zero-one laws and almost sure valuations of first-order logic in semiring semantics. In: Baier, C., Fisman, D. (eds.) LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, August 2–5, 2022, pp. 41:1–41:12. ACM (2022) (full version of this paper). Available from: https://arxiv.org/abs/2203.03425, https://doi.org/10.1145/3531130.3533358
30. Green, T.: Containment of conjunctive queries on annotated relations. Theory Comput. Syst. **49**(2), 429–459 (2011). https://doi.org/10.1007/s00224-011-9327-6
31. Green, T., Karvounarakis, G., Ives, Z., Tannen, V.: Update exchange with mappings and provenance. In: Proceedings of the 33rd International Conference on Very Large Data Bases, University of Vienna, September 23–27, 2007, pp. 675–686 (2007)
32. Green, T., Karvounarakis, G., Tannen, V.: Provenance semirings. In: Principles of Database Systems PODS, pp. 31–40. ACM (2007). https://doi.org/10.1145/1265530.1265535
33. Green, T., Ives, Z., Tannen, V.: Reconcilable differences. Theory Comput. Syst. **49**, 460–488 (2011). https://doi.org/10.1145/1514894.1514920
34. Herschel, M., Hernández, M.: Explaining missing answers to SPJUA queries. Proc. VLDB Endow. **3**(1), 185–196 (2010). Available from: http://www.vldb.org/pvldb/vldb2010/pvldb_vol3/R16.pdf, https://doi.org/10.14778/1920841.1920869
35. Herschel, M., Hernández, M., Tan, W.: Artemis: a system for analyzing missing answers. Proc. VLDB Endow. **2**(2), 1550–1553 (2009). Available from: http://www.vldb.org/pvldb/vol2/vldb09-1004.pdf, https://doi.org/10.14778/1687553.1687588
36. Ives, Z., Green, T., Karvounarakis, G., Taylor, N., Tannen, V., Talukdar, P., Jacob, M., Pereira, F.: The ORCHESTRA collaborative data sharing system. SIGMOD Rec. **37**(3), 26–32 (2008)
37. Karvounarakis, G., Ives, Z., Tannen, V.: Querying data provenance. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, June 6–10, 2010, pp. 951–962 (2010)
38. Meliou, A., Gatterbauer, W., Moore, K., Suciu, D.: WHY so? or WHY no? functional causality for explaining query answers. In: VLDB Workshop on Management of Uncertain Data (MUD 2010), vol. WP10-04. CTIT Workshop Proceedings Series, pp. 3–17 (2010)
39. Moschovakis, Y.: Elementary Induction on Abstract Structures. North Holland, Amsterdam (1974)
40. Naaf, M.: Computing least and greatest fixed points in absorptive semirings. arXiv:2106.00399 [cs.LO] (2021). Available from: https://arxiv.org/abs/2106.00399
41. Tannen, V.: Provenance propagation in complex queries. In: In Search of Elegance in the Theory and Practice of Computation - Essays Dedicated to Peter Buneman, pp. 483–493 (2013). https://doi.org/10.1007/978-3-642-41660-6_26
42. Xu, J., Zhang, W., Alawini, A., Tannen, V.: Provenance analysis for missing answers and integrity repairs. IEEE Data Eng. Bull. **41**(1), 39–50 (2018). Available from: http://sites.computer.org/debull/A18mar/p39.pdf

Reversify Any Sequential Algorithm

Yuri Gurevich

Darn the wheel of the world! Why must it continually turn over? Where is the reverse gear?

— Jack London

Abstract To reversify an arbitrary sequential algorithm A, we gently instrument A with bookkeeping machinery. The result is a step-for-step reversible algorithm that mimics A step-for-step and stops exactly when A does.

Without loss of generality, we presume that algorithm A is presented as an abstract state machine that is behaviorally identical to A. The existence of such representation has been proven theoretically, and the practicality of such representation has been amply demonstrated.

1 Introduction

In 1973, Charles Bennett posited that an"irreversible computer can always be made reversible" [2, p. 525]. To this end, he showed how to transform any one-tape Turing machine M that computes a function $F(x)$, into a reversible three-tape Turing machine M^R computing the function $x \mapsto (x, F(x))$. First, M^R emulates the computation of M on x, saving enough information to ensure step-for-step reversibility. If and when the output is computed, the emulation phase ends, and M^R proceeds to erase all saved information with the exception of the input.

Partially supported by the US Army Research Office under W911NF-20-1-0297.

Y. Gurevich (✉)
University of Michigan, Ann Arbor, MI, USA
e-mail: gurevich@umich.edu

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_22

Bennett's construction shows that, in principle, every sequential algorithm, is reversifiable.[1] In practice, you don't want to compile your algorithms to a one-tape Turing machine M and then execute the three-tape Turing machine M^R.

It had been discussed in the programming community, in particular by Edsger Dijkstra [7, pp. 351–354] and David Gries [9, pp. 265–274], which programs are reversible, but Bennett's reversification idea was either unknown to or neglected by programming experts.

The progress was led by physicists. They reversified Boolean circuits and other computation models. "We have shown", wrote Edward Fredkin and Tommaso Toffoli [8, p. 252], "that abstract systems having universal computing capabilities can be constructed from simple primitives which are invertible". The interest in reversible computations and especially in reversible circuit computations soared with the advent of quantum computing. This is related to the fact that pure (involving no measurements) quantum computations are reversible. There are books on reversible computations [1, 6, 16, 17]. The International Conference on Reversible Computation will have its 13th meeting in 2021 [18]

In this paper, we use sequential abstract state machines, in short sequential ASMs, to address the problem of practical reversification of arbitrary sequential algorithms. Why ASMs? Let us explain.

ASMs were introduced to faithfully simulate arbitrary algorithms on their natural abstraction levels [10]. One instructive early result was the formalization of the C programming language [13].

In [11], an ambitious ASM thesis was formulated: For every algorithm A, there is an ASM B that is behaviorally equivalent to A. If A is sequential, then B has the same initial states as A and the same state transition function. In [12], we axiomatized sequential algorithms and proved the ASM thesis for them.[2] Thus, semantically, sequential algorithms are sequential ASMs. In the meantime, substantial evidence has been accumulated to support the practicality of faithful ASM modeling. Some of it is found in the 2003 book [5].

The main result of the present paper is a simple construction, for every sequential ASM A, of a reversible sequential ASM B that step-for-step simulates A and stops when A does. B does exactly what A does plus some bookkeeping. If A uses some input and output variables and computes some function, then B uses the same input and output variables and computes the same function.

[1] We attempt to give a new useful meaning to the word reversify. To reversify an algorithm means to transform it into a reversible form (rather than to formulate it anew in verse, which is the current dictionary meaning of reversify).

[2] Later these results were generalized to other species of algorithms, e.g. to synchronous parallel algorithms [3] and interactive algorithms [4].

2 Preliminaries

The purpose of this section is to make the current paper self-contained.

2.1 Sequential Algorithms

By sequential algorithms we mean algorithms as the term was understood before modern computer science generalized the notion of algorithm in various directions, which happened in the final decades of the twentieth century. In this connection, sequential algorithms are also called classical.

While the term “sequential algorithm" is short and convenient, it also is too laconic. Some explication is in order. “Algorithms,” said Andrei Kolmogorov in a 1953 talk [15], “compute in steps of bounded complexity.” Let’s look more closely at the two aspects mentioned by Kolmogorov. One aspect is computing in steps, one step after another. Kolmogorov didn’t say “one step after another.” He didn’t have to. That was understood at the time.

The other aspect is a somewhat vague constraint: the bounded complexity of any one step of the algorithm. We prefer a related constraint, arguably a version of Kolmogorov’s constraint: the bounded resources of any one step of the algorithm. The bounded resources constraint, still informal, seems to us clearer and more suitable. It might have been Kolmogorov’s intention all along. We do not know exactly what Kolmogorov said during that talk.[3]

To summarize, sequential algorithms can be characterized informally as transition systems that compute in bounded-resources steps, one step after another.

In our axiomatization of sequential algorithms [12], the bounded resources constraint gives rise to the crucial bounded-exploration axiom. It is also used to justify that a sequential algorithm doesn’t hang forever within a step; time is a bounded resource.

In the following subsections, we recall some basic notions of mathematical logic in the form appropriate to our purposes.

2.2 Vocabularies

A *vocabulary* is a finite collection of function symbols where each symbol f is endowed with some metadata according to the following clauses (V1)–(V4). We interleave the four clauses with auxiliary definitions and explanations.

[3] Vladimir Uspensky, who chaired the Logic Department of Moscow State University after Kolmogorov’s death, admitted to me that the abstract [15] of Kolmogorov’s talk for the Moscow Mathematical Society was written by him (Uspensky) after many unsuccessful attempts to squeeze an abstract from Kolmogorov.

(V1) Each symbol f is assigned a natural number, the *arity* of f. ◁

Define *terms* (or *expressions*) by induction. If f is an r-ary symbol and $t_1, \ldots, t_r$ are terms, then $f(t_1, \ldots, t_r)$ is a term. (The case $r = 0$ is the basis of induction.)

(V2) Some symbols f are marked as *relational*. ◁

Clauses (V1) and (V2) are standard in logic, except that, traditionally, relations are viewed as separate category, not as special functions.

(V3) f may be marked as *dynamic*; if not then f is called *static*. Nullary static symbols are called *constants*; nullary dynamic symbols are called *variables*. ◁

Clause (V3) is related to our use of structures as states of algorithms. The intention is that, during computation, only dynamic functions may be assigned new values. We say that a term is *static* if it involves only static functions.

We presume that every vocabulary contains the following *obligatory* symbols which are all static.

1. Constants $\top$ and $\bot$ (read "true" and "false"), unary `Bool`, and the standard propositional connectives. All these symbols are relational.
2. Constant 0, and unary `Num`, `increment`, and `decrement`. Of these four symbols, only `Num` is relational.
3. Constant `nil` (called `undef` in [12] and other early papers) and the (binary) equality sign =. Of these two symbols, only the equality sign is relational.

(V4) Every dynamic symbol f is assigned a static term, the *default term* of f. If f is relational then so is its default term. ◁

Clauses (V2) and (V4) constitute rudimentary typing which is sufficient for our purposes in this paper. As a rule, the default term for any relational symbol is $\bot$. If a variable v is supposed to take numerical values, then typically the default term for v would be 0, but it could be 1. This concludes the definition of vocabularies.

If Υ and Υ' are vocabularies, we write $\Upsilon \subseteq \Upsilon'$, and we say that Υ is *included* in Υ' and that Υ' *includes* or *extends* Υ, if every Υ symbol belongs to Υ' and has the same metadata in Υ'.

2.3 Structures

A *structure* X of vocabulary Υ is a nonempty set $|X|$, the *universe* or *base set* of X, together with interpretations of the function symbols in Υ. The vocabulary Υ may be denoted $\mathrm{Voc}(X)$.

An r-ary function symbol f is interpreted as a function $f : |X|^r \to |X|$ and is called a *basic function* of X. If f is nullary then f is just a name of an element of (the universe of) X. If f is dynamic and d is the default term for f, then the value (denoted by) d is the *default value* of f.

If f is relational, then the elements $\top$ are $\bot$ are the only possible values of f. If $f(\bar{x}) = \top$, we say that f is true (or holds) at $\bar{x}$; otherwise we say that f is false (or fails) at $\bar{x}$. If f, g are relations of the same arity r, then f, g are *equivalent* in X if their values at every r-tuple of elements of X are the same.

Any basic relation f is the characteristic function of the set $\{x : f(x) = \top\}$. It is often convenient to treat f as that set. We will do that in Sect. 5.

Remark 1 (Names and Denotations) Syntactic objects often denote semantical objects. For example, vocabulary symbols denote basic functions. Different conventions may be used for disambiguation, e.g. a basic function may be denoted f_X. We will use no disambiguation convention in this paper. It should be clear from the context whether a symbol means a syntactic or semantic object. ◁

The equality sign has its usual meaning. `Bool` comprises (the values of) $\top$, $\bot$ which, together with the propositional connectives, form a two-element Boolean algebra.

Given a structure X, the *value* $\mathcal{V}_X\big(f(t_1, \ldots, t_r)\big)$ of a $\mathrm{Voc}(X)$ term $f(t_1, \ldots, t_r)$ in X is defined by induction:

$$\mathcal{V}_X(f(t_1, \ldots, t_r)) = f(\mathcal{V}_X(t_1), \ldots, \mathcal{V}_X(t_r)). \tag{1}$$

Again, the case $r = 0$ is the base of induction.

Instead of $\texttt{increment}(x)$, we will write $x + 1$. `Num` comprises the values of terms $0, 0 + 1, (0 + 1) + 1, \ldots$ which are all distinct. These values are denoted $0, 1, 2, \ldots$ respectively; we call them the *natural numbers* of structure X, and we say that these values are *numerical*. `decrement` is interpreted as expected as well. Instead of $\texttt{decrement}(x)$, we write $x - 1$. $\texttt{decrement}(0) = \texttt{nil}$. The value of `nil` is neither Boolean nor numerical.

Remark 2 (Totality) In accordance with Sect. 2.1, all basic functions are total. In applications, various error values may arise, in particular *timeout*. But, for our purposes in this paper (as in [12]), an error value is just another value.

A *location* in a structure X is a pair $\ell = (f, \bar{x})$ where f is a dynamic symbol in $\mathrm{Voc}(X)$ of some arity r and $\bar{x}$ is an r-tuple of elements of X. The value $f(\bar{x})$ is the *content* of location ℓ.

An *update of location* $\ell = (f, \bar{x})$ is a pair (ℓ, y), also denoted $(\ell \leftarrowtail y)$, where y is an element of X; if f is relational then y is Boolean. To *execute* an update $(\ell \leftarrowtail y)$ in X, replace the current content $\mathcal{V}_X(f(\bar{x}))$ of ℓ with y, i.e., set $f_X(\bar{x})$ to y. An update $(\ell \leftarrowtail y)$ of location $\ell = (f, \bar{x})$ is *trivial* if $y = f(\bar{x})$.

An *update of structure* X is an update of any location in X. A set Δ of updates of X is *contradictory* if it contains updates $(\ell \leftarrowtail y_1)$ and $(\ell \leftarrowtail y_2)$ with distinct y_1, y_2; otherwise Δ is *consistent*.

2.4 Sequential Abstract State Machines

Fix a vocabulary Υ and restrict attention to function symbols in Υ and terms over Υ.

Definition 1 (Syntax of Rules) *Rules* over vocabulary Υ are defined by induction.

1. An *assignment rule* or simply *assignment* has the form

$$f(t_1, \ldots, t_r) := t_0 \tag{2}$$

   where f, the *head* of the assignment, is dynamic, $r = \text{Arity}(f)$, and $t_0, \ldots, t_r$ are terms. If f is relational, then the head function of t_0 is relational. The assignment (2) may be called an f assignment.
2. A *conditional rule* has the form

$$\texttt{if } \beta \texttt{ then } R_1 \texttt{ else } R_2 \tag{3}$$

   where β is a Boolean-valued term and R_1, R_2 are Υ rules.
3. A *parallel rule* has the form

$$R_1 \quad \| \quad R_2 \quad \| \quad \cdots \quad \| \quad R_k \tag{4}$$

   where k is a natural number and $R_1, \ldots, R_k$ are Υ rules. In case $k = 0$, we write `Skip`. ◁

Definition 2 (Semantics of Rules) Fix an Υ structure X. Every Υ rule R generates a finite set Δ of updates in X. R *fails* in X if Δ is contradictory; otherwise R *succeeds* in X. To *fire* (or *execute*) rule R that succeeds in structure X means to execute all Δ updates in X.

1. An assignment $f(t_1, \ldots, t_r) := t_0$ generates a single update $(\ell, \mathcal{V}_X(t_0))$ where $\ell = \big(f, (\mathcal{V}_X(t_1), \ldots, \mathcal{V}_X(t_r))\big)$.
2. A conditional rule $\texttt{if } \beta \texttt{ then } R_1 \texttt{ else } R_2$ works exactly as R_1, if $\beta = \top$ in X, and exactly as R_2 otherwise.
3. A parallel rule $R_1 \quad \| \quad R_2 \quad \| \quad \cdots \quad \| \quad R_k$ generates the union of the update sets generated by rules $R_1, \ldots, R_k$ in X. ◁

Definition 3 A *sequential ASM* A is given by the following three components.

1. A vocabulary Υ, denoted $\text{Voc}(A)$.
2. A nonempty collection of $\text{Voc}(A)$ structures, closed under isomorphisms. These are the *initial states* of A. ◁
3. A $\text{Voc}(A)$ rule, called the *program* of A and denoted $\text{Prog}(A)$.

As we mentioned in Sect. 1, every sequential algorithm A is behaviorally identical to some sequential ASM B; they have the same initial states and the same state-transition function.

In the rest of the paper, by default, all ASMs are sequential.

Consider an ASM A. A $\mathrm{Voc}(A)$ structure X is *terminal* for A if $\mathrm{Prog}(A)$ produces no updates (not even trivial updates[4]) in X. A *partial computation* of A is a finite sequence $X_0, X_1, \ldots, X_n$ of Υ structures where

- X_0 is an initial state of A,
- every X_{i+1} is obtained by executing $\mathrm{Prog}(A)$ in X_i, and
- no structure in the sequence, with a possible exception of X_n, is terminal.

If X_n is terminal, then the partial computation is *terminating*. A *(reachable) state* of A is an $\mathrm{Voc}(A)$ structure that occurs in some partial computation of A.

A Boolean expression γ is a *green light* for an ASM A if it holds in the nonterminal states of A and fails in the terminal states.

Lemma 1 *Every ASM has a green light.*

Proof By induction on rule R, we construct a green light γ_R for any ASM with program R. If R is an assignment, set $\gamma_R = \top$. If R is the parallel composition of rules R_i, set $\gamma_R = \bigvee_i \gamma_{R_i}$. If $R = \texttt{if } \beta \texttt{ then } R_1 \texttt{ else } R_2$, set $\gamma_R = \big(\beta \wedge \gamma_{R_1}\big) \vee \big(\neg\beta \wedge \gamma_{R_2}\big)$. □

In examples and applications, typically, such conditions are easily available. Think of terminal states of finite automata or of halting control states of Turing machines. In a while-loop program, the while condition is a green light.

3 Reducts and Expansions

In mathematical logic, a structure X is a *reduct* of a structure Y if $\mathrm{Voc}(X) \subseteq \mathrm{Voc}(Y)$, the two structures have the same universe, and every $\mathrm{Voc}(X)$ symbol f has the same interpretations in X and in Y. If X is a reduct of Y, then Y is an *expansion* of X. For example, the field of real numbers expands the additive group of real numbers.

We say that an expansion Y of a structure X is *uninformative* if the additional basic functions of Y (which are not basic functions of X) are dynamic and take only their default values in Y. (The default values are defined in Sect. 2.3.) Clearly, X has a unique uninformative expansion to $\mathrm{Voc}(B)$.

Definition 4 An ASM B is a *faithful expansion* of an ASM A if the following conditions hold.

[4] In applications, trivial updates of A may mean something for its environment.

(E1) $\mathrm{Voc}(A) \subseteq \mathrm{Voc}(B)$. The symbols in $\mathrm{Voc}(A)$ are the *principal* symbols of $\mathrm{Voc}(B)$, and their interpretations in $\mathrm{Voc}(B)$ structures are *principal* basic functions; the other $\mathrm{Voc}(B)$ symbols and their interpretations are *ancillary*.
(E2) All ancillary symbols are dynamic, and the initial states of B are the uninformative expansions of the initial states of A.
(E3) If Y is a $\mathrm{Voc}(B)$ structure and X the $\mathrm{Voc}(A)$ reduct of Y, then the principal-function updates (including trivial updates) generated by $\mathrm{Prog}(B)$ in Y coincide with the those generated by $\mathrm{Prog}(A)$ in X, and the ancillary-function updates generated by $\mathrm{Prog}(B)$ in Y are consistent. ◁

Corollary 1 *Suppose that B is a faithful expansion of an ASM A, then the following claims hold.*

1. *If $X_0, \ldots, X_n$ is a partial computation of A then there is a unique partial computation $Y_0, \ldots, Y_n$ of B such that every X_i is the* $\mathrm{Voc}(A)$ *reduct of the corresponding Y_i.*
2. *If states $Y_0, \ldots, Y_n$ of B form a partial computation of B, then their* $\mathrm{Voc}(A)$ *reducts form a partial computation $X_0, \ldots, X_n$ of A.*
3. *The* $\mathrm{Voc}(A)$ *reduct of a state of B is a state of A.*

If an ASM A computes a function F, one would expect that any faithful expansion of A computes function F as well. To confirm this expectation, we need to formalize what it means to compute a function. In the context of this paper, every ASM state is endowed with a special copy of the set $\mathbb{N}$ of natural numbers. This makes the desired formalization particularly easy for numerical partial functions $F : \mathbb{N}^k \to \mathbb{N}$.

Corollary 2 *Suppose that an ASM A computes a partial numerical function $F : \mathbb{N}^k \to \mathbb{N}$ in the following sense:*

1. *A has input variables $\iota_1, \ldots, \iota_n$ taking numerical values in the initial states, and A has an output variable o,*
2. *all initial states of A are isomorphic except for the values of the input variables, and*
3. *the computation of A with initial state X eventually terminates if and only if F is defined at tuple $\bar{x} = \big(\mathcal{V}_X(\iota_1), \ldots, \mathcal{V}_X(\iota_n)\big)$, in which case the final value of o is $F(\bar{x})$.*

Then every faithful expansion of A computes F in the same sense. ◁

Corollary 2 can be generalized to computing more general functions and to performing other tasks, but this is beyond the scope of this paper.

An ASM may be faithfully expanded by instrumenting its program for monitoring purposes. For example, if you are interested how often a particular assignment σ fires, replace σ with a parallel composition

$$\sigma \quad \| \quad \kappa := \kappa + 1$$

where a fresh variable κ, initially zero, is used as a counter. A similar counter is used in our Reversibility Theorem below.

4 Reversibility

Definition 5 An ASM B is *reversible (as is)* if there is an ASM C which reverses all B's computations in the following sense. If $Y_0, Y_1, \ldots, Y_n$ is a partial computation of B, then $Y_n, Y_{n-1}, \ldots, Y_0$ is a terminating computation of C.

Theorem 1 (Reversification Theorem) *Every ASM A has a faithful reversible expansion.*

Proof Enumerate the (occurrences of the) assignments in $\mathrm{Prog}(A)$ in the order they occur:

$$\sigma_1, \sigma_2, \ldots, \sigma_N$$

It is possible that σ_i, σ_j are identical even though $i \neq j$. The metavariable n will range over numbers $1, 2, \ldots, N$. For each n, let f^n be the head of σ_n, $r_n = \mathrm{Arity}(f^n)$, and $t_0^n, t_1^n, \ldots, t_{r_n}^n$ the terms such that

$$\sigma_n = \left(f^n(t_1^n, \ldots, t_{r_n}^n) := t_0^n\right).$$

□

We construct an expansion B of A. The ancillary symbols of B are as follows.

1. A variable κ.
2. For every n, a unary relation symbol $\texttt{Fire}^n$.
3. For every n, unary function symbols $f_0^n, f_1^n, \ldots, f_{r_n}^n$.

The default term for κ is 0. The default term for all relations $\texttt{Fire}^n$ is $\bot$. The default term for all functions $f_0^n, f_1^n, \ldots, f_{r_n}^n$ is $\texttt{nil}$. Accordingly, the initial states of B are obtained from the initial states of A by setting $\kappa = 0$, every $\texttt{Fire}^n(x) = \bot$, and every $f_i^n(x) = \texttt{nil}$,

The intention is this. If $X_0, X_1, \ldots, X_l$ is a partial computation of B, then for each $k = 0, \ldots, l$ we have the following.

1. The value of κ in X_k is k, so that κ counts the number of steps performed until now; we call it a *step counter*.
2. $\texttt{Fire}^n(\kappa)$ holds in X_{k+1} if and only if σ_n fires in X_k.
3. The values of $f_1^n(\kappa), \ldots, f_{r_n}^n(\kappa)$ in X_{k+1} record the values of the terms $t_1^n, \ldots, t_{r_n}^n$ in X_k respectively, and the value of $f_0^n(\kappa)$ in X_{k+1} records the value of the term $f^n(t_1^n, \ldots, t_{r_n}^n)$ in X_k.

The program of B is obtained from Prog(A) by replacing every assignment σ_n with Instr(n) (an allusion to "instrumentation") where

$$\begin{aligned}&\text{Instr}(n) \;=\\ &\qquad \sigma_n \quad \| \quad \kappa := \kappa + 1 \quad \| \quad \texttt{Fire}^n(\kappa + 1) := \top \quad \|\\ &\qquad f_0^n(\kappa + 1) := f^n(t_1^n, \ldots, t_{r_n}^n) \quad \|\\ &\qquad f_1^n(\kappa + 1) := t_1^n \quad \| \quad \ldots \quad \| \quad f_{r_n}^n(\kappa + 1) := t_{r_n}^n\end{aligned}$$

It is easy to check that the conditions (E1)–(E3) of Definition 4 hold, and B is indeed a faithful expansion of A. In particular, if Y and X are as in (E3) and X is terminal, then no assignment σ_n fires in X, and therefore no Instr(n) fires in Y, so that Y is terminal as well.

Lemma 2 *If $Y_0, \ldots, Y_k$ is a partial computation of B, then*

- *$\kappa = k$ in Y_k and*
- *if $j > k$ then $\texttt{Fire}^n(j)$, $f_0^n(j), \ldots, f_{r_n}^n(j)$ have their default values in Y_k.*

Proof of Lemma Induction on k. □

Now, we will construct an ASM C which reverses B's computations. The vocabulary of C is that of B, and any Voc(C) structure is an initial state of C. The program of C is

$$\texttt{if } \kappa > 0 \texttt{ then } \Big(\kappa := \kappa - 1 \quad \| \quad \mathop{\mathbf{PAR}}_{n} \text{Undo}(n)\Big)$$

where **PAR** is parallel composition, n ranges over $\{1, 2, \ldots, N\}$, and

$$\begin{aligned}&\text{Undo}(n) =\\ &\qquad \texttt{if } \texttt{Fire}^n(\kappa) = \top \texttt{ then}\\ &\qquad\quad \texttt{Fire}^n(\kappa) := \bot \quad \|\\ &\qquad\quad f^n\Big(f_1^n(\kappa), \ldots, f_{r_n}^n(\kappa)\Big) := f_0^n(\kappa) \quad \| \quad f_0^n(\kappa) := \texttt{nil} \quad \|\\ &\qquad\quad f_1^n(\kappa) := \texttt{nil} \quad \| \quad \ldots \quad \| \quad f_{r_n}^n(\kappa) := \texttt{nil}\end{aligned}$$

Lemma 3 *Let Y be an arbitrary nonterminal* Voc(B) *structure such that all functions* $\texttt{Fire}^n$ *and* f_i^n *have their default values at argument $k = \mathcal{V}_Y(\kappa)$ in Y. If* Prog(B) *transforms Y to Y', then* Prog(C) *transforms Y' back to Y, i.e.,* Prog(C) *undoes the updates generated by* Prog(B) *and does nothing else.*

Proof of Lemma The updates generated by Prog(B) in Y are the updates generated by the rules Instr(n) such that σ_n fires in Y. Since $k = \mathcal{V}_Y(\kappa)$, we have $\mathcal{V}_{Y'}(\kappa) = k + 1 > 0$, and therefore Prog($C$) decrements κ. It also undoes the other updates generated by the rules Instr(n). Indeed, suppose that σ_n fires in Y.

To undo the update $\texttt{Fire}^n(k+1) \leftarrow \top$, Prog($C$) sets $\texttt{Fire}^n(k+1)$ back to $\bot$.

To undo the update $f^n\big(t_1^n, \ldots, t_{r_n}^n\big) \leftarrow t_0^n$, generated by σ_n itself, Prog(C) sets $f^n\Big(f_1^n(k+1), \ldots, f_{r_n}^n(k+1)\Big)$ to $f_0^n(k+1)$. Recall that $f_1^n(k+1), \ldots, f_{r_n}^n(k+1)$

record $t_1^n, \ldots, t_{r_n}^n$ in Y and $f_0^n(k+1)$ records the value of $f^n\big(t_1^n, \ldots, t_{r_n}^n\big)$ in Y. Thus, Prog(C) sets $f^n\big(t_1^n, \ldots, t_{r_n}^n\big)$ back to its value in Y.

To undo the updates of $f_0^n(k+1)$, $f_1^n(k+1)$, $\ldots$, $f_{r_n}^n(k+1)$, Prog(C) sets $f_0^n(k+1)$, $f_1^n(k+1), \ldots, f_{r_n}^n(k+1)$ back to `nil`.

Thus, being executed in Y', Prog(C) undoes all updates generated by the rules Instr(n) in Y. A simple inspection of Prog(C) shows that it does nothing else. Thus, Prog(C) transforms Y' to Y. □

Now suppose that $Y_0, \ldots, Y_n$ is a computation of B, $k < n$, $Y = Y_k$, and $Y' = Y_{k+1}$. Then Y is nonterminal and, by Lemma 2, all $\texttt{Fire}^n(\kappa)$ and $f_i^n(\kappa)$ have their default values in Y. By Lemma 3, Prog(C) transforms Y_{k+1} to Y_k. The Y_0 is a terminal state of C. Thus, C reverses all B's computations. □

The proof of Reversification Theorem uses notation and the form of Prog(B) which is convenient for the proof. In examples and applications, notation and Prog(B) can be simplified.

Remark 3 (Notation) Let σ_n be an assignment $\big(g(t_1^n, \ldots, t_r^n) := t_0^n\big)$ so that f^n is g. If σ_n is the only g assignment in Prog(A) or if every other g assignment σ_m in Prog(A) is just another occurrence of σ_n, then the ancillary functions f_i^n may be denoted g_i; no confusion arises. ◁

Recall that a green light for an ASM A is a Boolean-valued expression that holds in the nonterminal states and fails in the terminal states.

Remark 4 (Green Light and Step Counter) In Prog(B), every Instr(n) has an occurrence of the assignment $\kappa := \kappa + 1$. A green light for A provides an efficient way to deal with this excess. Notice that B increments the step counter exactly when the green light is on.

Case 1: Prog(A) has the form $\texttt{if}\ \gamma\ \texttt{then}\ (k := k+1 \parallel \Pi)$.
In this case, γ is a green light for A, and A has already a step counter, namely k. Without loss of generality, k is the step counter κ used by Prog(B); if not, rename one of the two variables. Notice that the assignment $\sigma_1 = (k := k+1)$ needs no instrumentation. There is no need to signal firings of σ_1 because σ_1 fires at every step. And, when a step is completed, we know the previous value of the step counter; there is no need to record it.
Let $\text{Instr}^-(\Pi)$ be the rule obtained from Π by first replacing every assignment σ_n with the rule Instr(n) defined in the proof of the program, and then removing all occurrences of $k := k+1$. Then the program

$$\texttt{if}\ \gamma\ \texttt{then}\ \big(k := k+1 \parallel \text{Instr}^-(\Pi)\big)$$

has only one occurrence of $k := k+1$ and is equivalent to Prog(B).
Case 2: $\text{Prog}(A) = \big(\texttt{if}\ \gamma\ \texttt{then}\ \Pi\big)$ where γ is a green light for A and the step counter κ of Prog(B) does not occur in Prog(A).

The modified program $\texttt{if}\ \gamma\ \texttt{then}\ \big(\kappa := \kappa + 1 \parallel \Pi\big)$,
where κ is the step counter of B, is a faithful expansion of Prog(A), and thus Case 2 reduces to Case 1.
Case 3 is the general case.

By Lemma 1, every ASM program has a green light. If γ is a green light for A and $\Pi = \text{Prog}(A)$, then the program $\texttt{if}\ \gamma\ \texttt{then}\ \Pi$ is equivalent to Prog(A), and thus Case 3 reduces to Case 2. ◁

The rules Instr(n) and Undo(n), described in the proof of the theorem, are the simplest in the case when $r_n = 0$. In such a case, σ_n has the form $v := t$ where v is a variable, so that $f^n = v$ and $t_0^n = t$. Then

$$\begin{aligned}
&\text{Instr}(n) = \\
&\quad \sigma_n \ \parallel\ \kappa := \kappa + 1 \ \parallel\ \texttt{Fire}^n(\kappa + 1) := \top \ \parallel\ v_0(\kappa + 1) := v\,, \\
&\text{Undo}(n) = \\
&\qquad \texttt{if Fire}^n(\kappa) = \top\ \texttt{then} \\
&\qquad\quad \texttt{Fire}^n := \bot \ \parallel\ v := v_0(\kappa) \ \parallel\ v_0(\kappa) := \texttt{nil}.
\end{aligned}$$

Lemma 4 *Suppose that an assignment σ_n to a variable v can fire only at the last step of A and that the update generated by σ_n is never trivial. Then,* Instr(n) *and* Undo(n) *can be simplified to*

$$\begin{aligned}
\text{Instr}(n) &= \quad \sigma_n \ \parallel\ \kappa := \kappa + 1 \\
\text{Undo}(n) &= \quad \texttt{if}\ v \neq d\ \texttt{then}\ v := d
\end{aligned}$$

where d is the default term for v in Voc(A).

We do not assume that σ_n fires at the last step of every computation of A, and so the expression $v \neq d$ is not necessarily a green light for A.

Proof Suppose that σ_n fires in state Y of B. Then Y is nonterminal, $v = d$ in Y, the next state Y' is terminal, and $v \neq d$ in Y'. It is easy to see the simplified version of Undo(n) indeed undoes the updates generated by the simplified version of Instr(n) in Y. □

5 Examples

To illustrate the reversification procedure of Sect. 4, we consider three simple examples. By the reversification procedure we mean not only the constructions in the proof of Reversification Theorem, but also Remarks 3 and 4 and Lemma 4. Unsurprisingly, in each case, the faithful reversible expansion produced by the general-purpose procedure can be simplified.

5.1 Bisection Algorithm

The well-known bisection algorithm solves the following problem where $\mathbb{R}$ is the field of real numbers. Given a continuous function $F : \mathbb{R} \to \mathbb{R}$ and reals a, b, ε such that $F(a) < 0 < F(b)$ and $\varepsilon > 0$, find a real c such that $|F(c)| < \varepsilon$. Here is a draft program for the algorithm:

```
if |F((a+b)/2)| ≥ ε then
    if      F((a+b)/2) < 0 then a := (a+b)/2
    elseif  F((a+b)/2) > 0 then b := (a+b)/2
elseif c = nil then c := (a+b)/2
```

The condition $c = \texttt{nil}$ in the last line ensures that computation stops when c is assigned a real number for the first time.

The Boolean expression $|F\big((a+b)/2\big)| \geq \varepsilon$ is not quite a green light for the algorithm. When it is violated for the first time, c is still equal to `nil`. But the equality $c = \texttt{nil}$ is a green light. With an eye on using Remark 4, we modify the draft program to the following program, our "official" program of an ASM A representing the bisection algorithm.

```
if   c = nil  then
    if      F((a+b)/2) < −ε  then a := (a+b)/2
    elseif  F((a+b)/2) > ε   then b := (a+b)/2
    else    c := (a+b)/2
```

Voc(A) consists of the obligatory symbols, the symbols in Prog(A), and the unary relation symbol `Real`. In every initial state of A, `Real` is (a copy of) the set of real numbers, the static functions of Prog(A) have their standard meaning, and $c = \texttt{nil}$.

Notice that Lemma 4 applies to Prog(A) with σ_n being $c := (a+b)/2$. Taking this into account, the reversification procedure of Sect. 4 gives us a reversible expansion B of A with the following program.

```
if c = nil then
    κ := κ + 1   ‖
    if F((a+b)/2) < −ε then
        a := (a+b)/2   ‖  Fire¹(κ+1) := ⊤   ‖  a₀(κ+1) := a
    elseif F((a+b)/2) > ε then
        b := (a+b)/2   ‖  Fire²(κ+1) := ⊤   ‖  b₀(κ+1) := b
    else c := (a+b)/2
```

This program can be simplified (and remain reversible). Notice that

- if $\texttt{Fire}^1(k+1) = \top$, then $\texttt{Fire}^2(k+1) = \bot$, the previous value of b is the current value of b, and the previous value of a is $2a - b$ where a, b are the current values; and

- if $\texttt{Fire}^1(k+1) = \bot$, then $\texttt{Fire}^2(k+1) = \top$, the previous value of a is the current value of a, and the previous value of b is $2b-a$ where a, b are the current values.

Thus, there is no need for functions a_0, b_0, recording the previous values of variables a, b, and there is no need for $\texttt{Fire}^2$. We get:

```
if c = nil then
    κ := κ + 1  ||
    if F((a + b)/2) < −ε then
        a := (a + b)/2  ||  Fire¹(κ + 1) := ⊤
    elseif F((a + b)/2) > ε then b := (a + b)/2
    else c := (a + b)/2
```

The corresponding inverse algorithm may have this program:

```
if κ > 0 then
  κ := κ − 1  ||  if c ≠ nil then c := nil
  ||  if Fire¹(κ) = ⊤ then (Fire¹(κ) := ⊥ || a := 2a − b)
      else b := 2b − a
```

5.2 *Linear-Time Sorting*

The information needed to reverse each step of the bisection algorithm is rather obvious; you don't have to use our reversification procedure for that. Such information is slightly less obvious in the case of the following sorting algorithm.

For any natural number n, the algorithm sorts an arbitrary array f of distinct natural numbers $< n$ in time $\leq 2n$. Let m be the length of an input array f, so that $m \leq n$. The algorithm uses an auxiliary array g of length n which is initially composed of zeroes.

Here is a simple illustration of the sorting procedure where $n = 7$ and $f = \langle 3, 6, 0\rangle$. Traverse array f setting entries $g[f[i]]$ of g to 1 for each index i of f, i.e., setting $g[3]$, $g[6]$ and $g[0]$ to 1, so that g becomes $\langle 1, 0, 0, 1, 0, 0, 1\rangle$. Each index j of g with $g[j] = 1$ is an entry of the input array f. Next, traverse array g putting the indices j with $g[j] = 1$—in the order that they occur—back into array f, so that f becomes $\langle 0, 3, 6\rangle$. Voila, f has been sorted in $m + n$ steps.

We describe an ASM A representing the sorting algorithm. Arrays will be viewed as functions on finite initial segments of natural numbers. The nonobligatory function symbols in $\mathrm{Voc}(A)$ are as follows.

0. Constants m, n and variables k, l.
1. Unary dynamic symbols f and g.
2. Binary static symbols $<, +, -$ where $<$ is relational.

In every initial state of A,

0. m and n are natural numbers such that $m \leq n$, and $k = l = 0$,
1. f, g are arrays of lengths m, n respectively, the entries of f are distinct natural numbers $< n$, and all entries of g are zero,
2. the arithmetical operations $+$, $-$ and relation $<$ work as expected on natural numbers.

In the following program of A, k is the step counter, and l indicates the current position in array f to be filled in.

$$
\begin{array}{l}
\texttt{if } k < m + n \texttt{ then} \\
\quad k := k + 1 \;\; \| \\
\quad \texttt{if } k < m \texttt{ then } g(f(k)) := 1 \\
\quad \texttt{elseif } g(k - m) = 1 \texttt{ then } \big(f(l) := k - m \parallel l := l + 1\big)
\end{array}
$$

The reversification procedure of Sect. 4 plus some simplifications described below give us a faithful reversible expansion B of A with a program

$$
\begin{array}{l}
\texttt{if } k < m + n \texttt{ then} \\
\quad k := k + 1 \;\; \| \\
\quad \texttt{if } k < m \texttt{ then } \big(g(f(k)) := 1 \parallel g_1(k + 1) := f(k)\big) \\
\quad \texttt{elseif } g(k - m) = 1 \texttt{ then} \\
\qquad f(l) := k - m \;\; \| \;\; l := l + 1 \;\; \| \;\; f_0(k + 1) := f(l)
\end{array}
$$

We made some simplifications of Prog(B) by discarding obviously unnecessary ancillary functions.

- It is unnecessary to record the firings of assignment $\sigma_2 = \big(g(f(k)) := 1\big)$ because, in the states of B, the condition $\texttt{Fire}^2(k) = \top$ is expressed by the inequality $k \leq m$.
- The ancillary function g_0 recording the previous values of g is unnecessary because those values are all zeroes.
- The final two assignments in Prog(A) fire simultaneously, so that one fire-recording function, say $\texttt{Fire}^3$, suffices. But even that one ancillary function is unnecessary because, in the states of B, the condition $\texttt{Fire}^3(k) = \top$ is expressed by $m < k \wedge g(k - m - 1) = 1$.
- The ancillary functions f_1 and l_0 recording the previous value of l are unnecessary because we know that value, it is $l - 1$.

The desired inverse algorithm C may be given by this program:

$$
\begin{array}{l}
\texttt{if } k > 0 \texttt{ then} \\
\quad k := k - 1 \\
\quad \| \;\; \texttt{if } k \leq m \texttt{ then } \big(g(g_1(k)) := 0 \parallel g_1(k) := \texttt{nil}\big) \\
\quad \| \;\; \texttt{if } m < k \texttt{ and } g(k - m - 1) = 1 \texttt{ then} \\
\qquad f(l) := f_0(k) \;\; \| \;\; l := l - 1 \;\; \| \;\; f_0(k) := \texttt{nil}
\end{array}
$$

Obviously, A is not reversible as is; its final state doesn't have information for reconstructing the initial f. But do we need both remaining ancillary functions?

Since f_0 is obliterated after the first n steps of C, f_0 seems unlikely on its own to ensure reversibility. But it does. The reason is that, after the first n steps of C, the original array f is restored. Recall that $g_1(k)$ records the value $f(k-1)$ of the original f for each positive $k \leq m$, but we can discard g_1 and modify Prog(C) to

$$\begin{array}{l}\texttt{if } k > 0 \texttt{ then } k := k-1 \\ \quad \| \texttt{ if } k \leq m \texttt{ then } g(f(k-1)) := 0 \\ \quad \| \texttt{ if } m < k \texttt{ and } g(k-m-1) = 1 \texttt{ then} \\ \qquad f(l) := f_0(k) \parallel l := l-1 \parallel f_0(k) := \texttt{nil}\end{array}$$

Alternatively, we can discard f_0 but keep g_1. Indeed, the purpose of the assignment $f(l) := f_0(k)$ in Prog(C) is to restore $f(l)$ to its original value. But recall that every $f(l)$ is recorded as $g_1(l+1)$. So we can modify Prog(C) to

$$\begin{array}{l}\texttt{if } k > 0 \texttt{ then } k := k-1 \\ \quad \| \texttt{ if } k \leq m \texttt{ then } \big(g(g_1(k)) := 0 \parallel g_1(k) := \texttt{nil}\big) \\ \quad \| \texttt{ if } m < k \texttt{ and } g(k-m-1) = 1 \texttt{ then} \\ \qquad f(l) := g_1(l+1) \quad \| \quad l := l-1\end{array}$$

5.3 External Functions and Karger's Algorithm

Until now, for simplicity, we restricted attention to algorithms that are isolated in the sense that their computations are not influenced by the environment. Actually, the analysis of sequential algorithms generalizes naturally and easily to the case when the environment can influence the computation of an algorithm [12, §8]. To this end, so-called external functions are used.

Syntactically, the item (V3) in Sect. 2.2 should be refined to say that a function symbol f may be dynamic, or static, or external. Semantically, external functions are treated as oracles[5] When an algorithm evaluates an external function f at some input $\bar{x}$, it is the environment (and typically the operating system) that supplies the value $f(\bar{x})$. The value is well defined at any given step of the algorithm; if f is called several times, during the same step, on the same input $\bar{x}$, the same value is given each time. But, at a different step, a different value $f(\bar{x})$ may be given.

To illustrate reversification involving an external function, we turn attention to Karger's algorithm [14]. In graph theory, a minimum cut of a graph is a cut (splitting the vertices into two disjoint subsets) that minimizes the number of edges crossing the cut. Using randomization, Karger's algorithm constructs a cut which is a minimum cut with a certain probability. That probability is small but only polynomially (in the number of vertices) small. Here we are not interested in the minimum cut problem, only in the algorithm itself.

[5] In that sense, our generalization is similar to the oracle generalization of Turing machines.

Terminology 1 *Let $G = (V, E)$ be a graph and consider a partition P of the vertex set V into disjoint subsets which we call* cells*; formally P is the set of the cells. The* P-ends *of an edge $\{x, y\}$ are the cells containing the vertices x and y. An edge is* inter-cell *(relative to P) if its P-ends are distinct.* ◁

Now we describe a version of Karger's algorithm that we call KA. Given a finite connected graph (V, E), KA works with partitions of the vertex set V, one partition at a time, and KA keeps track of the set Inter of the inter-cell edges. KA starts with the finest partition $P = \big\{\{v\} : v \in V\big\}$ and Inter $= E$. If the current partition P has > 2 cells, then Inter is nonempty because the graph (V, E) is connected. In this case, KA selects a random inter-cell edge e, merges the P-ends p, q of e into one cell, and removes from Inter the edges in $\big\{\{x, y\} : x \in p \wedge y \in q\big\}$. The result is a coarser partition and smaller Inter. When the current partition has at most two cells, the algorithm stops.

Next we describe an ASM A representing KA. There are many ways to represent KA as an ASM. Thinking of the convenience of description rather than implementation of KA, we chose to be close to naive set theory. Let U be a set that includes V and all subsets of V and all sets of subsets of V (which is much more than needed but never mind). The relation $\in$ on U has its standard meaning; the vertices are treated as atoms (or urelements), not sets.

The nonobligatory function symbols of Voc(A) are as follows.

0. Nullary variables P and Inter.
1. Unary static symbols V, E, $|.|$, and a unary external symbol R.
2. Binary static symbols $>$, $-$, Merge, and Intra, where $>$ is relational.

In every initial state of A,

- V, U and $\in$ are as described above (up to isomorphism). $|s|$ is the cardinality of a set s, and $-$ is the set-theoretic difference. The relation $>$ is the standard ordering of natural numbers
- E is a set of unordered pairs $\{x, y\}$ with $x, y \in V$ such that the graph (V, E) is connected. P is the finest partition $\big\{\{v\} : v \in V\big\}$ of V. Inter $= E$.
- If $e \in E$, S is a partition of V, and p, q are the S-ends of e, then
  - Merge$(e, S) = (S - \{p, q\}) \cup \{p \cup q\}$, and
  - Intra$(e, S) = \big\{\{x, y\} : x \in p \wedge y \in q\big\}$.

The external function R takes a nonempty set and returns a member of it. The program of A can be this:

```
if |P| > 2 then
    P := Merge(R(Inter), P)
    ‖  Inter := Inter − Intra(R(Inter), P)
```

Now we apply the reversification procedure of Theorem 1, taking Remark 4 into account. We also take into account that both assignments fire at every step of the

algorithm and so there is no need to record the firings. This gives us a faithful reversible expansion B of A with a program

```
if |P| > 2 then
    κ := κ + 1   ||
    P := Merge(R(Inter), P)   ||   P_0(κ + 1) := P   ||
    Inter := Inter − Intra(R(Inter), P)   ||   Inter_0(κ + 1) := Inter
```

The corresponding inverse ASM C may be given by the program

```
if κ > 0 then
    κ := κ − 1   ||
    P := P_0(κ)   ||   P_0(κ) := nil   ||
    Inter := Inter_0(κ)   ||   Inter_0(κ) := nil
```

Remark 5 A custom crafted faithful expansion may be more efficient in various ways. For example, instead of recording the whole P, it may record just one of the two P-ends of $R(\text{Inter})$. This would require a richer vocabulary.

6 Conclusion

We have shown how to reversify an arbitrary sequential algorithm A by gently instrumenting A with bookkeeping machinery. The result is a step-for-step reversible algorithm B whose behavior, as far as the vocabulary of A is concerned, is identical to that of A.

We work with an ASM (abstract state machine) representation of the given algorithm which is behaviorally identical to it. The theory of such representation is developed in [12], and the practicality of it has been amply demonstrated.

Acknowledgments Many thanks to Andreas Blass for generous sanity check.

References

1. Al-Rabadi, A.N.: Reversible Logic Synthesis: From Fundamentals to Quantum Computing. Springer, Berlin (2014)
2. Bennett, C.H.: Logical reversibility of computation. IBM J. Res. Dev. **17**(6), 525–532 (1973)
3. Blass, A., Gurevich, Y.: Abstract state machines capture parallel algorithms. ACM Trans. Comput. Logic **4**(4), 578–651 (2003). Correction and extension, ibid. **9**(3), Article 19 (2008)
4. Blass, A., Gurevich, Y., Rossman, B.: Interactive small-step algorithms. Parts I and II. Log. Methods Comput. Sci. **3**(4), Articles 3 and 4 (2007)
5. Börger, E., Stärk, R.: Abstract State Machines: A Method for High-Level System Design and Analysis. Springer, Berlin (2003)
6. De Vos, A.: Reversible Computing: Fundamentals, Quantum Computing, and Applications. Wiley-VCH, Weinheim (2010)

7. Dijkstra, E.W.: Selected Writings on Computing: A Personal Perspective. Springer, New York (1982)
8. Fredkin, E., Toffoli, T.: Concervative logic. Int. J. Theor. Phys. **21**(3/4), 219–253 (1982)
9. Gries, D.: The Science of Programming. Springer, New York (1981)
10. Gurevich, Y.: Evolving Algebras: An Introductory Tutorial. Bull. EATCS **43**, 264–284 (1991). And (slightly revised) In: Rozenberg, G., Salomaa, A. (eds.) Current Trends in Theoretical Computer Science: Essays and Tutorials, pp. 266–292. World Scientific, Singapore (1993) (Abstract state machines used to be called evolving algebras)
11. Gurevich, Y.: Evolving algebra 1993: Lipari guide. In: Börger, E. (ed.) Specification and Validation Methods, pp. 9–36. Oxford University Press (1995). Reprinted at arXiv. https://arxiv.org/abs/1808.06255 (Abstract state machines used to be called evolving algebras)
12. Gurevich, Y.: Sequential abstract state machines capture sequential algorithms. ACM Trans. Comput. Logic **1**(1), 77–111 (2000)
13. Gurevich, Y., Huggins, J.: The semantics of the C programming language. In: Börger, E., et al. (eds.) Proc. CSL'92, Computer Science Logic. Springer Lecture Notes in Computer Science, vol. 702, pp. 274–308. Springer, Berlin (1993)
14. Karger, D.: Global min-cuts in RNC and other ramifications of a simple mincut algorithm. In: Proc. 4th Annual ACM-SIAM Symposium on Discrete Algorithms (1993)
15. Kolmogorov, A.N.: On the concept of algorithm. Uspekhi Matematicheskikh Nauk **8**(4), 175–176 (1953) (Russian). English version in Uspensky, V., Semenov, A.: Algorithms: Main ideas and Applications, pp. 18–19. Kluwer (1993)
16. Morita, K.: Theory of Reversible Computing. Springer Japan, Tokyo (2017)
17. Perumala, K.S.: Introduction to Reversible Computing. CRC Press, Boca Raton (2014)
18. RC2021: 13th International Conference on Reversible Computation. https://reversible-computation-2021.github.io/

Gentzen in the 3- and 4-Valued Jungle

Gerhard Jäger

For János Makowsky, in honor of his 75th birthday

Abstract This article begins with recalling the three-valued characterizations of the cut-free and identity-free derivations in the usual Gentzen-style sequent calculus $\mathcal{LK}$. Later we turn to a three-valued logic à la Kleene and four-valued logics in the spirit of Belnap and Priest. In each case sound and complete sequent calculi will be presented. The Priest four-valued logic plays an important role in approaching the tetralemma and the Catuskoti—prominent in Indian logic—from a Western perspective.

1 Foreword

János Makowsky is a logician with a wide spectrum of research activities: They range from graph polynomials and combinatorics over model theory to many logic-based parts of computer science such as database theory, logic programming and computational complexity. In addition, János is interested in the history of logic and has also many non-technical publications. To sum up: It seems difficult to press such a complex personality into the narrow framework of classical two-valued logic. Therefore, a three- or even four-valued environment seems better suited for dealing with and reflecting his many views and opinions and allowing even some inconsistencies.

Of course, it would be beyond the scope of this publication to even begin to "describe János". Instead, we limit ourselves to examining one possible formal framework for such an enterprise in a little more detail.

G. Jäger (✉)
Institute of Computer Science, University of Bern, Bern, Switzerland
e-mail: gerhard.jaeger@unibe.ch

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_23

2 Introduction

Three-valued logic has a very long history, and I am not the right person to give even a short summary. Important contributions have been made, for example, by Aristoteles, Pierce, Łukasiewicz, Lewis, Post, Kleene, Bochvar, Priest, Avron, and many more. In the following we will only speak about a few three-valued logics in the context of Gentzen-style sequent calculi. After having introduced the standard sequent calculus $\mathcal{LK}$ for propositional logic, we recall two classical results about the characterization of cut-free and identity-free derivations in $\mathcal{LK}$.

Afterwards we turn to three-valued logic à la Kleene and introduce 3-Kleene valuations. Then we present the sequent calculus $\mathcal{S}3\mathcal{K}$ and prove that it is sound and complete with respect to the 3-Kleene valuations.

In the next section we move from three-valued to four-valued logic. In doing that we concentrate on Belnap-style four-valued logic. After having introduced the semantical apparatus we introduce the subsystem $\mathcal{S}4\mathcal{B}$ of $\mathcal{S}3\mathcal{K}$ and show that it is adequate for Belnap's four-valued logic.

The final section is dedicated to the tetralemma and the Catuskoti. Both play a prominent role in Indian logic, and it is interesting to look at them from a Western perspective. We do so in picking up ideas from Priest and develop a suitable four-valued framework, extending that of the previous section. Again our focus is on the interplay between semantical considerations and an adequate sequent calculus.

3 The Sequent Calculus $\mathcal{LK}$

$\mathcal{LK}$ is formulated in the standard language of propositional logic based on *atomic propositions* $P, Q, R, \ldots$ whose formulas $A, B, C, \ldots$ are generated by the following grammar:

$$A \;:=\; P \mid \neg A \mid (A \vee A) \mid (A \wedge A).$$

We ue the letters $\Gamma, \Delta, \Pi, \Sigma, \ldots$ to denote finite sequences of formulas, and *sequents* are expressions of the form

$$\Gamma \to \Delta.$$

By *theories* we mean finite or infinite collections of sequents. All the letters of these syntactical categories may also be used with subscripts or primed.

The sequent calculus $\mathcal{LK}$ is now given by the following rules of inference and identity axioms:

I. Structural Rules

Weakening

$$\frac{\Gamma \to \Delta}{\Gamma, A \to \Delta}\ (lW) \qquad \frac{\Gamma \to \Delta}{\Gamma \to \Delta, A}\ (rW)$$

Exchange

$$\frac{\Gamma_1, A, B, \Gamma_2 \to \Delta}{\Gamma_1, B, A, \Gamma_2 \to \Delta}\ (lE) \qquad \frac{\Gamma \to \Delta_1, A, B, \Delta_2}{\Gamma \to \Delta_1, B, A, \Delta_2}\ (rE)$$

Contraction

$$\frac{\Gamma, A, A \to \Delta}{\Gamma, A \to \Delta}\ (lC) \qquad \frac{\Gamma \to \Delta, A, A}{\Gamma \to \Delta, A}\ (rC)$$

II. Logical Rules

Negation

$$\frac{\Gamma \to \Delta, A}{\Gamma, \neg A \to \Delta}\ (l\neg) \qquad \frac{\Gamma, A \to \Delta}{\Gamma \to \Delta, \neg A}\ (r\neg)$$

Disjunction

$$\frac{\Gamma, A \to \Delta \qquad \Gamma, B \to \Delta}{\Gamma, A \vee B \to \Delta}\ (l\vee) \qquad \frac{\Gamma \to \Delta, A, B}{\Gamma \to \Delta, A \vee B}\ (r\vee)$$

Conjunction

$$\frac{\Gamma, A, B \to \Delta}{\Gamma, A \wedge B \to \Delta}\ (l\wedge) \qquad \frac{\Gamma \to \Delta, A \qquad \Gamma \to \Delta, B}{\Gamma \to \Delta, A \wedge B}\ (r\wedge)$$

III. Identity Axioms and Cut

$$\Gamma, A \to \Delta, A\ \ (id) \qquad \frac{\Gamma \to \Delta, A \qquad \Gamma, A \to \Delta}{\Gamma \to \Delta}\ (cut)$$

Definition 1 Let T be a theory.

1. Then the notion $T \vdash \Gamma \to \Delta$ is used to express that the sequent $\Gamma \to \Delta$ is derivable in $\mathcal{LK}$ from T; it is inductively defined as follows:

(a) If $\Gamma \to \Delta$ is an identity axiom or an element of T, then we have $T \vdash \Gamma \to \Delta$.
(b) If $T \vdash \Gamma_i \to \Delta_i$ for every premise $\Gamma_i \to \Delta_i$ of a structural, a logical rule, or a cut, then we have $T \vdash \Gamma \to \Delta$ for the conclusion $\Gamma \to \Delta$ of this rule.

2. We write $T \vdash_0 \Gamma \to \Delta$ if the sequent $\Gamma \to \Delta$ is *cut-free* provable form T, i.e. if there is a proof using only the identity axioms, the elements of T, the structural rules and the logical rules.
3. $T \Vdash \Gamma \to \Delta$, on the other hand, means that the sequent $\Gamma \to \Delta$ is provable form T by a proof which does not make use of the identity axioms. In this case we say that $\Gamma \to \Delta$ is *identity-free* provable from T.

It is well known that $\mathcal{LK}$ is sound and complete with respect to the usual two-valued semantics for classical propositional logic. In the next section we provide a semantic characterization of the cut-free and identity-free derivations in $\mathcal{LK}$.

4 Cut-Free and Identity-Free Derivations

From now on we work with three truth values: **t** (true), **f** (false), **u** (undetermined, undefined, unknown). A 3-*valued valuation* simply is a mapping $\mathcal{V}$ that assigns one of these truth values to any formula,

$$\mathcal{V} : \mathit{Formulas} \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}.$$

Given a 3-valued valuation $\mathcal{V}$, we turn now to the question what it means that $\mathcal{V}$ satisfies or is a model of a sequent. Here we distinguish between two cases.

Definition 2 Let $\mathcal{V}$ be an arbitrary 3-valued valuation and $\Gamma \to \Delta$ an arbitrary sequent.

1. We call $\mathcal{V}$ a *weak model* of $\Gamma \to \Delta$ iff $\mathcal{V}(A) \neq \mathbf{t}$ for some A from Γ or $\mathcal{V}(B) \neq \mathbf{f}$ for some B from Δ,

$$\mathcal{V} \models_w \Gamma \to \Delta \;:\Leftrightarrow\; \begin{cases} \mathcal{V}(A) \neq \mathbf{t} \text{ for some } A \text{ from } \Gamma \text{ or} \\ \mathcal{V}(B) \neq \mathbf{f} \text{ for some } B \text{ from } \Delta. \end{cases}$$

2. We call $\mathcal{V}$ a *strong model* of $\Gamma \to \Delta$ iff $\mathcal{V}(A) = \mathbf{f}$ for some A from Γ or $\mathcal{V}(B) = \mathbf{t}$ for some B from Δ,

$$\mathcal{V} \models_s \Gamma \to \Delta \;:\Leftrightarrow\; \begin{cases} \mathcal{V}(A) = \mathbf{f} \text{ for some } A \text{ from } \Gamma \text{ or} \\ \mathcal{V}(B) = \mathbf{t} \text{ for some } B \text{ from } \Delta. \end{cases}$$

The following remark indicates which role these two notions of 3-valued model will play in the analysis of the cut-free and identity free $\mathcal{LK}$-derivations.

Remark 1 Let $\mathcal{V}$ be an arbitrary 3-valued valuation.

1. Weak models do not validate (*cut*). Consider formulas A, B, C with $\mathcal{V}(A) = \mathbf{t}$, $\mathcal{V}(B) = \mathbf{u}$, and $\mathcal{V}(C) = \mathbf{f}$. Then

$$\mathcal{V} \models_w A \to B \quad \text{and} \quad \mathcal{V} \models_w B \to C, \quad \text{but} \quad \mathcal{V} \not\models_w A \to C.$$

2. Strong models do not validate (*id*). Consider a formulas A with $\mathcal{V}(A) = \mathbf{u}$. Then

$$\mathcal{V} \not\models_s A \to A.$$

Now we turn to the two crucial valuations of this section: weak and strong Schütte valuations and the corresponding subsystems of $\mathcal{LK}$. This sort of valuations goes back to Schütte [6], but has been developed there in the context of a semantic interpretation of Takeuti's conjecture.

We begin with what we call weak Schütte valuations and will see that they provide the semantic counterpart of cut-free derivations in $\mathcal{LK}$.

Definition 3 A 3-valued valuation $\mathcal{V}$ is called *weak Schütte* iff the following three conditions are satisfied for all formulas A and B:

- If $\mathcal{V}(\neg A) = \mathbf{f}$, then $\mathcal{V}(A) = \mathbf{t}$; if $\mathcal{V}(\neg A) = \mathbf{t}$, then $\mathcal{V}(A) = \mathbf{f}$.
- If $\mathcal{V}(A \vee B) = \mathbf{f}$, then $\mathcal{V}(A) = \mathbf{f}$ and $\mathcal{V}(B) = \mathbf{f}$; if $\mathcal{V}(A \vee B) = \mathbf{t}$, then $\mathcal{V}(A) = \mathbf{t}$ or $\mathcal{V}(B) = \mathbf{t}$.
- If $\mathcal{V}(A \wedge B) = \mathbf{f}$, then $\mathcal{V}(A) = \mathbf{f}$ or $\mathcal{V}(B) = \mathbf{f}$; if $\mathcal{V}(A \wedge B) = \mathbf{t}$, then $\mathcal{V}(A) = \mathbf{t}$ and $\mathcal{V}(B) = \mathbf{t}$.

Please observe that in the case of weak Schütte valuations the value of a complex formula is in general not determined by the values of its subformulas. If $\mathcal{V}(B) = \mathcal{V}(C) = \mathbf{t}$, then $\mathcal{V}(B \wedge C)$ can take the values $\mathbf{t}$ and $\mathbf{u}$. Weak Schütte valuations only propagate the definite truth values $\mathbf{t}$ and $\mathbf{f}$ from compound formulas to (some of) their subformulas.

Definition 4 Let $\mathcal{V}$ be weak Schütte, $\Gamma \to \Delta$ a sequent, and T a theory.

1. $\mathcal{V}$ is called a *weak model* of T if $\mathcal{V} \models_w \Pi \to \Sigma$ for all elements $\Pi \to \Sigma$ of T.
2. $\Gamma \to \Delta$ is called a *weak consequence* of T if every weak Schütte valuation which is a weak model of T is also a weak model of $\Gamma \to \Delta$; in this case we write $T \models_w \Gamma \to \Delta$.

The following is a famous result due to Girard which provides a semantic characterization of the cut-free derivations in $\mathcal{LK}$. It states their soundness and completeness with respect to weak Schütter valuations. For a detailed proof see Girard [3].

Theorem 1 (Girard) *For all sequents* $\Gamma \to \Delta$ *and all theories* T*,*

$$T \vdash_0 \Gamma \to \Delta \quad \Longleftrightarrow \quad T \models_w \Gamma \to \Delta.$$

Next we turn to identity-free derivations and introduce strong Schütte valuations as their semantic counterpart. Now the definite truth values propagate from (some of) the subformulas of a compound formula to the compound formula.

Definition 5 A 3-valued valuation $\mathcal{V}$ is called *strong Schütte* iff the following three conditions are satisfied for all formulas A and B:

- If $\mathcal{V}(A) = \mathbf{f}$, then $\mathcal{V}(\neg A) = \mathbf{t}$; if $\mathcal{V}(A) = \mathbf{t}$, then $\mathcal{V}(\neg A) = \mathbf{f}$.
- If $\mathcal{V}(A) = \mathbf{f}$ and $\mathcal{V}(B) = \mathbf{f}$, then $\mathcal{V}(A \vee B) = \mathbf{f}$; if $\mathcal{V}(A) = \mathbf{t}$ or $\mathcal{V}(B) = \mathbf{t}$, then $\mathcal{V}(A \vee B) = \mathbf{t}$.
- If $\mathcal{V}(A) = \mathbf{f}$ or $\mathcal{V}(B) = \mathbf{f}$, then $\mathcal{V}(A \wedge B) = \mathbf{f}$; if $\mathcal{V}(A) = \mathbf{t}$ and $\mathcal{V}(B) = \mathbf{t}$, then $\mathcal{V}(A \wedge B) = \mathbf{t}$.

Also in this case the values of the subformulas of a formula A do not necessarily determine the value of A. However, if the values of all immediate subformulas of A belong to $\{\mathbf{t}, \mathbf{f}\}$, then $\mathcal{V}(A)$ is determined and belongs to $\{\mathbf{t}, \mathbf{f}\}$ as well.

Definition 6 Let $\mathcal{V}$ be strong Schütte, $\Gamma \to \Delta$ a sequent, and T a theory.

1. $\mathcal{V}$ is called a *strong model* of T if $\mathcal{V} \models_s \Pi \to \Sigma$ for all elements $\Pi \to \Sigma$ of T.
2. $\Gamma \to \Delta$ is called a *strong consequence* of T if every strong Schütte valuation which is a strong model of T is also a strong model of $\Gamma \to \Delta$; in this case we write $T \models_s \Gamma \to \Delta$.

The following theorem is from Hösli and Jäger [4] and provides the desired semantical characterization of the identity-free derivations in $\mathcal{LK}$.

Theorem 2 (Hösli and Jäger) *For all sequents* $\Gamma \to \Delta$ *and all theories* T,

$$T \Vdash \Gamma \to \Delta \quad \Longleftrightarrow \quad T \models_s \Gamma \to \Delta.$$

5 3-Kleene Valuations

Kleene's motivation for extending two-valued logic by a third truth value **u** was to consider the determination of the truth value of a formula as a computational process: It may yield **t** or **f**, but it may also fail to terminate. In that case the truth value is undefined or undetermined, represented by **u**. The behaviour of this third value has to be compatible with any increase of information.

That is, if the value of some atomic proposition P is changed from **u** to either **t** or **f**, the value of any formula in which P occurs must never change from **t** to **f** or from **f** to **t**, though a change from **u** to **t** or **f** is possible. The following definition makes this principle precise.

Definition 7 A 3-valued valuation $\mathcal{V}$ is called a *3-Kleene valuation (3$\mathcal{K}$-valuation for short)* iff the following three conditions are satisfied for all formulas A and B:

- $\mathcal{V}(\neg A) = \begin{cases} \mathbf{t} \text{ if } \mathcal{V}(A) = \mathbf{f}, \\ \mathbf{f} \text{ if } \mathcal{V}(A) = \mathbf{t}, \\ \mathbf{u} \text{ if } \mathcal{V}(A) = \mathbf{u}. \end{cases}$
- $\mathcal{V}(A \vee B) = \begin{cases} \mathbf{t} \text{ if } \mathcal{V}(A) = \mathbf{t} \text{ or } \mathcal{V}(B) = \mathbf{t}, \\ \mathbf{f} \text{ if } \mathcal{V}(A) = \mathbf{f} \text{ and } \mathcal{V}(B) = \mathbf{f}, \\ \mathbf{u} \text{ otherwise.} \end{cases}$
- $\mathcal{V}(A \wedge B) = \begin{cases} \mathbf{t} \text{ if } \mathcal{V}(A) = \mathbf{t} \text{ and } \mathcal{V}(B) = \mathbf{t}, \\ \mathbf{f} \text{ if } \mathcal{V}(A) = \mathbf{f} \text{ or } \mathcal{V}(B) = \mathbf{f}, \\ \mathbf{u} \text{ otherwise.} \end{cases}$

Clearly, a $3\mathcal{K}$-valuation is completely determined by the values of its atomic propositions. We simply have to compute its values on complex formulas according to the rules (1)–(3) above.

A first and trivial observation is that there are no tautologies with respect to $3\mathcal{K}$-valuations. Given any formula A we can assign the truth value **u** to all atomic propositions that occur in A. Then A has the truth value **u** under this valuation.

Definition 8 Let $\mathcal{V}$ be a $3\mathcal{K}$-valuation, $\Gamma \to \Delta$ a sequent, and T a theory.

1. We call $\mathcal{V}$ a $3\mathcal{K}$-*model* of $\Gamma \to \Delta$ iff $\mathcal{V}(A) \neq \mathbf{t}$ for some A from Γ or $\mathcal{V}(B) = \mathbf{t}$ for some B from Δ,

$$\mathcal{V} \models_{3\mathcal{K}} \Gamma \to \Delta \ :\Leftrightarrow \begin{cases} \mathcal{V}(A) \neq \mathbf{t} \text{ for some } A \text{ from } \Gamma \text{ or} \\ \mathcal{V}(B) = \mathbf{t} \text{ for some } B \text{ from } \Delta. \end{cases}$$

2. $\mathcal{V}$ is called a $3\mathcal{K}$-*model* of T iff $\mathcal{V} \models_{3\mathcal{K}} \Pi \to \Sigma$ for all elements $\Pi \to \Sigma$ of T.
3. $\Gamma \to \Delta$ is a $3\mathcal{K}$-*consequence* of T iff every 3-model of T is also a $3\mathcal{K}$-model of $\Gamma \to \Delta$. In this case we write $T \models_{3\mathcal{K}} \Gamma \to \Delta$.

The following remark lists several important properties of $3\mathcal{K}$-models. Note especially the discrepancy in the treatment of the rules $(l\neg)$ and $(r\neg)$.

Remark 2

1. $3\mathcal{K}$-models validate (*cut*):

$$\mathcal{V} \models_{3\mathcal{K}} \Gamma \to \Delta, A \quad \text{and} \quad \mathcal{V} \models_{3\mathcal{K}} \Gamma, A \to \Delta \quad \Longrightarrow \quad \mathcal{V} \models_{3\mathcal{K}} \Gamma \to \Delta.$$

2. $3\mathcal{K}$-models validate (*id*):

$$\mathcal{V} \models_{3\mathcal{K}} \Gamma, A \to \Delta, A.$$

3. $3\mathcal{K}$-models validate $(l\neg)$:

$$\mathcal{V} \models_{3\mathcal{K}} \Gamma \to \Delta, A \quad \Longrightarrow \quad \mathcal{V} \models_{3} \Gamma, \neg A \to \Delta.$$

4. But $3\mathcal{K}$-models do not validate $(r\neg)$: For $\mathcal{V}(A) = \mathbf{t}$, $\mathcal{V}(B) = \mathbf{f}$, and $\mathcal{V}(C) = \mathbf{u}$ we have that

$$\mathcal{V} \models_{3\mathcal{K}} A, C \to B \qquad \text{but} \qquad \mathcal{V} \not\models_{3\mathcal{K}} A \to B, \neg C.$$

6 The Sequent Calculus $\mathcal{S}3\mathcal{K}$

The next step is to come up with a syntactic analogue of the semantical notion of $3\mathcal{K}$-consequence. We do this by introducing an appropriate sequent calculus. Our calculus $\mathcal{S}3\mathcal{K}$ is obtained from $\mathcal{LK}$ by a few modifications.

- We add the following left-sided identity axioms:

$$\Gamma, A, \neg A \to \Delta \quad (l\text{-}id)$$

- The rules $(l\neg)$ and $(r\neg)$ for negation are replaced by the following rules:

Negated Negation

$$\frac{\Gamma, A \to \Delta}{\Gamma, \neg\neg A \to \Delta}\ (l\neg\neg) \qquad \frac{\Gamma \to \Delta, A}{\Gamma \to \Delta, \neg\neg A}\ (r\neg\neg)$$

Negated Disjunction

$$\frac{\Gamma, \neg A, \neg B \to \Delta}{\Gamma, \neg(A \vee B) \to \Delta}\ (l\neg\vee) \qquad \frac{\Gamma \to \Delta, \neg A \qquad \Gamma \to \Delta, \neg B}{\Gamma \to \Delta, \neg(A \vee B)}\ (r\neg\vee)$$

Negated Conjunction

$$\frac{\Gamma, \neg A \to \Delta \qquad \Gamma, \neg B \to \Delta}{\Gamma, \neg(A \wedge B) \to \Delta}\ (l\neg\wedge) \qquad \frac{\Gamma \to \Delta, \neg A, \neg B}{\Gamma \to \Delta, \neg(A \wedge B)}\ (r\neg\wedge)$$

If T is a theory we write $T \vdash_{3\mathcal{K}} \Gamma \to \Delta$ in order to state that the sequent $\Gamma \to \Delta$ is derivable in $\mathcal{S}3\mathcal{K}$ from T.

Remark 3 I assume that the sequent calculus $3\mathcal{K}$ or a closely related system must have been known before, but I could not find such a system in the literature

(probably due to my ignorance). Anyway, I think the proof of the following theorem is interesting in its own and worth to be presented.

Our main result states that $3\mathcal{K}$ is sound and complete with respect to the notion of 3-consequence. The interesting part is the completeness proof. To show the completeness of $3\mathcal{K}$ we build so-called *deduction chains*, a technique that has also been used in Schütte [7] for various completeness proofs.

Theorem 3 *For all sequents* $\Gamma \to \Delta$ *and theories* T*,*

$$T \vdash_{3\mathcal{K}} \Gamma \to \Delta \quad \Longleftrightarrow \quad T \models_{3\mathcal{K}} \Gamma \to \Delta.$$

Proof The direction from left to right is obvious. To show the converse direction, let us assume that $T \nvdash_{3\mathcal{K}} \Gamma \to \Delta$. We first fix an enumeration $F_1, F_2, F_3, \ldots$of all formulas and then define an infinite sequence

$$\Gamma_0 \to \Delta_0,\ \Gamma_1 \to \Delta_1,\ \ldots,\ \Gamma_n \to \Delta_n,\ \ldots$$

of sequents—the deductive chain for $\Gamma \to \Delta$ —by the following distinction of cases:

(D0) $\Gamma_0 \to \Delta_0 \ := \ \Gamma \to \Delta$.

(D1) $n = 3m > 0$. Then we set

$$\Gamma_{n+1} \to \Delta_{n+1} \ := \ \begin{cases} \Gamma_n \to F_m, \Delta_n \text{ if } T \nvdash_{3\mathcal{K}} \Gamma_n \to F_m, \Delta_n, \\ F_m, \Gamma_n \to \Delta_n \text{ otherwise.} \end{cases}$$

(D2) $n = 3m + 1$. Then Γ_{n+1} is Γ_n and Δ_{n+1} is determined by the rightmost formula of Δ_n.

(D2.1) Δ_n is Δ'_n, P for some atomic proposition P. Then

$$\Delta_{n+1} \ := \ P, \Delta'_n.$$

(D2.2) Δ_n is $\Delta'_n, A \vee B$. Then

$$\Delta_{n+1} \ := \ A \vee B, \Delta'_n, A, B.$$

(D2.3) Δ_n is $\Delta'_n, A \wedge B$. Then

$$\Delta_{n+1} \ := \ \begin{cases} A \wedge B, \Delta'_n, A \text{ if } T \nvdash_{3\mathcal{K}} \Gamma_n \to A \wedge B, \Delta'_n, A, \\ A \wedge B, \Delta'_n, B \text{ otherwise.} \end{cases}$$

(D2.4) Δ_n is $\Delta'_n, \neg\neg A$. Then

$$\Delta_{n+1} \;:=\; \neg\neg A, \Delta_n', A.$$

(D2.5) Δ_n is $\Delta_n', \neg(A \vee B)$. Then

$$\Delta_{n+1} \;:=\; \begin{cases} \neg(A \vee B), \Delta_n', \neg A \text{ if } T \nvdash_{3\mathcal{K}} \Gamma_n \to \neg(A \vee B), \Delta_n', \neg A, \\ \neg(A \vee B), \Delta_n', \neg B \text{ otherwise.} \end{cases}$$

(D2.6) Δ_n is $\Delta_n', \neg(A \wedge B)$. Then

$$\Delta_{n+1} \;:=\; \neg(A \wedge B), \Delta_n', \neg A, \neg B.$$

(D3) $n = 3m + 2$. Then Δ_{n+1} is Δ_n and Γ_{n+1} is determined by the rightmost formula of Γ_n, analogously to (D2.1)–(D2.5).

We immediately see, by a straightforward induction, that for all natural numbers n,

(R0) $T \nvdash_{3\mathcal{K}} \Gamma_n \to \Delta_n$,
(R1) all elements of Γ_n occur in Γ_{n+1}, and all those of Δ_n in Δ_{n+1}.

Let K_0 be the collection of all formulas that occur in one of the Γ_n and let K_1 be the collection of all formulas that occur in one of the Δ_n. Then K_0 and K_1 have the following properties concerning the logical connectives:

(P0) $A \vee B \in K_0 \implies A \in K_0$ or $B \in K_0$,
(P1) $A \vee B \in K_1 \implies A \in K_1$ and $B \in K_1$,
(P2) $A \wedge B \in K_0 \implies A \in K_0$ and $B \in K_0$,
(P3) $A \wedge B \in K_1 \implies A \in K_1$ or $B \in K_1$,
(P4) $\neg\neg A \in K_0 \implies A \in K_0$,
(P5) $\neg\neg A \in K_1 \implies A \in K_1$.
(P6) $\neg(A \vee B) \in K_0 \implies \neg A \in K_0$ and $\neg B \in K_0$.
(P7) $\neg(A \vee B) \in K_1 \implies \neg A \in K_1$ or $\neg B \in K_1$.
(P8) $\neg(A \wedge B) \in K_0 \implies \neg A \in K_0$ or $\neg B \in K_0$.
(P9) $\neg(A \wedge B) \in K_1 \implies \neg A \in K_1$ and $\neg B \in K_1$.

In addition, as consequence of definition rule (D1), the identity axioms of $3\mathcal{K}$, and the properties (R0) and (R1) we also have:

(Q0) For every formula A we have $A \in K_0$ or $A \in K_1$.
(Q1) There is no formula which belongs to K_0 and K_1.
(Q2) There is no formula A with $A \in K_0$ and $\neg A \in K_0$.

In view of properties (Q0)–(Q2) we can now introduce a mapping $\mathcal{V}_0$ from the atomic propositions to $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ by setting

$$\mathcal{V}_0(P) \;:=\; \begin{cases} \mathbf{t} \text{ if } P \in K_0, \\ \mathbf{f} \text{ if } \neg P \in K_0, \\ \mathbf{u} \text{ if } P \notin K_0 \text{ and } \neg P \notin K_0. \end{cases}$$

Then $\mathcal{V}_0$ is extended to the 3-valued valuation $\mathcal{V}$ by following the rules of 3-valuations.

By induction of the length of A, making use of (P0)–(P9) above, it is easily shown that

(1) $A \in K_0 \implies \mathcal{V}(A) = \mathbf{t}$,
(2) $A \in K_1 \implies \mathcal{V}(A) \in \{\mathbf{f}, \mathbf{u}\}$.

Since all elements of Γ belong to K_0 and all elements of Δ to K_1 we can conclude that

$$\mathcal{V} \not\models_{3\mathcal{K}} \Gamma \to \Delta.$$

It remains to show that $\mathcal{V}$ is a 3-model of T. So let $\Pi \to \Sigma$ be an element of T. Since all $\Gamma_n \to \Delta_n$ are not provable from T, it follows that there is an A in Π that does not belong to K_0 or a B in Σ that does not belong to K_1. In the first case we have $A \in K_1$ because of (Q0) and, therefore, $\mathcal{V}(A) \in \{\mathbf{f}, \mathbf{u}\}$ according to (2), and the second case implies $\mathcal{V}(B) = \mathbf{t}$ in view of (Q0) and (1). Hence $\mathcal{V} \models_{3\mathcal{K}} \Pi \to \Sigma$.

So we have shown that for a sequent $\Gamma \to \Delta$ with $T \not\vdash_{3\mathcal{K}} \Gamma \to \Delta$ there exists a $3\mathcal{K}$-model of T which does not $3\mathcal{K}$-validate $\Gamma \to \Delta$, completing our proof. □

Remark 4 If you go back to the definition of $3\mathcal{K}$ then you see that the rule

$$\frac{\Gamma \to \Delta, A}{\Gamma, \neg A \to \Delta} \quad (l\neg).$$

does not belong to its rules of inference. However, it is easy to see (semantically) that it is admissible.

7 Adding the Fourth Truth Value b

Four valued logics play a certain role in many different areas. There is the wide field of philosophical logic, there are connections to other areas of logic such as relevance logic, and there are even applications in computer science. As in the case of three-valued logic, I refrain from going into details and only mention a few scientists who have made important contributions: Belnap, Dunn, Priest, Bimbó, and Avron, but there are many more. The following is based to a certain extent on Belnap [1, 2].

In four valued logic the three truth values **t**, **f**, **u** are supplemented by a fours truth value **b** which can be read as both: true and false. Sometimes it is also interpreted as representing the epistemic status of conflicting information. A simple way to deal with these truth values is to identify them with elements of $\{0, 1\} \times \{0, 1\}$ as follows:

$$\mathbf{t} = \langle 1, 0\rangle, \qquad \mathbf{f} = \langle 0, 1\rangle,$$
$$\mathbf{u} = \langle 0, 0\rangle, \qquad \mathbf{b} = \langle 1, 1\rangle.$$

On these pairs we then define the following operations:

$$\begin{aligned} -\langle x, y\rangle &:= \langle y, x\rangle, \\ \langle x_1, y_1\rangle \oplus \langle x_2, y_2\rangle &:= \langle \max(x_1, x_2), \min(y_1, y_2)\rangle, \\ \langle x_1, y_1\rangle \otimes \langle x_2, y_2\rangle &:= \langle \min(x_1, x_2), \max(y_1, y_2)\rangle. \end{aligned}$$

In analogy to three valued logic, a 4-*valued valuation* simply is a mapping $\mathcal{V}$ from the formulas to $\{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{b}\}$,

$$\mathcal{V} : \mathit{Formulas} \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{b}\}.$$

In the following we are only interested in specific 4-valued valuations, the so-called 4-Belnap valuations.

Definition 9 A 4-valued valuation $\mathcal{V}$ is called a 4-*Belnap valuation (4$\mathcal{B}$-valuation for short)* iff the following three conditions are satisfied for all formulas A and B:

- $\mathcal{V}(\neg A) = -\mathcal{V}(A)$.
- $\mathcal{V}(A \vee B) = \mathcal{V}(A) \oplus \mathcal{V}(B)$.
- $\mathcal{V}(A \wedge B) = \mathcal{V}(A) \otimes \mathcal{V}(B)$.

Please observe that any 3-Kleene valuation is also a 4-Belnap valuation. The set of *designated values* is the set $\mathcal{D} := \{\mathbf{t}, \mathbf{b}\}$.

Definition 10 Let $\mathcal{V}$ be a 4$\mathcal{B}$-valuation, $\Gamma \to \Delta$ a sequent, and T a theory.

1. We call $\mathcal{V}$ a 4$\mathcal{B}$-*model* of $\Gamma \to \Delta$ iff $\mathcal{V}(A) \notin \mathcal{D}$ for some A from Γ or $\mathcal{V}(B) \in \mathcal{D}$ for some B from Δ,

$$\mathcal{V} \models_{4\mathcal{B}} \Gamma \to \Delta \;:\Leftrightarrow\; \begin{cases} \mathcal{V}(A) \notin \mathcal{D} \text{ for some } A \text{ in } \Gamma \text{ or} \\ \mathcal{V}(B) \in \mathcal{D} \text{ for some } B \text{ in } \Delta. \end{cases}$$

2. $\mathcal{V}$ is called a 4$\mathcal{B}$-*model* of T if $\mathcal{V} \models_{4\mathcal{B}} \Gamma \to \Delta$ for all elements $\Gamma \to \Delta$ of T.
3. $\Gamma \to \Delta$ is a 4$\mathcal{B}$-*consequence* of T iff every 4$\mathcal{B}$-model of T is also an 4$\mathcal{B}$-model of $\Gamma \to \Delta$. In this case we write $T \models_{4\mathcal{B}} \Gamma \to \Delta$.

Let us briefly look at the relationship between 3-consequences and 4-consequences. It is easy to see that we have

$$\mathcal{V} \models_{3\mathcal{K}} \Gamma \to \Delta \quad\Longleftrightarrow\quad \mathcal{V} \models_{4\mathcal{B}} \Gamma \to \Delta$$

for all 3$\mathcal{K}$-valuations $\mathcal{V}$ and all $\Gamma \to \Delta$. This implies that

$$T \models_{4\mathcal{B}} \Gamma \to \Delta \quad\Longrightarrow\quad T \models_{3\mathcal{K}} \Gamma \to \Delta. \qquad (*)$$

Now consider the sequent $P, \neg P \rightarrow Q$ where P and Q are atomic propositions. Then $\mathcal{V} \models_{3\mathcal{K}} P, \neg P \rightarrow Q$ for all $3\mathcal{K}$-valuations. On the other hand, if we consider a $4\mathcal{B}$-valuation $\mathcal{V}$ with $\mathcal{V}(P) = \mathbf{b}$ and $\mathcal{V}(Q) = \mathbf{f}$, then $\mathcal{V} \not\models_{4\mathcal{B}} P, \neg P \rightarrow Q$. Hence the converse of (*) is not true in general.

8 The Sequent Calculus $\mathcal{S}4\mathcal{B}$

The previous observations indicate how we can proceed to obtain a calculus which is adequate for $4\mathcal{B}$-consequences. Obviously, $(l\text{-}id)$ creates "$4\mathcal{B}$-valued problems". So we simply drop it.

So let $\mathcal{S}4\mathcal{B}$ the subsystem of $\mathcal{S}3\mathcal{K}$ that is obtained by deleting the axioms $(l\text{-}id)$. If T is a theory we write $T \vdash_{4\mathcal{B}} \Gamma \rightarrow \Delta$ in order to state that the sequent $\Gamma \rightarrow \Delta$ is derivable in $\mathcal{S}4\mathcal{B}$ from T. Again we have the desired soundness and completeness result.

Theorem 4 *For all sequents* $\Gamma \rightarrow \Delta$ *and theories* T*,*

$$T \vdash_{4\mathcal{B}} \Gamma \rightarrow \Delta \quad \Longleftrightarrow \quad T \models_{4\mathcal{B}} \Gamma \rightarrow \Delta.$$

Proof As in the proof of Theorem 3 the direction from left to write is obvious. To show the converse direction, we proceed in analogy to the proof Theorem 3 and assume that $T \not\vdash_{4\mathcal{B}} \Gamma \rightarrow \Delta$. Then we fix an enumeration $F_1, F_2, F_3, \ldots$ of all formulas. Also the definition of the infinite sequence

$$\Gamma_0 \rightarrow \Delta_0,\ \Gamma_1 \rightarrow \Delta_1,\ \ldots,\ \Gamma_n \rightarrow \Delta_n,\ \ldots$$

is exactly as before. Therefore we have again that for all natural numbers n,

(R0) $T \not\vdash_{4\mathcal{B}} \Gamma_n \rightarrow \Delta_n$,
(R1) all elements of Γ_n occur in Γ_{n+1}, and all those of Δ_n in Δ_{n+1}.

With K_0 and K_1 defined as before we also have the properties (P0)–(P9) as well as (Q0) and (Q1). However, since $4\mathcal{B}$ does not comprise the axioms $(l\text{-}id)$, property (Q2) is no longer available.

The next step is to introduce a mapping $\mathcal{W}_0$ from the atomic propositions to $\{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{b}\}$ by setting

$$\mathcal{W}_0(P) := \begin{cases} \mathbf{t} \text{ if } P \in K_0 \text{ and } \neg P \notin K_0, \\ \mathbf{b} \text{ if } P \in K_0 \text{ and } \neg P \in K_0, \\ \mathbf{f} \text{ if } P \notin K_0 \text{ and } \neg P \in K_0, \\ \mathbf{u} \text{ if } P \notin K_0 \text{ and } \neg P \notin K_0. \end{cases}$$

Then $\mathcal{W}_0$ is extended to the 4-valuation $\mathcal{W}$ by following the rules of 4-valuations described in Definition 10. As before, the following two crucial properties

(1) $A \in K_0 \implies \mathcal{W}(A) \in \{\mathbf{t}, \mathbf{b}\}$,
(2) $A \in K_1 \implies \mathcal{W}(A) \in \{\mathbf{f}, \mathbf{u}\}$

are proved by induction on the length of A; this proof makes use of (Q1) and (P0)–(P9).
The final steps are again as before. Since all elements of Γ belong to K_0 and all elements of Δ to K_1 we can conclude that

$$\mathcal{W} \not\models_{4\mathcal{B}} \Gamma \to \Delta.$$

To verify that $\mathcal{W}$ is a $4\mathcal{B}$-model of T pick any element $\Pi \to \Sigma$ of T. Since all $\Gamma_n \to \Delta_n$ are not provable from T, it follows that there is an A in Π that does not belong to K_0 or a B in Σ that does not belong to K_1. In the first case we have $A \in K_1$ because of (Q0) and, therefore, $\mathcal{W}(A) \in \{\mathbf{f}, \mathbf{u}\}$ according to (2). The second case implies $\mathcal{W}(B) \in \{\mathbf{t}, \mathbf{b}\}$ in view of (Q0) and (1). Hence $\mathcal{W} \models_{4\mathcal{B}} \Pi \to \Sigma$.

So we have shown that for a sequent $\Gamma \to \Delta$ with $T \not\vdash_{4\mathcal{B}} \Gamma \to \Delta$ there exists a $4\mathcal{B}$-model of T which does not $4\mathcal{B}$-validate $\Gamma \to \Delta$, completing our proof. □

With this proof we end our section about Belnap's four-valued logic. For more about applications of this logic in computer science we refer to Belnap [2]. To conclude this article let us look at another four-valued logic.

9 The Tetralemma and the Catuskoti

Let us begin this section with a quotation from Wikipedia:

> The *tetralemma* is a figure that features prominently in the logic of India. It states that with reference to any a logical proposition A, there are four possibilities:
>
> | | | |
> |---|---|---|
> | (Po1) | A | affirmation |
> | (Po2) | $\neg A$ | negation |
> | (Po3) | $A \wedge \neg A$ | both |
> | (Po4) | $\neg(A \vee \neg A)$ | neither |
>
> The history of fourfold negation, the *Catuskoti* (Sanskrit), is evident in the logico-epistemological tradition of India, given the categorical nomenclature Indian logic in Western discourse. Subsumed within the auspice of Indian logic, "Buddhist logic" has been particularly focused in its employment of the fourfold negation, as evidenced by the traditions of Nagarjuna and the Madhyamaka, particularly the school of Madhyamaka given the retroactive nomenclature of Prasangika by the Tibetan Buddhist logico-epistemological tradition. Though tetralemma was also used as a form inquiry rather than logic in the Nasadiya Sukta of Rigveda (creation hymn) though seems to be rarely used as a tool of logic before Buddhism.

Clearly, this tetralemma does not make much sense on the basis of classical logic; for example, (P3) and (P4) are equivalent in classical logic. However, it is an important logical argument in Indian philosophy.

Catuskoti was employed particularly by Nagarjuna (ca. 150–250 AD), in connection with metaphysical questions about being and not-being: It requires a different "role of negation" and the possibility of a so-called *middle path*. In Christian theology there are some discussions in connection with the so-called *Ambivalenztoleranz* and the idea to reconcile life and death (resurrection).

So there is the question whether it is possible to make sense of the Catuskoti on the basis of more traditional (Western) logics?

Priest [5] addresses this question and provides an excellent general introduction into the area of the tetralemma and the Catuskoti. In the following we will also present a formal framework tailored for the Catuskoti which takes up some of Priest's ideas.

In particular, we follow him in extending our basic language of propositional logic to the language $\mathcal{L}_{\mathcal{T}}$ by a adding new unary operator $\mathcal{T}$. The formulas $A, B, C, \ldots$ of $\mathcal{L}_{\mathcal{T}}$ are generated by the following grammar:

$$A \;:=\; P \mid \neg A \mid (A \vee A) \mid (A \wedge A) \mid \mathcal{T}(A).$$

The *atomic formulas* of $\mathcal{L}_{\mathcal{T}}$ are the atomic propositions and all formulas of the form $\mathcal{T}(A)$. $\mathcal{T}$ is a sort of truth-predicate whose exact meaning will be specified below. From now on the letters $\Gamma, \Delta, \Pi, \Sigma, \ldots$ denote finite sequences of $\mathcal{L}_{\mathcal{T}}$ formulas, and sequents are extended accordingly.

Definition 11 A mapping $\mathcal{V}$ from the $\mathcal{L}_{\mathcal{T}}$-formulas to $\{\mathbf{t}, \mathbf{f}, \mathbf{b}, \mathbf{u}\}$ is called a 4-*Priest valuation (4$\mathcal{P}$-valuation for short)* iff it satisfies the following five properties for all $\mathcal{L}_{\mathcal{T}}$ formulas A and B:

- $\mathcal{V}(\neg A) = -\mathcal{V}(A)$.
- $\mathcal{V}(A \vee B) = \mathcal{V}(A) \oplus \mathcal{V}(B)$.
- $\mathcal{V}(A \wedge B) = \mathcal{V}(A) \otimes \mathcal{V}(B)$.
- $\mathcal{V}(A) \in \mathcal{D} \implies \mathcal{V}(\mathcal{T}(A)) \in \mathcal{D}$.
- $\mathcal{V}(A) \notin \mathcal{D} \implies \mathcal{V}(\mathcal{T}(A)) = \mathbf{f}$.

Thus a 4$\mathcal{P}$-valuation is a 4$\mathcal{B}$-valuation for $\mathcal{L}_{\mathcal{T}}$ with two extra clauses for the new atomic formulas. The following definition is the Priest equivalent of Definition 10.

Definition 12 Let $\mathcal{V}$ be a 4$\mathcal{P}$-valuation, $\Gamma \to \Delta$ a sequent, and T a theory.

1. We call $\mathcal{V}$ a 4$\mathcal{P}$-*model* of $\Gamma \to \Delta$ iff $\mathcal{V}(A) \notin \mathcal{D}$ for some A from Γ or $\mathcal{V}(B) \in \mathcal{D}$ for some B from Δ,

$$\mathcal{V} \models_{4\mathcal{P}} \Gamma \to \Delta \;:\Leftrightarrow\; \begin{cases} \mathcal{V}(A) \notin \mathcal{D} \text{ for some } A \text{ in } \Gamma \text{ or} \\ \\ \mathcal{V}(B) \in \mathcal{D} \text{ for some } B \text{ in } \Delta. \end{cases}$$

2. $\mathcal{V}$ is called a 4$\mathcal{P}$-*model* of T if $\mathcal{V} \models_{4\mathcal{P}} \Gamma \to \Delta$ for all elements $\Gamma \to \Delta$ of T.

3. $\Gamma \to \Delta$ is a $4\mathcal{P}$-*consequence* of T iff every $4\mathcal{P}$-model of T is also a $4\mathcal{P}$-model of $\Gamma \to \Delta$. In this case we write $T \models_{4\mathcal{P}} \Gamma \to \Delta$.

Before we present the sequent calculus $4\mathcal{P}$ for Priest consequences, let us list a few properties of Priest valuations that will motivate some axioms.

Remark 5 For all $4\mathcal{P}$-valuations $\mathcal{V}$ we have:

1. $\mathcal{V} \models_{4\mathcal{P}} \Gamma, A \to \Delta, \mathcal{T}(A)$.
2. $\mathcal{V} \models_{4\mathcal{P}} \Gamma, \mathcal{T}(A) \to \Delta, A$.
3. $\mathcal{V} \models_{4\mathcal{P}} \Gamma \to \Delta, \mathcal{T}(A), \neg\mathcal{T}(A)$.
4. In general, however, $\mathcal{V}(A) \neq \mathcal{V}(\mathcal{T}(A))$.

10 The Sequent Calculus $\mathcal{S}4\mathcal{P}$

The 4-valued sequent calculus $\mathcal{S}4\mathcal{P}$ is obtained from the calculus $\mathcal{S}4\mathcal{B}$—formulated, of course, in the language $\mathcal{L}_{\mathcal{T}}$—by adding the following $\mathcal{T}$-**axioms**:

$$\Gamma, A \to \Delta, \mathcal{T}(A) \qquad \mathcal{T}.1$$

$$\Gamma, \mathcal{T}(A) \to \Delta, A \qquad \mathcal{T}.2$$

$$\Gamma \to \Delta, \mathcal{T}(A), \neg\mathcal{T}(A) \qquad \mathcal{T}.3$$

If T is a theory in $\mathcal{L}_{\mathcal{T}}$ we write $T \vdash_{4\mathcal{P}} \Gamma \to \Delta$ to state that the sequent $\Gamma \to \Delta$ is derivable in $\mathcal{S}4\mathcal{P}$ from T.

Theorem 5 *For all $\mathcal{L}_{\mathcal{T}}$ sequents $\Gamma \to \Delta$ and all $\mathcal{L}_{\mathcal{T}}$ theories T,*

$$T \vdash_{4\mathcal{P}} \Gamma \to \Delta \quad \Longleftrightarrow \quad T \models_{4\mathcal{P}} \Gamma \to \Delta.$$

Proof For the direction from left to right, by Remark 5, all $\mathcal{T}$-axioms are $4\mathcal{P}$-valid; the rest is then straightforward.
For the converse direction let us begin literally as in the proof of Theorem 4: We assume that $T \nvdash_{4\mathcal{P}} \Gamma \to \Delta$ and fix an enumeration $F_1, F_2, F_3, \ldots$of all formulas. Then the definition of the infinite sequence

$$\Gamma_0 \to \Delta_0,\ \Gamma_1 \to \Delta_1,\ \ldots,\ \Gamma_n \to \Delta_n,\ \ldots$$

is exactly as before (with the only difference that the atomic propositions in (D2.1) are replaced by atomic $\mathcal{L}_{\mathcal{T}}$ formulas). We have again that for all natural numbers n,

(R0) $T \nvdash_{4\mathcal{P}} \Gamma_n \to \Delta_n$,
(R1) all elements of Γ_n occur in Γ_{n+1}, and all those of Δ_n in Δ_{n+1}.

We define K_0 and K_1 as before and have the properties (P0)–(P9) as well as (Q0) and (Q1). But now we deviate from the previous proof and turn to a mapping $\mathcal{V}_0$ from the atomic formulas of $\mathcal{L}_\mathcal{T}$ to $\{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{b}\}$ by setting, for all atomic α:

$$\mathcal{V}_0(\alpha) \;:=\; \begin{cases} \mathbf{t} & \text{if } \alpha \in K_0 \text{ and } \neg\alpha \notin K_0, \\ \mathbf{b} & \text{if } \alpha \in K_0 \text{ and } \neg\alpha \in K_0, \\ \mathbf{f} & \text{if } \alpha \notin K_0 \text{ and } \neg\alpha \in K_0, \\ \mathbf{u} & \text{if } \alpha \notin K_0 \text{ and } \neg\alpha \notin K_0. \end{cases}$$

Finally, $\mathcal{V}_0$ is extended to the mapping $\mathcal{V}$ from the $\mathcal{L}_\mathcal{T}$ formulas to $\{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{b}\}$ by employing the clauses (1)–(3) of the definition of $4\mathcal{P}$-valuations to fix the values of complex formulas.

We do not know yet whether $\mathcal{V}$ is a Priest valuation ; this will be shown later. However, as before we obtain—now for all $\mathcal{L}_\mathcal{T}$ formulas A—that

(1) $A \in K_0 \implies \mathcal{V}(A) \in \{\mathbf{t}, \mathbf{b}\}$,
(2) $A \in K_1 \implies \mathcal{V}(A) \in \{\mathbf{f}, \mathbf{u}\}$.

Now we show that $\mathcal{V}$ satisfies properties (4) and (5) of a $4\mathcal{P}$-valuation.

(i) So assume $\mathcal{V}(A) \in \mathcal{D}$. Because of (2) and (Q0) we conclude $A \in K_0$. Since none of the sequents $\Gamma_n \to \Delta_n$ is provable in $4\mathcal{P}$, this implies together with axiom ($\mathcal{T}$.1) that $\mathcal{T}(A) \notin K_1$, hence $\mathcal{T}(A) \in K_0$ because of (Q0). The definition of $\mathcal{V}_0$ thus tells us that $\mathcal{V}(\mathcal{T}(A)) \in \mathcal{D}$.
(ii) On the other hand, if $\mathcal{V}(A) \notin \mathcal{D}$, then $A \in K_1$ follows from (1) and (Q0). Hence $\mathcal{T}(A) \notin K_0$ because of ($\mathcal{T}$.2). Thus (Q0) yields $\mathcal{T}(A) \in K_1$. In addition, $\neg\mathcal{T}(A)$ cannot be an element of K_1 because of axiom ($\mathcal{T}$.3). Therefore, $\neg\mathcal{T}(A) \in K_0$ in view of (Q0). So we have $\mathcal{T}(A) \notin K_0$ and $\neg\mathcal{T}(A) \in K_0$. From the definition of $\mathcal{V}_0$ conclude that $\mathcal{V}(\mathcal{T}(A)) = \mathbf{f}$.

So we know that $\mathcal{V}$ is a $4\mathcal{P}$-valuation. But then we are done since as in the previous two completeness proofs we obtain $\mathcal{V} \not\models_{4\mathcal{P}} \Gamma \to \Delta$ and $\mathcal{V} \models_{4\mathcal{P}} T$ □

The truth-operator $\mathcal{T}$ provides the essential means to represent the Catuskoti. First we define

$$\mathcal{F}(A) := \mathcal{T}(\neg A)$$

and observe that $\mathcal{F}(A)$ is not the negation of $\mathcal{T}(A)$. However, intuitively we may interpret $\mathcal{T}(A)$ as "A is true" and $\mathcal{F}(A)$ as "A is false. Since $\mathcal{T}(A)$ and $\mathcal{F}(A)$ are independent of each other, this provides space for truth gaps. This independence of $\mathcal{T}(A)$ and $\mathcal{F}(A)$ gives us the possibility to define finer degrees of truth (and falsity).

Definition 13 (Degrees of Truth) We introduce the following *degrees of truth*:

$$\mathbb{T}(A) \;:\Leftrightarrow\; \mathcal{T}(A) \wedge \neg\mathcal{F}(A),$$

$$\mathbb{F}(A) \ :\Leftrightarrow \ \neg\mathcal{T}(A) \wedge \mathcal{F}(A),$$

$$\mathbb{B}(A) \ :\Leftrightarrow \ \mathcal{T}(A) \wedge \mathcal{F}(A),$$

$$\mathbb{U}(A) \ :\Leftrightarrow \ \neg\mathcal{T}(A) \wedge \neg\mathcal{F}(A).$$

It is now easy to check that every assertion A has exactly one degree of truth. The following two statements are $4\mathcal{P}$-valid:

$$\mathbf{C1} : \mathbb{T}(A) \vee \mathbb{F}(A) \vee \mathbb{B}(A) \vee \mathbb{U}(A),$$

$$\mathbf{C2} : \neg(\mathbb{D}_1(A) \wedge \mathbb{D}_2(A)),$$

where $\mathbb{D}_1$ and $\mathbb{D}_2$ are different truth degrees. In this sense one can say that the $4\mathcal{P}$-environment provides a satisfactory framework for the tetralemma and the Cutuskoti.

References

1. Belnap, N.D.: A useful four-valued logic. In: Dunn, J.M., Epstein, G. (eds.) Modern Uses of Multiple-Valued Logic, pp. 5–37. D. Reidl Publishing Company, Dordrecht (1977)
2. Belnap, N.D.: How a computer should think. In: Omori, H., Wansing, H. (eds.) New Essays on Belnap-Dunn Logic, pp. 35–53. Synthese Library, no. 418 (2019)
3. Girard, J.-Y.: Proof Theory and Logical Complexity, Studies in Proof Theory, Monographs, no. 1, Bibliopolis, Pittsburgh (1987)
4. Hösli, B., Jäger, G.: About dome symmetries of negation. J. Symb. Logic **59**(2), 473–485 (1994)
5. Priest, G.: The logic of the Catuskoti. Comp. Philos. **1**(2), 24–54 (2010)
6. Schütte, K.: Syntactical and semantical properties of simple type theory. J. Symb. Logic **25**(4), 305–326 (1960)
7. Schütte, K.: Proof Theory, Grundlehren der mathematischen Wissenschaften, vol. 225. Springer, Berlin (1977)

Characterizing Data Dependencies Then and Now

Phokion G. Kolaitis and Andreas Pieris

Abstract Data dependencies are integrity constraints that the data of interest must obey. During the 1980s, János Makowsky made a number of contributions to the study of data dependencies; in particular, he was the first researcher to characterize data dependencies in terms of their structural properties. The goal of this article is to first present an overview of Makowsky's work on characterizing certain classes of data dependencies and then discuss recent developments concerning characterizations of broader classes of data dependencies.

1 Introduction

Since E.F. Codd introduced the relational data model in 1970 [9], logic and databases have enjoyed a continuous and fruitful interaction, so much so that it has been said that "logic and databases are inextricably intertwined" [13]. There are two main uses of logic in databases: the use of logic as a declarative language to express queries posed on databases and the use of logic as a specification language to express data dependencies, i.e., integrity constraints that the data of interest must obey.

Codd [10] had the key insight that first-order logic can be used as a database query language, which he called relational calculus. Furthermore, Codd showed that the expressive power of relational calculus (i.e., first-order logic on databases) coincides with that of relational algebra, which is a procedural database query language

P. G. Kolaitis (✉)
University of California Santa Cruz, Santa Cruz, CA, USA

IBM Research, San Jose, CA, USA
e-mail: kolaitis@ucsc.edu

A. Pieris
University of Edinburgh, Edinburgh, UK

University of Cyprus, Nicosia, Cyprus
e-mail: apieris@inf.ed.ac.uk

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_24

based on five basic operations on relations (union, difference, cartesian product, projection, and selection). As regards data dependencies, Codd [11] introduced the class of functional dependencies, which to date constitute the most widely used such class of constraints. Soon after this, researchers introduced and studied several different classes of data dependencies, such as inclusion dependencies, join dependencies, and multi-valued dependencies. This plethora of types of data dependencies raised the question of identifying a unifying formalism for them. Logic came to the rescue as it was eventually realized that the class of *embedded implicational dependencies* (EIDs) provides such a formalism [4, 14, 29]. Embedded implicational dependencies comprise two classes of first-order sentences, the class of tuple-generating dependencies (tgds) and the class of equality-generating dependencies (egds); the latter generalizes functional dependencies, while the former generalizes, among others, inclusion, join, and multi-valued dependencies. Informally, a tgd asserts that if some tuples belong to some relations, then some other tuples must belong to some (perhaps different) relations, while an egd asserts that if some tuples belong to some relations, then two of the values occurring in some of these tuples must be equal. More precisely, a tgd is a universal-existential first-order sentence of the form

$$\forall \bar{x} \forall \bar{y} \, (\phi(\bar{x}, \bar{y}) \; \rightarrow \; \exists \bar{z} \psi(\bar{x}, \bar{z})),$$

where $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$ are conjunctions of atomic formulas. Furthermore, an egd is a universal first-order sentence of the form

$$\forall \bar{x} \, (\phi(\bar{x}) \; \rightarrow \; x_i = x_j),$$

where $\phi(\bar{x})$ is a conjunction of atomic formulas, and x_i, x_j are variables in $\bar{x}$.

During the 1980s, János Makowsky made remarkable contributions to the study of data dependencies. The first of these contributions concerns the implication problem for data dependencies, which by the late 1970s had emerged as the central problem in this area and, in fact, had been called *the fundamental problem of databases* (see [21]). This problem takes as input a finite set Σ of data dependencies and a data dependency σ, and asks whether Σ logically implies σ. Here, logical implication has two different versions: the first version is that σ is true on every database (finite or infinite) on which every member of Σ is true; the second version is that this implication holds in the finite, i.e., σ is true on every finite database on which every member of Σ is true. In the case of functional dependencies, it was known that the two versions of the implication problem coincide and that there is a polynomial-time algorithm for solving this problem. It was not known, however, whether the implication problem for EIDs was decidable or undecidable. Makowsky showed that both versions of the implication problem for EIDs are undecidable. This result appeared in a joint paper with Chandra and Lewis [7], who had independently arrived at the same solution. The proof was via a reduction from the halting problem for two-counter machines. A different reduction that uses undecidability results from equational logic was obtained by Beeri and Vardi [3] around the same time.

Chandra et al. [7] also showed that the implication problem is decidable for *full* tgds and egds, where a tgd is full if it has no existential quantifiers, i.e., it is a first-order sentence of the form

$$\forall \bar{x} \forall \bar{y} \, (\phi(\bar{x}, \bar{y}) \rightarrow \psi(\bar{x})),$$

where $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x})$ are conjunctions of atomic formulas.

Makowsky's second contribution to the study of data dependencies concerns characterizations of classes of data dependencies. To put Makowsky's work in a proper context, let us recall three important topics in mathematical logic. First, in his invited address at the 1950 International Congress of Mathematicians, Tarski articulated his interest in characterizing notions of "metamathematical origin" in "purely mathematical terms" [25]. A series of results in model theory eventually led to the following characterization of definability in first-order logic: a class C of structures is definable by a finite set of first-order sentences if and only if both the class C and its complement $\overline{C}$ are closed under isomorphisms and ultraproducts (see [8]). Thus, the "metamathematical" notion of definability by a finite set of first-order sentences can be characterized in terms of purely algebraic closure properties. Second, there is a body of results in model theory, known as *preservation theorems*, that characterize when a first-order sentence is logically equivalent to a first-order sentence of a restricted syntactic form. The prototypical preservation theorem is the Łoś-Tarski Theorem, which asserts that a first-order sentence is logically equivalent to a universal first-order sentence if and only if it is preserved under substructures. Third, Lindström [19] characterized first-order logic as a maximal logic that has some mild closure properties and satisfies the compactness theorem and the Skolem-Löwenheim Theorem. This intrinsic characterization of first-order logic became the catalyst for the development of abstract model theory, which aims at characterizing logical formalisms in terms of their properties.

Leveraging his expertise in mathematical logic, Makowsky worked on characterizations of classes of data dependencies. In [20], he focused on full tgds and egds over uni-relational databases, that is, all data dependencies considered were over a database schema consisting of a single relation symbol. Note that much (but not all) of the work on data dependencies at that time was about uni-relational databases, perhaps because the first data dependencies studied were the functional dependencies, and they involve a single relation symbol. Makowsky obtained a preservation theorem that characterizes when a first-order sentence is logically equivalent to a finite set of full tgds and egds. This preservation theorem entails closure under subdatabases and closure under direct products. It is worth pointing out that, while this result is aligned with Tarski's goal to characterize metamathematical notions in purely mathematical terms, Makowsky questioned its relevance to databases because, as he put it, "algebraic operations alone is not what is needed in data base theory". In other words, Makowsky argued that characterizations of data dependencies should involve notions that are more ubiquitous in database practice. With these considerations in mind, Makowsky went on to establish the main result in [20], which is a Lindström-type theorem for the

class of all full tgds and egds. Specifically, he showed that the class of all full tgds and egds is a maximal collection of first-order sentences that satisfy a locality condition, called *securability*, and admit *Armstrong relations*, which are relations that encapsulate precisely the full tgds and egds that are logically implied by a set of full tgds and egds.

After this, Makowsky and Vardi [22] studied data dependencies over multi-relational databases, that is, data dependencies over a database schema that consists of finitely many relation symbols. Makowsky and Vardi [22] characterized when a class of databases is axiomatizable by a set of full tgds and egds, and also when it is axiomatizable by a finite set of full tgds and egds. These characterizations involve closure under subdatabases, closure under direct products, suitable locality properties, and some other structural properties that will be discussed in the sequel.

During the past two decades, data dependencies have found new uses and applications in several different areas of data management and knowledge representation, including data exchange, data integration, and specification of ontologies. As a result, the interest in characterizing data dependencies has been rekindled [12, 26]; furthermore, new preservation theorems about data dependencies have been obtained [30, 31]. In particular, using a novel notion of locality, characterizations of arbitrary tgds and egds were obtained in [12], thus going well beyond the characterizations of full tgds and egds established in [22]. In this article, we present some of these developments in detail and discuss the motivating role played by the concepts and results in [22].

2 Relational Databases and Data Dependencies

Let $\mathbf{C}$ and $\mathbf{V}$ be disjoint countably infinite sets of constants and variables, respectively. For an integer $n > 0$, we may write $[n]$ for the set $\{1, \ldots, n\}$.

Relational Databases A *(relational) schema* $\mathbf{S}$ is a finite set of relation symbols (or predicates) with positive arity; let $\mathsf{ar}(R)$ be the arity of R. A *(relational) database* D over $\mathbf{S} = \{R_1, \ldots, R_n\}$, or simply $\mathbf{S}$-*database*, is a tuple $(\mathsf{dom}(D), R_1^D, \ldots, R_n^D)$, where $\mathsf{dom}(D) \subseteq \mathbf{C}$ is a finite domain and $R_1^D, \ldots, R_n^D$ are relations over $\mathsf{dom}(D)$, i.e., $R_i^D \subseteq \mathsf{dom}(D)^{\mathsf{ar}(R_i)}$ for $i \in [n]$. In logical terms, a database is simply a relational structure with finite domain. A *fact* of D is an expression of the form $R_i(\bar{c})$, where $\bar{c} \in R_i^D$. Let $\mathsf{facts}(D)$ be the set of facts of D. The *active domain* of D, denoted $\mathsf{adom}(D)$, is the set of elements of $\mathsf{dom}(D)$ that occur in at least one fact of D. An $\mathbf{S}$-database D' is a *subdatabase* of D, denoted $D' \preceq D$, if $\mathsf{dom}(D') \subseteq \mathsf{dom}(D)$ and, for each $R \in \mathbf{S}$, we have that $R^{D'} = R^D_{|\mathsf{dom}(D')}$, where $R^D_{|\mathsf{dom}(D')}$ is the *restriction of* R^D *over* $\mathsf{dom}(D')$, i.e., $R^D_{|\mathsf{dom}(D')} = \left\{\bar{c} \in R^D \mid \bar{c} \in \mathsf{dom}(D')^{\mathsf{ar}(R)}\right\}$. A *homomorphism* from an $\mathbf{S}$-database D to an $\mathbf{S}$-database D' is a function $h : \mathsf{dom}(D) \to \mathsf{dom}(D')$ with the following property: for each $i \in [n]$ and for each tuple $\bar{c} = (c_1, \ldots, c_m) \in R_i^D$, we have that $h(\bar{c}) = (h(c_1), \ldots, h(c_m)) \in R_i^{D'}$. We write $h : D \to D'$ to denote that h is a homomorphism from D to D'. We also write

$h(\mathsf{facts}(D))$ for the set $\{R(h(\bar{c})) \mid R(\bar{c}) \in \mathsf{facts}(D)\}$. Finally, we say that D and D' are *isomorphic*, written $D \simeq D'$, if there is a bijection $h : \mathsf{dom}(D) \to \mathsf{dom}(D')$ such that h is a homomorphism from D to D' and h^{-1} is a homomorphism from D' to D.

Tuple-Generating Dependencies An atom over $\mathbf{S}$ is an expression of the form $R(\bar{v})$, where $R \in \mathbf{S}$ and $\bar{v}$ is an $\mathsf{ar}(R)$-tuple of variables from $\mathbf{V}$. A *tuple-generating dependency* (tgd) σ over a schema $\mathbf{S}$ is a constant-free first-order sentence

$$\forall \bar{x} \forall \bar{y} \, (\phi(\bar{x}, \bar{y}) \ \rightarrow \ \exists \bar{z} \, \psi(\bar{x}, \bar{z})) \, ,$$

where $\bar{x}$, $\bar{y}$, $\bar{z}$ are tuples of variables of $\mathbf{V}$, $\phi(\bar{x}, \bar{y})$ is a (possibly empty) conjunction of atoms over $\mathbf{S}$, $\psi(\bar{x}, \bar{z})$ is a non-empty conjunction of atoms over $\mathbf{S}$, and every variable in $\bar{x}$ occurs in both $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$. For brevity, we write σ as $\phi(\bar{x}, \bar{y}) \ \rightarrow \ \exists \bar{z} \, \psi(\bar{x}, \bar{z})$. We refer to $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$ as the *body* and *head* of σ, denoted $\mathsf{body}(\sigma)$ and $\mathsf{head}(\sigma)$, respectively. By an abuse of notation, we may treat a tuple of variables as a set of variables, and we may also treat a conjunction of atoms as a set of atoms. A tgd is called *full* if it has no existentially quantified variables. Let us stress that, since the head of a tgd is by definition non-empty, full tgds mention at least one universally quantified variable, which in turn implies that their body is non-empty. An $\mathbf{S}$-database D *satisfies a tgd* σ as the one above, written $D \models \sigma$, if the following holds: whenever there is a function $h : \bar{x} \cup \bar{y} \ \rightarrow \ \mathsf{dom}(D)$ such that $h(\phi(\bar{x}, \bar{y})) \subseteq \mathsf{facts}(D)$ (as usual, we write $h(\phi(\bar{x}, \bar{y}))$ for the set $\{R(h(\bar{v})) \mid R(\bar{v}) \in \phi(\bar{x}, \bar{y})\}$), then there exists an extension h' of h such that $h'(\psi(\bar{x}, \bar{z})) \subseteq \mathsf{facts}(D)$. The $\mathbf{S}$-database D *satisfies a set* Σ *of tgds*, written $D \models \Sigma$, if $D \models \sigma$ for each $\sigma \in \Sigma$; in this case, we say that D is a *model* of Σ.

Equality-Generating Dependencies An *equality-generating dependency* (egd) θ over a schema $\mathbf{S}$ is a constant-free first-order sentence

$$\forall \bar{x} \left(\phi(\bar{x}) \ \rightarrow \ x_i = x_j\right) ,$$

where $\bar{x}$ is a tuple of variables of $\mathbf{V}$, $\phi(\bar{x})$ is a (non-empty) conjunction of atoms over $\mathbf{S}$, and $x_i, x_j \in \bar{x}$. For brevity, we write θ as $\phi(\bar{x}) \rightarrow x_i = x_j$. We refer to $\phi(\bar{x})$ as the *body* of θ, denoted $\mathsf{body}(\theta)$. An $\mathbf{S}$-database D *satisfies an egd* θ as the one above, written $D \models \theta$, if, whenever there is a function $h : \bar{x} \rightarrow \mathsf{dom}(D)$ such that $h(\phi(\bar{x})) \subseteq \mathsf{facts}(D)$, then $h(x_i) = h(x_j)$. The $\mathbf{S}$-database D *satisfies a set* Σ *of egds*, written $D \models \Sigma$, if $D \models \theta$ for each egd $\theta \in \Sigma$, and we say that D is a *model* of Σ.

Finite Axiomatizability Let C be a collection of databases. We say that C is *finitely axiomatizable* by tgds and egds if there are finite sets Σ_T and Σ_E of tgds and egds, respectively, such that for every database D, it holds that $D \in C$ if and only if $D \models \Sigma_T$ and $D \models \Sigma_E$. Note that in the literature we also have the notion of axiomatizability, where the sets Σ_T and Σ_E can be infinite, i.e., if we allow Σ_T and

Σ_E to be infinite, then we say that C is *axiomatizable* by tgds and egds. In this paper, we focus on finite axiomatizability since it is more relevant to database practice. In what follows, we consider only collections C of databases that are *closed under isomorphisms*, i.e., if $D \in C$ and D' is a database such that $D \simeq D'$, then $D' \in C$. In other words, for a collection of databases, we silently assume that it is closed under isomorphisms.

3 Finite Axiomatizability by Full TGDs and EGDs

As discussed in Sect. 1, one of Makowsky's main contributions to the study of data dependencies concerns characterizations of classes of data dependencies. The result that stands out in this context, which we will present here, is the characterization of finite sets of *full dependencies* obtained by Makowsky and Vardi in 1986 [22, Theorem 5]. By definition, a full dependency is a first-order sentence of the form $\forall\bar{x}\forall\bar{y}\,(\phi(\bar{x},\bar{y}) \rightarrow \psi(\bar{x}))$, where $\phi(\bar{x},\bar{y})$ is a non-empty conjunction of relational atoms and $\psi(\bar{x})$ is a non-empty conjunction of relational atoms and equality atoms. Clearly, every full dependency is logically equivalent to a finite set of full tgds and egds. Consequently, given a full dependency σ, there is a finite set Σ of full tgds and egds such that, for every database D, it holds that D satisfies σ if and only if D satisfies the tgds and egds of Σ. For this reason, in what follows we will state Theorem 5 of [22] in terms of finite sets of full tgds and egds.

As we shall see, Theorem 5 of [22] involves the property of closure under direct products, which is formally defined below. However, as already mentioned in Sect. 1, Makowsky questioned the relevance to databases of closure under algebraic operations, such as closure under direct products. With this in mind, we further present an alternative characterization of finite sets of full tgds and egds, which uses the property of closure under intersections instead of closure under direct products.

3.1 The Characterization by Makowsky and Vardi

We first introduce the properties that are needed for the characterization of interest and then recall the characterization by Makowsky and Vardi from [22].

3.1.1 Model-Theoretic Properties

In the sequel, fix an arbitrary schema $\mathbf{S} = \{R_1, \ldots, R_\ell\}$.

1-Criticality An $\mathbf{S}$-database $D = (\mathsf{dom}(D), R_1^D, \ldots, R_\ell^D)$ is said to be 1-*critical* if $|\mathsf{dom}(D)| = 1$, that is, the domain of D consists of a single element $c \in \mathbf{C}$, and $R_i^D = \{c\}^{\mathsf{ar}(R_i)}$ for each $i \in [\ell]$, i.e., $\mathsf{facts}(D) = \{R_i(c, \ldots, c) \mid i \in [\ell]\}$. Let us

clarify that in [22] 1-critical databases are called *trivial* databases. A collection of databases over $\mathbf{S}$ is 1-*critical* if it contains an 1-critical $\mathbf{S}$-database.

Domain Independence A collection C of databases over $\mathbf{S}$ is *domain independent* if, for every $\mathbf{S}$-database $D \in C$ and every database D' with $\mathsf{facts}(D) = \mathsf{facts}(D')$, it holds that $D' \in C$. In other words, C is domain independent if, for every two $\mathbf{S}$-databases that have the same set of facts, but not necessarily the same domain, either both are in C or neither of them is in C.

Modularity A collection C of databases over $\mathbf{S}$ is n-*modular*, for $n \geq 0$, if, for every $\mathbf{S}$-database $D \notin C$, there is an $\mathbf{S}$-database $D' \preceq D$ with $|\mathsf{dom}(D')| \leq n$ such that $D' \notin C$. We say that C is *modular* if it is n-modular for some $n \geq 0$. Roughly, n-modularity provides a "small" database with at most n domain elements as a witness to why a database does not belong to a collection of databases. Let us clarify that in [22], n-modularity is called n-*locality*. However, here we adopt the term n-modularity, which has been already used in [26] for a similar notion in the context of schema mappings, in order to avoid any confusion with the notion of (n, m)-locality from [12], which will be used in the next section to handle arbitrary tgds and egds.

Closure Under Subdatabases A collection C of databases is *closed under subdatabases* if, for every $\mathbf{S}$-database $D \in C$, it holds that $D' \in C$ for every $D' \preceq D$.

Closure Under Direct Products Assume that $D = (\mathsf{dom}(D), R_1^D, \ldots, R_\ell^D)$ and $D' = (\mathsf{dom}(D'), R_1^{D'}, \ldots, R_\ell^{D'})$ are two databases over a schema $\mathbf{S}$. The *direct product* of D and D', denoted $D \otimes D'$, is defined as the $\mathbf{S}$-database

$$D'' = (\mathsf{dom}(D''), R_1^{D''}, \ldots, R_\ell^{D''}),$$

where $\mathsf{dom}(D'') = \mathsf{dom}(D) \times \mathsf{dom}(D')$, and, for each $i \in [\ell]$, $R_i^{D''}$ is the relation

$$\left\{ \left((a_1, b_1), \ldots, (a_{r_i}, b_{r_i}) \right) \mid \left(a_1, \ldots, a_{r_i} \right) \in R_i^D \text{ and } \left(b_1, \ldots, b_{r_i} \right) \in R_i^{D'} \right\},$$

where $r_i = \mathsf{ar}(R_i)$ is the arity of the relation symbol R_i. A collection C of databases over $\mathbf{S}$ is *closed under direct products* if, for every two $\mathbf{S}$-databases $D, D' \in C$, it holds that $D \otimes D' \in C$.

3.1.2 The Characterization

We can now provide the characterization of interest from [22]. We do not discuss here the proof of the characterization since it will be derived from the more general characterization of finite sets of arbitrary tgds and egds presented in Sect. 4.

Theorem 1 (Theorem 5 in [22]) *Let C be a collection of databases. The following statements are equivalent:*

1. *C is finitely axiomatizable by full tgds and egds.*
2. *C is* 1*-critical, domain independent, modular, closed under subdatabases, and closed under direct products.*

3.2 The Alternative Characterization

We will present an alternative characterization of finite sets of full tgds and egds that avoids the use of closure under direct products. This is done by replacing in the characterization of Theorem 1 the property of closure under direct products with the property of closure under intersections defined as follows. Consider a schema $\mathbf{S} = \{R_1, \ldots, R_\ell\}$. The *intersection* of the $\mathbf{S}$-databases $D = (\mathsf{dom}(D), R_1^D, \ldots, R_\ell^D)$ and $D' = (\mathsf{dom}(D'), R_1^{D'}, \ldots, R_\ell^{D'})$, denoted $D \cap D'$, is the database

$$(\mathsf{dom}(D) \cap \mathsf{dom}(D'), R_1^D \cap R_1^{D'}, \ldots, R_\ell^D \cap R_\ell^{D'}).$$

A collection C of databases over $\mathbf{S}$ is *closed under intersections* if, for every two $\mathbf{S}$-databases $D, D' \in C$, it holds that $D \cap D' \in C$.

Before stating and proving the promised characterization, let us briefly discuss the two tools that will be used in the proof: Robinson's *method of diagrams* and McKinsey's *method of eliminating disjunctions*. In his Ph.D. thesis in 1949, Abraham Robinson introduced the notion of a *diagram* of a relational structure, which, together with its variants, became a standard tool in constructing models that contain an isomorphic copy of some structure of interest. In model theory, the method of diagrams is typically combined with the compactness theorem to construct such models (see [8, 24]). In characterizing data dependencies, however, the method of diagrams is combined with closure properties and with suitable notions of locality, instead of the compactness theorem. In 1943, J.C.C. McKinsey [23] introduced the method of eliminating disjunctions, which makes it possible to show that, when a class of structures satisfies certain closure properties, then, on such a class of structures, a first-order sentence of the form $\forall \bar{x}(\phi(\bar{x}) \rightarrow \bigvee_{i=1}^{k} \psi_i(\bar{x}))$ implies one of the sentences $\forall \bar{x}(\phi(\bar{x}) \rightarrow \psi_j(\bar{x}))$, for some $j \in [k]$; as a consequence of this, on such a class of structures, the sentence $\forall \bar{x}(\phi(\bar{x}) \rightarrow \bigvee_{i=1}^{k} \psi_i(\bar{x}))$ is equivalent to the sentence $\forall \bar{x}(\phi(\bar{x}) \rightarrow \psi_j(\bar{x}))$. McKinsey's original application of this method used closure under direct products, but the method is flexible enough to adapt to other closure properties, such as closure under intersections. It should be noted that the method of diagrams and the method of eliminating disjunctions were already utilized by Makowsky and Vardi [22] in the proof of Theorem 1.

Theorem 2 *Let C be a collection of databases. The following are equivalent:*

1. *C is finitely axiomatizable by full tgds and egds.*
2. *C is* 1*-critical, domain independent, modular, closed under subdatabases, and closed under intersections.*

Proof The direction (1) $\Rightarrow$ (2) is easy and we leave it as an exercise. We focus on the direction (2) $\Rightarrow$ (1). We first need to introduce a couple of auxiliary notions.

A *disjunctive dependency* (dd) δ over a schema $\mathbf{S}$ is a constant-free sentence

$$\forall \bar{x} \left(\phi(\bar{x}) \rightarrow \bigvee_{i=1}^{k} \psi_i(\bar{x}_i) \right),$$

where $\bar{x}$ is a tuple of variables of $\mathbf{V}$, the expression $\phi(\bar{x})$ is a (non-empty) conjunction of atoms over $\mathbf{S}$, and, for each $i \in [k]$, $\bar{x}_i \subseteq \bar{x}$ and the expression $\psi_i(\bar{x}_i)$ is either an equality formula $y = z$ with $\bar{x}_i = \{y, z\}$, or an atom over $\mathbf{S}$. An $\mathbf{S}$-database D *satisfies the dd* δ, written $D \models \delta$, if, whenever there is a function $h : \bar{x} \rightarrow \mathsf{dom}(D)$ with $h(\phi(\bar{x})) \subseteq \mathsf{facts}(D)$, then there is $i \in [k]$ such that, if $\psi_i(\bar{y}_i)$ is $y = z$, then $h(y) = h(z)$, and if $\psi_i(\bar{y}_i)$ is $R(\bar{x}_i)$, then $h(R(\bar{x}_i)) \in \mathsf{facts}(D)$. We say that the database D *satisfies a set* Σ of dds or that D is a *model* of Σ, written $D \models \Sigma$, if $D \models \delta$ for each $\delta \in \Sigma$.

We further need the notion of the diagram of a database. Let D be an $\mathbf{S}$-database such that $\mathsf{facts}(D) \neq \emptyset$. Let A be the set of atomic formulas that can be formed using predicates from $\mathbf{S}$ and constants from $\mathsf{dom}(D)$. The *diagram of* D, denoted Δ_D, is

$$\bigwedge_{\alpha \in \mathsf{facts}(D)} \alpha \ \wedge \bigwedge_{\alpha \in A \setminus \mathsf{facts}(D)} \neg\alpha \ \wedge \bigwedge_{c,d \in \mathsf{dom}(D) \text{ and } c \neq d} \neg(c = d).$$

Note that, since $\mathsf{facts}(D) \neq \emptyset$, the conjunction $\bigwedge_{\alpha \in \mathsf{facts}(D)} \alpha$ is non-empty. □

Let C be collection of databases over $\mathbf{S}$ possessing the properties in the second statement of Theorem 2. In particular, since C is modular, there is some integer $n \geq 0$ such that C is n-modular. Moreover, since C is closed under subdatabases, we have that C contains the empty database, hence n must be a positive integer.

Let $\Sigma^{\vee}$ be the set of all dds over $\mathbf{S}$ with at most n variables that are satisfied by every database $D \in C$. It is clear that $\Sigma^{\vee}$ is finite (up to logical equivalence). We proceed to show the following technical lemma.

Lemma 1 *For each* $\mathbf{S}$*-database* D*, we have that* $D \in C$ *if and only* $D \models \Sigma^{\vee}$.

Proof of Lemma 1 Fix an $\mathbf{S}$-database D. From the definition of $\Sigma^{\vee}$, $D \in C$ implies $D \models \Sigma^{\vee}$. For the other direction, we will show that if $D \notin C$, then $D \not\models \Sigma^{\vee}$. Assume that $D \notin C$. Since C is n-modular, there exists an $\mathbf{S}$-database D_n such that $D_n \preceq D$, $|\mathsf{dom}(D_n)| \leq n$, and $D_n \notin C$. Since C is domain independent, we can assume that $\mathsf{dom}(D_n) = \mathsf{adom}(D_n)$. We will show that $D_n \not\models \Sigma^{\vee}$, which in turn implies that $D \not\models \Sigma^{\vee}$ since every sentence in $\Sigma^{\vee}$ is universal and it is well-known that universal first-order sentences are preserved under subdatabases.

To show that $D_n \not\models \Sigma^{\vee}$, it suffices to find a dd $\delta \in \Sigma^{\vee}$ such that $D_n \not\models \delta$. We will use the diagram of D_n to construct this desired δ. Let Δ_{D_n} be the diagram of D_n, and let $\Phi_{D_n}(\bar{x})$ be the quantifier-free first-order formula obtained from Δ_{D_n}

by replacing each $c \in \mathsf{dom}(D_n)$ with a new variable $x_c \in \mathbf{V}$. We claim that the following statements are true:

1. $D_n \models \exists \bar{x}\, \Phi_{D_n}(\bar{x})$.
2. For every $D' \in C$, we have that $D' \models \neg\exists \bar{x}\, \Phi_{D_n}(\bar{x})$.
3. The sentence $\neg\exists \bar{x}\, \Phi_{D_n}(\bar{x})$ is logically equivalent to a dd δ.

The first statement is obviously true by the construction of the sentence $\exists \bar{x}\, \Phi_{D_n}(\bar{x})$ (each existential quantifier $\exists x_c$ in $\exists \bar{x}$ can be witnessed by the element c in $\mathsf{dom}(D_n)$). For the second statement, if there is some database $D' \in C$ such that $D' \models \exists \bar{x}\, \Phi_{D_n}(\bar{x})$, then D' must contain a subdatabase D'' that is isomorphic to D_n. Since C is closed under subdatabases and isomorphisms, it follows that $D_n \in C$, which is a contradiction. For the third statement, observe first that $\mathsf{facts}(D_n) \neq \emptyset$ because D_n is not in C, while C, being closed under subdatabases, contains the empty database. This implies that Δ_{D_n} contains at least one positive relational atom α; hence, the conjunction $\bigwedge_{\alpha \in \mathsf{facts}(D_n)} \alpha$ is non-empty. Furthermore, since D_n is not in C and since C is 1-critical and closed under isomorphisms, we conclude that D_n is not an 1-critical database. This implies that either $|\mathsf{dom}(D_n)| \geq 2$ or $|\mathsf{dom}(D_n)| = 1$ and there is a relation symbol $R \in \mathbf{S}$ such that R^{D_n} is empty. Therefore, Δ_{D_n} contains either the negation of an equality atom β or the negation of a relational atom γ. Consequently, $\neg\exists \bar{x}\, \Phi_{D_n}(\bar{x})$ is logically equivalent to a dd δ. Therefore, all three statements have been established.

Taken together, these three statements imply that δ is a dd in $\Sigma^{\vee}$ that is false on D_n, and hence, $D_n \not\models \Sigma^{\vee}$. This completes the proof of Lemma 1. □

Observe that the property of closure under intersections was not used in the proof of Lemma 1. This property will be used in the proof of the next technical lemma.

Lemma 2 *Assume that $\forall \bar{x}(\phi(\bar{x}) \rightarrow \bigvee_{i=1}^{k} \alpha_i(\bar{x}_i))$ is a disjunctive dependency that belongs to $\Sigma^{\vee}$. There is $j \in [k]$ such that $\forall \bar{x}(\phi(\bar{x}) \rightarrow \alpha_j(\bar{x}_j))$ also belongs to $\Sigma^{\vee}$.*

Proof of Lemma 2 Note that for every $j \in [k]$, the sentence $\forall \bar{x}(\phi(\bar{x}) \rightarrow \alpha_j(\bar{x}_j))$ is either a full tgd or an egd, hence it is a disjunctive dependency. Towards a contradiction, assume that none of these sentences belongs to $\Sigma^{\vee}$. Since $\Sigma^{\vee}$ consists of all disjunctive dependencies that are satisfied by every database in C, it follows that for every $j \in [k]$, there is a database $D_j \in C$ such that $D_j \not\models \forall \bar{x}(\phi(\bar{x}) \rightarrow \alpha_j(\bar{x}_j))$. Hence, for every $j \in [k]$, there is a tuple $\bar{b}^j$ of elements in $\mathsf{dom}(D_j)$ such that $D_j \models \phi(\bar{b}^j) \wedge \neg\alpha_j(\bar{b}^j_j)$. Since C is closed under isomorphisms, we may assume that these tuples are the same tuple $\bar{b}$, i.e., we may assume that for every j and ℓ with $j, \ell \in [k]$, we have that $\bar{b}^j = \bar{b}^\ell = \bar{b}$. Let $D_\cap$ be the intersection of the databases D_j, that is, $D_\cap = \bigcap_{j \in [k]} D_j$. Since C is closed under intersections, we have that $D_\cap$ belongs to C; consequently, we have that $D_\cap \models \forall \bar{x}(\phi(\bar{x}) \rightarrow \bigvee_{i=1}^{k} \alpha_i(\bar{x}_i))$. Since the elements of the tuple $\bar{b}$ belong to $\mathsf{dom}(D_j)$ for every $j \in [k]$, it follows that they also belong to $\mathsf{dom}(D_\cap)$. Therefore, and since also $\phi(\bar{b})$ holds, there is some $j \in [k]$ such that $\alpha_j(\bar{b}_j)$ holds, which contradicts the fact that $\neg\alpha_j(\bar{b}_j)$ holds

(since $\bar{b}$ witnesses that $D_j \not\models \forall\bar{x}(\phi(\bar{x}) \rightarrow \alpha_j(\bar{x}_j))$. This completes the proof that there is a $j \in [k]$ such that the formula $\forall\bar{x}(\phi(\bar{x}) \rightarrow \alpha_j(\bar{x}_i))$ belongs to $\Sigma^\vee$. □

With the two preceding lemmas at hand, we are now ready to complete the proof of Theorem 2. Let Σ be the set of full tgds and egds in $\Sigma^\vee$, i.e.,

$$\Sigma = \left\{\delta \in \Sigma^\vee \mid \delta \text{ is a full tgd or an egd}\right\}.$$

Note that Σ is a finite set because it is a subset of the finite set $\Sigma^\vee$. We proceed to show that, for each $\mathbf{S}$-database D, it holds that $D \in C$ if and only if $D \models \Sigma$.

($\Rightarrow$) If $D \in C$, then, by the definition of $\Sigma^\vee$, we have that $D \models \Sigma^\vee$; hence, $D \models \Sigma$ since $\Sigma \subseteq \Sigma^\vee$.

($\Leftarrow$) Assume now that D is an $\mathbf{S}$-database such that $D \models \Sigma$. We need to show that $D \in C$. By Lemma 1, it suffices to show that $D \models \Sigma^\vee$. Let $\forall\bar{x}(\phi(\bar{x}) \rightarrow \bigvee_{i=1}^{k} \alpha_i(\bar{x}_i))$ be a disjunctive dependency in $\Sigma^\vee$. By Lemma 2, there is $j \in [k]$ such that $\forall\bar{x}(\phi(\bar{x}) \rightarrow \alpha_j(\bar{x}_j))$ belongs to $\Sigma^\vee$. Since this sentence is either a full tgd or an egd, we have that it belongs to Σ; hence, $D \models \forall\bar{x}(\phi(\bar{x}) \rightarrow \alpha_j(\bar{x}_j))$. It is obvious that $\forall\bar{x}(\phi(\bar{x}) \rightarrow \alpha_j(\bar{x}_j))$ logically implies the disjunctive dependency $\forall\bar{x}(\phi(\bar{x}) \rightarrow \bigvee_{i=1}^{k} \alpha_i(\bar{x}_i))$, and therefore, $D \models \forall\bar{x}(\phi(\bar{x}) \rightarrow \bigvee_{i=1}^{k} \alpha_i(\bar{x}_i))$. This completes the proof of the above claim.

Thus, we have established that C is axiomatizable by the set Σ, which is a finite set of full tgds and egds. This completes the proof of Theorem 2. □

4 Finite Axiomatizability by TGDs and EGDs

The fact that Theorems 1 and 2 deal only with full tgds, i.e., tgds without existentially quantified variables, leads to the following natural question: is there an analogous characterization for arbitrary tgds and egds? Such a characterization has been recently established in [12] for the case of arbitrary (finite or infinite) relational structures. In what follows, we present the characterization from [12] for databases, i.e., finite relational structures. Apart from the standard properties of 1-criticality and closure under direct products, already used in Sect. 3, we are going to use a novel locality property. The latter locality property was introduced in [12] for arbitrary structures, but we are going to adapt it for databases. We proceed to introduce this notion of locality and show that every collection C of databases that is finitely axiomatizable by tgds and egds enjoys this property. We then present the characterization of arbitrary tgds and egds and discuss how it allows us to derive the characterization for full tgds and egds from [22], namely Theorem 1.

4.1 Locality

The locality property of interest relies on the notion of local embedding of a collection of databases in a database. Roughly speaking, a collection C of databases over a schema $\mathbf{S}$ is locally embeddable in an $\mathbf{S}$-database D if, for every subdatabase E of D with a bounded number of active domain elements (i.e., domain elements that occur in $\mathsf{facts}(E)$), we can find a database $D_E \in C$ such that every local neighbour of E in D_E (i.e., subdatabases of D_E that contain E and have a bounded number of additional active domain elements that do not occur in $\mathsf{facts}(E)$), can be embedded in D while preserving E. We call the collection C of databases local if, for every $\mathbf{S}$-database D, the fact that C is locally embeddable in D implies that D belongs to C. We proceed to formalize the above high-level description.

Recall that the active domain of a database D, denoted $\mathsf{adom}(D)$, is the set of elements of $\mathsf{dom}(D)$ that occur in at least one fact of D. Consider an $\mathbf{S}$-database D' and an $\mathbf{S}$-database $D'' \subseteq D'$, i.e., $\mathsf{facts}(D'') \subseteq \mathsf{facts}(D')$. The *$m$-neighbourhood of D'' in D'* is defined as the set of $\mathbf{S}$-databases

$$\{E \mid \mathsf{adom}(D'') \subseteq \mathsf{adom}(E),\ E \preceq D' \text{ and } |\mathsf{adom}(E)| \leq |\mathsf{adom}(D'')| + m\},$$

that is, all the subdatabases of D' such that their facts contain constants from $\mathsf{adom}(D'')$ and at most m additional elements not occurring in $\mathsf{adom}(D'')$. It is easy to verify that the m-neighbourhood of D'' in D' is actually the set of $\mathbf{S}$-databases

$$\{E \mid D'' \subseteq E,\ E \preceq D' \text{ and } |\mathsf{adom}(E)| \leq |\mathsf{adom}(D'')| + m\},$$

i.e., the set all subdatabases of D' that contain D'' and their facts mention at most m additional elements not occurring in the facts of D''. Indeed, if $D'' \subseteq D'$ and $E \preceq D'$, then we have that $\mathsf{adom}(D'') \subseteq \mathsf{adom}(E)$ if and only if $D'' \subseteq E$.

Consider a collection C of databases over $\mathbf{S}$ and an $\mathbf{S}$-database D. For integers $n, m \geq 0$, we say that *C is (n, m)-locally embeddable in D* if, for every $E \preceq D$ with $|\mathsf{adom}(E)| \leq n$, there is $D_E \in C$ such that $E \subseteq D_E$, and for every D' in the m-neighbourhood of E in D_E, there is a function $h : \mathsf{adom}(D') \to \mathsf{adom}(D)$ such that h is the identity on $\mathsf{adom}(E)$ and $h(\mathsf{facts}(D')) \subseteq \mathsf{facts}(D)$. The notion of (n, m)-local embeddability is illustrated in Fig. 1; the circles represent the set of facts of the databases. Clearly, for every $\mathbf{S}$-database E, we have that $\mathsf{adom}(E) \subseteq \mathsf{dom}(E)$, and this containment may be a proper one. Observe, however, that the notion of a collection C of databases being (n, m)-locally embeddable in D only depends on $\mathsf{adom}(E)$ of E and the set of facts of E, and not on $\mathsf{dom}(E)$ of E. In turn, this implies that, when showing that C is (n, m)-locally embeddable in D, it suffices to focus our attention on subdatabases E of D such that $\mathsf{adom}(E) = \mathsf{dom}(E)$. We state this observation as a separate lemma, which will be used later in the paper.

Lemma 3 *Consider a collection C of databases, integers $n, m \geq 0$, and a database D. The following statement are equivalent:*

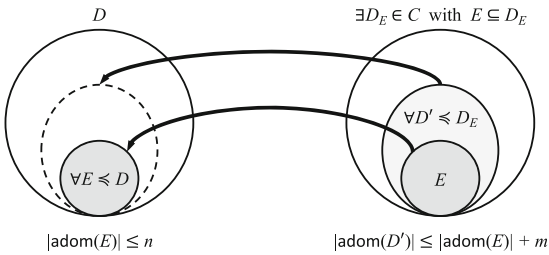


Fig. 1 C is (n, m)-locally embeddable in D

1. *C is (n, m)-locally embeddable in D.*
2. *For every $E \preceq D$ with $\mathsf{adom}(E) = \mathsf{dom}(E)$ and $|\mathsf{adom}(E)| \leq n$, there is $D_E \in C$ such that $E \subseteq D_E$, and for every D' in the m-neighbourhood of E in D_E, there is a function $h : \mathsf{adom}(D') \to \mathsf{adom}(D)$ such that h is the identity on $\mathsf{adom}(E)$ and $h(\mathsf{facts}(D')) \subseteq \mathsf{facts}(D)$.*

We are now ready to give the definition of the central notion of locality.

Definition 1 (Locality) A collection C of databases over $\mathbf{S}$ is (n, m)-*local*, for $n, m \geq 0$, if, for every $\mathbf{S}$-database D, the following holds: C is (n, m)-locally embeddable in D implies $D \in C$. We say that C is *local* if it is (n, m)-local for some $n, m \geq 0$.

We can now show the following technical lemma that essentially states that every collection of databases that is finitely axiomatizable by tgds with at most n universally and m existentially quantified variables, and egds with at most n universally quantified variables, is (n, m)-local. A tgd is called (n, m)-*tgd*, for $n \geq 0$ and $m > 0$, or $n > 0$ and $m \geq 0$, if it mentions at most n universally and m existentially quantified variables. Moreover, an egd is called n-*egd*, for $n > 0$, if it mentions at most n universally quantified variables. For notational convenience, we also define the corner cases of $(0, 0)$-tgd and 0-egd as the truth value true, i.e., as a tautology.

Lemma 4 *For integers $n, m \geq 0$, every collection C of databases that is finitely axiomatizable by (n, m)-tgds and n-egds is (n, m)-local.*

Proof Let C be a collection of databases over a schema $\mathbf{S}$ that is finitely axiomatizable by (n, m)-tgds and n-egds. By definition, there is a finite set Σ_T of (n, m)-tgds and a finite set Σ_E of n-egds such that, for every $\mathbf{S}$-database D, we have that $D \in C$ iff $D \models \Sigma_T$ and $D \models \Sigma_E$. Consider an $\mathbf{S}$-database D and assume that C is (n, m)-locally embeddable in D. We proceed to show that $D \in C$, i.e., $D \models \Sigma_T$ and $D \models \Sigma_E$.

We first show that $D \models \Sigma_T$. Consider a tgd $\sigma \in \Sigma_T$, other than the $(0, 0)$-tgd that is trivially satisfied by D, of the form $\phi(\bar{x}, \bar{y}) \to \exists \bar{z}\, \psi(\bar{x}, \bar{z})$, and assume that there exists a function $h : \bar{x} \cup \bar{y} \to \mathsf{dom}(D)$ such that $h(\phi(\bar{x}, \bar{y})) \subseteq \mathsf{facts}(D)$. We show that there exists an extension λ of h such that $\lambda(\psi(\bar{x}, \bar{z})) \subseteq \mathsf{facts}(D)$; the existence of λ is illustrated in Fig. 2. Let E be the database $(\mathsf{dom}(E), R_1^E, \ldots, R_\ell^E)$,

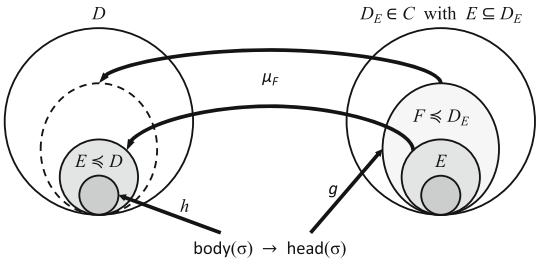


Fig. 2 The function $\lambda = \mu_F \circ g$ in the proof of Lemma 4

where $\mathsf{dom}(E)$ is the set of constants occurring in $h(\phi(\bar{x}, \bar{y}))$, and, for each $i \in [\ell]$, $R_i^E = R_i^D{}_{|E}$. It is clear that $E \preceq D$ with $|\mathsf{adom}(E)| \leq n$ since $\phi(\bar{x}, \bar{y})$ mentions at most n variables. Since, by hypothesis, C is (n, m)-locally embeddable in D, we get that there exists $D_E \in C$ such that $E \subseteq D_E$, and, for every D' in the m-neighbourhood of E in D_E, there is a function $\mu_{D'} : \mathsf{adom}(D') \to \mathsf{adom}(D)$, which is the identity on $\mathsf{adom}(E)$, such that $\mu_{D'}(\mathsf{facts}(D')) \subseteq \mathsf{facts}(D)$. It is clear that $h(\phi(\bar{x}, \bar{y})) \subseteq \mathsf{facts}(D_E)$. Since $D_E \in C$, or, equivalently, $D_E \models \Sigma$, there exists an extension g of h such that $g(\psi(\bar{x}, \bar{z})) \subseteq \mathsf{facts}(D_E)$.

Let $F = (\mathsf{dom}(F), R_1^F, \ldots, R_\ell^F)$, where $\mathsf{dom}(F)$ are the constants occurring in $h(\phi(\bar{x}, \bar{y})) \cup g(\psi(\bar{x}, \bar{z}))$, and, for each $i \in [\ell]$, $R_i^F = R_i^{D_E}{}_{|\mathsf{dom}(F)}$. It is clear that F is in the m-neighbourhood of E in D_E since $\bar{z}$ has at most m variables. Therefore, there is a function $\mu_F : \mathsf{adom}(F) \to \mathsf{adom}(D)$, which is the identity on $\mathsf{adom}(E)$, such that $\mu_F(\mathsf{facts}(F)) \subseteq \mathsf{facts}(D)$. Consider the function $\lambda = \mu_F \circ g$. Since g is an extension of h, and μ_F is the identity on the elements occurring in $h(\phi(\bar{x}, \bar{y}))$, we get that $\lambda(v) = h(v)$ for each variable v in $\phi(\bar{x}, \bar{y})$, and thus, λ is an extension of h. Moreover, since $g(\psi(\bar{x}, \bar{z})) \subseteq \mathsf{facts}(F)$, we get that $\lambda(\psi(\bar{x}, \bar{z})) \subseteq \mathsf{facts}(D)$.

We now show that $D \models \Sigma_E$. Consider an egd $\theta \in \Sigma_E$, other than the 0-egd that is trivially satisfied by D, of the form $\phi(\bar{x}) \to x_i = x_j$, and assume there exists a function $h : \bar{x} \to \mathsf{dom}(D)$ such that $h(\phi(\bar{x})) \subseteq \mathsf{facts}(D)$. We show that $h(x_i) = h(x_j)$. Let $E = (\mathsf{dom}(E), R_1^E, \ldots, R_\ell^E)$, where $\mathsf{dom}(E)$ is the set of constants occurring in $h(\phi(\bar{x}))$, and, for each $i \in [\ell]$, $R_i^E = R_i^D{}_{|E}$. It is clear that $E \preceq D$ with $|\mathsf{adom}(E)| \leq n$ since $\phi(\bar{x})$ mentions at most n variables. Since, by hypothesis, C is (n, m)-locally embeddable in D, there exists $D_E \in C$ such that $E \subseteq D_E$. Observe that $E \subseteq D_E$ implies that $h(\phi(\bar{x})) \subseteq E_E$. Since $D_E \models \Sigma_E$, we get that $h(x_i) = h(x_j)$. □

It is useful to observe that locality implies domain independence.

Lemma 5 *If a collection of databases is local, then it is also domain independent.*

Proof Assume that C is a collection of databases over $\mathbf{S}$ that is (n, m)-local for $n, m \geq 0$, and let D be an $\mathbf{S}$-database in C. Consider an $\mathbf{S}$-database D' such that $\mathsf{facts}(D) = \mathsf{facts}(D')$. We need to show that $D' \in C$. This is done by showing that C is (n, m)-locally embeddable in D', which implies that $D' \in C$ since, by hypothesis, C is (n, m)-local. Consider an $\mathbf{S}$-database $D'' \preceq D'$ with $|\mathsf{adom}(D'')| \leq n$. Since $\mathsf{facts}(D) = \mathsf{facts}(D')$, it is clear that $D'' \subseteq D$. Since

$D \in C$, it suffices to show that, for every E in the m-neighbourhood of D'' in D, there is a function $h_E : \mathsf{adom}(E) \to \mathsf{adom}(D')$, which is the identity on $\mathsf{adom}(D'')$, such that $h_E(\mathsf{facts}(E)) \subseteq \mathsf{facts}(D')$. Since $E \subseteq D$, which means that $\mathsf{facts}(E) \subseteq \mathsf{facts}(D)$, we get that $\mathsf{facts}(E) \subseteq \mathsf{facts}(D')$. Therefore, h_E is simply the identity on $\mathsf{adom}(E)$. □

4.2 The Characterization for TGDs and EGDs

We are now ready to provide the characterization of interest, which was originally established for arbitrary relational structures in [12], and here is presented for databases (i.e., finite structures).

Theorem 3 *Consider a collection C of databases, and integers $n, m \geq 0$. The following statements are equivalent:*

1. *C is finitely axiomatizable by (n, m)-tgds and n-egds.*
2. *C is 1-critical, (n, m)-local, and closed under direct products.*

Clearly, the above result implies a characterization for arbitrary tgds and egds:

Corollary 1 *Consider a collection C of databases. The following are equivalent:*

1. *C is finitely axiomatizable by tgds and egds.*
2. *C is 1-critical, local, and closed under direct products.*

The direction (1) $\Rightarrow$ (2) of Theorem 3 is easy and we leave it as an exercise to the reader. We proceed to discuss the non-trivial direction (2) $\Rightarrow$ (1). To this end, we need to introduce a couple of auxiliary notions, namely existential disjunctive dependencies and relative diagrams of databases.

Existential Disjunctive Dependencies Existential disjunctive dependencies generalize disjunctive dependencies, already used in the proof of Theorem 2, with existential quantification in the right-hand side of the implication. Moreover, the hypothesis and the conclusion of the implication can be empty. More precisely, an *existential disjunctive dependency* (edd) δ over a schema $\mathbf{S}$ is a constant-free sentence

$$\forall \bar{x} \left(\phi(\bar{x}) \to \bigvee_{i=1}^{k} \psi_i(\bar{x}_i) \right),$$

where $\bar{x}$ is a tuple of variables of $\mathbf{V}$, the expression $\phi(\bar{x})$ is a (possibly empty) conjunction of atoms over $\mathbf{S}$, and, for each $i \in [k]$, $\bar{x}_i \subseteq \bar{x}$ and the expression $\psi(\bar{x}_i)$ is either an equality formula $y = z$ with $\bar{x}_i = \{y, z\}$, or a constant-free formula $\exists \bar{y}_i \, \chi_i(\bar{x}_i, \bar{y}_i)$ with $\bar{y}_i$ being a tuple of variables from $\mathbf{V} \setminus \bar{x}$ and $\chi_i(\bar{x}_i, \bar{y}_i)$ is a (possibly empty) conjunction of atoms over $\mathbf{S}$. When the conclusion of δ is empty (i.e., there are no disjuncts), δ is essentially the sentence $\forall \bar{x} \left(\phi(\bar{x}) \to \mathsf{false} \right)$. Now,

when both the hypothesis and the conclusion of δ are empty, then δ is essentially the truth value false, i.e., a contradiction. Assuming that the conclusion of δ is non-empty, a database D *satisfies* δ if, whenever there exists a function $h : \bar{x} \to \mathsf{dom}(D)$ such that $h(\phi(\bar{x})) \subseteq \mathsf{facts}(D)$, then there is $i \in [k]$ such that, if $\psi_i(\bar{y}_i)$ is $y = z$, then $h(y) = h(z)$; otherwise, if $\psi_i(\bar{y}_i)$ is of the form $\exists \bar{y}_i \, \chi_i(\bar{x}_i, \bar{y}_i)$, then there is an extension h' of h such that $h'(\chi_i(\bar{x}_i, \bar{y}_i)) \subseteq \mathsf{facts}(D)$. In case the conclusion of δ is empty and δ is not a contradiction, D *satisfies* δ if there is no function $h : \bar{x} \to \mathsf{dom}(D)$ such that $h(\phi(\bar{x})) \subseteq \mathsf{facts}(D)$. We write $D \models \delta$ for the fact that D satisfies δ. The database D *satisfies a set* Σ of edds, written $D \models \Sigma$ (D is a *model* of Σ), if $D \models \delta$ for each $\delta \in \Sigma$.

Relative Diagram of a Database Consider an integer $\ell \geq 0$, an $\mathbf{S}$-database E, and an $\mathbf{S}$-database D such that $E \preceq D$. We are going to define the notion of the *ℓ-diagram of E relative to D*, which can be regarded as a refinement of the standard notion of diagram of a database, already used in the proof of Theorem 2. Let $A_{E,\ell}$ be the set of all atomic formulas of the form $R(\bar{u})$ that can be formed using predicates from $\mathbf{S}$, constants from $\mathsf{dom}(E)$, and ℓ distinct variables $y_1, \ldots, y_\ell$ from $\mathbf{V}$, i.e., $R \in \mathbf{S}$ and $\bar{u} \in (\mathsf{dom}(E) \cup \{y_1, \ldots, y_\ell\})^{\mathsf{ar}(R)}$. Let $C_{E,\ell}$ be the set of all conjunctions of atomic formulas from $A_{E,\ell}$. Note that both $A_{E,\ell}$ and $C_{E,\ell}$ are finite sets because $\mathsf{dom}(E)$ is finite. Given a formula $\gamma(y_1, \ldots, y_\ell) \in C_{E,\ell}$, we can naturally talk about the satisfaction of the sentence $\exists y_1 \cdots \exists y_\ell \, \gamma(y_1, \ldots, y_\ell)$ by a database D', in which case we simply write $D' \models \exists y_1 \cdots \exists y_\ell \, \gamma(y_1, \ldots, y_\ell)$. The *$\ell$-diagram of E relative to D*, denoted $\Delta^D_{E,\ell}$, is the first-order sentence

$$\bigwedge_{\alpha \in \mathsf{facts}(E)} \alpha \; \wedge \bigwedge_{\substack{c,d \in \mathsf{dom}(E), \\ c \neq d}} \neg(c = d) \; \wedge \bigwedge_{\substack{\gamma(y_1,\ldots,y_\ell) \in C_{E,\ell}, \\ D \not\models \exists y_1 \cdots \exists y_\ell \, \gamma(y_1,\ldots,y_\ell)}} \neg \left(\exists y_1 \cdots \exists y_\ell \, \gamma(y_1, \ldots, y_\ell)\right).$$

In fact, we are interested in the first-order formula $\Phi^D_{E,\ell}(\bar{x})$ obtained from the first-order sentence $\Delta^D_{E,\ell}$ by replacing each constant element $c \in \mathsf{dom}(E)$ with a new variable $x_c \in \mathbf{V} \setminus \{y_1, \ldots, y_\ell\}$. As was the case with the formulas of $C_{E,\ell}$, we can naturally talk about the satisfaction of $\exists \bar{x} \, \Phi^D_{E,\ell}(\bar{x})$ by a database D', in which case we simply write $D' \models \exists \bar{x} \, \Phi^D_{E,\ell}(\bar{x})$. It is straightforward to verify that:

Lemma 6 *For every integer $\ell \geq 0$, $\mathbf{S}$-database E, and $\mathbf{S}$-database D with $E \preceq D$, it holds that $D \models \exists \bar{x} \, \Phi^D_{E,\ell}(\bar{x})$.*

Let $\mathsf{E}_{n,m}$, for some integers $n, m \geq 0$, be the set of all edds over $\mathbf{S}$ of the form $\forall \bar{x} \left(\phi(\bar{x}) \to \bigvee_{i=1}^{k} \psi_i(\bar{x}_i)\right)$ such that $\bar{x}$ consists of at most n distinct variables, and, for each $i \in [k]$, the formula $\psi_i(\bar{x}_i)$ mentions at most $n + m$ distinct variables. The latter means that, if $\psi_i(\bar{x}_i)$ is a formula of the form $\exists \bar{y}_i \, \chi_i(\bar{x}_i, \bar{y}_i)$ (i.e., it is not an equality expression), then $\bar{y}_i$ consists of at most m distinct variables. Note that $\mathsf{E}_{n,m}$ is a finite set (up to logical equivalence) since $\mathbf{S}$ is finite and the number of variables in each element of $\mathsf{E}_{n,m}$ is finite. We can show the following auxiliary lemma.

Lemma 7 *Let $n, m \geq 0$ and assume that D, E are two* **S***-databases such that* $\mathsf{adom}(E) = \mathsf{dom}(E)$, $|\mathsf{adom}(E)| \leq n$, *and* $E \preceq D$. *It holds that there exists a sentence* $\delta \in \mathsf{E}_{n,m}$ *such that* $\delta \equiv \neg\exists\bar{x}\, \Phi^D_{E,m}(\bar{x})$.

Proof Recall that the formula $\Phi^D_{E,m}(\bar{x})$ is obtained from the m-diagram of E relative to D by renaming each $c \in \mathsf{dom}(E)$ to a new variable x_c; let ρ be the renaming function, i.e., $\rho(c) = x_c$ for each $c \in \mathsf{dom}(E)$. Thus, $\Phi^D_{E,m}(\bar{x})$ is of the form

$$\underbrace{\bigwedge_{\alpha \in \mathsf{facts}(E)} \rho(\alpha)}_{\Psi_1} \wedge \underbrace{\bigwedge_{\substack{c,d \in \mathsf{dom}(E),\\ c \neq d}} \neg(\rho(c) = \rho(d)) \wedge \bigwedge_{\substack{\gamma(\bar{y}) \in C_{E,m},\\ D \not\models \exists\bar{y}\, \gamma(\bar{y})}} \neg\exists\bar{y}\, \rho(\gamma(\bar{y}))}_{\Psi_2}.$$

Note that Ψ_1 and Ψ_2 might be empty (i.e., they have no conjuncts). In particular, Ψ_1 is empty if E is empty, whereas Ψ_2 is empty if D is 1-critical, and thus, $D \models \exists\bar{y}\, \gamma(\bar{y})$ for each $\gamma(\bar{y}) \in C_{E,m}$. It is clear that $\neg\exists\bar{x}\, \Phi^D_{E,m}(\bar{x})$ is equivalent to the sentence

$$\delta \;=\; \forall\bar{x}\, (\phi(\bar{x}) \;\rightarrow\; \psi(\bar{x}))\,,$$

where

$$\phi(\bar{x}) \;=\; \bigwedge_{\alpha \in \mathsf{facts}(E)} \rho(\alpha)$$

and the shape of $\psi(\bar{x})$ depends on whether Ψ_1 and Ψ_2 are empty or not. If both are empty, then δ is the truth constant false, i.e., a contradiction, and thus, $\delta \in \mathsf{E}_{n,m}$. If Ψ_1 is non-empty and Ψ_2 is empty, then, for an **S**-database D', it holds that $D' \models \neg\exists\bar{x}\, \Phi^D_{E,m}(\bar{x})$ if there is no function $h : \bar{x} \to \mathsf{dom}(D')$ such that $h(\phi(\bar{x})) \subseteq \mathsf{facts}(D')$. Hence, in this case, δ is the edd $\forall\bar{x}\, \big(\phi(\bar{x}) \rightarrow \mathsf{false}\big)$. Since, by hypothesis, $|\mathsf{adom}(E)| \leq n$, we get that $\phi(\bar{x})$ mentions at most n variables, and thus, $\delta \in \mathsf{E}_{n,m}$. Now, if Ψ_1 is either empty or non-empty, and Ψ_2 is non-empty, then we have that

$$\psi(\bar{x}) \;=\; \bigvee_{\substack{c,d \in \mathsf{dom}(E),\\ c \neq d}} \rho(c) = \rho(d) \vee \bigvee_{\substack{\gamma(\bar{y}) \in C_{E,m},\\ D \not\models \exists\bar{y}\, \gamma(\bar{y})}} \exists\bar{y}\, \rho(\gamma(\bar{y})),$$

and hence, δ is an edd. It remains to show that $\delta \in \mathsf{E}_{n,m}$. To this end, we need to show the following two statements: (i) each variable in $\psi(\bar{x})$ is either existentially quantified or appears in $\phi(\bar{x})$, and (ii) δ mentions at most n universally quantified variables and at most m existentially quantified variables. Statement (i) is true since, by hypothesis, $\mathsf{adom}(E) = \mathsf{dom}(E)$. Concerning statement (ii), δ has at most n universally quantified variables since, by hypothesis, $|\mathsf{adom}(E)| \leq n$, and δ has

m existentially quantified variables because so does the formula $\neg\Phi^D_{E,m}$ by the construction of $\Phi^D_{E,m}$. □

We can now discuss the non-trivial direction (2) $\Rightarrow$ (1) of Theorem 3. Assume that C is a collection of databases over $\mathbf{S}$. The proof proceeds in two main steps:

1. We construct a finite set $\Sigma^\vee$ of edds over $\mathbf{S}$, which mention at most n universally and at most m existentially quantified variables, and show that $\Sigma^\vee$ has the following property: a database D is in C if and only if D satisfies $\Sigma^\vee$. To this end, we exploit the fact that C is (n, m)-local.
2. We then show that there is a finite set $\Sigma^{\exists,=}$ of (n, m)-tgds and n-egds over $\mathbf{S}$ such that a database D is in C if and only if D satisfies $\Sigma^{\exists,=}$; in fact, $\Sigma^{\exists,=}$ is the set of the tgds and egds occurring in $\Sigma^\vee$. To this end, we exploit the fact that C is 1-critical and closed under direct products.

In what follows, we give the details for each of the above steps.

Step 1: The Finite Set $\Sigma^\vee$ of Edds that Axiomatizes C

Recall that $\mathsf{E}_{n,m}$ is the set of all edds over $\mathbf{S}$ of the form $\forall\bar{x}\left(\phi(\bar{x}) \rightarrow \bigvee_{i=1}^{k} \psi_i(\bar{x}_i)\right)$ such that $\bar{x}$ consists of at most n distinct variables, and, for each $i \in [k]$, the formula $\psi_i(\bar{x}_i)$ mentions at most $n + m$ distinct variables. The latter means that, if $\psi_i(\bar{x}_i)$ is a formula of the form $\exists\bar{y}_i\, \chi_i(\bar{x}_i, \bar{y}_i)$ (i.e., it is not an equality expression), then $\bar{y}_i$ consists of at most m distinct variables. Recall also that $\mathsf{E}_{n,m}$ is finite (up to logical equivalence). Define $\Sigma^\vee$ to be the set of all edds from $\mathsf{E}_{n,m}$ that are satisfied by every database of C, that is,

$$\Sigma^\vee = \left\{\delta \in \mathsf{E}_{n,m} \mid \text{ for each } D \in C, \text{ it holds that } D \models \delta\right\}.$$

It is clear that $\Sigma^\vee$ is finite (up to logical equivalence) since $\Sigma^\vee \subseteq \mathsf{E}_{n,m}$. We will show that C is precisely the set of databases that satisfy $\Sigma^\vee$.

Lemma 8 *For every* $\mathbf{S}$*-database* D*, we have that* $D \in C$ *if and only if* $D \models \Sigma^\vee$.

Proof The ($\Rightarrow$) direction follows from the definition of $\Sigma^\vee$. We proceed with the ($\Leftarrow$) direction. We have to show that if $D \models \Sigma^\vee$, then $D \in C$. We will show that C is (n, m)-locally embeddable in D, which, since C is (n, m)-local, will imply that $D \in C$. By Lemma 3, to show that C is (n, m)-locally embeddable in D, we have to show that, for every $E \preceq D$ with $\mathsf{adom}(E) = \mathsf{dom}(E)$ and $|\mathsf{adom}(E)| \leq n$, there is a database $D_E \in C$ such that $E \subseteq D_E$, and for every D' in the m-neighbourhood of E in D_E, there is a function $h : \mathsf{adom}(D') \rightarrow \mathsf{adom}(D)$ such that h is the identity on $\mathsf{adom}(E)$ and $h(\mathsf{facts}(D')) \subseteq \mathsf{facts}(D)$.

Let E be a database such that $E \preceq D$, $\mathsf{adom}(E) = \mathsf{dom}(E)$, and $|\mathsf{adom}(E)| \leq n$. By Lemma 7, the sentence $\neg\exists\bar{x}\, \Phi^D_{E,m}(\bar{x})$ is logically equivalent to a sentence $\delta \in \mathsf{E}_{n,m}$. We claim that $\delta \notin \Sigma^\vee$. Indeed, otherwise, we would have that $D \models \delta$, which is a contradiction, since, by Lemma 6, $D \models \exists\bar{x}\, \Phi^D_{E,m}(\bar{x})$. Since $\delta \notin \Sigma^\vee$, there must exist a database $D_E \in C$ such that $D_E \not\models \delta$, which means that $D_E \models \exists\bar{x}\, \Phi^D_{E,m}(\bar{x})$. Thus, there exists a database $E' \subseteq D_E$ such that $E \simeq E'$,

i.e., E and E' are isomorphic databases. Therefore, and without loss of generality, we can assume that $E \subseteq D_E$. To show that C is (n, m)-locally embeddable in D, it remains to show that for every database D' in the m-neighbourhood of E in D_E, there exists a function $h : \mathsf{adom}(D') \to \mathsf{adom}(D)$ such that h is the identity on $\mathsf{adom}(E)$ and $h(\mathsf{facts}(D')) \subseteq \mathsf{facts}(D)$. Towards a contradiction, assume that there exists D' in the m-neighbourhood of E in D_E for which there is no function $h : \mathsf{adom}(D') \to \mathsf{adom}(D)$ that is the identity on $\mathsf{adom}(E)$ with $h(\mathsf{facts}(D')) \subseteq \mathsf{facts}(D)$. Let F be the $\mathbf{S}$-database defined as the difference between D' and E, i.e., F is such that $\mathsf{facts}(F) = \mathsf{facts}(D') \setminus \mathsf{facts}(E)$, while $\mathsf{dom}(F)$ consists of all the constants occurring in $\mathsf{facts}(D') \setminus \mathsf{facts}(E)$, and hence, $\mathsf{dom}(F) = \mathsf{adom}(F)$. Clearly, there is no function $h : \mathsf{adom}(F) \to \mathsf{adom}(D)$ that is the identity on $\mathsf{adom}(E)$ and $h(\mathsf{facts}(F) \subseteq D$. Observe that $|\mathsf{adom}(F) \setminus \mathsf{adom}(E)| \leq m$; we assume that $\mathsf{adom}(F) \setminus \mathsf{adom}(E) = \{d_1, \ldots, d_{m'}\}$ for $m' \leq m$. Let $\gamma(\bar{y})$ be the formula obtained from $\bigwedge_{\alpha \in \mathsf{facts}(F)} \alpha$ after renaming each constant d_i to the variable y_i; thus, $\bar{y} = y_1, \ldots, y_{m'}$. Since there is no function $h : \mathsf{adom}(F) \to \mathsf{adom}(D)$ that is the identity on $\mathsf{adom}(E)$ and $h(\mathsf{facts}(F) \subseteq D$, we can conclude that $D \not\models \exists \bar{y}\, \gamma(\bar{y})$. Observe now that, by construction, $\neg\exists \bar{y}\, \gamma(\bar{y})$ is a conjunct of $\Delta^D_{E,m}$, which in turn implies that the formula $\neg\exists \bar{z}\, \gamma(\bar{z})$ obtained from $\neg\exists \bar{y}\, \gamma(\bar{y})$ after renaming each constant $c \in \mathsf{dom}(E) = \mathsf{adom}(E)$ to the variable x_c is a conjunct of $\Phi^D_{E,m}(\bar{x})$. Since $F \subseteq D_E$, we conclude that $D_E \models \exists \bar{z}\, \gamma(\bar{z})$, which implies that $D_E \not\models \exists \bar{x}\, \Phi^D_{E,m}(\bar{x})$. But this contradicts the fact that $D_E \models \exists \bar{x}\, \Phi^D_{E,m}(\bar{x})$. This completes the proof that C is (n, m)-embeddable in D. □

Step 2: The Finite Set $\Sigma^{\exists,=}$ of tgds and egds that Axiomatizes C

The next lemma, which is an elimination-of-disjunctions result, provides a stepping stone towards proving that the class C is finitely axiomatizable by tgds and egds.

Lemma 9 *Assume that $\forall \bar{x}(\phi(\bar{x}) \to \bigvee_{i=1}^{k} \psi_i(\bar{x}_i))$, where, for each $i \in [k]$, $\psi_i(\bar{x}_i)$ is either an equality formula or a non-empty conjunction of atoms, is an edd that belongs to $\Sigma^{\vee}$. There is $j \in [k]$ such that $\forall \bar{x}(\phi(\bar{x}) \to \psi_j(\bar{x}_j))$ also belongs to $\Sigma^{\vee}$.*

Proof Note that for every $j \in [k]$, the sentence $\forall \bar{x}(\phi(\bar{x}) \to \psi_j(\bar{x}_j))$ is either an egd or a tgd, hence it is an edd. Towards a contradiction, assume that none of these sentences belongs to $\Sigma^{\vee}$. Since $\Sigma^{\vee}$ consists of all edds in $\mathsf{E}_{n,m}$ that are satisfied by every database in C, for every $j \in [k]$, there is an $\mathbf{S}$-database $D_j \in C$ such that $D_j \not\models \forall \bar{x}\, (\phi(\bar{x}) \to \psi_j(\bar{x}_j))$, or, equivalently, $D_j \models \exists \bar{x}\, (\phi(\bar{x}) \wedge \neg\psi_j(\bar{x}_j))$. Let

$$E \;=\; D_1 \otimes \cdots \otimes D_k.$$

Since, by hypothesis, C is closed under direct products, we have that $E \in C$. We will show that $E \not\models \delta$, which leads to a contradiction since $\delta \in \Sigma^{\vee}$ and $E \in C$, which means that $E \models \delta$.

Since $D_j \models \exists \bar{x}\, (\phi(\bar{x}) \wedge \neg\psi_j(\bar{x}_j))$ for each $j \in [k]$, there is a function $h_j : \bar{x} \to \mathsf{dom}(D_j)$ such that $h_j(\phi(\bar{x})) \subseteq \mathsf{facts}(D_j)$ and $D_j \models \neg\psi_j(h_j(\bar{x}_j))$. Let $h : \bar{x} \to \mathsf{dom}(E)$ be the function such that, for each variable $x \in \bar{x}$, we have $h(x) = (h_1(x), \ldots, h_k(x))$. By the definition of direct products, we have that $h(\phi(\bar{x})) \subseteq$

$\mathsf{facts}(E)$. It remains to show that, for each $j \in [k]$, if $\psi_j(\bar{x}_j)$ is an equality formula $y = z$, then $h(y) \neq h(z)$, while if $\psi_j(\bar{x}_j)$ is not an equality formula, then there is no extension h' of h such that $h'(\psi_j(\bar{x}_j)) \subseteq \mathsf{facts}(E)$. We proceed by case analysis on the form of $\psi_j(\bar{x}_j)$.

Case 1 Assume first that $\psi_j(\bar{x}_j)$ is the equality expression $y = z$. By the properties of the function h_j, we have that $h_j(y) \neq h_j(z)$. Hence, by the definition of h, we conclude that $h(y) \neq h(z)$.

Case 2 Assume now that $\psi_j(\bar{x}_j)$ is a formula of the form $\exists \bar{y}_j \, \chi_j(\bar{x}_j, \bar{y}_j)$, and assume, by contradiction, that there exists an extension h' of h such that $h'(\chi_j(\bar{x}_j, \bar{y}_j)) \subseteq \mathsf{facts}(E)$. For a constant $c \in \mathsf{dom}(E)$, we write $c[i]$ for the i-th component of c, which is actually a constant from $\mathsf{dom}(D_i)$. We define the function $h'_j : \bar{x} \cup \bar{y}_j \to \mathsf{dom}(D_j)$ such that $h'_j(v) = h'(v)[j]$, for each variable $v \in \bar{x} \cup \bar{y}_j$. Since h' is an extension of h and $h_j(v) = h(v)[j]$, for each variable $v \in \bar{x}$, we conclude that h'_j is an extension of h_j. By hypothesis, $h'(\alpha) \in \mathsf{facts}(E)$ for each conjunct α of $\chi_j(\bar{x}_j, \bar{y}_j)$. Assume that $h'(\alpha) = R(\bar{c}_i, \ldots, \bar{c}_m)$. By the definition of the direct product, we conclude that $R(\bar{c}_1[j], \ldots, c_m[j]) \in \mathsf{facts}(D_j)$. Consequently, $h'_j(\psi_j(\bar{x}_j)) \subseteq \mathsf{facts}(D_j)$, which in turn implies that $D_j \models \psi_j(h_j(\bar{x}_j)$, which contradicts the fact that $D_j \not\models \psi_j(h_j(\bar{x}_j)$. This completes the proof of Lemma 9.

□

We are now ready to complete the proof of Theorem 3. Define $\Sigma^{\exists,=}$ to be the set of all tgds and egds occurring in $\Sigma^{\vee}$, that is,

$$\Sigma^{\exists,=} = \left\{ \delta \in \Sigma^{\vee} \mid \delta \text{ is a tgd or an egd} \right\}.$$

Note that $\Sigma^{\exists,=}$ is a finite set because it is a subset of the finite set $\Sigma^{\vee}$. We proceed to show that, for each $\mathbf{S}$-database D, it holds that $D \in C$ if and only if $D \models \Sigma^{\exists,=}$.

($\Rightarrow$) If $D \in C$, then, by definition, $D \models \Sigma^{\vee}$; hence, $D \models \Sigma^{\exists,=}$ since $\Sigma^{\exists,=} \subseteq \Sigma^{\vee}$.

($\Leftarrow$) Assume now that D is an $\mathbf{S}$-database such that $D \models \Sigma^{\exists,=}$. We need to show that $D \in C$. By Lemma 8, it suffices to show that $D \models \Sigma^{\vee}$. Let δ be a sentence in $\Sigma^{\vee}$. Since C is 1-critical, we get that δ is not an implication with the truth constant false being its conclusion. Therefore, δ is an edd of the form $\forall \bar{x}(\phi(\bar{x}) \to \bigvee_{i=1}^{k} \psi_i(\bar{x}_i))$, where each expression $\psi_i(\bar{x}_i)$ is either an equality $y = z$ with y and z among the variables in $\bar{x}$ or a constant-free formula $\exists \bar{y}_i \, \chi_i(\bar{x}_i, \bar{y}_i)$ with $\chi_i(\bar{x}_i, \bar{y}_i)$ non-empty, the variables in $\bar{x}_i$ among those in $\bar{x}$, and the variables in $\bar{y}_i$ different from those in $\bar{x}$. By Lemma 9, there is $j \in [k]$ such that the sentence $\forall \bar{x}(\phi(\bar{x}) \to \psi_j(\bar{x}_j))$ belongs to $\Sigma^{\vee}$. Since this sentence is either an egd or a tgd, we have that it belongs to $\Sigma^{=,\exists}$, and hence, $D \models \forall \bar{x}(\phi(\bar{x}) \to \psi_j(\bar{x}_j))$. It is obvious that the formula $\forall \bar{x}(\phi(\bar{x}) \to \psi_j(\bar{x}_j))$ logically implies the edd $\forall \bar{x}(\phi(\bar{x}) \to \bigvee_{i=1}^{k} \psi_i(\bar{x}_i))$, hence $D \models \forall \bar{x}(\phi(\bar{x}) \to \bigvee_{i=1}^{k} \psi_i(\bar{x}_i))$. This completes the proof of the above claim.

Therefore, we have established that the class C is axiomatizable by the set $\Sigma^{=,\exists}$, which is a finite set of tgds and egds. This completes the proof of Theorem 3 .

4.3 From Theorem 3 to Theorem 1

We conclude this section by discussing how we can derive the characterization for full tgds and egds from [22], that is, Theorem 1 in Sect. 3, by exploiting the characterization for (n, m)-tgds and n-egds presented above, that is, Theorem 3. We first present the following consequence of Theorem 3, which forms a relevant characterization for finite sets of full tgds and egds in its own right.

Corollary 2 *Consider a collection C of databases. The following are equivalent:*

1. *C is finitely axiomatizable by full tgds and egds.*
2. *C is 1-critical, $(n, 0)$-local for some integer $n \geq 0$, and closed under direct products.*

Observe that for obtaining Theorem 1 it suffices to replace in Corollary 2 the property of $(n, 0)$-locality for some integer $n \geq 0$ with the properties of domain independence, modularity, and closure under subdatabases. We proceed to show that this is indeed the case. But let us first state a simple fact that provides a useful characterization of the notion of $(n, 0)$-local embeddability.

Recall that, for a schema $\mathbf{S}$ and an integer $m \geq 0$, the m-neighbourhood of an $\mathbf{S}$-database D'' in an $\mathbf{S}$-database D' with $D'' \subseteq D'$ is the set of $\mathbf{S}$-databases

$$\{E \mid D'' \subseteq E,\ E \preceq D' \text{ and } |\mathsf{adom}(E)| \leq |\mathsf{adom}(D'')| + m\}.$$

Note that if $D'' \subseteq E$, we have that $\mathsf{adom}(D'') \subseteq \mathsf{adom}(E)$. Furthermore, if E is in the 0-neighbourhood of D'' in D', then $|\mathsf{adom}(E)| \leq |\mathsf{adom}(D'')|$, and thus, $\mathsf{adom}(D'') = \mathsf{adom}(E)$. Since both D'' and E are contained in D', it follows that $\mathsf{facts}(E) = \mathsf{facts}(D' \restriction \mathsf{adom}(D'')) = \mathsf{facts}(D'')$, where $D' \restriction \mathsf{adom}(D'')$ is the restriction of D' over $\mathsf{adom}(D'')$, that is, the database $\{R(\bar{c}) \in D' \mid \bar{c} \in \mathsf{adom}(D'')^{\mathsf{ar}(R)}\}$. It follows that the 0-neighborhood of D'' in D' is the set of $\mathbf{S}$-databases

$$\{E \mid E \preceq D' \text{ and } \mathsf{facts}(E) = \mathsf{facts}(D' \restriction \mathsf{adom}(D'')) = \mathsf{facts}(D'')\}.$$

Consequently, we have that the following holds.

Lemma 10 *Consider a collection C of databases over a schema $\mathbf{S}$. The following are equivalent:*

1. *An $\mathbf{S}$-database D is $(n, 0)$-locally embeddable in C.*
2. *For every $E \preceq D$ with $\mathsf{adom}(E) = \mathsf{dom}(E)$ and $|\mathsf{adom}(E)| \leq n$, there is an $\mathbf{S}$-database $D_E \in C$ such that $\mathsf{facts}(D_E \restriction \mathsf{adom}(E)) = \mathsf{facts}(E)$.*

By exploiting the above lemma, we can show that $(n, 0)$-locality can be replaced by the properties of domain independence, modularity, and closure under subdatabases.

Proposition 1 *Consider a collection C of databases over a schema $\mathbf{S}$ and an integer $n \geq 0$. The following statements are equivalent:*

1. *C is $(n, 0)$-local.*
2. *C is domain independent, n-modular, and closed under subdatabases.*

Proof We first show the direction (1) $\Rightarrow$ (2). Since, by hypothesis, C is $(n, 0)$-local, Lemma 5 implies that C is domain independent. To show that C is n-modular, let D be an $\mathbf{S}$-database such that every $D' \preceq D$ with $\mathsf{dom}(D') \leq n$ belongs to C. We proceed to show that $D \in C$ by showing that C is $(n, 0)$-locally embeddable in D. This follows from Lemma 10 by taking $D_E = D'$. Finally, to show that C is closed under subdatabases, consider an arbitrary $\mathbf{S}$-database $D \in C$ and let D' be a subdatabase of D. We proceed to show that $D' \in C$ by showing that C is $(n, 0)$-locally embeddable in D'. The latter again follows from Lemma 10 by taking $D_E = D$.

We now show (2) $\Rightarrow$ (1). Assume that C is domain independent, n-modular, and closed under subdatabases. We need to show that if C is $(n, 0)$-locally embeddable in an $\mathbf{S}$-database D, then $D \in C$. By Lemma 10, for every $E \preceq D$ with $\mathsf{adom}(E) = \mathsf{dom}(E)$ and $|\mathsf{adom}(E)| \leq n$, there is an $\mathbf{S}$-database $D_E \in C$ such that $\mathsf{facts}(D_E \restriction \mathsf{adom}(E)) = \mathsf{facts}(E)$. Since C is closed under subdatabases, we have that $D_E \restriction \mathsf{adom}(E)$ belongs to C. Since C is domain independent and $\mathsf{facts}(D_E \restriction \mathsf{adom}(E)) = \mathsf{facts}(E)$, we get that $E \in C$. Thus, every subdatabase E of D with at most n elements in its active domain is in C. Since C is n-modular, it follows that D is also in C. □

Theorem 1 readily follows from Corollary 2 and Proposition 1.

5 Concluding Remarks

When János Makowsky retired as President of the European Association for Computer Science Logic (EACSL), he gave an invited address at the 2011 CSL conference in which he reflected on his experience in research [21]. In particular, he reflected on his work in database theory, highlighted the undecidability of the implication problem for EIDs, and concluded the section on his contributions to database theory by writing that "After that I tried to learn the true problems of database theory. However, J. Ullman changed his mind and declared that Dependency Theory and Design Theory had run their course. As a result, papers dealing with these topics were almost banned from the relevant conferences." Here, Makowsky refers to Ullman's invited talk and paper, titled "Database Theory: Past and Future", at the 1987 ACM Symposium on Principles of Database Systems (PODS) [28]. When it came to the past of dependency theory, Ullman stated that

"We quickly learned far more about the subject than was necessary, a process that continues to this day" and then, contemplating the future, he delegated dependency theory to the section titled "Last Gasps of the Dying Swans".

The pronouncement of the demise of dependency theory, however, turned out to be premature. Indeed, about a decade later, data dependencies found numerous uses in formalizing and analyzing data inter-operability tasks, such as data exchange and data integration (see, e.g., the surveys [16, 18] and the books [1, 17]). As a matter of fact, the study of data dependencies has enjoyed a renaissance that continues to date. In particular, tgds have been used to specify data transformation tasks (i.e., how data structured under one schema should be transformed into data structured under a different schema) and as building blocks of ontology languages [2, 5, 6]. In the case of data exchange and data integration, a subclass of tgds, called *source-to-target tgds* (s-t tgds), has played a central role. These are the tgds of the form $\forall \bar{x} \forall \bar{y}(\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z} \psi(\bar{x}, \bar{z}))$, where all relations in $\phi(\bar{x}, \bar{y})$ are from a source schema, while all relations in $\psi(\bar{x}, \bar{z})$ are from a disjoint target schema. The study of s-t tgds led to the discovery of new structural properties for this class of tgds, the most prominent of which is the existence of *universal* solutions in data exchange [15]. In turn, this motivated the pursuit of characterizations of s-t tgds and of natural subclasses of them using structural properties relevant to data inter-operability, such as the existence of universal solutions and conjunctive query rewriting [26, 27]. And now we have come full circle with the structural characterizations of arbitrary tgds and egds discussed in this paper. In conclusion, János Makowsky's work on data dependencies was well ahead of its time.

References

1. Arenas, M., Barceló, P., Libkin, L., Murlak, F.: Foundations of Data Exchange. Cambridge University Press, Cambridge (2014)
2. Baget, J.-F., Leclère, M., Mugnier, M.-L., Salvat, E.: On rules with existential variables: walking the decidability line. Artif. Intell. **175**(9–10), 1620–1654 (2011)
3. Beeri, C., Vardi, M.Y.: The implication problem for data dependencies. In: ICALP, pp. 73–85 (1981)
4. Beeri, C., Vardi, M.Y.: A proof procedure for data dependencies. J. ACM **31**(4), 718–741 (1984)
5. Calì, A., Gottlob, G., Lukasiewicz, T.: A general datalog-based framework for tractable query answering over ontologies. J. Web Semant. **14**, 57–83 (2012)
6. Calì, A., Gottlob, G., Pieris, A.: Towards more expressive ontology languages: the query answering problem. Artif. Intell. **193**, 87–128 (2012)
7. Chandra, A.K., Lewis, H.R., Makowsky, J.A.: Embedded implicational dependencies and their inference problem. In: STOC, pp. 342–354 (1981)
8. Chang, C.C., Keisler, H.J.: Model Theory, 3rd edn., vol. 73. Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam (1992)
9. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**(6), 377–387 (1970)
10. Codd, E.F.: Relational completeness of data base sublanguages. Research Report/RJ/IBM/San Jose, RJ987 (1972)

11. Codd, E.F.: Further normalization of the data base relational model. Data Base Syst. **6**, 33–64 (1972)
12. Console, M., Kolaitis, P.G., Pieris, A.: Model-theoretic characterizations of rule-based ontologies. In: PODS, pp. 416–428 (2021)
13. Date, C.J.: Logic and Relational Theory. Technics Publications, Sedona (2020)
14. Fagin, R.: Horn clauses and database dependencies. J. ACM **29**(4), 952–985 (1982)
15. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theor. Comput. Sci. **336**(1), 89–124 (2005)
16. Kolaitis, P.G.: Schema mappings, data exchange, and metadata management. In: PODS, pp. 61–75 (2005)
17. Kolaitis, P.G., Lenzerini, M., Schweikardt, N. (eds.): Data Exchange, Integration, and Streams, vol. 5. Dagstuhl Follow-Ups. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Wadern (2013)
18. Lenzerini, M.: Data integration: a theoretical perspective. In: PODS, pp. 233–246 (2002)
19. Lindström, P.: On extensions of elementary logic. Theoria **35**(1) (1969)
20. Makowsky, J.A.: Characterizing data base dependencies. In: ICALP, pp. 86–97 (1981)
21. Makowsky, J.A.: Model theory in computer science: my own recurrent themes. In: CSL, pp. 553–567 (2011)
22. Makowsky, J.A., Vardi, M.Y.: On the expressive power of data dependencies. Acta Inform. **23**(3), 231–244 (1986)
23. McKinsey, J.C.C.: The decision problem for some classes of sentences without quantifiers. J. Symb. Logic **8**(3), 61–76 (1943)
24. Robinson, A.: Introduction to Model Theory and to the Metamathematics of Algebra. North-Holland Publishing Company, Amsterdam (1963)
25. Tarski, A.: Some notions and methods on the borderline of algebra and metamathematics. In: Proceedings of the International Congress of Mathematicians, vol. 1, pp. 705–720 (1950)
26. ten Cate, B., Kolaitis, P.G.: Structural characterizations of schema-mapping languages. In: ICDT, pp. 63–72 (2009)
27. ten Cate, B., Kolaitis, P.G.: Structural characterizations of schema-mapping languages. Commun. ACM **53**(1), 101–110 (2010)
28. Ullman, J.D.: Database theory: past and future. In: PODS, pp. 1–10 (1987)
29. Yannakakis, M., Papadimitriou, C.H.: Algebraic dependencies (extended abstract). In: FOCS, pp. 328–332 (1980)
30. Zhang, H., Jiang, G.: Characterizing the program expressive power of existential rule languages. In: AAAI, pp. 5950–5957 (2022)
31. Zhang, H., Zhang, Y., Jiang, G.: Model-theoretic characterizations of existential rule languages. In: IJCAI (2020)

On Consistency of Graphically Defined Specifications

Katerina Korenblat and Elena V. Ravve

Abstract A software bug is defined as an error, flaw, or fault in a system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways. Traditionally, verification is a task of bugs' detection. The main traditional question is whether a system correctly implements the expected behavior, described in its specification. In this paper, we deal rather with a different question, which is: are we sure that our specification is even sound? In order to effectively answer the question, we limit ourselves to the case of analysis of definition of Graphical User Interface (GUI) of information systems, where the frontend (GUI) in many cases may be defined as a "walk" between different screens. For this kind of systems, we present a methodology for partial capturing of a GUI specification using a graphical presentation, which is a formal model of such a system at the abstraction level of its GUI. The model is subsequently saved as a program graph (Kripke structure). That graph is then translated into ProMeLA and can be checked against LTL properties. We define a set of properties specifying sanity checks of the specifications. The LTL properties (commandments) are aimed to cover standard verification notions such as consistency, absence of ambiguity, etc. We develop a tool helping the user to model a GUI specification and execute the sanity checks, which are done using standard verification machinery. The contribution presents a fresh use of temporal logics in this context. Other approaches that tackle the same problem (for example, based on UML diagrams) have not been used yet for formal verification of software specifications.

K. Korenblat
Braude College of Engineering, Karmiel, Israel
e-mail: katerina@braude.ac.il

E. V. Ravve (✉)
Department of Software Engineering, ORT Braude College, Karmiel, Israel
e-mail: cselena@braude.ac.il

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_25

1 Introduction

Traditionally for software systems, we have a specification of a system and its model (implementation). Then, we start verification of the model. The model (program) may be presented as a logical structure $\mathcal{P}_{system}$ and the specification is assumed to be formulated as a formula φ_{spec}, expressed in some logical formalism. In a logical notation, the verification question is now formulated as $\mathcal{P}_{system} \models \varphi_{spec}$? What is wrong with such approach?

As a rule, such formula φ_{spec} is manually formulated and extremely complicated. The first problem that immediately follows from the fact is: are we sure that φ_{spec} does not contain contradictions? Otherwise the verification task is absolutely meaningless. Even if the formula is not a contradiction, can we formulate a (minimal) set of requirements that such a formula must satisfy? It means: what is the set of requirements that a specification is expected to satisfy?

In fact, any specification is naturally divided into two parts: functionality and a user interface. The last one may be presented as a Graphical User Interface (GUI) or Command Line Interface (CLI). Typically, the definition of a specification starts from a detailed description of the functionality and only then the definition of an interface is mostly dictated by the functionality. In many verification tasks, the question whether $\mathcal{P}_{system} \models \varphi_{spec}$ addresses the verification of the functionality of the software product rather than verification of the user interface. If the interaction with the finite user is limited to CLI then as a rule the interface is pretty simple and its verification is rather straight-forward. However, in many cases the verification of the GUI implementation is also omitted with no such a reason.

On the other hand, if we consider Information Systems (IS) then even from the most basic and classical point of view (cf. [14]), the systems contain at least: hardware, data, software, procedure and people. It means that a proper implementation of the interaction with the finite users (the people) becomes to be not less important than the proper implementation of the functionality (procedure). Moreover, without ignoring the importance of the functionality, the (financial) success of modern IS software products is more affected by the usability of the GUI than the functionality, which is in many cases is almost the same among the competitors.

From the practical (software engineering) point of view, in the definition of a modern IS product, we have the following team: a customer, say, a bank expert, who understands the needs of her/his clients; an algorithms' expert, who knows how to solve the functionality challenges; a programmer, who implements both the functionality and the GUI. The algorithms' expert and the programmer have a common technical background (common technical language), coming back to the common basic courses of their education programs. The customer (the bank expert) and the clients also have a common background (common language) based on their particular (financial) terminology. The differences in the terminology of the software engineering language and the customer's language finally bring to

numerous misunderstandings, which are hardly discovered in the next steps of implementation and testing.[1]

The only part of the specification that may be somehow formulated in a common (graphical) language is the GUI, which may be expressed (in IS's) as some kind of "walk" between different screens. The observations justify our claim that in modern IS software products, the GUI specification must precede functionality specification or at least must be formulated in parallel.

1.1 A Motivation Example

Let us perform a very simple experiment together. Take your smartphone, go to *Settings*, turn *Airplane Mode* ON. What do you observe? Once the *Airplane Mode* is turned ON, the *Wi-Fi* and *Bluetooth* are turned OFF *automatically*. The experiment may lead you to an assumption that *Wi-Fi* and *Bluetooth must* be turned OFF, when the *Airplane Mode* is ON.

Now, go to *Wi-Fi* or *Bluetooth* option and try to turn them to ON (each one at a time or both) manually? Did you succeed? Most (if not all of you) did. Congratulations! Finally, you should face the situation, shown on Fig. 1, where only *Wi-Fi* option is turned ON, when *Airplane Mode* is turned ON as well.

Now, our question is: whether it is correct that *Wi-Fi* and/or *Bluetooth* options are turned ON while *Airplane Mode* is ON as well? Is the situation, shown on Fig. 1, a bug, or rather a "feature"? The answer, according to the classical definition of bugs, is that it depends upon what is written in the specification of the corresponding program! In our specific case, it seems.[2] that it is just *NOT* written in the specification of the corresponding program!!!

Fig. 1 *Wi-Fi* option is turned ON while *Airplane Mode* is ON

[1] We have even much more differences than the differences in terminology: the common understanding of the task for people with different background can be different.

[2] We do not have the specification of the product that is why we only may guess that this is the case.

1.2 Classical Formal Verification and Software Engineering

Software verification is used to establish that the implementation of the software under consideration possesses certain properties. The properties to be validated can be quite basic, e.g., software should never be able to reach a situation, where no progress can be made (a deadlock scenario). More specific properties under verification may correspond to what was defined in the software specification.

In this paper, we adopt classical formal verification techniques to software engineering tasks. The software system is defined by its specification, which "in general case" is a free style text describing functionality of the system. Thus, the main question under consideration, motivated by the demonstrated example, is: whether it is clear that the specification is correct? Or more precisely, are we sure that *a model of the specification* $\mathcal{M}_{spec}$ *is correct*?

In the logical notation it should be presented as: $\mathcal{M}_{spec} \models \varphi_{sanity}$. It means that we should define the model $\mathcal{M}_{spec}$ of the specification as well as the formal definition of its minimal correctness: φ_{sanity}. It is obvious that the total "correctness" of a software specification is also related to how well it matches the needs of the intended users and satisfies their requirements. Moreover, the chosen specification domain of the GUI of IS applications focuses on user interface behavior only, which seldom to never would be sufficient for a complete specification of any application. This is the reason our approach verifies GUI specification logics for IS applications only, expressed as φ_{sanity}, and not total specification "correctness" for any application. We define five such φ_{sanity}'s formulae in Sect. 4 and we call them *Five Commandments for* GUI *design of* IS *Systems*.

In terms of the presented motivation example, the question under our limited consideration may be expressed as follows: is it written somewhere in the corresponding specification that *Wi-Fi* and $Bluetooth$ options must be always turned OFF while *Airplane Mode* is turned ON? If it is, there exists a bug in implementation in the classical formulation; otherwise, the specification is probably not complete (describes all possible scenaria) or even sound (does not contain contradictions)! Our main interest lies in the second situation. Below, we will use the software engineering term of *"sanity check"*, which means a basic suite of requirements to be quickly evaluated in order to verify whether the claim can possibly be true, cf. [5].

Our approach may be formulated as follows: we propose to check whether each sanity check property is satisfied on the model of the system specification. Thus, the approach falls into the standard pattern. Moreover, we specifically recall that when telecommunication protocols were checked in the old days, they were called specifications although from the model checking point of view they were models. Hence, we expand and adopt the approach to formal verification of software and program specifications. The problem of definition of the sanity check properties is real and serious. A partial solution can consist of listing some requirements of general nature that every system should satisfy.

In our paper, such an example is the described property: "all parameters always must be consistent". Furthermore, the next immediate question is: what is the model $\mathcal{M}_{spec}$? In the classical approach, we are dealing with the logical interpretation of specification in the form of logical formulae rather than logical structures!!! In this contribution, we deal with GUI design of information systems and partially answer the question (what is the model $\mathcal{M}_{spec}$?) in this case.

1.3 Why GUI of Information Systems?

Altogether, the definition of correctness of a general form of a specification, while it is written as a free style text, is a complicated problem, notwithstanding there is some experience with other special cases of specification, investigated, for example, in [12]. However, information systems, like applications for smartphones or different booking systems are mostly of a very specific kind. In fact, how do such systems work? The user clicks a GUI element, displayed on a screen and either the given screen is changed, or a transfer to another screen is executed. Information systems are programs, the essential part of the specification for which describes screens, transitions between the screens, and GUI action, available on these screens.

For such systems, we propose a new approach, where the specification may be formulated as a logical model of the system. So therefore, the model can be analyzed in order to answer the following question: are we sure that our specification is even sound? In a formal way, it means that the expected behavior of the system may be described in its specification as a set of GUI actions. Such observation leads us to an idea to present the model of the specifications of the systems ($\mathcal{M}_{spec}$) as a graph structure, where each vertex represents a specific screen and a combination of values corresponding to its GUI elements and edges represent transitions between them, see Fig. 2.

Now, the next question is: what is the formula φ_{sanity}? In fact, we propose to use a set of formulae φ_i. Each formula expresses a property that must be true in any spec of an information system. We call them: $commandments$. One example of such a formula is the consistency of values: *Wi-Fi* and $Bluetooth$ options must be always turned OFF while *Airplane Mode* is turned ON. We give more examples of the formulae in Sect. 4. The set of formulae may be extended, according to the detected failures, and it should be considered the standard, such that each such a system must satisfy.

1.4 Structure of the Paper

Novelty of the contribution is described in Sect. 2. The detailed description of our approach and its implementation is rather provided in Sect. 3. In Sect. 4, which is the

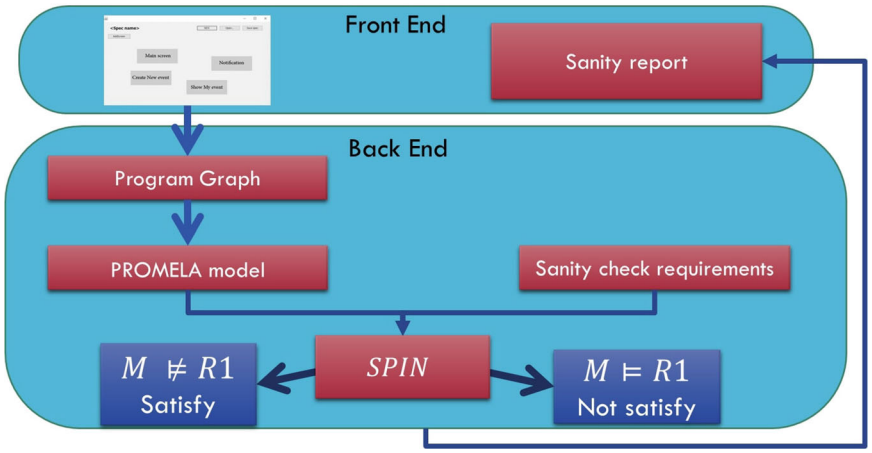


Fig. 2 General description of the method

main section of our paper, we explicitly define the commandments to be checked for each information system, formulated as LTL formulae. Section 5 summarizes the paper and gives outlook. We provide the reader with a brief review of the basic definitions in Appendix.

2 Novelty of the Contribution

As it was mentioned in one of the recent publications of [7], *"inconsistency in Requirement Engineering (RE) for Computer-Based Systems (CBSs) has received much attention over the years in the RE literature"*; see also [9, 15]. Inconsistency in RE for a CBS is defined as "any situation in which [at least] two parts of a requirements specification [for the CBS] do not obey some relationship that should hold between them".

As it was claimed in [6], *"in practice maintaining consistency at all times is an intractable problem."* The aim of our contribution is a **formal definition** of the relationships, which these parts of a requirements specification should obey in the case of the IS software. Our approach is strongly stimulated by *"preliminary empirical findings highlighting existing perceptions and attitudes of practitioners toward inconsistency"* of [6] and the proposed dimensions for their classification.

In fact, exploitation of the special kind of software in not really new indeed and goes back to presenting GUIs in a formal manner as described, for example, in [3]. The recent review of the verification and automated testing of GUI programs may be found in [13].

Go back to [4], where more than 500 examples of property specifications are collected. Based on this repository, the authors proposed patterns of the properties.

The approach was extended to the notion of Property Specification Patterns (PSPs), which was further developed in [10]. The main difference between this approach and our approach is that we limit ourselves by the special kind of specifications. This limitation allows using formulae, which must hold in each such a specification rather than patterns, which must be aligned to any particular case manually.

An attempt to look at an existing design methodology, i.e., user-centered design, examine the types of processes and artefacts that are used, and find ways of incorporating these into a formal process, was made in [2]. In the referred paper, a presentation model is used to formally capture the meaning of an informal design artefact such as a scenario, storyboard, or prototype; and the corresponding syntax is provided. Some examples are outlined of how one can use the presentation model in dealing with design concerns such as consistency are provided. However, GUI may consist of many different screens, and various options of moving between these different screens are an integrated part of the specification of the functionality of the software. The presentation model of [2] provides a static view of the design, which does not hold enough information to extend its use to dynamic behavior.

Yet, every software system development starts from the system specification. Before one starts implementation, the minimal correctness (consistency) of the specification must be confirmed. Specifications of IS applications are of a very specific type: they mostly describe transfers from one set of parameters values of a screen to another one. We use the distinctiveness of the specs in order to verify their correctness. This very special kind of the applications allows us presenting their models as graph structures ($\mathcal{M}_{spec}$).

To be more practical, we built a toy tool that allows defining specifications of IS applications graphically. The user defines the specifications in the corresponding terms: screens, screens' elements, and transfers between the screens. The tool directly converts the specifications to models of the specification ($\mathcal{M}_{spec}$) formulated in a graphs' notation: vertices (locations) are the screens associated with the corresponding particular values of the parameters; edges (transitions) are marked within GUI actions, which cause transitions between the screens. We use Program Graph (PG), cf. [11], as a formal model of the specification: $\mathcal{M}_{spec}$. Our tool translates the spec, defined graphically by the developer, to a PG, which is coded in ProMeLa model language

Next, our tool gets a pre-defined list of commandments: formulae φ_i, which demonstrates the correctness of the specification and which must be verified on $\mathcal{M}_{spec}$. The requirements are automatically translated to LTL formulae by the tool. Model checking is aimed answering the question of whether the formulae are satisfied on the PG: $\mathcal{M}_{spec} \models \varphi_i$ for each commandment. In our tool, the verification is executed using SPIN, which is the state-of-the-art model checker. For a particular commandment, it results in either a confirmation message, or in a counterexample presenting a path, where the commandment failed. We are not aware of another similar method to verify correctness of the *specifications*.

3 Detailed Description of the Approach and Its Implementation

In this paper, we propose a method and its naive partial implementation as a tool that allows graphical presentation and definition of specification of a IS system. The obtained specification is formally described as a Program Graph ($\mathcal{M}_{spec}$). Then, the PG is translated to a transition system, which is eventually translated to a Kripke structure. Doing so, we may then verify formally a predefined set of sanity check commandments (presented as LTL formulae φ_i on the model of the specification), which in logical notation precisely means: $\mathcal{M}_{spec} \models \varphi_i$ for each commandment.

The presented tool is naturally divided into two layers: frontend and backend. Frontend and backend are terms used to characterize program interfaces and services relative to the initial user of these interfaces and services. A frontend layer of an application is one that application users interact with directly.

A backend layer of an application serves indirectly in support of the frontend services. General description of the approach is presented on Fig. 2. We use the specification of a real smartphone application, called $BoPo$,[3] to demonstrate the ability of the tool.

BoPo application is aimed at helping people to find partners for mutual interests, not necessarily from their circle of friends or relatives, as well as helping them spend time together by organizing events. This application not only provides a service of event planning and management but also allows to find transportation solutions for participants, and more as well as helping them spend time together by organizing events, which are based on their mutual interests and hobbies.

This application not only provides a service of event planning and management but also allows to find transportation solutions for participants, and sends a reminder to the participants about the event enough time before (Fig. 3).

3.1 Defining Specification as a Set of Screens and Transfers Between Them

The basic ability of the frontend is adding of a new screen to a specification of an application. In order to add a new screen (see Fig. 4a), the user should press "add screen" button of the tool. Then, one sets the screen location inside an application window, defines the screen name and its description. Then, the user can also add a GUI element to a screen (see Fig. 4b). In this way, we can choose types of elements of screens. In our particular tool, the GUI element may be of the following types: On/Off, (Standard) Button, List, Empty/NotEmpty. Certainly, the selection of the types may be extended, when needed. Hereafter, a detailed description of the

[3] BoPo is not a commercial application. It was developed and implemented by our students.

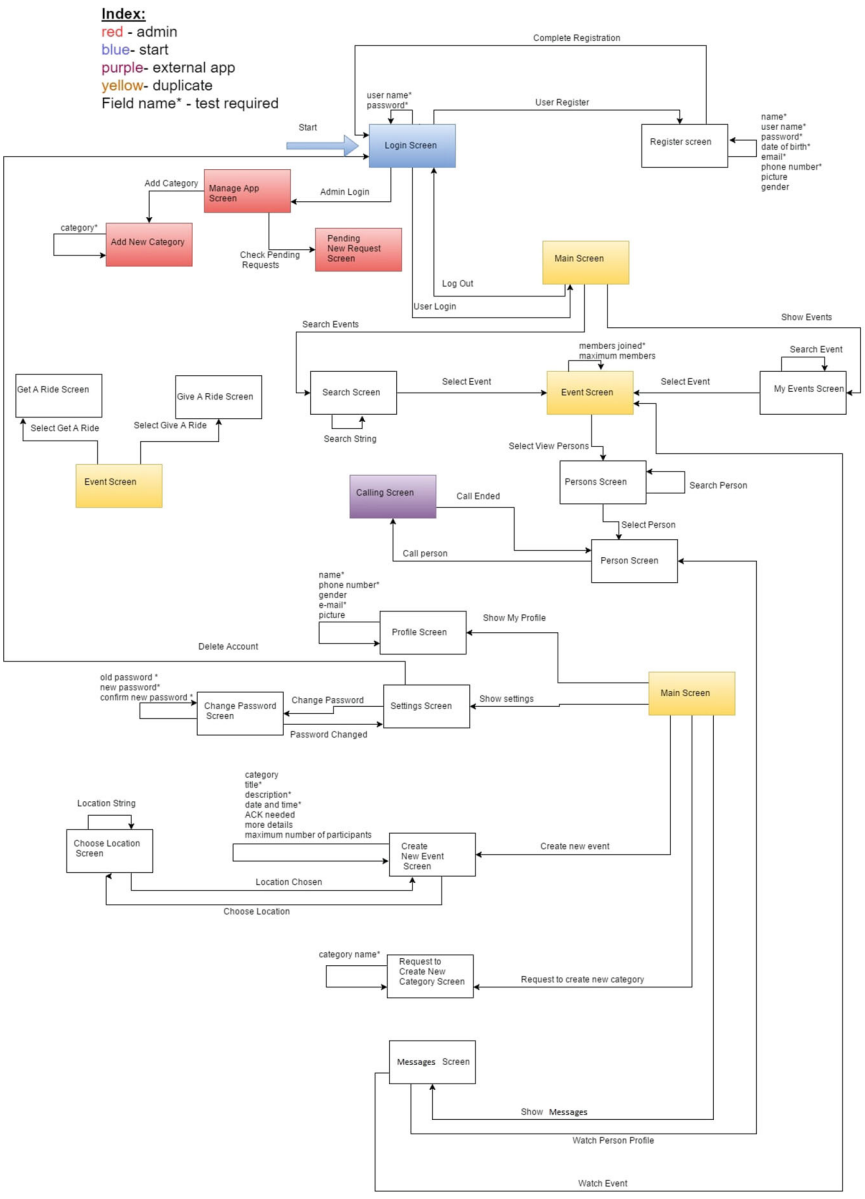


Fig. 3 The screen diagram of BoPo application

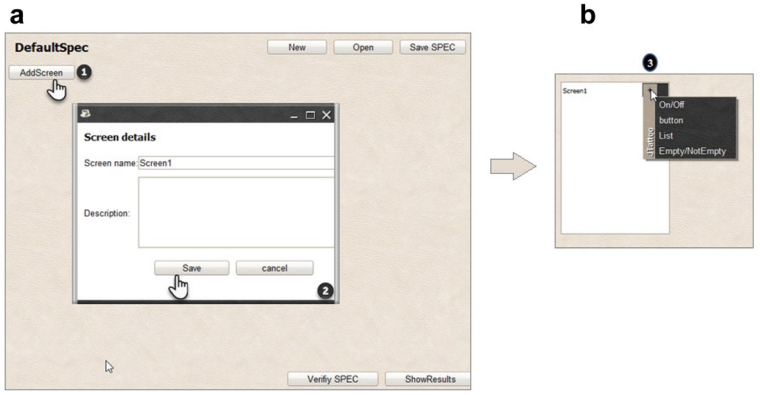


Fig. 4 (**a**) Adding a new screen. (**b**) Adding a new element to the screen

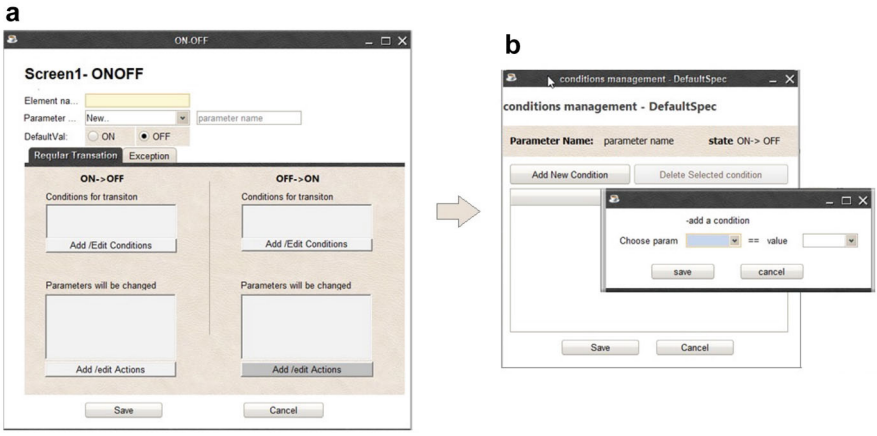


Fig. 5 (**a**) Definition of an On/Off element. (**b**) Definition of an On/Off action

handling of On/Off elements in both frontend and backend is presented. Handling of other types of elements is similar and is omitted.

On-Off Elements This type of element allows activation or deactivation of some features, like *Wi-Fi* , *Bluetooth*, and *Airplane Mode* from our motivation example. For this element type, the user should specify an element name, and an action, where she can select parameters, which must be changed, when the one changes the value of the considered element. In the motivation example, values of *Wi-Fi* and *Bluetooth* must be turned OFF, when *Airplane Mode* is turned ON, see Figs. 5a, b.

Fig. 6 (**a**) Adding On/Off element *ack*. (**b**) Create New Event screen after adding *ack* element

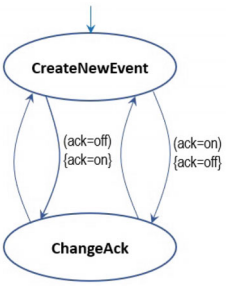


Fig. 7 Part of PG of the On/Off element *ack* in screen *Create New Event*

```
#define ON 1
#define OFF 0
mtype={CreateNewEvent, ChangeAck};
bool ack=OFF;
mtype state=CreateNewEvent;
create proctype vm()
{
do
::state=CreateNewEvent ->
        if
        :: ack==OFF -> atomic{ack=ON; state=ChangeAck}; state= CreateNewEvent;
        :: ack==ON -> atomic{ack=OFF; state=ChangeAck}; state= CreateNewEvent;
        fi
od
}
```

Fig. 8 PG of Fig. 7 coded in ProMeLa language

Frontend On Figs. 6a, b, one can see a way of presentation of On/Off element on the example of BoPo application. *Create New Event* screen contains On/Off element *ack*.

Backend In the backend, the tool creates a location (vertex) in PG for each defined screen. Then, the tool adds a location to the PG for each parameter, changed in the given On/Off element. Each change of the parameter is presented as a transition between these locations. On Figs. 7 and 8, the location *Create New Event* is the screen name, and *ChangeAck* is a location, built for tracking the parameter changes.

List Elements If the user knows a final predefined set of values of some parameter, she adds them as a List element. In such a case, the user specifies a field for Element Name, a field for Parameter Name, list of values and a default value by choosing it from the list. Moreover, the user may add an action button, where she can select parameters, which must be changed if one selects a value for the given List element.

Standard Button This type of element is used to enable moving from one screen to another. In this case, the user should specify a field for $Name$, $Conditions$, and the next screen to $Move\ to$.

Empty/Not Empty Button In this case, when some feature contains free text content, we are not supposed to analyze its real value. However, we want to reflect the fact of absence/presence of a value or any change of the value. To do so, we define an element of type $Empty/NotEmpty$. For such an element, we specify an *Element name, Parameter name*, existence of a *Default value*, and an option to add action for changing other parameters as for other types of elements. In particular, we specified a field for parameter $Name$ and an option button for $Empty$ or $NotEmpty$.

4 On Consistency of Graphically Defined Specifications

Linear Temporal Logic (LTL), cf. [1], is a convenient formalism for specifying and verifying properties of reactive systems. The underlying nature of time in LTL is linear, i.e., at each moment in time there is a single successor moment. In this section, we introduce a standard of properties for GUI design of IS applications, written as LTL formulae, which must be true throughout specifications of all such systems. We call this set of formulae: *Five Commandments for GUI design of IS Systems*.

Let us consider a spec, containing a set of screens $Screens = \{screen_1, \ldots, screen_n\}$ and a set of parameters $Parameters = \{X_1, \ldots, X_m\}$. We introduce the following set of formulae to be checked. Certainly, the set may be extended upon request.

4.1 Consistency Between Screens

The formulation is: *It is impossible to reach $screen_j$ from $screen_i$ without changing or defining the given parameter X_l*. In LTL, we code the property as:

$$\Box(screen_i \rightarrow screen_i\ U(\neg screen_i \wedge \neg(\neg Change X_l\ U screen_j))).$$

Let us consider this property on the following example: We have two modes of working—View and Edit and a parameter $Mode$ which define permission to edit. We can come from ViewScreen to EditScreen or directly, or by login of user with the corresponding privilege level. It is natural that in direct transition from ViewScreen to EditScreen we change the parameter $Mode$, but to change it also in a case of login is less obvious. In the property P4.1 we check that in any way of coming

from ViewScreen to EditScreen, the parameter mode is changed. The corresponding formula

$$\Box(ViewScreen \rightarrow$$

$$ViewScreen\ U(\neg ViewScreen \wedge \neg(\neg ChangeMode\ U EditScreen))).$$

means that for some scenario or we never come to EditScreen, or if we come, it is not correct to do it without changing of $Mode$.

4.2 Legal Values of Parameters

The formulation is: *Parameter X_l never receives value that is not defined in the list of its possible values.* Surely, implementation of this requirement depends upon the type of the parameter. In LTL, the property may be coded as:

On/Off

$$\Box((X_l = ON \vee X_l = OFF) \rightarrow \neg\Diamond(\neg(X = ON) \wedge \neg(X = OFF))).$$

Empty/NotEmpty

$$\Box((X_l = Empty \vee X_l = NotEmpty) \rightarrow$$

$$\neg\Diamond(\neg(X = Empty) \wedge \neg(X = NotEmpty))).$$

List Given a set of values $\{L_1, \ldots, L_k\}$ of X_l

$$\Box((X_l = L_1 \vee \ldots X_l = L_k) \rightarrow \neg\Diamond(\neg(X = L_1) \wedge \ldots \neg(X = L_k))).$$

The last requirement may be considered as trivial from the logical point of view. However, for specifications of information systems, where we do not define explicitly a set of parameters, each screen might use incorrect values. It means that it is important to check that, in different screens, we do not require different types of the same parameter.

For example, in a welcome screen, we ask to enter a *product ID*, and in another screen, we need to know that the *product was chosen*. It is important to show the developer such inconsistency in order to refine the specification. In fact, the developer should enter an additional parameter of the product (*was chosen*) and define dependencies between these two parameters. For this property, the classical form is “assignments should always satisfy the range annotation of parameters”.

4.3 Presence of Predefined Values of Parameters

The formulation is: *A certain list of parameters must be defined in order to enter the screen.* More precisely: Given a Standard Button B defining a transition from $screen_i$ to $screen_j$. If B contains subcondition $(X_l = NotEmpty)$, then in LTL, the property may be coded as:

$$\Box(screen_j \rightarrow (X_l = NotEmpty)).$$

In fact, this property shows another type of parameter consistency: if we need a value of parameter X_l to appear in $screen_j$, then it is natural to require that this value should not depend upon the way chosen to reach this screen. This property is similar to the classical requirement formulated as: "if a procedure is annotated with an *assert*, then check that the *assert* never fails" just rephrased for the graphical language.

4.4 The Desired Changing Values of Parameters Only

The formulation is: *Parameters values cannot be changed unless it was intended to do so in its path.* In LTL, the property may be coded as:

$$\bigwedge_{X_l \in Parameters} \bigwedge_{val} \Box((X_l = val) \rightarrow$$

$$\neg \circ (\neg ChangeX_l) U (\neg(X_l = val)) \wedge (\neg ChangeX_l))).$$

The requirement means that if a parameter is changed in a specific state, then the change should be updated wherever the parameter is used.

4.5 All Parameters Always Must Be Consistent

The formulation is: *For GUI element X_l associated with an action $\{X_1 = val_1, \ldots, X_j = val_j\}$, there is a dependency between value of parameter X_l and the values of parameters from the action.* In LTL, the property may be coded as:

$$\Box((X_l = val_l) \rightarrow (\bigwedge_m ((\bigvee_k ChangeX_k) U (X_m = val_m)))) \bigwedge$$

$$\Box(((\bigvee_m \neg(X_m = val_m)) \wedge (\neg \bigvee_k ChangeX_k)) \rightarrow \neg(X_l = val_l)).$$

This property for our motivation example described in Sect. 1.1 can be formulated as follows: For GUI element *Airplane Mode* , we have an action *Wi-Fi = Off* , when *Airplane Mode* is *On*, and the corresponding formula

$$\Box((AirplaneMode = On) \rightarrow (Change\ WiFi\ U(WiFi = Off))) \bigwedge$$

$$\Box((\neg(WiFi = Off) \wedge \neg(Change\ WiFi)) \rightarrow \neg(AirplaneMode = On))$$

if it holds, should prevent the inconsistent situation in our motivation example (Fig. 1).

5 Conclusion

In this paper, we introduced a method and its implementation as a toy tool. The tool allows graphical definition of specifications for GUI design of IS applications. Given a list of requirements, which should be satisfied in every such specification and which are expressible in LTL. The tool allows *the sanity check* of specifications for these applications. The specification composer may check satisfaction of the requirements in the spec, using formal verification machinery. The method allows prevention of bugs even before coding, at the stage of the specification definition but not during QA. As it was shown with different examples, the method allows extracting a model, coded in ProMeLa, of a specification of each information system methodically and in complete way.

Previously, such specifications were presented as shown on Fig. 9 and transformation to the corresponding model was executed manually. The question of how to come up with suitable LTL properties is more complicated and goes back to more general question: what is the complete set of properties for any formal verification suit? The question is widely studied in the literature and is out the scope of our contribution. We propose only a small-scale example of such a suit.

We emphasize that all the properties *does not depend on a particular* specification and must hold in all specifications of the considered kind. Our toy tool translates the specification to PG, and the requirements to LTL formulae. Then, PG is coded, using ProMeLa language; and the verification is done, using $SPIN$. Our approach may be extended to partial automatic GUI code generation as described, for example, in [8].

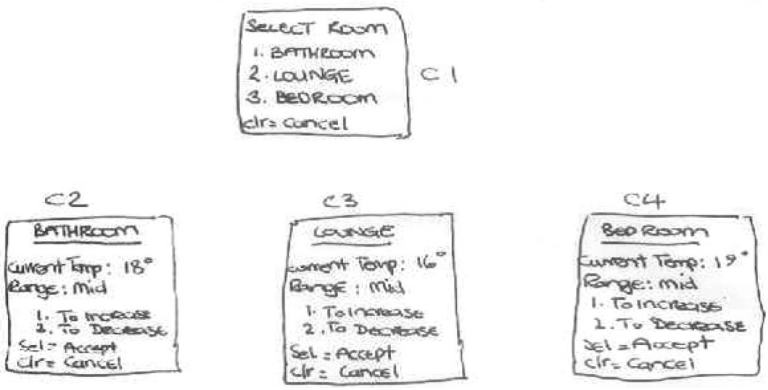


Fig. 9 Design for Mobile Phone Application UI, cf. [2]

Acknowledgments We would like to thank our students S. Namih, and A. Mnasra for implementation of the tool.

We would like to thank the anonymous reviewer for the comments, which helped us to significantly improve the paper.

Disclaimer
Here, we explicitly declare that the proposed paper contains a significant common part of definitions and examples of [8], taken almost verbatim. In fact, we do use the same approach. However, while [8] is concentrated on the automatic code generation, this contribution is dedicated to a completely different practical impact of theoretical computer science.

Appendix

Program Graph and Transition System

Program graph as a logical structure, cf. [2], is defined over a set Var of typed variables. Let $Eval(Var)$ denote a set of (variable) evaluations that assign values to variables. Let $Cond(Var)$ be a set of Boolean conditions over Var.

Definition 1 A Program Graph over set Var of typed variables is a tuple $(Loc, Act, Effect, \rightarrow, Loc_0, g_0)$ where:

- Loc is a set of locations,
- Act is a set of actions,
- $Effect : Act \times Eval(Var) \rightarrow Eval(Var)$ is the effect function,
- $\rightarrow \subseteq Loc \times Cond(Var) \times Act \times Loc$ is the conditional transition relation,
- $Loc_0 \subseteq Loc$ is a set of initial locations,
- $g_o \in Cond(Var)$ is the initial condition.

For verification purposes, a simpler model of the system, called a *Transition System*, cf. [2], may be used.

Definition 2 A Transition System (TS) is a tuple $(S, Act, \rightarrow, I, AP, L)$, where

- S is a set of states,
- Act is a set of actions,
- $\rightarrow \subseteq S \times Act \times S$ is a transition relation,
- $I \subseteq S$ is a set of initial states.
- AP is a set of atomic propositions,
- $L : S \rightarrow 2^{AP}$ is a labeling function.

$L(s)$ intuitively stands for exactly those atomic propositions $a \in AP$, which are satisfied on the state s.

Each program graph can be interpreted as a transition system. The underlying transition system of a program graph results from unfolding. Its states consist a location of the program graph, together with an evaluation η of the variables. Initial states are initial locations that satisfy the initial condition g_0. In order to formulate properties of the system described by a program graph, the set AP of atomic propositions is comprised of locations (to be able to state at which control location the system currently is), and Boolean conditions for the variables.

Linear Temporal Logic LTL

LTL, cf. [1], is a convenient formalism for specifying and verifying properties of reactive systems. The underlying nature of time in temporal logics is linear, i.e., at each moment in time there is a single successor moment. LTL formulae over the set AP of atomic proposition are formed according to the following grammar:

$$\varphi := true | a | \varphi_1 \wedge \varphi_2 | \neg\varphi | O\varphi | \varphi_1 U \varphi_2, \text{ where } a \in AP.$$

The basic ingredients of LTL-formulae are atomic propositions (state labels $a \in AP$), the Boolean connectors like conjunction $\wedge$, and negation $\neg$, as well as two basic temporal modalities O (pronounced "next") and U (pronounced "until").

LTL formulae stands for properties of paths (or in fact their trace). This means that a path can either satisfies an LTL-formula or not. The semantics of LTL formula is defined as a language $Words(\varphi)$ that contains all infinite words over the alphabet 2^{AP}, which satisfy φ. For each LTL formula, a single property is associated. Then, the semantics is extended to an interpretation over paths and states of a transition system. Several model-checking tools use LTL as a property specification language. The model checker $SPIN$ is a prominent example of such an automated verification tool.

SPIN Tool

SPIN is a popular verification tool for analyzing the logical consistency of asynchronous systems, used by thousands of people worldwide. It was developed at Bell Labs in the UNIX group of the Computing Sciences Research Center, starting in 1980. The tool can be used for the formal verification of multi-threaded software applications. *SPIN* can perform interactive, guided, or random simulations of the system execution.

ProMeLa Language

ProMeLa language (Process or Protocol Meta Language) is a verification modeling language. The language allows for the dynamic creation of concurrent processes to model, for example, distributed systems. In ProMeLa models, communication via message channels can be defined to be synchronous or asynchronous. ProMeLa models can be analyzed with the *SPIN* model checker, to verify that the modeled system produces the desired behavior. ProMeLa programs consist of processes, message channels, and variables. Processes are global objects that represent the concurrent entities of the distributed system. Processes specify behavior, channels and global variables define the environment, in which the processes run.

References

1. Baier, C., Katoen, J.P.: Principles of Model Checking (Representation and Mind Series). The MIT Press, Cambridge (2008)
2. Bowen, J., Reeves, S.: Formal models for informal GUI designs. Electron. Notes Theor. Comput. Sci. **183**, 57–72 (2007)
3. Duke, D., Fields, R., Harrison, M.: A case study in the specification and analysis of design alternatives for a user interface. Formal Asp. Comput. **11**, 107–131 (1999)
4. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Patterns in property specifications for finite-state verification. In: Proceedings of the 21st International Conference on Software Engineering. ICSE '99, pp. 411–420. Association for Computing Machinery, New York (1999)
5. Fecko, M., Lott, C.: Lessons learned from automating tests for an operations support system. Softw. Pract. Exp. **32**, 1485–1506 (2002)
6. Hadar, I., Zamansky, A.: Cognitive factors in inconsistency management. In: 2015 IEEE 23rd International Requirements Engineering Conference (RE), pp. 226–229 (2015)
7. Hadar, I., Zamansky, A., Berry, D.M.: The inconsistency between theory and practice in managing inconsistency in requirements engineering. Empir. Softw. Eng. **24**(6), 3972–4005 (2019)
8. Korenblat, K., Ravve, E.: Automatic code generator for screen based systems. In: Welzer, T., Eder, J., Podgorelec, Viliand Wrembel, R., Ivanović, M., Gamper, J., Morzy, M., Tzouramanis, T., Darmont, J., Kamišalić Latifić, A. (eds.) New Trends in Databases and Information Systems, pp. 253–265. Springer International Publishing, Cham (2019)

9. Levy, M., Hadar, I., Aviv, I.: A requirements engineering methodology for knowledge management solutions: integrating technical and social aspects. Requir. Eng. **24**(4), 503–521 (2019)
10. Narizzano, M., Pulina, L., Tacchella, A., Vuotto, S.: Property specification patterns at work: verification and inconsistency explanation. Innov. Syst. Softw. Eng. **15**(3–4), 307–323 (2019)
11. Peled, D., Gries, D., Schneider, F. (eds.): Software Reliability Methods. Springer, Berlin (2001)
12. Ravve, E.: Template based automatic generation of DRC and LVS runsets. Int. J. Adv. Softw. **10**, 143–154 (2017)
13. Rodríguez-Valdés, O., Vos, T.E.J., Aho, P., Marín, B.: 30 years of automated gui testing: a bibliometric analysis. In: Paiva, A.C.R., Cavalli, A.R., Ventura Martins, P., Pérez-Castillo, R. (eds.) Quality of Information and Communications Technology, pp. 473–488. Springer International Publishing, Cham (2021)
14. Silver, M., Markus, M., Beath, C.: The information technology interaction model: a foundation for the mba core course. MIS Q. **19**, 361–390 (1995)
15. Zamansky, A., Hadar, I., Berry, D.M.: Reasoning about inconsistency in RE - separating the wheat from the chaff. In: ENASE (2016)

The Path-bifurcation Hierarchy Does Not Collapse to Σ^1 in Infinite Abelian Groups

Mihai Prunescu

Abstract Some general transfer principles for unit cost complexity problems are proved. As an application, we show that for infinite abelian groups $\Sigma^2 \cap \Pi^2 \neq \Sigma^1 \cup \Pi^1$. This has been shown for the group of real numbers by Herve Fournier and Pascal Koiran using the problem Twenty Questions. As this problem is not appropriate for infinite abelian groups in general, it was replaced here by Null-sack.

1 Introduction

The unit cost model of computation has been introduced by Blum, Shub and Smale in [1] as a computation model for real numbers. The model achieved maturity already in their monograph [2] and has been generalized for arbitrary algebraic structures by Poizat in [3]. This theory is a substantial connection between the classical theory of computational complexity and other branches of logic, like model-theory. One of the first papers published in this direction was Meer's article [4]. For an interesting survey, see Makovsky and Meer [5].

As the principles of unit cost complexity are very well presented in the monographs cited above, specially by Poizat in [3], I will recall them only informally. Also, I will try to keep notations as simple as possible.

By unit cost machine we understand a multi-tape Turing machine, which is able to work with the elements of some algebraic structure. The description of the machine is best given if we start with a finite signature L. L contains finitely many operation symbols of given arities, finitely many relation symbols of given arities, and finitely many constant symbols. The machine works on a presumable (so far unspecified) L-structure. The elements of this unspecified L-structure can be used

M. Prunescu (✉)
Research Center for Logic, Optimization and Security (LOS), Faculty of Mathematics and Computer Science, University of Bucharest, Bucharest, Romania

Simion Stoilow Institute of Mathematics of the Romanian Academy, Research Unit 5, Bucharest, Romania
e-mail: mihai.prunescu@fmi.unibuc.ro; mihai.prunescu@imar.ro

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*, Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_26

like letters of a possibly infinite alphabet. The input of the machine is a string built by structure elements which stay in consecutive cells on some tape.

An L-machine has only finitely many states, and has a finite description consisting of transition rules, as follows. In a given state and for given positions of the heads, the machine can:

- delete an element in a tape-cell. A cell can be also empty.
- write in a tape cell an element of the structure. This element can interpret an L-constant, can be copied from another cell (tape) or may be the result of some computation done by the machine. In the last case the arguments of this computation are elements read by machine heads.
- verify if a tuple of elements, which are located in cells viewed by given heads, or which interpret L-constants, satisfy or not a relation. This test can be also an equality test. According to the result of this test, the machine changes its state.
- move (or not) different heads.

Any such action takes exactly one computation step. Like classical Turing machines, a unit cost machine has a finite set of states, and has final states which may be accepting or rejecting. Also, the machine might have only one final state, and may output a string of elements on a tape.

So far, an L-machine is just an abstract object which cannot do anything. Only once we specify an L-structure S - meaning that there are operations and relations in S interpreting the corresponding L-symbols, and once given fixed elements of S interpreting the constant L-symbols - an L-machine can work on S according to the given interpretation of L.

For an L-structure S, we may speak about problems over S and about machines over S, but in both cases the corresponding signature L and its interpretation must be already fixed. A problem over S is a set of finite strings of elements of S. The machine over S is able to decide if an arbitrary string of elements belongs to a problem. We are interested in the situation in which the machine is able to make such a decision after performing a number of steps which is bounded by some function of n, where n is the length of the string. If this is the case, and the function is a polynomial, the set of problems decided by polynomial-time machines over the L-structure S build together the complexity class $P(S, L)$.

Like in the classical theory of computational complexity, there are also non-deterministic machines. The generalisation is done such that if one takes $L = (0, 1)$ and $S = \{0, 1\}$, one gets back the classical complexity classes. We follow the convention used in model-theory, where the equality sign is not mentioned in the signature as it is considered to belong to the logical symbols. An L-machine is always able to check if the content of two different cells is equal or different.

A classical non-deterministic Turing machine has non-deterministic states, in which the machine may continue its computation along two different paths, and chooses randomly one of them. If the work time of a classical non-deterministic machine is bounded by some function on the length of the input, those machines can be organized as follows. In a non-deterministic stage, the machine writes sufficiently many random bits on an extra tape. Then the machine goes over in the deterministic

stage. At any path bifurcation, the machine read the random bit written down during the non-deterministic stage and chooses the continuation (deterministically!) according to this bit.

Suppose that the signature L contains at least one constant, that we may call 0. Suppose that a unit cost machine has non-deterministic states like the classical machines. This machine can be also transformed in a machine with a non-deterministic stage followed by a deterministic stage. During the non-deterministic stage, the machine guesses (picks randomly) a sufficient number of structure elements, then starts working with its original input. If the machine comes in a (formerly) non-deterministic state, it reads the next element written down during the non-deterministic stage and tests its equality with 0. If true, the machine continues along the left-hand computation path. If false, it continues along the right-hand one. This leads to the definition of the boolean complexity class NP, denoted by Poizat $NBP(S, L)$. A problem A belongs to $NBP(S, L)$ if and only if there is some problem $B \in P(S, L)$ and some polynomial q such that for every $n > 0$,

$$(x_1, \ldots, x_n) \in A \longleftrightarrow \exists y_1, \ldots, y_{q(n)} \in \{0, 1\} \quad (\mathbf{x}, \mathbf{y}) \in B.$$

Here we denoted with 1 those guessed elements which are different from 0. In the case that the signature L contains at least two constants, they can be denoted by 0 and 1. The boolean non-determinism can be performed even with no constants in L, provided that S hat at least two elements. Instead of guessing k bits, one may guess $2k$ elements from the structure. When it comes to choose a computation path after $s - 1$ bifurcations, one continues along the left-hand path if $y_{2s} = y_{2s+1}$ and along the right-hand one if $y_{2s} \neq y_{2s+1}$.

For the existential non-determinism, one only arbitrarily guesses elements in the structure and uses them in further computations. A problem A belongs to $NP(S, L)$ if and only if there is some problem $B \in P(S, L)$ and some polynomial q such that for every $n > 0$,

$$(x_1, \ldots, x_n) \in A \longleftrightarrow \exists y_1, \ldots, y_{q(n)} \in S \quad (\mathbf{x}, \mathbf{y}) \in B.$$

The existential non-determinism is more general then the boolean non-determinism, and this naturally implies that:

$$P(S, L) \subseteq NBP(S, L) \subseteq NP(S, L).$$

In the next section, further complexity classes are defined.

Meer studied in [6] the linear computation model over real numbers. In this model, it is possible to multiply an element with a parameter of the machine or to add it to a parameter of the machine, or to other elements existent in the tape cells. He found out that according to this model of computation, $P(\mathbb{R}) \neq NBP(\mathbb{R})$ with real parameters. This implies that $P(\mathbb{R}) \neq NBP(\mathbb{R})$ with real parameters is also true for the additive group $(\mathbb{R}, +, -, 0, (c \in \mathbb{R}))$. Poizat added in [3] other results like $P \neq NBP$ for atom-less boolean algebras with arbitrary elements as parameters or

for the group $\mathbb{H}_2 = \oplus_\omega \mathbb{Z}_2$, also with arbitrary elements as parameters. The book contains also many other results, as structures with $P = NBP$ and connections with the classical complexity problems. The cited results were generalised by the author for all infinite boolean algebras with parameters [7] and for all infinite abelian groups with parameters [8]. A question of Poizat is solved in [9] (and presented in [10] with even more details): there is a structure with $P(S) = NP(S)$. The structure displayed has however only unary operations, and the existence of a function with binary operations, as originally asked by Poizat, is still open.

The paper [11] by Fournier and Koiran deserves a lot of attention. The paper contains three great results. First, that in the ordered group of the reals $(\mathbb{R}, +, -, <, 0)$ one has $P = NBP$ if and only if $P = NP$ classically. This means that in order to solve the Knapsack problem, one has basically to perform only additions and order tests. This result was generalized in [12] for other ordered structures with addition. Second, that for the group of the reals, without considering the order, $\Sigma^2 \cap \Pi^2 \neq \Sigma^1 \cup \Pi^1$. We will concentrate here on this result. Third, a lot of implications and transfers from the classical polynomial hierarchy to the polynomial hierarchy over the group of real numbers without order.

Warning There is a difference between the way how I understand L-machines in this article and the way it is understood in various classical texts, like [2] or [3]. In these books and in many articles one considers that any element of the structure may be used as constants (parameters) in various machines, without mentioning it. As in the present paper, I focus on transfer results, and as these results are naturally got parameter-free, I use the following convention. *Every time that a structure is discussed, it is presented as an L-structure for some signature L. When I speak about L-machines, only those constants which are mentioned in the signature L may be used as machine parameters.*

2 Hierarchy Classes

Here we define polynomial hierarchies, both existential and boolean. To write down some complexity class, one can use four quantifier symbols: $\exists_b$, $\exists$, $\forall_b$, $\forall$. The quantifiers indexed with b are boolean:

$$\exists_b x\ \varphi :\longleftrightarrow \varphi(0) \vee \varphi(1),$$

$$\forall_b x\ \varphi :\longleftrightarrow \varphi(0) \wedge \varphi(1).$$

Let L be a finite set of operation symbols, relation symbols and constants and let S be an L-structure. By definition $\Sigma^0(S, L) = \Pi^0(S, L) = P(S, L)$. For symmetry, the quantifiers which run over the whole universe of the structure S may be denoted by $\forall_s$ and $\exists_s$.

Definition 1 Consider the set of problems $\Pi^k_\iota(S, L)$, where $k \geq 0$ and the index $\iota \in \{b, s\}^*$ is a word of $|\iota| = k$. Then the set $\Sigma^{k+1}_{b\iota}(S, L)$ consists of the problems B such that there is a polynomial $p(n)$ and a problem $A \in \Pi^k_\iota(S, L)$ such that for all $n \in \mathbb{N}$:

$$\forall\, x_1, \ldots\, x_n \in S \quad (\mathbf{x} \in B \longleftrightarrow \exists_b\, \varepsilon_1, \ldots, \varepsilon_{p(n)} \quad (\mathbf{x}, \boldsymbol{\varepsilon}) \in A).$$

Also, the set $\Sigma^{k+1}_{s\iota}(S, L)$ consists of the problems B such that there is a polynomial $p(n)$ and a problem $A \in \Pi^k_\iota(S, L)$ such that for all $n \in \mathbb{N}$:

$$\forall\, x_1, \ldots\, x_n \in S \quad (\mathbf{x} \in B \longleftrightarrow \exists_s\, \varepsilon_1, \ldots, \varepsilon_{p(n)} \quad (\mathbf{x}, \boldsymbol{\varepsilon}) \in A).$$

The sets of problems $\Pi^{k+1}_{b\iota}(S)$ and $\Pi^{k+1}_{s\iota}(S)$ consist of the complements of the problems contained in the corresponding Σ classes.

Consequently $NBP(S, L) = \Sigma^1_b(S, L)$, $NP(S, L) = \Sigma^1_s(S, L)$, $coNBP(S, L) = \Pi^1_b(S, L)$, $coNP(S, L) = \Pi^1_s(S, L)$.

There are also various mixed classes, like $\Sigma^3_{sbs}(S, L)$:

$$(x_1, \ldots, x_n) \in A \longleftrightarrow$$

$$\longleftrightarrow \exists_s\, y_1, \ldots, y_{p(n)}\ \forall_b\, z_1, \ldots, z_{q(n)}\ \exists_s\, w_1, \ldots, w_{r(n)} \quad (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) \in B.$$

Here the index sbs means that the quantifier blocks are in this order standard, boolean, standard. But as boolean quantification is only a particular case of formula with quantifiers, it follows that $\Sigma^3_{sbs}(S, L) \subseteq \Sigma^3(S, L)$, where the former is defined with standard quantifiers only, and similarly $\Sigma^k_\iota(S, L) \subseteq \Sigma^k(S, L)$, $\Pi^k_\iota(S, L) \subseteq \Pi^k(S, L)$, for every $k \in \mathbb{N}$ and $\{s, b\}$-index ι.

3 P = NBP Is Unlikely

Not all problems are equally good for transfer results from unit-cost complexity to classical complexity. Take for example the problem Twenty Questions, presented in [2] as a candidate to prove that $P \neq NBP$ in the field $(\mathbb{C}, +, -, \cdot, 0, 1)$ of the complex numbers.

$$TQ = \{(x_1, \ldots, x_n) \,|\, n > 0, x_1 \in \{0, 1, 2, \ldots, 2^n - 1\}\} =$$

$$= \{(x_1, \ldots, x_n) \,|\, n > 0,\ \exists\, \varepsilon_0, \ldots, \varepsilon_{n-1}\ \ x_1 = 2^0\varepsilon_0 + \cdots + 2^{n-1}\varepsilon_{n-1}\}.$$

The problem is of course in NBP. One can produce the powers of two in unit cost polynomial time by successive addition of the current value with itself, as $1 \rightsquigarrow$

$2 \rightsquigarrow 4 \rightsquigarrow \cdots \rightsquigarrow 2^k \rightsquigarrow 2^{k+1} \rightsquigarrow \ldots$. But surprisingly, the restriction of this problem to natural numbers produces a trivial problem. If one gets a sequence of natural numbers represented by bit strings, one has only to count the bits of the first number.

For other problems, the transfer from unit cost complexity to classical complexity works very easy, as in the following example.

Definition 2 For some string s, let $|s|$ mean its length. Finite sequences of natural numbers $n_1, \ldots n_k$ are represented as follows: the numbers are represented binary and encoded such that the digit 0 is represented as 00, the digit 1 is represented as 10 and the comma is represented as 11.

Theorem 1 *Let $L = (+, o_1, \ldots, o_k, R_1, \ldots, R_m, 0, 1)$ be a signature containing a binary operation "+", finitely many other operations o_s of arities a_s, finitely many relations R_t of arities b_t, and two constants. Let S be an L-structure such that:*

- *$\mathbb{N} \leq_L S$ is a substructure; "+" restricted to $\mathbb{N}$ is the addition of natural numbers; $0, 1 \in \mathbb{N}$ have their usual meaning.*
- *Every operation o_s restricted to $\mathbb{N}$ is polynomial time computable in the total bit-length of the input $(n_1, \ldots, n_{a_s})$.*
- *For every operation o_s restricted to $\mathbb{N}$, (as also for the addition of natural numbers) there is a $k_s \in \mathbb{N}$ such that*

$$o_s(n_1, \ldots, n_{a_s}) < k_s \max(n_1, \ldots, n_{a_s}).$$

- *Every relation R_t restricted to $\mathbb{N}$ is polynomial time decidable in the total bit-length of the input $(n_1, \ldots, n_{b_t})$ according to the defined encoding.*

If $P(S, L) = NBP(S, L)$ (according to the signature L, so considering only parameter-free machines) then $P = NP$ in the classic sense.

Proof We consider the unit cost Knapsack problem in S:

$$KS = \{(x_1, \ldots, x_n) \mid n > 0, \exists\, J \subseteq \{1, \ldots, n-1\}, J = \{j_1 < \cdots < j_k\},$$

$$(\ldots((x_{j_1} + x_{j_2}) + x_{j_3}) + \cdots + x_{j_{k-1}}) + x_{j_k} = x_n\}.$$

As we did not say that the addition is associative in S, we have to use the parentheses.

As $P(S, L) = NBP(S, L)$, there is a unit cost L-machine able to decide KS in polynomial time $p(n)$ running over S. Then there exists an algorithm able to solve the classic Knapsack problem in polynomial time. Let $(n_1, \ldots, n_k)$ be a sequence of natural numbers building an instance of Knapsack. Let n be the length of this string, which is encoded according to our convention. One has $k < n$. If we consider $(n_1, \ldots, n_k)$ to be an instance of the unit cost problem KS, the decision about this instance is taken in unit cost time $p(k)$. But this computation can be simulated by a classical deterministic machine. Let q be a polynomial such that every computation

of an operation o_s and every decision of a relation R_k is bounded by corresponding values of q. For the addition of natural numbers, one can take the constant $k_+ = 3$. Let $k := \max(k_+, k_1, \ldots, k_s)$. Let M be the largest value in the set $\{n_1, \ldots, n_k\}$. The biggest possible value that can be computed, is $< Mk^{p(n)}$. The total space used by all computed values is:

$$< \log(M + Mk + Mk^2 + \cdots + Mk^{p(n)}) < \log(2Mk^{p(n)+1}) \approx A + p(n)B.$$

The total computation time is:

$$< \sum_{i=1}^{p(n)} q(1 + \log M + i(1 + \log k)) <$$

$$< p(n)q(1 + \log M + p(n)(1 + \log k)) < p(n)q(n + np(n)),$$

and this bound is polynomial. It follows that $P = NP$. □

The multiplication of natural numbers does not act as in the hypothesis. We observe that for a ring extending $\mathbb{Z}$, not every unit cost machine can be simulated by a classical machine, so in general the transfer is difficult also from unit cost to classical.

4 Elementary Equivalence

A transfer result referring to groups and to the complexity classes P and NBP, introduced in [8], is generalized here for arbitrary hierarchy classes and arbitrary structures. In this section, L is some signature containing finitely many operation-, relation- and constant-symbols.

Lemma 1 *Let S be an L-structure and let $A \in P(S, L)$ be a polynomial-time unit cost problem over S. Then for every $n \in \mathbb{N}$ there is a quantifier-free L-formula $\psi_n(x_1, \ldots, x_n)$, that does not contain other free variables as $x_1, \ldots, x_n$, such that:*

$$\forall\, x_1, \ldots, x_n \in S \quad (x_1, \ldots, x_n) \in A \longleftrightarrow \psi_n(x_1, \ldots, x_n).$$

Proof Consider the deterministic unit-cost machine M that decides the problem A in some polynomial time $p(n)$. For every input-length n, and for every possible computation path ending with an accepting final state, one writes down the tests determining the current computation path. Those tests are quantifier-free formulas of one of the following shapes: $t_1(\mathbf{x}) = t_2(\mathbf{x})$, $t_1(\mathbf{x}) \neq t_2(\mathbf{x})$, $R_i(t_1(\mathbf{x}), \ldots, t_{b_i}(\mathbf{x}))$ respectively $\neg R_i(t_1(\mathbf{x}), \ldots, t_{b_i}(\mathbf{x}))$. Here $t_s(\mathbf{x})$ are L-terms. They are determined by the computations done along the path so far. If $\tau_1(\mathbf{x}), \ldots, \tau_k(\mathbf{x})$ are the tests satisfied along some accepting path π, then $k \leq p(n)$ and we associate the path π

with the conjunction:

$$\pi(\mathbf{x}) := \tau_1(\mathbf{x}) \wedge \cdots \wedge \tau_k(\mathbf{x}).$$

The formula $\psi_n(\mathbf{x})$ is then the disjunction of the formulas $\pi(\mathbf{x})$ over all accepting computation paths π. □

Lemma 2 *Let S be an L-structure and let B be some problem in a given complexity class C, where C is $\Sigma_\iota^k(S, L)$ or $\Pi_\gamma^m(S, L)$ and ι and γ are corresponding $\{s, b\}$-indexes. Then there is a sequence of L-formulas in prenex form $\delta_n(x_1, \ldots, x_n)$ such that for all $n \in \mathbb{N}$:*

$$\forall\, x_1, \ldots, x_n \in S \quad (x_1, \ldots, x_n) \in B \longleftrightarrow \delta_n(x_1, \ldots, x_n).$$

The quantifier prefix of δ_n corresponds to the definition of the corresponding hierarchy class C.

Proof The complexity class C is the set of problems B such that there is a problem $A \in P(S, L)$ with the property that for every $n > 0$,

$$(x_1, \ldots, x_n) \in B \longleftrightarrow Pr_\iota^k(\mathbf{y}^1, \ldots, \mathbf{y}^k) \quad (\mathbf{x}, \mathbf{y}^1, \ldots, \mathbf{y}^k) \in A.$$

Here the alternating prefix Pr_ι^k consists of blocks of quantifiers according to the corresponding complexity class. There are polynomials $p_1(n), \ldots, p_k(n)$ such that $|\mathbf{y}^s| = p_s(n)$. We apply Lemma 1. Let (ψ_n) be the corresponding sequence of formulas corresponding to the problem A. Then:

$$(x_1, \ldots, x_n) \in B \longleftrightarrow Pr_\iota^k(\mathbf{y}^1, \ldots, \mathbf{y}^k) \quad \psi_{n+p_1(n)+\cdots+p_k(n)}(\mathbf{x}, \mathbf{y}^1, \ldots, \mathbf{y}^k)$$

□

Theorem 2 *Let S_1 and S_2 be two elementary equivalent L-structures. Let C' and C'' be two hierarchy complexity classes, defined as Σ or Π with some $\{s, b\}$-indexes. If $C'(S_1, L) = C''(S_1, L)$ then $C'(S_2, L) = C''(S_2, L)$.*

Proof Suppose that $C'(S_1, L) = C''(S_1, L)$. Let $B(S_1) \in C'(S_1, L)$ be some problem. By Lemma 2, there is a sequence of formulas (δ'_n) such that any tuple $\mathbf{x} \in S_1$ belongs to $B(S_1)$ if and only if $\delta'_{|\mathbf{x}|}(\mathbf{x})$ is true in S_1. Let (δ''_n) be the sequence of formulas corresponding to the problem $B(S_1)$ as element of the complexity class $C''(S_1, L)$. We define the closed first order statement:

$$\alpha_n : \quad \forall\, x_1, \ldots, x_n \quad \delta'_n(x_1, \ldots, x_n) \longleftrightarrow \delta''_n(x_1, \ldots, x_n).$$

One has a corresponding sequence of L-statements for every problem in the set $C'(S_1, L) = C''(S_1, L)$, and for all n, $S_1 \models \alpha_n$. By elementary equivalence, for all n, $S_2 \models \alpha_n$, so the corresponding problem $B(S_2) \in C'(S_2, L)$ and $B(S_2) \in$

$C''(S_2, L)$. If some problem was in $C'(S_2, L) \setminus C''(S_2, L)$, then by transfer from S_2 to S_1, one gets $C'(S_1, L) \neq C''(S_1, L)$. Contradiction. □

5 Fields of Fractions

Let D be a commutative domain with 1 and let Q be its field of fractions. In the following, we deal with parameter-free unit cost computations over the additive groups of these rings. Only boolean non-determinism is considered.

Lemma 3 *Let $x_1, \ldots, x_n \in Q$ be a tuple of elements. Then there is a subgroup D' of the additive group of K such that $x_1, \ldots, x_n \in D'$ and the additive groups D' and D are isomorphic.*

Proof The elements x_i are represented in the form $x_i = y_i/z$, where all $y_i, z \in D$ and z is a common denominator of the fractions $x_1, \ldots, x_n$. Then $x_1, \ldots, x_n \in D'$ where $D' = (1/z)D$. We observe that the additive groups D' and D are isomorphic. □

Theorem 3 *Let D be a commutative domain with 1 and let Q be its field of fractions. We consider parameter-free unit cost computations over the groups D and Q with the signature $L = (+, -, 0)$. Let C' and C'' be two hierarchy classes, such that their (s, b)-indexes contain only the letter b (this means, again, that one has only path-bifurcation non-determinism, i.e. all quantifiers act over $\{0, 1\}$). Then:*

$$C'(D, L) = C''(D, L) \Longleftrightarrow C'(Q, L) = C''(Q, L).$$

Proof Let C'' be the class defined with the longer prefix. $C'' = C'$ means that for every C'' alternate L-machine M'', the problem X recognised by this machine is also recognised by a C' alternate L-machine M'.

Suppose that $C'(Q, L) = C''(Q, L)$. Take a C'' alternate L-machine M'' which defines a problem $X(Q)$ when it runs over Q. We know that there is a C' alternate L-machine M' which recognizes the same problem $X(Q)$ when it runs over Q. As $D \subset Q$ is an L-substructure, and $X(Q) \cap D = X(D)$, the problem recognized by M'' running over D is exactly $X(D)$ and the same problem is recognized by M' when it runs over D. It follows that $C'(D, L) = C''(D, L)$.

Suppose now that $C'(D, L) = C''(D, L)$. Take a C'' alternate L-machine M'' which defines a problem $X(D)$ when it runs over D. We know that there is a C' alternate L-machine M' which recognizes the same problem $X(D)$ when it runs over D. Let $X(Q)$ be the problem recognized by M'' when it runs over Q. For every input $\mathbf{x} \in Q$ there is an additive subgroup $D' = aD$ isomorphic with D that contains x. Any element computed by M'' or by M' at a given step is an integer combination $a_1x_1 + \cdots + a_nx_n \in D' = aD$. We observe that:

$$X(Q) = \bigcup_{a \in Q \setminus \{0\}} X(aD),$$

and that $C'(aD, L) = C''(aD, L)$ for every $a \in Q \setminus \{0\}$. It follows that the machine M' running over such an additive subgroup aD recognizes the problem $X(aD)$, so M' running over Q recognizes $X(Q)$. It follows that $C'(Q, L) = C''(Q, L)$ □

6 Additive Groups of Infinite Fields

The additive groups of infinite fields are just an intermediate step towards the general infinite abelian groups. The groups we need further are introduced in the last corollaries.

In this section, K is an infinite field. We consider unit cost computations over the additive group of K, meaning that the machines use the signature $L = (+, -, 0)$. In the case $K = \mathbb{R}$, Fournier and Koiran proved that the corresponding polynomial hierarchy has $\Sigma^1 \neq \Sigma^2$. In this section, we generalize this result to all infinite fields. The problem Twenty Questions TQ is not appropriate for this generalisation, because in characteristic p the range of values taken by x_1 is bounded. We consider instead another problem. The problem Null-sack, used by the author also in [8] to prove that $P \neq NBP$ over the infinite abelian groups, is defined as follows:

$$N = \{(x_1, \ldots, x_n) \mid n > 0, \exists \varepsilon_1, \ldots, \varepsilon_n \in \{0, 1\}$$

$$(\varepsilon_1 \neq 0 \vee \cdots \vee \varepsilon_n \neq 0) \wedge \varepsilon_1 x_1 + \cdots + \varepsilon_n x_n = 0\}.$$

Also, we consider the following problem $N\overline{N}$:

$$N\overline{N} = \{(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n}) \mid n > 0, (x_1, \ldots, x_n) \in N$$
$$\wedge\ (x_{n+1}, \ldots, x_{2n}) \notin N\}.$$

Definition 3 A subset $H \subseteq K^n$ defined as $H = \{(x_1, \ldots, x_n) \mid a_1 x_1 + \cdots + a_n x_n = 0\}$, where $a_1, \ldots, a_n \in K$, is called a hyperplane. The complement of a finite union of proper hyperplanes will be called a dense set.

Lemma 4 *Let K be an infinite field. A set is definable with parameters in the structure $(K, +, -, 0)$ if and only if it is a boolean combination of hyperplanes with coefficients in $\mathbb{Z}$ and a free term which is a linear combination of parameters with coefficients in $\mathbb{Z}$. If K has finite characteristic p, all the coefficients from above are consequently in $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$.*

Proof Every formula in prenex form is a boolean combination of linear equations in variables and parameters with coefficients in $\mathbb{Z}$, preceded by the quantifier blocks. But these formulas allow quantifier elimination, for example by Gauss' Algorithm. □

Theorem 4 *For every infinite field K, for $L = (+, -, 0, (c_i))$ with finitely many constants, every class $\Sigma_t^k(K, L) = \Sigma^k(K, L)$.*

Proof Pascal Koiran proved in [13] for $K = \mathbb{R}$ that $\Sigma_b^1(\mathbb{R}, L) = \Sigma_s^1(\mathbb{R}, L)$, in other words that $NBP(\mathbb{R}, L) = NP(\mathbb{R}, L)$. A proof is given also in Poizat's monograph [3]. The proof uses only the fact that $\mathbb{R}$ is an infinite field. Further, Cucker and Koiran proved in [14] that even for the signature $L \cup \{<\}$, one has that $\Sigma_\iota^k(\mathbb{R}, L) = \Sigma^k(\mathbb{R}, L)$ for every $\{b, s\}$-index ι. This is of course true also without order, and again they did not use other property as that $\mathbb{R}$ is an infinite field and that there is no multiplication between variables, between machine constants, and so on. □

Because of the result above, in this section there will be no indexes in the complexity class notation if we refer to infinite fields.

The next two results have proofs that go along the proofs of the similar results about Twenty Questions in $\mathbb{R}$ from [11].

Theorem 5 *Let K be an infinite field and $L = (+, -, 0, (c_i))$ with finitely many constants. Then $N \in \Sigma^1(K, L) \setminus \Pi^1(K, L)$.*

Proof By definition $N \in \Sigma_b^1(K, L) = \Sigma^1(K, L)$. Suppose that $N \in \Pi^1(K, L)$. As $\Pi^1(K, L) = \Pi_b^1(K, L)$, $N \cap K^n$ is an intersection of finitely many sets A_i with $A_i = P_i \cap K^n$, where $P_i \in P(K, L)$ (without parameters as well). Since $N \cap K^n$ is not dense in K^n and all A_i are boolean combinations of hyperplanes, there is an A_0 which is not dense in K^n as well. We look at the generic path for the unit cost machine deciding P_0 in polynomial time. (The generic path is the computation path obtained by answering no to all not trivially true equality tests.) The final answer for an input that goes along the generic path must be negative, and it follows that $N \cap K^n$ must be contained in a union of at most $p(n)$ hyperplanes. But $N \cap K^n$ consists of $2^n - 1$ hyperplanes, and this is a contradiction. □

Theorem 6 *Let K be an infinite field and $L = (+, -, 0, (c_i))$ with finitely many constants.*

Then $N\overline{N} \in \Sigma^2(K, L) \cap \Pi^2(K, L)$ and $N\overline{N} \notin \Sigma^1(K, L) \cup \Pi^1(K, L)$.

Proof The problem $N\overline{N}$ can be defined by the formula:

$$\exists\, \varepsilon_1, \ldots, \varepsilon_n \in \{0, 1\}\, \forall\, \varepsilon_{n+1}, \ldots, \varepsilon_{2n} \in \{0, 1\}$$

$$(\varepsilon_1 \neq 0 \vee \cdots \vee \varepsilon_n \neq 0) \wedge \varepsilon_1 x_1 + \cdots + \varepsilon_n x_n = 0 \wedge$$

$$\wedge\ [(\varepsilon_{n+1} \neq 0 \vee \cdots \vee \varepsilon_{2n} \neq 0) \longrightarrow \varepsilon_{n+1} x_{n+1} + \cdots + \varepsilon_{2n} x_{2n} \neq 0].$$

The two quantifier blocks can be commuted, so $N\overline{N}$ belongs to both $\Sigma^2(K, L)$ and $\Pi^2(K, L)$.

Suppose that $N\overline{N} \in \Sigma^1(K, L) = \Sigma_b^1(K, L)$. This means that $N\overline{N} \cap K^{2n}$ is a finite union of sets A_i, where every $A_i = P_i \cap K^{2n}$ with $P_i \in P(K, L)$ without parameters. Consider a hyperplane consisting of solutions of the left N-problem, like $S = \{\mathbf{x} \mid x_1 + x_2 = 0\}$. As $N\overline{N} \cap S$ is dense in S, there is a set A_0 which is dense in S. With the generic path method applied to P_0 for S, we see that $(K^{2n} \setminus A_0) \cap S$ is

contained in a union of $p(2n)$ hyperplanes. But $(K^{2n} \setminus N\overline{N}) \cap S$ contains $\geq 2^n - 1$ hyperplanes, and this is a contradiction.

Suppose that $N\overline{N} \in \Pi^1(K, L) = \Pi^1_b(K, L)$. This means that $N\overline{N} \cap K^{2n}$ is a finite intersection of sets A_i, where every $A_i = P_i \cap K^{2n}$ with $P_i \in P(K, L)$ without parameters. Consider a hyperplane in which the right $\overline{N}$-problem is dense, and which differs from the $2^n - 1$ hyperplanes of which the left N-problem consists, like $S = \{\mathbf{x} \mid x_{n+1} + x_{n+2} = a\}$ where $a \neq 0$. As $N\overline{N} \cap S$ is not dense in S, there must be an A_0 which is not dense in S. With the generic path method applied to P_0 for S, we see that $A_0 \cap S$ is contained in an union of $p(2n)$ hyperplanes. But $N\overline{N} \cap S$ is contained in $2^n - 1$ hyperplanes, and is dense in any of them. This is a contradiction. □

Corollary 1 *For the group of integers $\mathbb{Z}$ and the language $L = (+, -, 0)$ without parameters, the following hold:*

1. $\Sigma^1_b(\mathbb{Z}, L) \neq \Pi^1_b(\mathbb{Z}, L)$,
2. $\Sigma^2_{bb}(\mathbb{Z}, L) \cap \Pi^2_{bb}(\mathbb{Z}, L) \neq \Sigma^1_b(\mathbb{Z}, L) \cup \Pi^1_b(\mathbb{Z}, L)$.

Proof For the infinite field $\mathbb{Q}$ of rational numbers with the signature $L = (+, -, 0)$, the following statements hold: $\Sigma^k_\iota(\mathbb{Q}, L) = \Sigma^k(\mathbb{Q}, L)$ because of Theorem 4 which is more general, as it admits also computations with parameters, $\Sigma^1(\mathbb{Q}, L) \neq \Pi^1(\mathbb{Q}, L)$ because the Theorem 5 and $\Sigma^2(\mathbb{Q}, L) \cap \Pi^2(\mathbb{Q}, L) \neq \Sigma^1(\mathbb{Q}, L) \cup \Pi^1(\mathbb{Q}, L)$ because of Theorem 6.

Using the first relation, the second and the third relation are written as: $\Sigma^1_b(\mathbb{Q}, L) \neq \Pi^1_b(\mathbb{Q}, L)$ and as $\Sigma^2_{bb}(\mathbb{Q}, L) \cap \Pi^2_{bb}(\mathbb{Q}, L) \neq \Sigma^1_b(\mathbb{Q}, L) \cup \Pi^1_b(\mathbb{Q}, L)$ respectively.

The last two relations are relations between purely boolean hierarchy classes and they are transferred to the group $\mathbb{Z}$ because of the transfer Theorem 3, as $\mathbb{Q}$ is the field of fractions of $\mathbb{Z}$. □

Definition 4 Let p be a prime natural number. We define the additive group:

$$\mathbb{H}_p := \bigoplus_{\omega} \mathbb{Z}_p,$$

where $\mathbb{Z}_p$ is the group $\mathbb{Z}/p\mathbb{Z}$.

Corollary 2 *Let p be a prime number. For the group $\mathbb{H}_p$ and the signature $L = (+, -, 0)$ without parameters, the following hold:*

1. $\Sigma^k_\iota(\mathbb{H}_p, L) = \Sigma^k(\mathbb{H}_p, L)$,
2. $\Sigma^1(\mathbb{H}_p, L) \neq \Pi^1(\mathbb{H}_p, L)$,
3. $\Sigma^2(\mathbb{H}_p, L) \cap \Pi^2(\mathbb{H}_p, L) \neq \Sigma^1(\mathbb{H}_p, L) \cup \Pi^1(\mathbb{H}_p, L)$.

Proof Consider a countable infinite field K of characteristic p. Its additive group $(K, +, -, 0)$ satisfies (1), (2) and (3) as shown respectively in Theorem 4, in Theorem 5 and in Theorem 6. But K contains the field $\mathbb{Z}_p$, so is a vector space

over $\mathbb{Z}_p$. As such, K has a countable vector basis over $\mathbb{Z}_p$, being a countable direct sum of copies of $\mathbb{Z}_p$. It follows that $(K, +, -, 0)$ is isomorphic with $(\mathbb{H}_p, +, -, 0)$. □

7 Infinite Abelian Groups

As most infinite abelian groups have no quantifier elimination in the given language, it is not to expect that the full hierarchy (based on general quantifiers only) does collapse at any level. However, for the cases with quantifier elimination, it is important to see that some complexity classes keep their conjectured difficulty. This will be shown in the next Theorem.

We will transfer some of the properties established above to general infinite abelian groups and to computations with parameters. The next Lemma was proved in [8].

Lemma 5 *Let $(G, +, -, 0, (k_i))$ be an infinite abelian group with finitely many constants interpreted in G. Then there is a group G^* such that:*

1. *$G \leq G^*$ and $(G, +, -, 0, (k_i))$ is elementary equivalent with $(G^*, +, -, 0, (k_i))$.*
2. *G^* contains a subgroup H such that $H \cong \mathbb{Z}$ or $H \cong \mathbb{H}_p$ for some prime number p, and $H \cap G = \{0\}$.*

Here is the main result:

Theorem 7 *Let $(G, +, -, 0, (k_i))$ be an infinite abelian group with finitely many constants interpreted in G. If $L_k = (+, -, 0, (k_i))$, then:*

1. $\Sigma^1_b(G, L_k) \neq \Pi^1_b(G, L_k)$,
2. $\Sigma^2_{bb}(G, L_k) \cap \Pi^2_{bb}(G, L_k) \neq \Sigma^1_b(G, L_k) \cup \Pi^1_b(G, L_k)$.

Proof The results will be proved for (G^*, L_k) and then they will be transferred to (G, L_k) using Theorem 2. We denote with $S(M)$ the set of finite tuples (strings) over a set M. Recall that H is the special subgroup given by Lemma 5. We observe the following facts:

- *Elimination of constants. Let L be again the signature $L = (+, -, 0)$. If $A \in P(G^*, L_k)$ then $A \cap S(H) \in P(H, L)$.*

  Indeed, as $G \cap H = \{0\}$, any test of the form $h + g = 0$ with $h \in H$ and $g \in G$ can be positive only if $h = 0$ and $k = 0$. For every L_k-machine, there is an L-machine that decides over H the same problem as the L_k-machine.
- *It follows that the implication $A \in C(G^*, L_k)$ then $A \cap S(H) \in C(H, L)$ is also true for the following classes C: Σ^1_b, Π^1_b, Σ^2_{bb}, Π^2_{bb}.*

  This is because the boolean non-determinism is only a way of writing quantifier-free formulas in a more compact way, and the result of the formula evaluation depends only on the input, and not on a bigger set where it might be considered.

– *The Null-sack problem* $N(G^*) \in \Sigma_b^1(G^*, L_k) \setminus \Pi_b^1(G^*, L_k)$.
$N(G^*) \in \Sigma_b^1(G^*, L_k)$ is true by definition. If $N(G^*) \in \Pi_b^1(G^*, L_k)$, then by the property proved above, $N(G^*) \cap S(H) \in \Pi_b^1(H, L)$, so $N(H) \in \Pi_b^1(H, L)$. Contradiction with the Corollaries 1 and 2.
– *The combined problem*

$$N\overline{N}(G^*) \in (\Sigma_{bb}^2(G^*, L_k) \cap \Pi_{bb}^2(G^*, L_k)) \setminus (\Sigma_b^1(G^*, L_k) \cup \Pi_b^1(G^*, L_k)).$$

While $N\overline{N}(G^*) \in \Sigma_{bb}^2(G^*, ,_k) \cap \Pi_{bb}^2(G^*, L_k)$ is true by definition and by the fact that the quantifier blocks commute. Along the same lines with the proof that $N \notin \Pi_b^1(G^*, L_k)$ we get that it does not belong to $\Sigma_b^1(G^*, L_k) \cup \Pi_b^1(G^*, L_k)$.
– *The results* $\Sigma_b^1(G^*, L_k) \neq \Pi_b^1(G^*, L_k)$ *and* $\Sigma_{bb}^2(G^*, L_k) \cap \Pi_{bb}^2(G^*, L_k) \neq \Sigma_b^1(G^*, L_k) \cup \Pi_b^1(G^*, L_k)$ *transfer to G by Theorem 2.*

□

Acknowledgments Dedicated to Janos Makowsky to his 75-th. anniversary.

References

1. Blum, L., Shub, M., Smale, S.: On a theory of computation and complexity over the real numbers. Bull. A.M.S. **21** (1989)
2. Blum, L., Cucker, F., Shub, M., Smale, S.: Complexity and Real Computation. Springer-Verlag, New-York (1998)
3. Poizat, B.: Les petits cailloux. Aleas, Lyon (1995)
4. Meer, K.: Real number models under various sets of operations. J. Complexity **9** (1993)
5. Makowsky, J., Meer, K.: P versus NP over arbitrary structures. The ESSLLI-2014 Lectures, Tübingen, Germany, 11–15 August, 2014. https://csaws.cs.technion.ac.il/~janos/COURSES/ESSLLI-2014/
6. Meer, K.: A note on a P $\neq$ NP result for a restricted class of real machines. J. Complexity **8**(4), 451–453 (1992)
7. Prunescu, M.: $P \neq NP$ for all infinite boolean algebras. Math. Logic Q. **49**(2), 210–213 (2003)
8. Prunescu, M.: A model-theoretic proof for $P \neq NP$ over all infinite abelian groups. J. Symb. Logic **67**(1), 235–238 (2002)
9. Prunescu, M.: Structure with fast elimination of quantifiers. J. Symb. Logic **71**(1), 321–328 (2006)
10. Prunescu, M.: Fast elimination of quantifiers means P = NP. In: Beckmann, A., Berger, U., Löwe, B., Tucker, J.V. (eds.) Logical Approaches to Computational Barriers. Lecture Notes in Computer Science. Springer Verlag, vol. 3988, pp. 459–471 (2006)
11. Fournier, H., Koiran, P.: Lower bounds are not easier over the reals: inside PH [Research Report]. LIP RR-1999-21, Laboratoire de l'informatique du parallélisme, Lyon (1999)
12. Prunescu, M.: Two situations with unit-cost: ordered abelian semi-groups and some commutative rings. J. Complexity **21**(4), 579–592 (2005)
13. Koiran, P.: Computing over the reals with addition and order. Theor. Comput. Sci. **133**, 35–47 (1994)
14. Cucker, F., Koiran, P.: Computing over the reals with addition and order: higher complexity classes. J. Complexity **11**, 358–376 (1995)

Data with Logical and Statistical Constraints

Michel de Rougemont

Abstract Descriptive Complexity and Algorithmic Complexity theory both use an analysis in the worst-case. However, hard problems such as SAT become much easier when we relax the worst-case condition. We introduce the notion of statistical queries which take finite structures as inputs and return distributions on finite domains. A statistical constraint is a relation between statistical queries. We use the notion of a stochastic approximation (Mathieu and Rougemont, Network Sci. 9(4):403–424, 2021) for structures which satisfy a statistical constraint and can be generated with a distribution μ. A hard problem is approximable with an algorithm A if A is correct on YES instances with high probability, and on NO instances generated by μ with high probability. We explain how a generalization of Maxclique is easy on graphs which follow a power law degree distribution, even if the graph is given as a stream of edges.

1 Introduction

Logical constraints are sentences in some logic $\mathcal{L}$, mostly First and Second-order Logic. The *Consistency* problem decides if there is a model which satisfies these constraints. The *Entailment* problem decides if a constraint is a logical consequence of a set $\phi_1, ...\phi_k$ of constraints. These problems are central both for Logic and Computer Science, more specifically for *Finite Model theory* and *Data Base theory* which share many central concepts. Janos Makowsky made central contributions at the intersection of these areas, which have led to new fundamental notions, published for example in [6, 7, 13].

The importance of finite structures and the link to Complexity theory are central for the understanding of what *efficiently computable* is. In the classical Complexity theory, the SAT problem has been a fundamental NP-complete problem with natural extensions to optimization problems such as Maxsat, extended versions

M. de Rougemont (✉)
University Paris II, Paris, France
e-mail: mdr@irif.fr

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_27

such as QBF (Quantified Boolean Formula) and counting version such as #SAT. The Descriptive Complexity characterizes all these problems with Logic-based languages. They are considered hard because we do not know polynomial time algorithms for their exact solutions. Logic-based approaches such as [7] identified subclasses for which there exist solutions in polynomial time.

The SAT competition gives a different point of view however. Very efficient heuristics have solved these problems for larger and larger instances over the years. In [3], a recent history of these heuristics is presented. In general, a SAT instance is first analyzed with statistical methods and the space of inputs is then partitioned into several areas. Different heuristics are used for each area and provide efficient solutions in practice.

The notion of *effective computability* therefore needs to be better understood. An $O(n)$-time algorithm must be linear for all inputs of size n, for both Computational and Descriptive Complexity which both use the *worst-case complexity*. It is however possible that an algorithm A would require $O(2^n)$ time on a few inputs, while all other inputs require only $O(n)$-time. How easy are these inputs to generate? For SAT, there are hard instances, but they are either randomly generated or hard to generate with a deterministic algorithm. There have been at least three possible approaches for a better understanding of efficient algorithms when problems are hard in the worst-case:

- Find classes of inputs for which hard problems become easy. Bounded Clique-width for graphs is an example which introduces a new parameter k, as shown by Janos Makowsky and his co-authors in [7],
- Use Approximations, on inputs and outputs,
- Relax the Worst-case Complexity.

This paper presents an approach based on a *statistical property* which relaxes the Worst-case Complexity. We show how some NP-hard problems can become easy on finite structures which satisfy such a property, using the notion of a *1-sided stochastic randomized* approximation algorithm A of a decision problem, defined in [15]. We consider random inputs of size n which follow some statistical property, generated by a distribution μ. On Yes instances, the algorithm accepts with high probability, and on NO instances generated by μ, the algorithm rejects with high probability (typically $2/3$ or $1 - \delta$). We only guarantee a correct answer on NO instances when the input follows the statistics and the probabilistic space is the product of $\mu \times \Omega$ where Ω is the probabilistic space of the algorithm.

As an example, we take graphs whose degree distribution follows a power law (a specific statistical constraint) and we explain how a generalization of Maxclique can become easy when the graph is given as a stream of edges. We survey this approach, based on statistical constraints, which restricts the class of inputs and take Words, Graphs and relational databases as a source of structures for which there are natural statistical constraints.

In the second section, we review some classical results concerning logical constraints, classical approximations for search and decision problems and average complexity. In the third section, we present the statistical queries and constraints,

for classes of Words, Graphs and Datawarehouses, i.e. relational structures used for data analysis. In the fourth section, the notion of a stochastic approximation for structures which satisfy some statistical constraint is presented and we discuss the case of graphs which follow a power law degree distribution.

2 Classical Approaches

We review three different approaches to understand the complexity of search and decision problems and the existence of efficient algorithms.

2.1 Logical Constraints

The importance of finite structures for the *Entailment* problem was stressed in [6]. It was shown that for some class of constraints called *Embedded Implicational Depencies*, the *Entailment* problem on finite structures is co-recursively enumerable complete. It is one of first results on *Finite Model theory*, which has become a mainstream subject.

The study of density functions of graph properties definable in Monadic Second Order Logic started with the work of Blatter and Specker [4]. This rich subject on MSO definable graph properties was extended in [13]. The case of relations of arity 4, studied in[8], shows a very different situation, relevant for database theory where relations may have a large arity.

If MSO-definable properties on graphs can be NP-complete, hence hard, which additional constraint implies a feasible solution? In [7], Bruno Courcelle, Janos Makowsky and Udi Rotics propose the bounded Clique-width property and proved that MSO properties on graphs with bounded Clique-width have a linear time solution. This influential result has started an entire new research area.

Other graph properties such as bounded treewidth have similar properties and lead to study the time complexity of an algorithm as a function of n and other graph parameters, hence *parametrized complexity*. In particular, when the time complexity is polynomial in n and exponential in the graph parameters [17].

2.2 Approximation

A search problem returns a numerical value and defines a function f such that on input x, we have $f(x) = y$. A randomized algorithm A (ε, δ)-approximates the search problem if $\mathbb{P}rob[\mid A(x) - f(x) \mid < \varepsilon] > 1 - \delta$ for an additive error and $\mathbb{P}rob[\mid A(x) - f(x) \mid < \varepsilon.f(x)] > 1 - \delta$ for a multiplicative error.

For a decision problem Q, this definition is inadequate and we need to shift the approximation on the input. Assume a distance dist on the inputs x, such as the Edit distance on Words, Trees and Graphs, and a model where we query the input: given a predicate P of arity k and arguments $a_1, ...a_k$, we ask if $P(a_1, ...a_k)$ is true or not. We extend the distance to a property Q as $\mathsf{dist}(x, Q) = Min_{x' \in Q}\mathsf{dist}(x, x')$ and say that x is ε-far from Q if $\mathsf{dist}(x, Q) \geq \varepsilon$. An (ε, δ)-Tester is a randomized algorithm A such that:

- If $x \in Q$ then A accepts with probability 1,
- If x is ε-far from Q then $I\!P rob[A(x)\text{rejects}] > 1 - \delta$.
- The query complexity is independent of n, the size of x, and depends only on ε and δ.

This definition is 1-sided, but can be generalized to a 2-sided version. The query complexity can also be extended to some sublinear function of n. Both definitions assume a worst-case situation.

2.3 Average-case Complexity

The natural approach, originated by Levin [14], considered a distribution $\mathcal{D}$ on the inputs and required that the expected time complexity on $\mathcal{D}$ be bounded by a function of n. Given a reduction between *Distributional problems*, problems with a distribution $\mathcal{D}$, [14] presented a complete problem for polynomial time computable distributions. More advanced results are presented in [5].

On the algorithmic side, the *non worst-case analysis* [19] presents the analysis of algorithms on specific distributions. In Sect. 4, we consider inputs which satisfy statistical constraints and a distribution μ on these inputs. We then consider a probabilistic space which is a product of $\mu \times \Omega$ where Ω is the probabilistic space of the algorithm. The algorithm is correct only on inputs generated by μ.

3 Statistical Constraints

A statistical query generalizes the notion of a *query* on a class $\mathcal{K}$ of finite structures, as a function which takes a finite structure $U_n \in \mathcal{K}$, whose domain is of size n, as input and returns a relation on U_n of arity r. A *statistical query* takes a finite structure $U_n \in \mathcal{K}$ as input and returns a multivariate distribution δ on U_n of arity $r \geq 1$. A *statistical constraint* is a relation between distributions, for example the equality or the proximity $\mathsf{dist}(\delta, \delta') \leq \varepsilon$ relation for some dist function between distributions, and is similar to a boolean query. On relational structures, statistical queries are often called *OLAP* (OnLine Analytical Processing) queries and defined in SQL with the *GROUP BY* expression.

In general, we take a class $\mathcal{K}$ of finite structures of size n augmented with the set Q of rationals with the basic arithmetical operations as parameters. We construct terms and formulas to define the statistical queries. The use of specific distances between distributions is a central element of this approach. The difference with the Average-case complexity is that the distribution of inputs concerns inputs of size n, which satisfy some statistical constraints and not arbitrary distributions $\mathcal{D}$ on the inputs. In Sect. 4, we introduce the *1-sided-stochastic approximation*, similar to the approximation in Property Testing.

Let $\mathcal{L}$ be a logic on a class $\mathcal{K}$ of finite structures with several domains. Consider a First-order Σ_1 formula $\psi(x_1)$ with the free variable $x_1 \in D$ and some existential quantifier. We may have several domains, we write:

$$\psi(x_1 \in D) : \exists y_1 \in D' \ \varphi(x_1, y_1)$$

to specify that x_1 ranges over D and y_1 ranges over $D' = \{1, 2...n\}$. We say that $\psi(x_1)$ is *separating* on a relational structure $U = (D, D', R_1, ...R_k)$ if the sets $W_a = \{b : \varphi(a, b)\}$ are disjoint for each $a \in D$. Observe that if there is a functional dependency $y_1 \rightarrow x_1$, then $\psi(x_1)$ is always *separating*.

The *counting formula* $\psi_c(x_1)$ defines the function which associates to each $a \in D$ the number of distinct values of y_1, i.e. $|W_a|$ and we write:

$$\psi_c(x_1) : \#y_1 \ \varphi(x_1, y_1)$$

We can also write $\psi_c(a) =\mid \{b : \varphi(a, b)\} \mid$.

When the formula $\psi(x_1)$ is separating for the domain D of cardinality m, we can introduce the *distribution formula* $\psi_d(x_1)$ which defines the distribution of values for $x_1 = a_1, .., a_m$:

$$\psi_d(x_1) : \%y_1 \ \varphi(x_1, y_1)$$

We define $\psi_d(a) = \frac{\psi_c(a)}{\sum_a \psi_c(a)} = \frac{\psi_c(a)}{n}$ where n is the size of D', the domain of y_1, assuming the dependency $y_1 \rightarrow x_1$. In this case, $\sum_a \psi_d(a) = 1$ and $\psi_d(x_1)$ is a 1-dimension distribution. In general we may have $\psi_d(x_1, x_2, ...x_d)$ for a distribution of dimension d. On Words, Graphs and Datawarehouses (specific relational structures), there are functional dependencies which guarantee that the formulas are separating and therefore define statistical queries. On Datawarehouses (see Sect. 3.3), if the free variables $x_1, x_2, ...x_d$ of the query are *analysis variables*, then the distributions are well defined because

$$y_1 \rightarrow x_1, x_2, ...x_d$$

Given a distance dist between 2 distributions, a *Statistical constraint* is a relation between two distributions δ_1, δ_2, either equality $\delta_1 = \delta_2$ or $\mathsf{dist}(\delta_1, \delta_2) < \varepsilon$ for a specific distance dist between distributions.

3.1 Words

A binary word $w_n \in \{0,1\}^n$ is classically represented by the finite structure:

$$W_n = (\{1, 2...n\}, P_0, P_1, <)$$

where the domain is the set $\{1, 2...n\}$ ordered with the binary predicate $<$, and unary predicates P_i for $i = 0, 1$ such that $P_i(j)$ is true if $w_n[j] = i$. On a large alphabet $\Sigma_m = \{a_1, a_2...a_m\}$ where each a_i is a symbol, it is more convenient to consider a word w_n of length n as a finite structure with two domains such as:

$$U_n = (\{1, 2...n\}, \Sigma_m, P, <)$$

where the binary relation $P \subseteq \Sigma_m \times \{1, 2...n\}$ is defined by $P(i, j)$ is true if $w_n[j] = a_i$, i.e. the letter a_i appears in position j.

A statistical query gives, for example, the distribution of the occurrences of each letter, where the occurrence function $\mathsf{occ} : \Sigma_m \to \{1, 2...n\}$ is such that for each letter $a_i \in \Sigma_m$, $\mathsf{occ}(a_i) = \#a_i$ where $\#a_i$ is the number of occurrences of the letter a_i. The frequency function $f : \{1, 2...n\} \to \{0, 1, 2...n\}$ gives the *ordered* occurrences, i.e. $f(i+1) \leq f(i)$ holds for $i = 1, ..n-1$. So $f(1) = \#a_i$ is the number of occurrences of the most frequent letter a_i and $f(n) = \#a_j$ is the number of occurrences of the least frequent letter. The relative occurrence occ' is defined as $\mathsf{occ}'(a_i) = \mathsf{occ}(a_i)/n$ and the relative frequency f' is defined as $f'(i) = f(i)/n$. Both functions take values on the rationals $\mathbb{Q}$. Consider the two words *aaabb* and *bbbaa*: they have the same frequencies f but different occurrence functions occ.

Both occ' and f' are distributions as $\sum_i \mathsf{occ}'(i) = \sum_i f'(i) = 1$. The typical application is when the size m of the alphabet is large and n is very large. Typically, Large language Models read the whole internet, $m \simeq 3.10^4$ is the number of *tokens*, the basic elements defined by the Byte-Pair Encoding algorithm [9], and $n \simeq 10^{12}$. A strict statistical constraint states that occ' is for example the uniform distribution, i.e.

$$\forall i \;\; \mathsf{occ}(a_i) = n/m$$

when m divides n. In general, we accept rounding errors and a statistical constraint is the property:

$$\forall i \;\; \mathsf{occ}'(i) \simeq 1/m$$

where $\simeq$ is the classical approximation on $\mathbb{Q}$. Hence the class of structures is:

$$V_n = (\{1, 2...n\}, \Sigma_m, P, <; \mathbb{Q}, +, *, /)$$

as the set of rational numbers $\mathbb{Q}$ and the arithmetical functions $+, *, /$ can be used to define basic terms. Another statistical constraint would apply on f', for example f' is a Zipf distribution:

$$\forall i \quad f'(i) \simeq c/i^2$$

where c is a constant which depends on n, the size of the structures. The term c/i^2 takes values in $\mathbb{Q}$.

A k-gram is the generalization of occ' to factors of length k, i.e. consecutive letters. We write $\mathsf{ustat}_k(w_n)$ as the function from $(\Sigma_m)^k$ into $[0, 1]$. It gives the probability to observe a k-factor in a word of length n when sampling a uniform position $1 \leq i \leq n - k + 1$, hence the name *uniform statistics*.

$$\mathsf{ustat}_k(w_n) = \frac{1}{n-k+1}. \begin{pmatrix} \#w_1 \\ \#w_2 \\ \ldots \\ \#w_p \end{pmatrix}$$

where w_i is the i-th k-factor ordered lexicographically. As an example:

$$\mathsf{ustat}_2(aaabb) = \frac{1}{4}. \begin{pmatrix} 2 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \frac{1}{4}. \begin{pmatrix} \#aa = 2 \\ \#ab = 1 \\ \#ba = 0 \\ \#bb = 1 \end{pmatrix}.$$

A k-gram defines the next distribution: given a $(k-1)$-factor what is the distribution of the next letter? For example, for $w_5 = aaabb$, $k = 2$ and the factor a, the distribution of the next letter is:

$$\mathsf{next}_a(aaabb) = \frac{1}{3}. \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

There are 3 factors of length 2 which start with an a: two of them have an a as the next letter letter and one of them has a b as the next letter. Similarly, if the factor starts with a b, the distribution is:

$$\mathsf{next}_b(aaabb) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

We can then consider $\mathsf{ustat}_2(w_n)[x_1, x_2]$ where $x_1, x_2 \in \Sigma$ as the probability that a uniform random factor u of size 2 is such that $u_1 = x_1 \wedge u_2 = x_2$. We write:

$$\mathsf{ustat}_2[x_1, x_2] = \mathbb{P}rob_u[u_1 = x_1 \wedge u_2 = x_2].$$

We then use the notation:

$$\mathsf{ustat}_2[x_2 \mid x_1 = a] = I\!Prob_u[u_2 = x_2 \mid u_1 = x_1 = a] = \mathsf{next}_a.$$

We use conditional probabilities as projections. Similarly, for a distribution of arity r with variables $x_1 ... x_r$. Consider the formula:

$$\psi_2(x_1, x_2 \in \Sigma_m) : \exists y_1, y_2 \in \{1, 2...n\} \;\; P(x_1, y_1) \wedge y_2 = y_1 + 1 \wedge P(x_2, y_2)$$

The formula $\psi(x_1)$ is separating for the domain Σ_m, because of the functional dependency $y_1, y_2 \rightarrow x_1, x_2$: there is only one letter on each position. We can replace the existential quantifier $\exists y_1, y_2$ by the counting $\#y_1$ and the distribution $\%y_1$ quantifiers and obtain the formulas $\psi_c(x_1, x_2)$ and $\psi_d(x_1, x_2)$. It defines precisely the distribution $\mathsf{ustat}_2[x_1, x_2]$.

If the size m of the alphabet is large, the number of potential k-factors is m^k, i.e. exponential in k. The distribution next is sparse in general but its space representation is too large, although it can be compressed. A neural network or a transformer constructs a compressed version [21] of a distribution next', related to next, of size $(p \cdot k)^2 \cdot d$ which approximates this distribution, where p is the size of the embedding of the tokens and d the depth of the circuit. We have d layers where each layer is a matrix of size $p \cdot k$. A node at layer i applies a non linear function to a linear combination of the values of the nodes at the previous layer $i - 1$, for $i > 1$. This distribution is central for Large Language Models and generative A.I. in general.

3.2 Graphs

A finite graph is a structure $G_n = (\{1, 2...n\}, E)$, where $E \subseteq \{1, 2...n\} \times \{1, 2...n\}$ is the Edge relation. As in the previous case, we extend the structure with the rationals and obtain:

$$G_n = (\{1, 2...n\}, E, ; \mathbb{Q}, +, *, /)$$

Let $d(x, i)$ the relation expressing that a node x has i distinct neighbors. In this case, we have the functional dependency $x \rightarrow i$ and we can define the natural degree distributions. The degree function degree defines the number of nodes of a degree i, i.e. for $i, j \in \{1, 2...n\}$, $\mathsf{degree}(i) = j$ if exactly j nodes have degree i. The degree distribution is:

$$\mathsf{degree}_d(i) = \mathsf{degree}(i)/n$$

which gives the probability that a random uniform node has degree i.

We can use the Logic FO+ C (Counting). We extend the first-order quantifiers with new quantifiers $\exists^i, \exists^{>i}, \exists^{<i}$, which quantify the existence of exactly i, more than i, less than i witnesses. Define:

$$d(x,i) : \exists^i y \; E(x,y)$$

$$degree(i) : \#x \; d(x,i)$$

$$degree_d(i) : \%x \; d(x,i)$$

This last formula formula defines the degree distribution of the graph because of the functional dependency $x \rightarrow i$. One can show that we precisely need this extension of First order Logic. An important statistical constraint is that the degree' distribution follows a Zipf law of parameter $\alpha > 1$:

$$\mathsf{degree}_d(i) \simeq c/i^\alpha$$

The $\simeq$ symbol refers to some distance, between two distributions, discussed in Sect. 3.4.

This last formula formula gives the degree distribution of the graph because if the functional dependency $x \rightarrow i$. If we express that this degree distribution is ε-close to a Zipf distribution for the Fréchet distance introduced in Sect. 3.4, we write:

$$\%x \;\; d(x,i) \;\simeq_{F,\varepsilon} \;\; c/i^2 \;\text{ for }\; \mathsf{dist}_F(\%x \;\; d(x,i), c/i^2) < \varepsilon$$

In Sect. 4, we consider the constraint $\%x \;\; d(x,i) = \;\; c/i^2$, i.e the graphs follow a power law degree distribution of parameter $\alpha = 2$.

3.3 Relational Databases and Datawarehouses

Consider the two tables of Figs. 1 and 2. The *Product* relation lists different products with a key PID, and the *Buy* relation lists the Sales of the products with a date and a price. The *Buy* relation is often called a *Datawarehouse*, as it may grow to a very large table and there are functional dependencies between the attributes of the *Buy* relation and some attributes of other tables, called the *analysis attributes*. An OLAP schema defines all the functional dependencies between the attributes of the Datawarehouse relation and the analysis attributes. In this example: $PID, DATE, PRICE \rightarrow TYPE, AGE$ and the two attributes $TYPE, AGE$ are analysis variables.

It is natural to analyze the number of Sales by TYPE, as a unary distribution Q_1^d, or the number of Sales by TYPE and AGE as a binary distribution Q_2^d. In both cases, we count the number of tuples of the relation *Buy* for different values of the analysis variables, called *dimensions* in the OLAP terminology. It is also natural to analyze

Fig. 1 Table product

	PID	TYPE	AGE
Product	P1	A	O
	P2	A	N
	P3	B	O
	P4	C	N

Fig. 2 Table buy

	PID	DATE	PRICE
Buy	P1	Jan 1.	15
	P2	Jan 2.	30
	P1	Jan 2.	20
	P3	Jan 3.	45
	P4	Jan 5.	30
	P1	Jan 6.	10

the global Sales, i.e the Sum of the PRICE attribute with the same dimensions. They correspond to the Count or Sum Aggregation operators in SQL, followed by the GROUP BY construction. We can define these statistical queries with simple First Order Formulas in the relational language $Product(x, t, a)$, $Buy(x, y, z)$.

- Q_1^d: number of sales per TYPE.

$$Q_1^d(t) : \%(x, y, z)\exists a \;\; Product(x, t, a) \wedge Buy(x, y, z)$$

- Q_2^d number of sales per TYPE and AGE.

$$Q_1^d(t, a) : \%(x, y, z) \;\; Product(x, t, a) \wedge Buy(x, y, z)$$

Both distributions are well defined because $x, y, z \rightarrow t, a$. For the sum of Sales, we sum on the PRICE attribute and write for the first query:

$$Q_1^d(t) : \%Sum(x, y, z).z \; \exists a \;\; Product(x, t, a) \wedge Buy(x, y, z)$$

A possible Statistical constraint is to fix one of these distributions. For example, the number of Sales by AGE is close to: δ_0: (O: 2/3, N: 1/3). It is true for the instance of Figs. 1 and 2. If the distributions are close for the L_1 distance, we write:

$$\%(x, y, z)\exists t \;\; Products(x, t, a) \wedge Buy(x, y, z) \simeq_{1,\varepsilon} (O : 2/3, N : 1/3)$$

Statistical constraints are similar to the distributions introduced in the Probablistic Relational models of [12]. We now introduce in Sect. 3.4 a central notion: the distance between distributions.

3.4 Distances Between Distributions

Given two distributions δ_1, δ_2 on the same domain, there are many possible distances. Classical distances include L_p distances, EMD (Earth-Moving Distance), Fréchet and other pseudo distances such as KL (Kullback-Liebler).

The L_1 distance, also called the *variational distance* between two distributions δ_1, δ_2 on a domain with n elements is defined as:

$$\mathsf{dist}_1(\delta_1, \delta_2) = \frac{1}{2} \cdot \sum_i \mid \delta_1(i) - \delta_2(i) \mid .$$

If we relabel the domain with a permutation π we may have a smaller variation $\sum_i \mid \delta_1(i) - \delta_2(\pi(i)) \mid$ and [10] introduces the *Variation distance up to relabeling* $\mathsf{VDR}(\delta_1, \delta_2)$[1] as the minimum over π of $\frac{1}{2} \cdot \sum_i \mid \delta_1(i) - \delta_2(\pi(i)) \mid$. It is also the L_1 distance between the frequencies of the distributions, i.e. ordered by decreasing values. Notice, that the distribution of the frequencies is invariant by relabeling, hence used for the definition of statistical constraints.

The *Fréchet* distance considers the distributions as points x, y in two dimensions and defines the *Fréchet distance* as the minimum d such that for each point (x, y) of δ_i there is a point of δ_j at an Euclidian distance less than d. For the *relative Fréchet* distance, consider an extended Box $(x \cdot (1 \pm \varepsilon_1), y \cdot (1 \pm \varepsilon_2))$ associated with each point x, y, as in the Fig. 3. The *relative Fréchet distance* distance dist_F is the minimum $\varepsilon_1, \varepsilon_2$ such that for each point (x, y) of δ_i there is a point of δ_j in the extended Box $(x \cdot (1 \pm \varepsilon_1), y \cdot (1 \pm \varepsilon_2))$.

We concentrate on two distances L_1 and relative Fréchet. Consider a stream of Graph edges and two inputs determined by the stream of edges up to times t_1 for G_1, and between time t_1 and $t_2 > t_1$ for G_2 (assume $t_2 \simeq 2.t_1$) with approximately the same number of nodes. The two degree distributions δ_1 and δ_2 represented in Fig. 3 seem *close* to a power law.

The L_1 distance between the degree distributions δ_1 and δ_2 of Fig. 3 is however not small because for large degrees i, one of the two distributions has a value 0 depending on i. In this example, the two distributions are close for the relative Fréchet distance because the points x, y of each distributions are relatively close for both x and y, and are also close to a power law. In practice, a statistical law assumes the data is only close to a predefined distribution, as in this example.

The L_1 distance is useful for distributions with small supports and the *relative Fréchet* distance is adapted for large supports of size $O(n)$. These distances are completely different from the Edit distance on the data. Indeed structures of very different sizes are far for the Edit distance but could be close for L_1 and *relative Fréchet*.

[1] For a distribution δ, let the histogram h_δ of δ be the function $[0, 1] \rightarrow N$ such that $h_\delta(x) = |\{i : \delta(i) = x\}|$, the number ef elements with probability x. Then [10] shows the connection betwen VDR and the Earth-Moving Distance EMD of the histograms: $\mathsf{VDR}(\delta_1, \delta_2) = \mathsf{EMD}(h_{\delta_1}, h_{\delta_2})/2$.

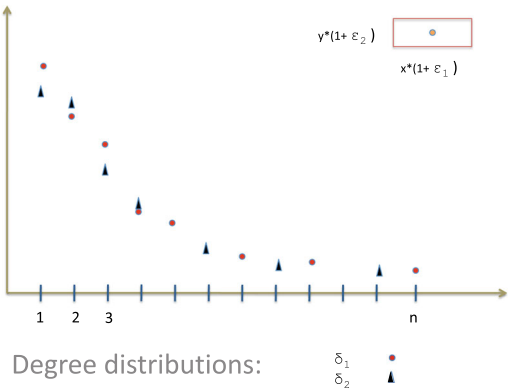


Fig. 3 Degree distributions of G_1 and G_2 on the same stream of graph edges

4 Stochastic Approximation

In the classical setting, discussed in Sect. 2.2, we approximate an optimization problem such as Maxclique with a possible (ϵ, δ) randomized algorithm A which returns on an input x of size n an integer value in $\{1, 2...n\}$ such that:

$$\forall x \mid x \mid > c \rightarrow \mathbb{P}rob_{\Omega}[\ \mid A(x) - \mathsf{Maxclique}(x) \mid \leq \varepsilon] \geq 1 - \delta$$

In this definition, the randomized algorithm A has a probabilistic space Ω and guarantees a good answer in the worst-case for large enough inputs x, *i.e.* $\mid x \mid > c$. We want to relax this last condition and only consider random inputs which satisfy some statistical property. Assume a distribution μ over structures which satisfy some statistical property and a decision problem where the answer is Yes or No.

1-sided-stochastic Approximation For Yes instances we consider the worst-case, but for No instances we only consider random inputs for μ. A *δ-1-sided stochastic randomized* algorithm A for a language L satisfies the following two conditions:

- For all YES instances x, $Prob_{\Omega}[A(x)\ accepts] \geq 1 - \delta$
- For NO instances x drawn from μ, $Prob_{\mu \times \Omega}[A(x)\ rejects] \geq 1 - \delta$

where Ω is the set of possible choices of the algorithm.

2-sided-stochastic Approximation A *δ-2-sided stochastic randomized* algorithm A for a language L satisfies the following two conditions:

- For YES instances x drawn from μ, $Prob_{\mu \times \Omega}[A(x)\ accepts] \geq 1 - \delta$
- For NO instances x drawn from μ, $Prob_{\mu \times \Omega}[A(x)\ rejects] \geq 1 - \delta$

Both definitions depend on μ and we assume that μ is uniform unless it is explicitely specified.

4.1 Large Dense Subgraphs of Graphs Which Follow a Power Law Degree Distribution

Consider a stream of Graph edges (v_i, v_j) and we want to decide if there is a large dense subgraph in the underlying graph G. The approximation of dense subgraphs is well studied in [2] and an $\Omega(n)$ space lower bound is known [1]. A classical density is the ratio $\rho = |E[S]|/|S|$ but we want a much higher density $\gamma = 2.|E[S]|/|S|(|S| - 1)$, hence the expression *very dense*. If $\gamma = 1$, we have a clique, and in practice we look for clusters where $\gamma < 1$.

Definition 1 The (γ, δ)-*large very dense subgraph* problem, where the parameters $\gamma, \delta \leq 1$, takes as input a graph $G = (V, E)$ and decides whether there exists an induced subgraph $S \subseteq V$ such that $|S| > \delta\sqrt{n}$ and $|E[S]| > \gamma|S|(|S| - 1)/2$.

A *very dense subgraph* is also called a γ-clique, as the density is greater than γ. The parameter δ concerns the size of the cluster. The (γ, δ)-*large very dense subgraph* problem is NP-hard and hard to approximate [11], as it contains the maximum clique problem as the special case when $\gamma = 1$. This leads us to use a new notion of approximation, adapted to a specific distribution of inputs. Social graphs define a specific regime where graphs approximately follow a power law degree distribution, precisely the statistical constraint considered in Sect. 3.2.

We proposed in [15] a streaming algorithm which uses $O(\sqrt{n}.\log n)$ space, reads one edge each time and approximates this hard problem on graphs which follow a power law degree distribution. The distribution μ is the uniform distribution on graphs which satisfy this statistical constraint, for each size n.

Theorem 1 *There is a δ-1-sided stochastic randomized streaming algorithm A which uses $O(\sqrt{n}.\log n)$ space for the (γ, δ)-large very dense subgraph problem, on inputs which follow a power law degree distribution where the parameters $\gamma, \delta \leq 1$, takes as input a graph $G = (V_n, E)$ and decides whether there exists an induced subgraph $S \subseteq V$ such that $|S| > \delta\sqrt{n}$ and $|E[S]| > \gamma|S|(|S| - 1)/2$.*

The algorithm uses a Reservoir sampling [20] and the analysis relies on the existence of giant components for random graphs generated in a 2-stage process: we first take the configuration model of random graphs which follow a power law degree distribution and then consider an Erdös-Renyi model where edges are uniformly sampled. We then use the Molloy-Reed [18] analysis for the existence of giant components in the Reservoir. If there is a large Connected component in the Reservoir of size $O(\sqrt{n}.\log n)$, the algorithm A accepts, else it rejects.

In this approach, it is important to efficiently decide if some data follows a statistical property. In [16], we give sufficient conditions on the frequency distributions of a streams of elements taken from a set $\{e_1, ...e_n\}$, so that the frequency distributon can be tested Online in space $O(poly(\log n))$, in the sense of a property Tester, using the relative Fréchet distance between distributions.

5 Conclusion

Data have a logical structure with many dependencies which are often expressed in First and Second-order Logic. They also have statistical properties, as defined in this paper by simple relations between statistical queries. Both Logical and Statistical constraints are useful to analyze the data and predict their evolution, but the techniques used are quite different.

We gave the example of graphs which follow a power law degree distribution, as a statistical property. This statistical property is definable in FO+ Counting. On these graphs, a generalization of Maxclique, definable in MSO, becomes easy with a δ-1-sided stochastic approximation. In this framework, both logical and statistical constraints can be combined to solve hard problems in the worst case.

Acknowledgments I thank Richard Lassaigne for fruitful discussions on the notions of statistical constraints.

References

1. Bahmani, B., Kumar, R., Vassilvitskii, S.: Densest subgraph in streaming and mapreduce. Proc. VLDB Endow. **5**(5), 454–465 (2012)
2. Bhattacharya, S., Henzinger, M., Nanongkai, D., Tsourakakis, C.E.: Space- and time-efficient algorithm for maintaining dense subgraphs on one-pass dynamic streams (2015). CoRR, abs/1504.02268
3. Biere, A., Fleury, M., Froleyks, N., Marijn Heule, J.H.: The sat museum. In: Proceedings of the 14th International Workshop on Pragmatics of SAT (SAT 2003). CEUR Workshop Proceedings, vol. 3545, pp. 72–87 (2023)
4. Blatter, C., Specker, E.: Modular periodicity of combinatorial sequences. Abstracts Am. Math. Soc. **4** (1983)
5. Bogdanov, A., Trevisan, L.: Average-case complexity (2021). arXiv:cs/0606037
6. Chandra, A.K., Lewis, H.R., Makowsky, J.A.: Embedded implicational dependencies and their inference problem. In: Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing, STOC, pp. 342–354. Association for Computing Machinery, New York (1981)
7. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory Comput. Syst. **9**(4), 125–150 (2000)
8. Fischer, E.: The Specker-Blatter theorem does not hold for quaternary relations. J. Comb. Theory Series A **103**(1), 121–136 (2003)
9. Gage, P.: A new algorithm for data compression. C Users J. **12**(2), 23–38 (1994)
10. Goldreich, O., Ron, D.: On the Relation Between the Relative Earth Mover Distance and the Variation Distance (an Exposition), pp. 141–151. Springer International Publishing, Berlin (2020)
11. Hastad, J.: Clique is hard to approximate within $n^{1-\epsilon}$. In: Proceedings of the 37th Annual Symposium on Foundations of Computer Science, FOCS '96, p. 627. IEEE Computer Society (1996)
12. Koller, D.: Probabilistic relational models. In: Džeroski, S., Flach, P. (eds.) Inductive Logic Programming, pp. 3–13. Springer, Berlin/Heidelberg (1999)
13. Kotek, T., Makowsky, J.A.: Definability of Combinatorial Functions and Their Linear Recurrence Relations, pp. 444–462. Springer, Berlin/Heidelberg (2010)

14. Levin, L.: Average case complete problems. SIAM J. Comput. **15**(1), 285–286 (1986)
15. Mathieu, C., de Rougemont, M.: Large very dense subgraphs in a stream of edges. Network Sci. **9**(4), 403–424 (2021)
16. Mathieu, C., de Rougemont, M.: Testing frequency distributions in a stream. arXiv, 2309.11175 (2023)
17. Meeks, K., Scott, A.: The parameterised complexity of list problems on graphs of bounded treewidth. Inf. Comput. **251**, 91–103 (2016)
18. Molloy, M., Reed, B.: The size of the giant component of a random graph with a given degree sequence. Comb. Probab. Comput. **7**(3), 295–305 (1998)
19. Roughgarden, T.: Beyond the Worst-Case Analysis of Algorithms. Cambridge University Press, Cambridge (2021)
20. Vitter, J.S.: Random sampling with a reservoir. ACM Trans. Math. Softw. **11**(1), 37–57 (1985)
21. Yang, Y., Mandt, S., Theis, L.: An introduction to neural data compression. Found. Trends Comput. Graph. Vision **15**(2), 113–200 (2023)

Relating Information and Knowledge

Anatol Slissenko

To Janos with my gratitude for sharing with me his vision of research and not just

Abstract The basic notion of information in mathematics is the entropy of a random variable, and the value of entropy is a number. Individual pieces of information are not analyzed explicitly, and the theory works well as it is. In colloquial speech "information" is usually a statement, a piece of information. Such a statement, that is mathematically a logical formula, should be justified by a proof, that constitutes the knowledge we refer to. The formula with the proof represents, in a way, the information received. How to measure its quantity? The paper discusses this question for a simple model of such knowledge. The approach uses an entropy-like function called entropic weight. Entropic weight, contrary to the classical entropy, is monotone and corresponds to our intuition in the context under treatment. The paper is mainly a conceptual discussion.

1 Introduction

One can see that the meaning of the word "information" in colloquial speech and in mathematics differs. Information in colloquial speech is usually a statement like "X is a winner of a competition C", and in mathematics information is, first of all (e.g., [2], Chapter 2), a number based on evaluation of chances, more precisely it is the entropy, a measure of uncertainty based on a probabilistic measure. It is a measure of *quantity* of information without naming the pieces of information involved. Such an approach gives a productive theory that works well.

A. Slissenko (✉)
Laboratory for Algorithmics, Complexity and Logic (LACL), University Paris East Créteil (UPEC), Créteil, France
e-mail: slissenko@u-pec.fr

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*, Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_28

This discrepancy is well known and was a subject of discussions by philosophers, see e.g., [1, 3] where one can find further references.

In this paper a piece of information is viewed as a formula together with its proof in a simplistic model of knowledge represented as a set of proofs. The quantity of information is evaluated on the basis of a probabilistic measure over proofs with the help of a function called *entropic weight* that has a flavor of entropy. However, it is different from the classical entropy, it is monotone, and corresponds to our intuition in the context under discussion.

2 Motivational Example

Three men named *Bok*, *Dok*, *Fok* participate in a competition where there is only one winner. The winner is announced by different sources at the same time (with this assumption we avoid mentioning the time moments).

The classical view on the quantity of information received by different persons X, Y and Z may look as follows. There are three (in our example, different) probabilistic spaces representing the worlds of X, Y and Z.

Someone called X estimates that

Bok wins with probability $\frac{1}{4}$,
Dok wins with probability $\frac{1}{4}$,
Fok wins with probability $\frac{1}{2}$.

Someone called Y estimates that

Bok wins with probability $\frac{1}{8}$,
Dok wins with probability $\frac{7}{16}$,
Fok wins with probability $\frac{7}{16}$.

Someone called Z estimates the chances of all outcomes as equiprobable.

In mathematics the information of X about the winner is
$-(\frac{1}{4}\log\frac{1}{4} + \frac{1}{4}\log\frac{1}{4} + \frac{1}{2}\log\frac{1}{2}) = \frac{3}{2} = 1.5$,
the information of Y about the winner is
$-(\frac{1}{8}\log\frac{1}{8} + \frac{7}{16}\log\frac{7}{16} + \frac{7}{16}\log\frac{7}{16}) \approx 1.42$
the information of Z about the winner is
$-(3 \cdot \frac{1}{3}\log\frac{1}{3}) = \log 3 \approx 1.58$

Do the numbers 1.5, 1.4 and 1.58 give the information about the winner we are interested in?—No. They evaluate the uncertainty of determining the winner in the vision (or model) of particular individuals, and these uncertainties are different.

Suppose that *Bok* is the winner. Suppose that *in some way* this information was received by X, Y and Z. It is the same for all of them.

What is "Bok is the winner" mathematically? It is a logical formula (that is, clearly, not a number).

"Bok is the winner" is of value if it comes with a *proof* that *Bok* is the winner. Such a proof may be of the following kind.

"Radio station ***R*** always gives a truthful information about competitions. It broadcasted that *Bok* is the winner of the competition".

Look at it more formally.

2.1 Inference System and Proofs for the Example

Constants

$\mathcal{P} = \{Bok, Dok, Fok\}$ is a set of participants. $\mathcal{S} = \{R_1, R_2, R_3\}$ is a finite set of information sources (some of them are always truthful, others are always deceitful or may be sometimes truthful, sometimes deceitful.) Fri is the name of a particular day of week, namely $Friday$.

Predicates and Functions

Day is the name of the day we speak about.
$Win(\alpha)$, where $\alpha \in \mathcal{P}$, says that α is the winner.
$Brd(R, \alpha)$ says that $R \in \mathcal{S}$ broadcasts that $\alpha \in \mathcal{P}$ is the winner.

Axioms

If $R \in \mathcal{S}$ is truthful and R broadcasts Φ then Φ.
If R is deceitful and R broadcasts Φ then $\neg\Phi$.
Source R_1 is always truthful.
Source R_2 is truthful on Fridays and deceitful on other days.
Source R_3 is always deceitful.
There is always a winner : $(Win(Bok) \vee Win(Dok) \vee Win(Fok))$.
There is at most one winner : $(Win(\alpha) \rightarrow \neg Win(\beta))$ for $\alpha \neq \beta$.

User's Data (Possible User's Axioms)

$Brd(R, \alpha)$, i.e., R broadcasts that α is the winner, where $R \in \mathcal{S}$ and $\alpha \in \mathcal{P}$.
$Day = Fri$, $Day \neq Fri$.
As inference rules we use that of predicate logic.

Proofs

Any proof starts with user's data.

If a user says something about a broadcast, then for the user it is truthful. We assume that in the set of proofs we consider, all the proofs have the final formula of the form $Win(\alpha)$, though the inference system outlined above permits proofs with other final formulas, e.g., $(Win(Bok) \vee Win(Fok))$.

Here is the set of proofs we consider. We do not make explicit the analysis, i.e., by what inference rule this or that formula is obtained, it is evident.

***QB*1**: $Day = Fri$, $Brd(R_2, Bok)$, $Win(Bok)$.
***QB*2**: $Day \neq Fri$, $Brd(R_2, Dok)$, $Brd(R_1, Bok)$, $Win(Bok)$.

$\boldsymbol{QB3}$: $Day \neq Fri$, $Brd(R_2, Dok)$, $Win(Bok) \vee Win(Fok)$, $Brd(R_3, Fok)$, $\neg Win(Fok)$, $Win(Bok)$.
$\boldsymbol{QD1}$: $Brd(R_1, Dok)$, $Win(Dok)$.
$\boldsymbol{QD2}$: $Day = Fri$, $Brd(R_3, Fok)$, $Brd(R_2, Dok)$, $Win(Dok)$.
$\boldsymbol{QD3}$: $Day = Fri$, $Brd(R_3, Fok)$, $Brd(R_1, Dok)$, $Brd(R_2, Dok)$, $Win(Dok)$.
$\boldsymbol{QF1}$: $Day \neq Fri$, $Brd(R_2, Dok)$, $Win(Bok) \vee Win(Fok)$, $Brd(R_3, Bok)$, $\neg Win(Bok)$, $Win(Fok)$.

3 Informativeness of Proofs

How much of information one has in a proof? First, we describe a possible approach in terms related to the example, and after that in Sect. 4 we give an abstract set-theoretic framework that does not mention proofs.

The proofs we consider are proofs of responses to information queries. Any information query is something like $\boldsymbol{Find}\, x\, \Phi(x)$ where $\Phi(x)$ is a formula, an information property. An answer to such a query is a formula $\Phi(\alpha)$ with a constant α. An information proof is a proof of $\Phi(\alpha)$. Below Φ is fixed.

3.1 Entropic Weight

In our setting all sets are finite.

Notations:

- $\mathcal{A}$ is a set of constants that are used in answers to queries.
- $\mathcal{F} = \{\Phi(\alpha)\}_\alpha$ is a set of answers to Φ, $\alpha \in \mathcal{A}$. An answer $\Phi(\alpha)$ may have different proofs.
- $\mathcal{Q}$ is the set of proofs under consideration. A proof is a list of formulas; a proof of a formula Φ is a proof with the last formula Φ.
- $\mathcal{Q}_{\Phi(\alpha)} = \mathcal{Q}_\alpha = \{Q \in \mathcal{Q} : Q \text{ is a proof of } \Phi(\alpha)\}$, $\mathcal{Q}_\alpha$ is a set of proofs of $\Phi(\alpha)$.
- $q_\alpha = |\mathcal{Q}_\alpha|$, $M = |\mathcal{A}|$.
- Probabilistic measure $\boldsymbol{P}$ on the proofs: $\boldsymbol{P}(\mathcal{Q}_\alpha) = \frac{1}{M}$, $\boldsymbol{P}(Q) = \frac{1}{M \cdot q_\alpha}$ for $Q \in \mathcal{Q}_\alpha$.

The chosen probabilistic measure is technically simple. One can say that the choice is not well justified (in passing just notice that the uniform distribution models maximal uncertainty). Very likely, some kind of general probabilistic measure will also work but this question has not been studied. On the other hand, we are out of context of real applications, so there is no reason to complicate the technicalities.

Strictly speaking, a proof consists of formulas and of analysis, i.e., of references to the rules applied. However, for simplicity we represent a proof as just a set of formulas. For the proofs related to the example the analysis is evident.

We wish to measure 'informativeness' of a given proof, more specifically, how one gets more and more information by taking into account bigger and bigger subsets of formulas of the proof.

To do it, for a given subset of formulas of a given proof we introduce entropic weight—a measure with a flavor of entropy that has properties corresponding to the intuition in the context under consideration.

For a subset S of formulas of a proof we set $E(S) = \{Q : S \subseteq Q\}$, and define its *entropic weight* $\mathcal{D}(S)$:

$$\mathcal{D}(S) = \mathcal{D}(E(S)) = -\sum_{\alpha} \boldsymbol{P}(E(S) \cap \mathcal{Q}_\alpha) \log \frac{\boldsymbol{P}(E(S) \cap \mathcal{Q}_\alpha)}{\boldsymbol{P}(E(S))}, \tag{1}$$

here and below $\log$ is $\log_2$.

Taking into account that the sets $(E(S) \cap \mathcal{Q}_\alpha)$ are disjoint, $\bigcup_\alpha \mathcal{Q}_\alpha = \mathcal{Q}$, $\boldsymbol{P}(\mathcal{Q}) = 1$, and thus $\sum_\alpha \boldsymbol{P}(E(S) \cap \mathcal{Q}_\alpha) = \boldsymbol{P}(E(S))$, we can rewrite formula (1) for $\mathcal{D}(S)$ as:

$$\mathcal{D}(S) = -\sum_{\alpha} \boldsymbol{P}(E(S) \cap \mathcal{Q}_\alpha) \log \boldsymbol{P}(E(S) \cap \mathcal{Q}_\alpha) + \boldsymbol{P}(E(S)) \log \boldsymbol{P}(E(S)) \tag{2}$$

Notice that the notation $\mathcal{D}(S)$ with argument S, and not $E(S)$, is in a way misleading: when S increases, the argument $E(S)$, that is in fact used, grows down (non strictly).

Entropic weight $\mathcal{D}(S)$ has the following properties:

(D1) $\mathcal{D}(\emptyset) = \log M$ (maximal uncertainty)
(D2) $\mathcal{D}(S) = 0$ for any $\alpha \in \mathcal{A}$ and any $S \subseteq \mathcal{Q}_\alpha$ such that $E(S) \subseteq \mathcal{Q}_\alpha$ (maximal certainty)
(D3) $\mathcal{D}(S)$ is non-increasing when S grows: if $S \subseteq S'$ then $\mathcal{D}(S) \geq \mathcal{D}(S')$ (the uncertainty does not grow with getting more and more of information).

The properties are proved in Proposition 1 in Sect. 4 below.

In order to evaluate the evolution of informativeness we look at what happens with entropic weight when the size of subsets S increases. How to choose such subsets? We do it according the principle of maximal uncertainty. Imagine that the choice is being done by an adversary who tries to maximize the uncertainty concerning the result.

Look at the example.

3.2 Entropic Weight for the Example

The measure $\boldsymbol{P}$ of each proof $\boldsymbol{QBi}$, $\boldsymbol{QDi}$, $1 \leq i \leq 3$, is $\frac{1}{9}$, and that of $\boldsymbol{QF1}$ is $\frac{1}{3}$;
$\mathcal{Q}_{Bok} = \{\boldsymbol{QBi}\}_{i=1,2,3}$, $\mathcal{Q}_{Dok} = \{\boldsymbol{QDi}\}_{i=1,2,3}$, $\mathcal{Q}_{Fok} = \{\boldsymbol{QF1}\}$.

Consider $\boldsymbol{QB1}$. For one-element subsets $U_0 = \{Brd(R_2, Bok)\}$ and $U_0 = \{Win(Bok)\}$ it is, respectively, $E(U_0) = \{\boldsymbol{QB1}\}$ and $E(U_0) = \{\boldsymbol{QB1}, \boldsymbol{QB2}, \boldsymbol{QB3}\}$, and thus $\mathcal{D}(U_0) = 0$ as follows from (D2). Such a choice of one-element subsets does not give the maximal entropic weight for one-element subsets.

Take the remaining one-element subset, namely, $U_1 = \{Day = Fri\}$. We have $E(U_1) = \{\boldsymbol{QB1}, \boldsymbol{QD2}, \boldsymbol{QD3}\}$, and $\boldsymbol{P}(E(U_1)) = \frac{1}{9} + \frac{1}{9} + \frac{1}{9} = \frac{1}{3}$.

The measures of intersections are: $\boldsymbol{P}(E(U_1) \cap Q_{Bok}) = \boldsymbol{P}(\{\boldsymbol{QB1}\}) = \frac{1}{9}$,

$\boldsymbol{P}(E(U_1) \cap Q_{Dok}) = \boldsymbol{P}(\{\boldsymbol{QD2}, \boldsymbol{QD3}\}) = \frac{2}{9}$, $\boldsymbol{P}(E(U_1) \cap Q_{Fok}) = \boldsymbol{P}(\emptyset) = 0$.

With (2) we get

$\mathcal{D}(U_1) = -\frac{1}{9}\log\frac{1}{9} - \frac{2}{9}\log\frac{2}{9} + \frac{1}{3}\log\frac{1}{3} \approx 0.31$.

If we extend the set U_1 in any way to U_1' we get $\mathcal{D}(U_1') = 0$ as follows from (D2). So for 1-element subsets of $\boldsymbol{QB1}$ the maximal entropic weight is approximately 0.31, and for 2-element subsets of $\boldsymbol{QB1}$ the maximal entropic weight is 0. The passage from 0.3 to 0 shows the speed of convergence to complete certainty.

Consider $\boldsymbol{QB3}$, and three sets

$S_1 = \{Brd(R_2, Dok)\}$,

$S_2 = \{Day \neq Fri,\ Brd(R_2, Dok)\}$,

$S_3 = \{Day \neq Fri,\ Brd(R_2, Dok), Win(Bok) \vee Win(Fok)\}$.

These sets maximize the entropic weight for the sets of size respectively 1, 2, 3. We have $E(S_1) = \{\boldsymbol{QB2}, \boldsymbol{QB3}, \boldsymbol{QD2}, \boldsymbol{QD3}, \boldsymbol{QF1}\}$, $E(S_2) = \{\boldsymbol{QB2}, \boldsymbol{QB3}, \boldsymbol{QF1}\}$,

$E(S_3) = \{\boldsymbol{QB3}, \boldsymbol{QF1}\}$, and $\boldsymbol{P}(E(S_1)) = \frac{4}{9} + \frac{1}{3} = \frac{7}{9}$, $\boldsymbol{P}(E(S_2)) = \frac{2}{9} + \frac{1}{3} = \frac{5}{9}$, $\boldsymbol{P}(E(S_3)) = \frac{1}{9} + \frac{1}{3} = \frac{4}{9}$.

For intersections of $E(S_1)$ with Q_α we have $E(S_1) \cap Q_{Bok} = \{\boldsymbol{QB2}, \boldsymbol{QB3}\}$, $E(S_1) \cap Q_{Dok} = \{\boldsymbol{QD2}, \boldsymbol{QD3}\}$, $E(S_1) \cap Q_{Fok} = \{\boldsymbol{QF1}\}$, and $\boldsymbol{P}(E(S_1) \cap Q_{Bok}) = \boldsymbol{P}(E(S_1) \cap Q_{Dok}) = \frac{2}{9}$, $\boldsymbol{P}(E(S_1) \cap Q_{Fok}) = \frac{1}{3}$.

For intersections of $E(S_2)$ we have $E(S_2) \cap Q_{Bok} = \{\boldsymbol{QB2}, \boldsymbol{QB3}\}$, $E(S_2) \cap Q_{Dok} = \emptyset$, $E(S_2) \cap Q_{Fok} = \{\boldsymbol{QF1}\}$, and $\boldsymbol{P}(E(S_2) \cap Q_{Bok}) = \frac{2}{9}$, $\boldsymbol{P}(E(S_2) \cap Q_{Dok}) = 0$, $\boldsymbol{P}(E(S_2) \cap Q_{Fok}) = \frac{1}{3}$.

For intersections of $E(S_3)$ we have $E(S_3) \cap Q_{Bok} = \{\boldsymbol{QB3}\}$, $E(S_3) \cap Q_{Dok} = \emptyset$, $E(S_3) \cap Q_{Fok} = \{\boldsymbol{QF1}\}$, and $\boldsymbol{P}(E(S_3) \cap Q_{Bok}) = \frac{1}{9}$, $\boldsymbol{P}(E(S_3) \cap Q_{Dok}) = 0$, $\boldsymbol{P}(E(S_3) \cap Q_{Fok}) = \frac{1}{3}$.

We use (2) to calculate the values of $\mathcal{D}$:

$\mathcal{D}(S_1) = -2\frac{2}{9}\log\frac{2}{9} - \frac{1}{3}\log\frac{1}{3} + \frac{7}{9}\log\frac{7}{9} \approx 1.21$

$\mathcal{D}(S_2) = -\frac{2}{9}\log\frac{2}{9} - \frac{1}{3}\log 3 + \frac{5}{9}\log\frac{5}{9} \approx 0.54$

$\mathcal{D}(S_3) = -\frac{1}{9}\log\frac{1}{9} - \frac{1}{3}\log 3 + \frac{4}{9}\log\frac{4}{9} \approx 0.36$

For bigger subsets S of $\boldsymbol{QB3}$ we get $\mathcal{D}(S) = 0$.

The sequence of $\mathcal{D}(S_k)$, above corresponds to the values of function $\delta(\boldsymbol{QB3}, k)$ from Sect. 4 for $k = 1, 2, 3$.

Notice that $\mathcal{D}(S_i) = \max\{\mathcal{D}(S) : S \text{ is a subset of } \boldsymbol{QB3} \wedge |S| = i\}$. The sequence $\mathcal{D}(S_1), \mathcal{D}(S_2), \mathcal{D}(S_3)$ shows the speed of convergence of entropic weight to 0 for $\boldsymbol{QB3}$.

4 Abstract Definition of Entropic Weight

Notations:

- $\mathcal{F}$ is a set (it corresponds to the set of formulas above).
- $\mathcal{G} \subset \mathcal{F}$ is a subset of $\mathcal{F}$ (these are goals, it corresponds to $\{\Phi(\alpha)\}_\alpha$ above), $M = |\mathcal{G}|$.
- $\mathcal{Q}$ is a set of subsets of $\mathcal{F}$ (corresponds to the set Q of proofs above); each $Q \in \mathcal{Q}$ contains exactly one element of $\mathcal{G}$, its goal, and (for simplicity) each element of $\mathcal{G}$ belongs to some $Q \in \mathcal{Q}$.
- $\mathcal{Q}_\phi = \{Q \in \mathcal{Q} : \phi \in Q\}$ for $\phi \in \mathcal{G}$ (corresponds to sets Q_α above).
- Probabilistic measure $\boldsymbol{P}$ on $\mathcal{Q}$: $\boldsymbol{P}(\mathcal{Q}_\phi) = \dfrac{1}{M}$ for $\phi \in \mathcal{G}$;
  $\boldsymbol{P}(Q) = \dfrac{1}{M \cdot |\mathcal{Q}_\phi|}$ for $Q \in \mathcal{Q}_\phi$.
- *Entropic weight.* For a $Q \in \mathcal{Q}$ and $S \subseteq Q$ set $E(S) = \{Q' \in \mathcal{Q} : S \subseteq Q'\}$ and

$$\mathcal{D}(S) = -\sum_{\phi \in \mathcal{G}} \boldsymbol{P}(E(S) \cap \mathcal{Q}_\phi) \log \frac{\boldsymbol{P}(E(S) \cap \mathcal{Q}_\phi)}{\boldsymbol{P}(E(S))}, \tag{3}$$

or equivalently

$$\mathcal{D}(S) = -\sum_{\phi \in \mathcal{G}} \boldsymbol{P}(E(S) \cap \mathcal{Q}_\phi) \log \boldsymbol{P}(E(S) \cap \mathcal{Q}_\phi) + \boldsymbol{P}(E(S)) \log \boldsymbol{P}(E(S)). \tag{4}$$

As measures of convergence one can take the following functions.

- $\delta(Q, k) = \max\{\mathcal{D}(S) : S \subseteq Q \wedge |S| = k\}$ for $Q \in \mathcal{Q}$. This function is non-increasing when k grows, it may be used to characterize speed of convergence to certainty.
- $\zeta(Q) = \min\{k : \delta(Q, k) = 0\}$. Clearly, $0 < \zeta(Q) \leq |Q|$. This is the minimal size of subsets of a proof that guarantees certainty in the worst case.
- $\frac{1}{|Q|} \sum_{i=1}^{i=|Q|} \delta(Q, i)$ is the average entropic weight of Q.
- $\frac{1}{\zeta(Q)-1} \sum_{i=1}^{i=\zeta(Q)-1} \big(\delta(Q, k)(i) - \delta(Q, k)(i+1)\big)$ is, in a way, average speed of convergence to certainty.

Proposition 1

(D1) $\mathcal{D}(\emptyset) = \log M$ (maximal uncertainty)
(D2) $\mathcal{D}(S) = 0$ for any $\phi \in \mathcal{G}$ and any $S \subseteq \mathcal{Q}_\phi$ such that $E(S) \subseteq \mathcal{Q}_\phi$
(maximal certainty)
(D3) $\mathcal{D}(S)$ (in fact, $\mathcal{D}(E(S))$) is non-increasing when its argument S grows:
if $S \subseteq S'$ then $\mathcal{D}(S) \geq \mathcal{D}(S')$
(the uncertainty does not grow with getting more information).

Proof

(D1). Indeed, $E(\emptyset) = \mathcal{Q}$, $\boldsymbol{P}(\mathcal{Q}) = 1$,
$\mathcal{D}(\mathcal{Q}) = -\sum_\alpha \boldsymbol{P}(\mathcal{Q}_\alpha) \log \frac{\boldsymbol{P}(\mathcal{Q}_\alpha)}{1} = -\sum_\alpha \frac{1}{M} \log \frac{1}{M} = \log M$.

(D2). Take $E(S) \subseteq \mathcal{Q}_\alpha$. Then $\mathcal{D}(S) = -\boldsymbol{P}(E(S)) \log \frac{E(S)}{E(S)} = 0$,
$E(S) \cap \mathcal{Q}_\beta = \emptyset$ for $\beta \neq \alpha$ and thus $\boldsymbol{P}(E(S) \cap \mathcal{Q}_\beta) = 0$.

(D3). To prove (D3) we follow the lines of the proof of a similar property for the entropic weight introduced in [4].

Take any function of continuous time $S(t) \subseteq \mathcal{F}$ such that $S(t_0) \subseteq S(t_1)$ for some $t_0 \leq t_1$. Then $E(S_1) \subseteq E(S_0)$.

Let $x_\phi(t)$ be a differentiable function, non-increasing when t goes from t_0 to t_1, such that $x_\phi(t_0) = \boldsymbol{P}(E(S(t_0)) \cap \mathcal{Q}_\phi)$ and $x_\phi(t_1) = \boldsymbol{P}(E(S(t_1) \cap \mathcal{Q}_\phi)$. Clearly, such a function exists and even can be easily constructed.
We have $\sum_\phi x_\phi(t_j) = \boldsymbol{P}(E(S(t_j)))$, $j = 0, 1$.

Set
$p(t) = -\sum_\phi x_\phi(t) \log x_\phi(t) + \big(\sum_\phi x_\phi(t)\big) \log \big(\sum_\phi x_\phi(t)\big)$.
Then from (4) we see that $p(t_j) = \mathcal{D}(S(t_j))$.
We have $0 \leq x_\phi(t) \leq \frac{1}{M}$ and $0 \leq \sum_\phi x_\phi(t) \leq 1$.

Assume that $S(t_0)$ is not empty, otherwise (D3) is trivial because of (D1). In this case $0 < x_\phi(t_1) \leq x_\phi(t) \leq x_\phi(t_0)$. Take the derivative of $p(t)$ over t (recall that $\log z = \frac{\ln z}{\ln 2}$):

$$p'(t) = -\sum_\phi \Big(x'_\phi \log x_\phi + x_\phi \frac{x'_\phi}{x_\phi \cdot \ln 2}\Big) +$$

$$\Big(\sum_\phi x'_\phi\Big) \log \Big(\sum_\phi x_\phi\Big) + \Big(\sum_\phi x_\phi\Big) \frac{\Big(\sum_\phi x'_\phi\Big)}{\Big(\sum_\phi x_\phi\Big) \ln 2} =$$

$$-\sum_\phi \Big(x'_\phi \log x_\phi + \frac{x'_\phi}{\ln 2}\Big) + \sum_\phi \Big(x'_\phi \log \Big(\sum_\phi x_\phi\Big) + \frac{x'_\phi}{\ln 2}\Big) =$$

$$\sum_\phi x'_\phi \cdot \log \frac{\Big(\sum_\phi x_\phi\Big)}{x_\phi} \qquad (5)$$

The functions x_ϕ are non-increasing, thus $x'_\phi \leq 0$. As $\sum_\phi x_\phi \geq x_\phi$ the value of (5) is non-positive, hence $p(t)$ is non-increasing when $S(t)$ increases. □

Remark With respect to the sets $E(S)$ the function $\mathcal{D}(E(S))$ is non-decreasing: if $E(S) \subseteq E(S')$ then $\mathcal{D}(E(S)) \leq \mathcal{D}(E(S'))$. When S grows from $\emptyset$ to $\mathcal{Q}_\phi$ the values of $\mathcal{D}(S)$ decrease from $\log M$ to 0, so this decreasing is 'strict on the whole'.

Acknowledgments I am thankful to the anonymous referee for the remarks and especially for the critical ones. I particularly appreciate the remark "that only exact deductions are taken into account" is a missing feature. But I cannot take it into consideration immediately.

References

1. Adriaans, P.: Information. In: Zalta, E.N. and Nodelman, U. (ed.) The Stanford Encyclopedia of Philosophy. Fall 2013 edn. (2013), ISSN 1095-5054
2. Cover, T., Thomas, J.: Elements of Information Theory, 2nd edn. Wiley-Intersience, New York (2006)
3. Floridi, L.: Semantic conceptions of information. In: Zalta, E.N. and Nodelman, U. (ed.) The Stanford Encyclopedia of Philosophy. Spring 2013 edn. (2013), ISSN 1095-5054
4. Slissenko, A.: On entropic convergence of algorithms. In: Blass, A., Cégielski, P., Dershowitz, N., Droste, M., Finkbeiner, B. (eds.) Fields of Logic and Computation III. Lecture Notes in Computer Science, vol 12180, pp. 291–304. Springer, Cham (2020)

Science and Practice of Modelling

Bernhard Thalheim

Devoted to Janos Makowsky

Abstract We conceive this birthday contribution as an eye-opener and an introduction to a new sub-discipline of Computer Science, model science and practice—modelology. It actually represents the fourth dimension of Computer Science, but it is so fundamental that all other dimensions must refer to it. A sub-discipline can only be effective if it is also practical and workable. Therefore we have to understand modelology not as representation science or even as a drawing tool for the visualization of complicated things, but as a broadly everywhere applicable systematic science and practice of modeling. Models should not only facilitate understanding but above all accompany system development and perfection.

1 Computer Science on the Way to Science

Computer Science has led to a real revolution after the industrial revolution. However, it is still in the early stages of its development and work in progress. If we follow Peter Denning's assertions, e.g. [4], Computer Science still has a long way to go to become a science one day. Although already technologically relevant to much of today's infrastructure as a pre-science and pre-engineering, it still takes generations of effort to turn it into a science combined with engineering. Besides many successful examples, there are enough failures. For example, object-oriented database technology and active database systems have become bogged down and have bitterly disappointed expectations, not to mention the new hypes.[1]

[1] See, for instance, the analysis of potential and capacity of AI in [9, 30].

B. Thalheim (✉)
Christian Albrechts University Kiel, Kiel, Germany
e-mail: bernhard.thalheim@email.uni-kiel.de

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_29

Computer Science is full of unwritten laws, creeds, postulates and principles about which you can rarely read anything, but which you need to know in order to understand anything at all more precisely. Two examples are the typical database approach of global–as–design and the strict separation with the principle syntax–first–semantics–then–on–syntax–basis.

There are many more surprising examples of this shirtlessness in Computer Science. Even for the basics like Turing computability, the postulates, paradigms and principles are hidden [29] although there are dozen important ones. The same situation prevails in the database world. A typical database technology assumption is global–as–design, where it is assumed that a global schema integrates all local viewpoints of application harmoniously at the same level of detail, so that from the global schema all local and implementation viewpoints can be derived within a constrained language. To avoid the pitfalls and shortcomings associated with this, star and snowflake schemes are then considered as materialisable derivations for local computations and necessary evaluations. The implementation viewpoints are based on barely traceable additions of the highest professionalism. Perhaps this work should be completely rethought while preserving the heritage that the new challenges subsumed under big data already show.

In natural languages, syntax, semantics and pragmatics form a unit. Computer Science, on the other hand, following mathematical logic, believes that syntax must be considered as the primary part and then semantics can be subordinated based on the definition procedure of syntax. This principle is also followed by the database world. The syntax is defined first and everything additional is shifted to the semantics, especially by means of functional dependencies.[2] One hopes also still additionally that one can neglect the pragmatics of the use and the applications for the time being. An example, which shows, how problematic such an approach is, is the Boyce-Codd normal form for database structure specification [19, 26, 27].

Database theory research has led to an elaboration of differences between the simple normal forms based on sets of functional dependencies and the Boyce-Codd normal form, which at least guarantees a simple treatment of integrity with relational database systems. These differences, however, leave us with the question of why they must exist as they do.

But one can also ask such questions once in the context of the starting point. Is not perhaps the assumed structuring the cause of this confusion? There is an answer to this: One simply searches once for the design error and in particular the hidden structure within the attributes [17]. And already there is a solution that convinces. The solution in this case was relatively simple: break the problematic attributes into their real components and model the structure again.

[2] The semantics definition of database structures over sets of functional and key dependencies collapses—for the worst case and even for the average case—due to exponential complexity in the number of attributes [3] where in practice typical relational database structures use at least several dozen of attributes.

This solution leads to a general principle, which can be represented for computer science as follows:

The Makowsky Principle

Whenever a problem entails only complex studies, then first of all the problem itself and its setting should be investigated in more detail.[3]

This principle was actually on the street, because mathematics has long known that a wrongly posed problem rarely leads to anything sensible. It was just not yet well-known for Computer Science.

As an early stage development, Computer Science knowledge is still taught as cookbook knowledge, with the cookbook being rewritten and re-taught with each new programming language, although with algorithmics [1, 7] and systems development in LaTeX form [12], role models have already existed for more than 30 years for systematic development.

Older science and engineering disciplines start university education by working out the paradigms and dimensions of these disciplines, so that special branches can then be understood as refinements and extensions of these paradigms and dimensions. We can currently identify four main dimensions that characterize all informatics research and practice: state, transformation, collaboration, and modeling both at micro and macro levels or even at meso level in addition [32]. Two of the other dimensions could be approximation and HCI (better human (web-based) systems, e.g. [25]). We are not yet ready to present a systematic of all dimensions, although the first three dimensions already have a good and expandable basis. A model science as science, art and tested and well-developed practice does not yet exist. We are at the beginning of such a art[4] and lore. This art and lore will lead to a new sub-discipline of Computer Science, modelology [28] ("Modellkunde"), which we will introduce in the following section.

We will, however, limit ourselves to the central approaches of modelology, knowing that we are just at the beginning here as well. In addition, there is the currently still misleading belief in the descriptive character of models, in which with a description and visualization models are used at best as an initial step or even worse only as inspiration for the programmer and for this reason are disposable goods after the system realisation. That is why today "models" are also called by many other names, e.g. "digital twins", so that modeling seems to lose its importance. This situation is exacerbated by selling models as study-based theories, thereby forgetting that models focus on some, mostly essential, aspects while celebrating exclusion of others. Additionally, models are combined with other models and other findings, so that they always appear as model societies

[3] The cited work and this principle has been extended in a whole series of works e.g. [18, 22] and applied many times. Therefore, this principle could also be called the Makowsky/Ravve principle.

[4] In the sense of [13].

or model suites,[5] so that the big context is preserved. There are further "fast", first models and also "slow", well-developed and thought-out models just as there is the fast and slow thinking [10]. To show the real potential of modeling, we will then illustrate the path to prescription models in a small case study for object-relational database structuring development. Models grow and thrive with their use in the appropriate environment in appropriate scenarios for appropriately trained and skilled users. Subsequently, we will briefly and very selectively discuss the theoretical foundations of modelology.

2 Modelology: An Approach to Systematic Modelling

Models occur everywhere, in science, engineering, education, and even in daily life. Everything can become a model, so we have to ask ourselves what being a model actually is. With the apparent decline in the importance of models, we should ask the Makowsky question of whether perhaps the expression of model lore has not so far led us astray. In this paper, we omit the presentation of the fundamentals of modelology and focus on some facets of this discipline.

2.1 The Conception of the Model-Being

Let us briefly remember notions to modelling in [28] (see also [30–34]): [6])
"A **model** *is a well-formed, adequate, and dependable instrument that represents origins and that functions in utilisation scenarios."* [6, 28]

"Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its ... CoP within some context and correspond to the functions that a model fulfills in utilisation scenarios.

The model should be well-formed according to some well-formedness criterion. As an instrument or more specifically an artifact a model comes with its *background*, e.g. paradigms, assumptions, postulates, language, thought community, etc. The background its often given only in an implicit form. The background is often implicit and hidden.

A well-formed instrument is *adequate* for a collection of origins if it is *analogous* to the origins to be represented according to some analogy criterion, it is more *focused* (e.g. simpler, truncated, more abstract or reduced) than the origins being modelled, and it sufficiently satisfies its *purpose*.

[5] See https://citeseerx.ist.psu.edu/ with search phrase "Model Engineering: Model Suites".

[6] We omit here a detailed bibliography of the more than 4000 works on models and further on the history of modeling already considered by us (see 40+ papers in [28]) and refer especially to [5, 20, 21, 23, 27, 35].

Well-formedness enables an instrument to be *justified* by an empirical corroboration according to its objectives, by rational coherence and conformity explicitly stated through conformity formulas or statements, by falsifiability or validation, and by stability and plasticity within a collection of origins.

The instrument is *sufficient* by its *quality* characterisation for internal quality, external quality and quality in use or through quality characteristics such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc. Sufficiency is typically combined with some assurance evaluation (tolerance, modality, confidence, and restrictions).

A well-formed instrument is called *dependable* [and thus reliable and trustable] "if it is sufficient and is justified for some of the justification properties and some of the sufficiency characteristics." [28]

2.2 The Makowsky Principle Applied

It is often lamented that modeling is receiving less and less attention and is slowly becoming obsolete due to other approaches in Computer Science. This is in harsh contradiction to the claim that modeling is the fourth dimension of Computer Science. We will list here some of the reasons that make the complaint false. The NGram in Fig. 1[7] shows the waning interest in American-English-language literature. The situation is similar for British-English-language or German-language literature.

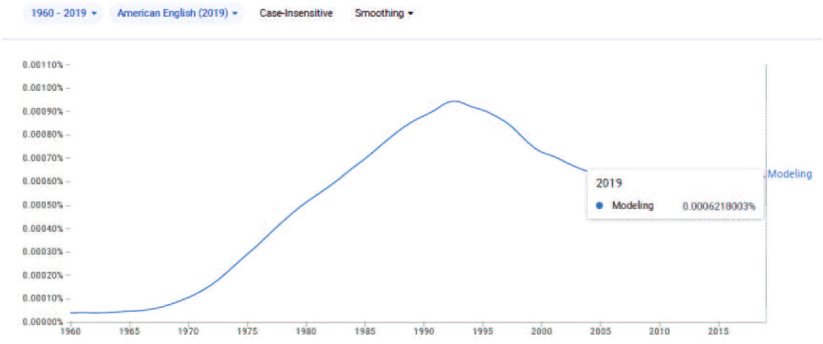


Fig. 1 The evolution of interest in modeling between 1960 and 2019

However, there are many misunderstandings here, e.g. the following:

- Modeling has outlived its usefulness because modern methods replace it. However, modeling is an essential part of all activities in computer science. It is just not called that. The systems are much too complex to be overlooked holistically.

[7] Google Books Ngram Viewer https://books.google.com/ngrams/ assessed Febr. 23, 2023.

- It is believed that after system development the models can just be thrown away and forgotten. Programs are self-explanatory for the specialist. And the modification of models is too time-consuming anyway.
- Modeling is waste without real RoI. You might as well program agile. But programs are also models, even if for a limited circle of users.
- Models can be developed after the realisation, if they are needed at all. One forgets about the further development of systems by other actors who seem to need only a minimal understanding.

There are also aberrations and thus misconceptions, and even worse misdevelopments in modeling:

- Over the last decades, modeling has been reduced to description models. Those models are rather enlightenment and illumination models. Models should simply still be representations of origins to be able to discuss them. They then serve as throwaway inspiration and outlive themselves.
- Models are mainly visualizations, and the visualization languages have acquired a complexity and imprecision that makes them almost unusable, so that only some central constructs are taken seriously.
- Models seem to have to reflect only the dream of future systems without supporting the further activities of programming.

These observations are already more than 20 years old, but continued to go unnoticed [16]. It is therefore high time to turn back, e.g. with the *Makowsky principle* in its extended form:[8]

Think the Modeling from the End

Instead of limiting oneself to description models, one should focus much more on system development and develop models that serve as templates for programming, i.e., prescription models.

2.3 An Answer to Lost Interest in Modelling: Prescription Models

We claim that a prescription model is the king in system development and maintenance and not the description model. A description model is indicative and tells what there is a system and what are ideas or requirements for some current or future system. It may say what is, how, why, when, and where. It might provide an analysis and explanation. It is based on postulates, paradigms, and principles that are commonly accepted in the application domain. Both types of model are additionally

[8] See also [14].

guided by an instruction and action model in which the concrete design of the work and especially modelling processes is specified as a steering guideline.

There are two development scenarios. In greenfield development, a description model is usually created first and then a prescription model is created as an extension based on this. In brownfield development, on the other hand, a prescription model is distilled from the system model or the system, with the description model later being created as a derived model from the system model or from the prescription model.

In addition, the concern of a description model is quite different. It is essentially primarily the understanding of the origins combined with a suitable representation, comprehension, documentation, perhaps even description of a solution. The scenario is then the finding, agreement and explanation of the modeling after an analysis of the origins and a model synthesis. In addition, there is a conceptualization of the constructs used in the model. An extended scenario also focuses on reconciling and negotiating the different views and points of view. The worksheet in Fig. 2 then describes the central tasks of modelling for a description model that supports both formation (e.g. different variants for structuring) and functionalization (e.g. different variants for using the model depending on the type of user) for solution development. We can specify the parameters of the model constellation directly, e.g. *comprehension*.

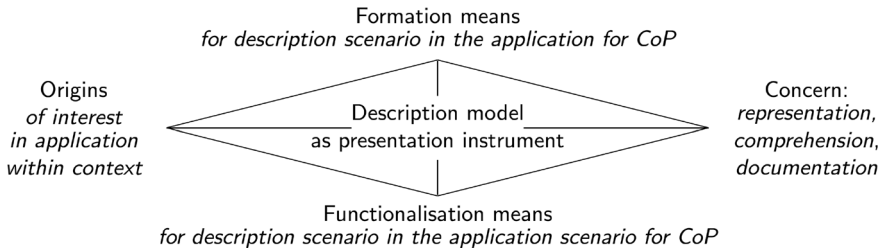


Fig. 2 Constellation worksheet for description models in application for the community of practice (CoP)

If one is additionally oriented towards a highly qualitative description, then a description model also includes the consolidation and documentation of the idea and a detailed justification of this more precise description, e.g. also in a coherent form and a control of the comprehensibility on the basis of a more precise testimony, so that the model also becomes resilient and robust.

A prescription model satisfies a different concern in greenfield development compared to a description model: it is commanding and obligating by telling what is assumed to exist in future and how to do something. It is often based on description models and on models that describe essentials of realisation environments. It should provide a prediction of an augmented or at least revised reality. It must also give explicit prescriptions such as methods, techniques, and assumptions. It is typically both intentional and extensional by extensionally describing and reflecting the system as seen from outside and by intentionally declaring and explaining what

is being modelling within the given environment and why it has the augmentation that it might have.

From one side, this prescription model contains a specification of externally observable properties of elements and behaviour of the system according to the requirements that business users wish them to be. From the other side, it also includes from the other side a specification of means that should be used for realisation of an effective and efficient system. Description models may well remain approximate. Prescription models, on the other hand, must be precise and unambiguous. They have to have a well-defined semantics and should support justification of model correctness and quality control.

A prescription model for greenfield system realisation is a model that is used as

(a) an instrument in a system realisation scenario (its nature)
(b) for construction and coding based on the negotiated blueprint
(c) of structuring of the system and its behaviour while covering of various viewpoints, and
(d) functions well as prescription for development, planning, coding, assess, and prediction of a realisation of database systems.

More generally, prescription describes that what should be an augmented or future situation or what are the objectives.

Prescription for greenfield realisation

(α) inherits the application landscape with the mission, nature, brand and portfolio of the system including postulates of this landscape and is biased by the platform environment with its capacity, potential, and supportable application portfolios,
(β) (implicitly) incorporates the disciplinary matrix with its paradigms, and
(γ) uses supporting and enabling means from its workshop with its specific postulates within the settings.
(δ) It is typically a model suite consisting of several well-associated models and
(ϵ) is based on a description model (suite).

These requirements for a prescription model also show that such a model not only has to be more accurate and correct, but also has to contain additional elements that fit the realization possibilities. We can represent the additional elements of such models in two steps, following common programming languages like C++:

- on the one hand the model is precisely captured in all details e.g. also details about data types, event types and the support of semantics as well as processes and
- on the other hand references to the possibilities of a realization within the platform.

We call the first addition directives and the second addition pragmas in the style of C++.

Figure 3 now shows the constellation for prescription models. The concern, intention and will is now oriented towards system realization, whereby a description model is assumed as the second origin. Formation and functionalisation means are

now based on the implementation scenario, in which the model enriched with directives and pragmas allows at least an initial program to be produced as a template.

Example 1 We can illustrate this for prescription models used for greenfield database application development. Database programming includes besides the automatic transformation or compilation of the actual structures also maintenance of the integrity and consistency of the database. The directives specify the way in which the individual constructs are to be translated. The pragmas set the generated structures into the context of the DBMS, so that with a three-step compilation

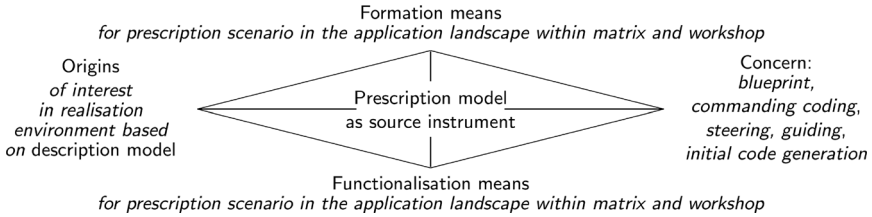


Fig. 3 Constellation worksheet for prescription models for greenfield system realisation scenario

(structure, conversion of the directives, optimization with the pragmas) to a large extent automatically also the entire realisation can be generated including the trigger networks and the macro state-based treatment (e.g. initial, running, archived).

In brownfield development, the concern of a prescription model is again completely different. In the simplest case, a system model is derived from a system, with which the structuring and behavior of the system (or some part of that) is determined on the basis of the analysis and modeling interests and concerns of this system. This can be condensed in a documentation scenario can then be used immediately to derive an informative model for illustration of system capabilities.

A completely different form can be used for prescription models in a system modification scenario. Modification scenarios [11, 36] then determine the form of such models. They often first contain a description or at least documentation of the existing system. However, there are also modification parts with the respective approaches for the step-by-step realization and improvement of the system over a longer period of time with different obligations for the use of the system. In the case of modification, a prescription model is derived from a description of the defects and problems of the system as an additional origin to eliminate the problems, whereby the model then contains a prescription model as a component that describes the original system in its realization peculiarities. This means that the prescription model often consists of at least two sub-models and is a model suite.

We note that analogous investigations can be made for many other model applications, e.g., problem solving, documentation, analysis, synthesis, communication, forecasting, simulation, tuning and negotiation, conceptualizations, explanation, theory development, and subsumption. We can also consider the descriptive representation of user-convenient queries with VisualSQL [8].

3 From Description Models to Prescription Models: A Case Study

As a slightly more complex example, let's look at a simple extended entity-relationship model [27][9] for a company `BookingPortal Ltd.`. For this example, we will limit ourselves to the use case of a direct booking. This example is generally known because well-known portals to support hotel bookings exist in many variants as websites [25]. This creates a database schema in Fig. 4 as a description model (here: without additional integrity constraints) that already reflects many of the company's specific decisions. We will omit the relevant details here, as the schema already speaks for itself.

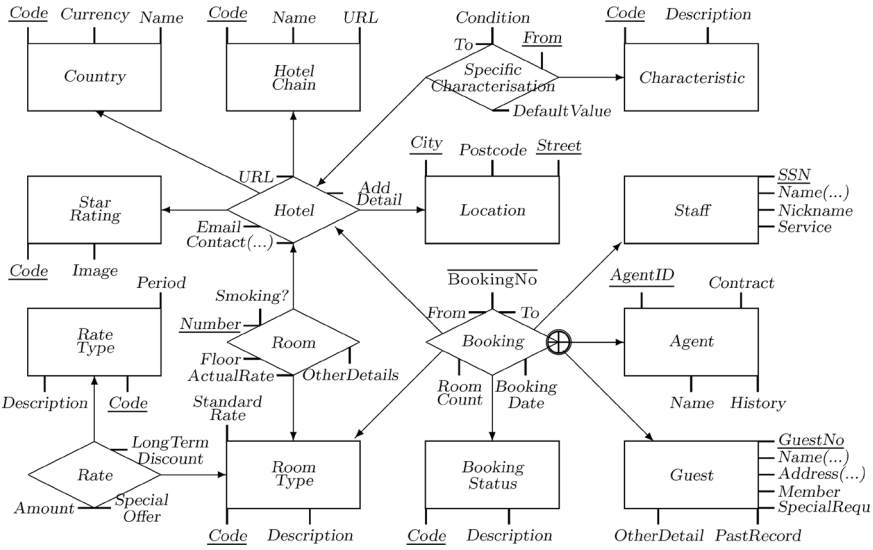


Fig. 4 A description model for the business case of room booking in a hotel chain

The database developer community is still often fobbed off with the description model, although a lot of knowledge exists that can be used to specify the directives at the same time. A simple known extension is the direct specification of the

[9] Our eER language (Higher-Order Entity-Relationship Modelling language (HERM) in [27]) uses attribute types for simple properties, entity types for basic units, relationship types for relationships between types and cluster types as a disjunctive union and thus generalization of component types such as the component that refers to the booker. Relationship types can be hierarchically layered, such as a first-order relationship type *Hotel*, a second-order type such as *Room* as well as *Booking*, which are defined on lower-order types or entity types. Identifying attributes are underlined. There are also derived attributes such as *RoomCount*. Additionally, integrity constraints for all constructs can be attached.

realization data types, e.g. `string[40]` for *Hotel.Chain.Name* or `boolean` for *Smoking?*. Sometimes corresponding patterns are also used as defaults for the integrity constraints. Defaults are always convenient, but are particularly scrutinized and often overwritten by developers.

A problematic feature of the DBMS culture is the treatment of values with NULL markers. This requires both appropriate directives and pragmas [24].

There are also a number of path dependencies. For example, the current rate for a *RoomType* depends on the booking and in particular on the guest's specifics. This cannot be recorded directly in the description model and is then specified separately as a semantic integrity constraint.

There are also implicit type specifications that come directly from the application, such as the decision of `BookingPortal Ltd.` to only serve hotel chains, so that a *Hotel* belongs to a *HotelChain* in a specific *Location*.

All constructs of a description model can be underlaid with directives for realization. In the case of entity types, it has become common practice to use set semantics for the classes, although other semantics such as multi-set semantics or list semantics are also possible. Set semantics are also used for relationship types, although reference semantics are more suitable for high-performance systems. If one is not interested in complex identification mechanisms for the individual data elements, then surrogate keys are used either directly, such as many types of such keys for *Code*, or indirectly added as an extension of the description model.

Things get exciting when a key becomes too complex. For example, *Booking* is identified by a *Hotel*, a *RoomType*, the booker (as either *Agent* or *Guest*) and a further identifier component such as *BookingDate*. In addition, there is the alternative with the booker, so that a special mapping would have to be found here for this alternative in the key. As this is too complex, a *BookingNo* is used as an overwriting primary key, from which the other—now secondary—key follows. All this automatically results in a bundle of integrity constraints for this case as a directive.

In addition, there are also the special types of triggers, stored procedures and other implementation routines that are already given with a special DBMS. These are precisely those pragmas that do not characterise the application but are predefined by the platform and also provide high-performance database support. These are usually well known to the coders, but should at least be supplied with a prescription model. Triggers to support semantic and syntactic integrity conditions are a typical example. Almost all tools for the direct interpretation of structures are problematic here. However, this can easily be improved by specifying explicit trigger nets in the prescription model. Further, interpretation is usually insufficient and should be replaced by a proper compilation.

A first variant of a compiler for the translation of eER schemas into relational schemas was developed in [2, 15]. The algorithms described in [27] are applied to a 3-pass compiler. This compiler can also be extended for translation into SQL schemas. We assume a prescription model with its directives and pragmas based on the description model in Fig. 4 and based on object-relational DBMS such as Sybase

(or Oracle). In order to remain as general as possible for many versions of DBMS, we orient ourselves to SQL2 and only use SQL3 if necessary.

For example, we get the following result:

```
create table HOTEL
(
 Hotel_ID  INTEGER  not null,
 COUNTRY_CODE  CHAR(3)  not null,
 STAR_RATING_ID  INTEGER  not null,
 Hotel_CHAIN_ID  INTEGER  null ,
 Hotel_CODE  CHAR(10)  null ,
 Hotel_NAME  VARCHAR2(40)  null ,
 constraint PK_Hotel primary key (Hotel_ID)
)
/
```

A compiler generally also uses parametric procedural patterns, which can then be used simply by adjusting the parameters. For example, we can use such a procedure for cluster types as well as for the variety of relationship types:

```
if(type = 'cluster'){
 if(no directive known){
    ask user for directive;   directive = developer decision;
    }
    else if(directive = 'combination'){
      use identification directly;
      introduce an explicit separator;
      }
      else if (directive = 'separation'){
        decompose cluster type into new types;
        create disjoint union view on the separated types;
        }
        else if (directive = 'surrogate'){
          introduce surrogate key for the cluster;
          build subtypes depending on the surrogate key;
          }
          else if (directive = 'view'){
            represent the cluster by a view;
            }
            else if (directive = 'union'){
              extend cluster type by a surrogate key;
              set identifiers of components NULL except one;
              }
  } ...
```

This translation uses alternatives that can be specified for cluster types, such as combination into one type, disjoint separation into several types, introduction of a surrogate key and surrogate types with corresponding integrity constraints, treatment by a superimposed view instead of a type, or unification into a universal type.

In the global-as-design approach, a global schema is used with a series of business schemata that can be derived from the global schema. With VisualSQL

[8] we can formulate these view schemata as complex queries and thus apply the same directives and pragmas to the view schemas, so that a first and at the same time sophisticated database description for Sybase is derived.

The translation can also make a number of adjustments, such as the following: Realization styles and tactics, configuration parameters for coding, generic operations for maintaining the database, a variety of predefined queries for the business process, hints for the execution of queries, performance tricks, integrity constraint enforcement, and import to or export from other database systems. In addition, several macro states of database runtime operating such as initial setup, working regime, maintenance regime or archiving can be predefined.

Furthermore, this approach to translation can also be applied to other DBMSs that are not object-relational, so that big data systems can be supported as well as local-as-design or generalized-global-as-design. The latter forms no longer require a homogeneous granularity for all data, but can then use different granularities as well as aggregations for data, as is common for OLAP databases. We have realized such a transformation especially for the development of websites [25].

4 Finally: Modelology of Model Kinds

In this paper, we have restricted ourselves to prescription models and traced the way in which prescription models can be developed as *composite models* from description models by *superposition* of models. This two-step procedure (first presentation models and then through advancement then activity models) is not the only one for developing prescription models. This procedure, however, has the advantage that one can compile an initial realisation semi-automatically by enrichment and thus refinement as well as an additional abstraction and extension by implementation patterns.

Prescription models can also arise directly, as is common for many actions, to which illustrative models are additionally derived for didactic presentation. This direct procedure is common in software development (e.g. agile and extreme one), where different descriptions are then derived from ideas about code development to understand the code. The craft primarily uses prescription models that are assembled with abstractions, if necessary, to form usage models.

With our example it seems that description and prescription models are mainly structural models. However, in all sciences and also in Computer Science there are also description and prescription models which describe the dynamic behavior. Sometimes a description model includes only the behavior, e.g. in the development of programs for analog computers or for modeling the human excitation system.

Typical mathematical simulation models, e.g. for quantitative weather forecasting, have only a prescription character and lagging illustrative models. There are also such superposition procedures for deriving exploratory or explaining models, e.g. in cardiac research. Conversely, there are also pure presentation models, which

are then assembled to complex models for the representation of circumstances, e.g. ceremony statues or toys.

Since models offer us a convenient way out of the complexity of the world by concentrating and focusing on some aspects, models are also combined into *model suites* in which the individual models become associated through appropriate dependencies. The different models in a model suite co-evolve and co-develop together with different commitments in case of further development of one of these models.

References

1. Brassard, G., Bratley, P.: Algorithmics - Theory and Practice. Prentice Hall, London (1988)
2. Dahanayake, A., Pastor, O., Thalheim, B.: Modelling to Program: Second International Workshop, M2P 2020, Lappeenranta, Finland, March 10–12, 2020, Revised Selected Papers. CCIS, vol. 1401. Springer Nature, Berlin (2021)
3. Demetrovics, J., Katona, G.O.H., Miklós, D., Seleznjev, O., Thalheim, B.: Asymptotic properties of keys and functional dependencies in random databases. Theor. Comput. Sci. **190**(2), 151–166 (1998)
4. Denning, P.J.: The science in computer science. Commun. ACM **56**(5), 35–38 (2013)
5. Eck, C., Garcke, H., Knabner, P.: Mathematische Modellierung. Springer, Berlin (2008)
6. Embley, D., Thalheim, B. (eds.): The Handbook of Conceptual Modeling: Its Usage and Its Challenges. Springer, Berlin (2011)
7. Harel, D.: Algorithmics: The Spirit of Computing. Addison-Wesley, Reading (1987)
8. Jaakkola, H., Thalheim, B.: Visual SQL - high-quality er-based query treatment. In: IWCMQ'2003, LNCS 2814, pp. 129–139. Springer, Berlin (2003)
9. Jaakkola, H., Henno, J., Mäkelä, J., Thalheim, B.: Artificial intelligence yesterday, today and tomorrow. In: Proceedings of the MIPRO 2019, pp. 860–867. IEEE (2019)
10. Kahneman, D.: Thinking, Fast and Slow. MacMillan, New York (2011)
11. Klettke, M., Thalheim, B.: Evolution and migration of information systems. In: The Handbook of Conceptual Modeling: Its Usage and Its Challenges, chap. 12, pp. 381–420. Springer, Berlin (2011)
12. Knuth, D.E.: The METAFONTbook. Addison-Wesley (1986)
13. Knuth, D.E.: The Art of Programming I-VI. Addison-Wesley, Reading (1968–2015)
14. Korenblat, K., Ravve, E.V.: Automatic code generator for screen based systems. In: Proceedings of the ADBIS 2019, Short Papers. CCIS, vol. 1064, pp. 253–265. Springer, Berlin (2019)
15. Kramer, F., Thalheim, B.: Holistic conceptual and logical database structure modelling with adoxx. In: Karagiannis, D., Mayr, H.C., Mylopoulos, J. (eds.) Domain-specific Conceptual Model, pp. 269–290. Springer, Cham (2016)
16. Ludewig, J.: Modelle im Software Engineering - eine Einführung und Kritik. Lecture Notes in Informatics, GI **P-12**, 7–22 (2002)
17. Makowsky, J.A., Ravve, E.V.: Translation schemes and the fundamental problem of database design. In: Proceedings of the ER'96, LNCS 1157, pp. 5–26. Springer, Berlin (1996)
18. Makowsky, J.A., Ravve, E.V.: BCNF via attribute splitting. In: Conceptual Modelling and Its Theoretical Foundations - Essays Dedicated to Bernhard Thalheim on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, vol. 7260, pp. 73–84. Springer, Berlin (2012)
19. Mancas, C.: Conceptual Data Modeling and Database Design: A Fully Algorithmic Approach. Apple Academic Press, Cambridge (2015)

20. Müller, R.: Model history is culture history. From early man to cyberspace (2016). http://www.muellerscience.com/ENGLISH/model.htm. Assessed Oct 29, 2017
21. Ortlieb, C.P., von Dresky, C., Gasser, I., Günzel, S: Mathematische Modellierung: Eine Einführung in zwölf Fallstudien. Vieweg, Munich (2009)
22. Ravve, E.V.: Decomposition of databases with translation schemes. Ph.D. Thesis, Technion - Israel Institute of Technology, Israel, 1999
23. Samarskii, A.A., Mikhailov, A.P.: Principles of Mathematical Modelling: Ideas, Methods, Examples (Translated from Russian, 1997). CRC Press, Boca Raton (2001)
24. Schewe, K.-D., Thalheim, B.: NULL value algebras and logics. In: Information Modelling and Knowledge Bases, vol. XXII, pp. 354–367. IOS Press, Amsterdam (2011)
25. Schewe, K.-D., Thalheim, B.: Design and Development of Web Information Systems. Springer, Chur (2019)
26. Silberschatz, A., Korth, H.F., Sudarshan, S.: Introduction to Data base Management System, 7th edn. Tata McGraw Hill, New Delhi (2013)
27. Thalheim, B.: Entity-relationship Modeling – Foundations of Database Technology. Springer, Berlin (2000)
28. Thalheim, B.: Models, to model, and modelling. *collections of papers.* https://www.researchgate.net (search keyphrase "Towards a theory of models, especially conceptual models and modelling"), also academia.edu, 2009–2021
29. Thalheim, B.: The conception of the model. In: BIS. Lecture Notes in Business Information Processing, vol. 157, pp. 113–124. Springer, Berlin (2013)
30. Thalheim, B.: Artificial intelligence enhanced by modelling. Intellectual Syst. Theory Appl. **26**(1), 360–366 (2022)
31. Thalheim, B.: Model-based reasoning for investigating the heart capability. In: Lohff, B., Schaefer, J. (eds.) Cardio-Physiology Challenging Empirical Philosophy, vol. II, pp. 162–199. BoD. Norderstedt (2022)
32. Thalheim, B.: Models: the fourth dimension of computer science – towards studies of models and modelling. Software Syst. Model. **21**(1), 9–18 (2022)
33. Thalheim, B.: Auf dem Wege zur Modellkunde. In: Modellierung 2022. LNI, vol. P-324, pp. 11–32. Gesellschaft für Informatik e.V., Bonn (2023)
34. Thalheim, B.: Modellkunde: kurz & knapp. In: Loeben, C.L. (ed.) Modellkunde und Ägyptologie im Dialog. Essays. Kulturverlag Kadmos, Berlin (2023)
35. Thalheim, B., Nissen, I. (eds.): Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung. De Gruyter, Boston (2015)
36. Thalheim, B., Wang, Q.: Towards a theory of refinement for data migration. In: ER. Lecture Notes in Computer Science, vol. 6998, pp. 318–331. Springer, Berlin (2011)

Graph Polynomials and Local Graph Operations

Peter Tittmann

Abstract This chapter gives an overview of local graph operations, recursions and reductions used in the computation of graph polynomials. It also gives a list of new results, including new local graph operations and their applications, recursive relations for known graph polynomials, the definition of more general graph polynomials that allow the derivation of new recursions for the domination, edge cover, acyclic, and covered component polynomial. Graph polynomials are considered as *generating functions* for some sequences of numbers of certain (induced, spanning, or general) subgraphs of a graph. A *local graph operation* assigns to any graph G another graph H such that G and H differ only in the neighborhood of a vertex or an edge. Local graph operations occur in recursive relations for graph polynomials. They are also the basis for graph reductions.

1 Introduction

The chromatic polynomial, introduced by George David Birkhoff in 1912, was one of the first graph polynomials ever considered. Since then, the number of different graph polynomials under study in algebraic graph theory has been steadily increasing. Why should we still have an interest in graph polynomials today? There are several good reasons:

- Most graph polynomials can be considered as generating functions for some sequences of numbers of certain (induced, spanning, or general) subgraphs of a graph. So they provide a tool for solving subgraph counting problems.
- Relations between different graph polynomials offer a way to a deeper understanding of graph structure and graph properties.
- There is some hope that open problems in graph theory could be solved by using graph polynomials:

P. Tittmann (✉)
Hochschule Mittweida, Mittweida, Germany
e-mail: peter@hs-mittweida.de

K. Meer et al. (eds.), *Model Theory, Computer Science, and Graph Polynomials*,
Trends in Mathematics, https://doi.org/10.1007/978-3-031-86319-6_30

  - A deeper understanding of the chromatic polynomial could provide a shorter and algebraic proof of the Four-Color Theorem.
  - If we could find a formula for the domination polynomial of the Cartesian product of two graphs, then we would be able to settle Vizing's conjecture.
  - Tutte's famous nowhere-zero-five-flow conjecture may find its proof based on the flow polynomial.
- There are interesting applications of graph polynomials outside graph theory, for instance in network reliability, percolation theory, statistical mechanics, and knot theory.

In his seminal paper [46], Johann A. Makowsky took the first step towards unifying and classifying graph polynomials. This seems to be an important undertaking, as there is an unmanageably large list of graph polynomials that continues to grow. However, to hope that we can gather all graph polynomials under one roof is perhaps too much to expect. We will provide here a study of a certain class of graph polynomials that is characterized by the property of satisfying recurrence relations with respect to local graph operations.

Let $\mathcal{G}$ be the set of all finite graphs. A *local graph operation* is a map $\phi : \mathcal{G} \to \mathcal{G}$ that assigns to any graph G another graph H such that G and H differ only in the neighborhood of a vertex or an edge. This excludes operations such as forming the complement of a simple graph or constructing the line graph of a graph. Well-known examples of local graph operations are deletion and contraction of edges or removal of vertices. One of the early used recurrence relations for graph polynomials is the deletion–contraction formula for the chromatic polynomial,

$$P(G, x) = P(G - e, x) - P(G/e, x).$$

More general polynomial graph invariants have been studied in the literature [17, 33, 49]. In most cases, the graph operation results in a graph having a smaller number of vertices and/or edges. This provides a way of recursive calculation of the graph polynomial in question. If we find a first-order recurrence, for instance of the form $f(G) = \psi(f(\phi(G)))$ and $\phi(G)$ is a graph that is simpler (smaller) than G, then we call the pair (ϕ, ψ) a *reduction*. Frequently used forms of reductions are *series* and *parallel reductions*. We might consider a graph polynomial f as a map $f : \mathcal{G} \to \mathbb{Z}[x_1, \ldots, x_r]$ that assigns to each graph a polynomial from the ring $\mathbb{Z}[x_1, \ldots, x_k]$. The map $\psi : \mathbb{Z}[x_1, \ldots, x_r] \to \mathbb{Z}[x_1, \ldots, x_r]$ should satisfy the diagram:

$$\begin{array}{ccc} \mathcal{G} & \xrightarrow{f} & \mathbb{Z}[x_1, \ldots, x_r] \\ {\scriptstyle\phi}\downarrow & & \uparrow{\scriptstyle\psi} \\ \mathcal{G} & \xrightarrow{f} & \mathbb{Z}[x_1, \ldots, x_r] \end{array}$$

There are many approaches to graph recurrences and graph reductions. In [32, 34], recurrences for graph polynomials are studied in the language of finite model theory. This approach proves most useful for the study of complexity of graph polynomials.

We consider in this paper polynomial graph invariants of finite undirected graphs such as, for instance, the Tutte polynomial, the domination polynomial or the matching polynomial. We exclude all generalizations of graph polynomials to matroids, delta-matroids, hypergraphs, embedded graphs, or digraphs. We restrict our investigation to polynomials with integer coefficients and a small fixed number of variables. In fact all polynomials in this paper have at most three variables. The graphs considered in this paper are not necessarily simple, i.e. they might have loops and parallel edges. We assume that whenever to graphs G and H are isomorphic and f is some graph polynomial, then we have $f(G) = f(H)$, with other words, the polynomial f is a graph invariant. We expect that the reader is familiar with the language of graph theory such as presented in textbooks like [19]. Suppose G is a graph with vertex set $V(G)$ and edge set $E(G)$, $v \in V(G)$, and $W \subseteq V(G)$. We just present some notation that will be used in the following:

$N(v)$ or $N_G(v)$	*Open neighborhood* of v, set of vertices adjacent to v
$N[v]$ or $N_G[v]$	*Closed neighborhood* of v, $N[v] = N(v) \cup \{v\}$
∂W	*Edge boundary* of W, $\partial W = \{\{u, v\} \in E(G) \mid u \in W, v \in V(G) \setminus W\}$
∂v	Set of edges incident to v, $\partial v = \partial\{v\}$
$E(W)$ or $E_G(W)$	Set of edges of G that have both end vertices in W
$k(G)$	Number of components of G,
$\mathrm{Ends}(F)$	Set of end vertices of edges in F, $F \subseteq E(G)$
$\mathrm{iso}(G)$	Number of isolated vertices of G
$c(G)$	Number of covered components, $c(G) = k(G) - \mathrm{iso}(G)$
$\langle F \rangle$	Spanning subgraph of G with edge set F
$G[W]$	Induced subgraph of G with vertex set W

This article has three purposes. It presents some new local graph operations based on loops in Sect. 2. A new graph polynomial is introduced in Sect. 3. Some special results concerning representations and reductions of graph polynomials are presented in the same section. Two special reductions, series and parallel reduction, are introduced in Sect. 6 in the context of multivariate extensions. We will show that a local graph operation or a reduction corresponds to a partition of the range of the defining sum of a graph polynomial (the state space). Section 5 presents a method of reducing terms in a recursion by introducing vertex colors.

To render the paper self-contained, we need to introduce a list of graph polynomials and local graph operations. For ease comprehension of the topic, we will also give some proofs or proof ideas for known facts about graph polynomials.

The New Results Presented Here Are

- introduction of loop operations, Sect. 2,

- proof of a recursion for the clique partition polynomial (adjoint polynomial), Sect. 3.6,
- introduction of a extended cut polynomial in Sect. 3.8, derivation of recursive formulae and reductions,
- new recursions for the edge cover polynomial, Theorems 5 and 6,
- a new recursion for the acyclic polynomial, Theorem 8,
- derivation of a simple recursion for the domination polynomial by implementation of a generalized polynomial, Sect. 3.14,
- proving that vertex-labeled graphs can reduce the number of terms in a recursion for the covered components polynomial, Sect. 5.

2 Local Graph Operations

Deletion of an Edge, $G-e$ Let G be a graph and e an edge of G. The graph $G-e$ obtained from G by removing e from the edge set of G is denoted by $G-e$.

Contraction of an Edge, G/e Let G be a graph and $e=\{u,v\}$ an edge of G. The contraction of e is a graph operation that assigns a new graph G/e to G that is obtained from $G-e$ by *merging* the two end vertices u and v of e. This means that u and v are replaced by a single vertex uv which is incident to each edge that was incident to u or to v in G. Consequently, we have

$$\deg_{G/e}(uv)=\deg_G(u)+\deg_G(v)-2.$$

Contraction with Parallel Replacement, $G/\!\!/e$ Let G be a graph and $e=\{u,v\}$ an edge of G. The graph $G/\!\!/e$ is obtained from G/e by replacing each pair of parallel edges resulting from the contraction of e by a single edge. If G is a simple graph, then $G/\!\!/e$ is also a simple graph.

Modified Edge Contraction, $G\wr e$ Let G be a graph and $e=\{u,v\}$ an edge of G. The graph $G\wr e$ is obtained from G/e by removing all single edges emanating from the merged vertex uv and substituting each pair of parallel edges incident to uv by a single edge.

Removal of a Vertex, $G-v$ Let G be a graph and v a vertex of G. The graph $G-v$ is obtained from G by deleting all edges that are incident to v and removing v from the vertex set of G.

Edge Extraction, $G\dagger e$ Let G be a graph and $e=\{u,v\}$ an edge of G. We define $G\dagger e=G-u-v$, that is as the graph obtained from G by removal of both end vertices of e.

Neighborhood removal, $G-N[v]$ Let G be a graph and v a vertex of G. The graph $G-N[v]$ is obtained from G by removal of v and all vertices adjacent to v.

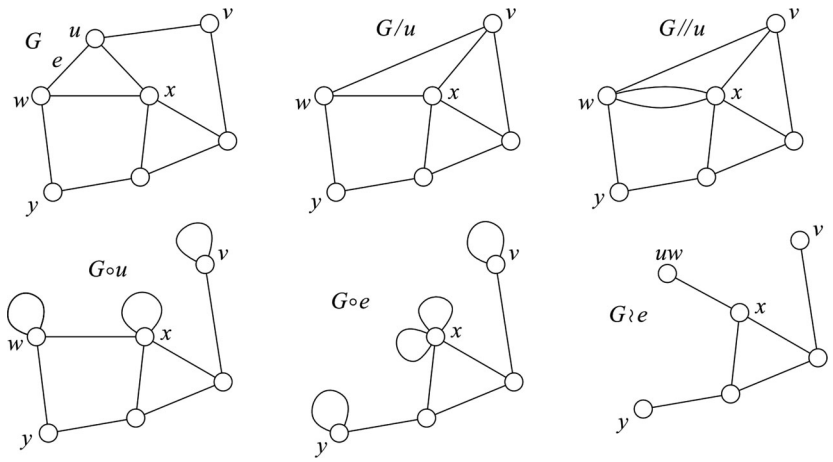


Fig. 1 Illustration of local graph operations

Vertex Completion, G/v Let G be a simple graph and v a vertex of G. The graph G/v is obtained from G by insertion of an edge between any two non-adjacent neighbor vertices of v and removal of v. This implies that the open neighborhood $N_G(v)$ of v in G induces a clique in G/v.

Vertex Resolution, $G /\!\!/ v$ The *resolution* of v of G is a graph $G /\!\!/ v$ obtained from G by removal of v and insertion of edges between all pairs of neighbors of v. The result of this operation is in general a non-simple graph, as we might introduce parallel edges to existing ones. However, we will always avoid the introduction of more than two parallel edges. The reason for this convention is that we need to have a way to represent cycles of length 2. If the vertex to be removed, say v is connected by two edges to a vertex w, then we attach a loop to w in $G /\!\!/ v$.

Vertex Loop Operation, $G \circ v$ Let G be a graph and v a vertex of G. The graph $G \circ v$ is obtained from $G - v$ by attaching one loop to each vertex of $N_G(v)$.

Edge Loop Operation, $G \circ e$ For any edge $e = \{u, v\} \in E(G)$, we define $G \circ e = G \circ u \circ v$.

Figure 1 shows some local graph operations defined above.

Edge Pivot, G^{uv} Let G be a loopless graph and $e = uv$ an edge of G. We define three vertex sets

$$A = N(u) \setminus N[v]$$
$$B = N(v) \setminus N[u]$$
$$C = N(u) \cap N(v).$$

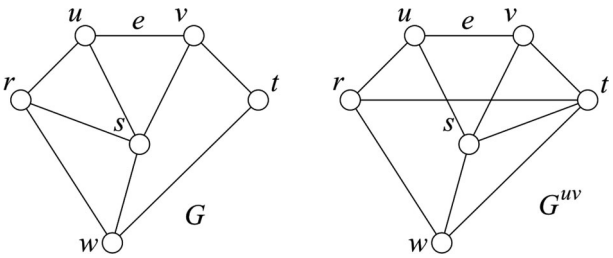


Fig. 2 The edge pivot operation

The *pivot operation* on uv transforms G into a new graph G^{uv} by switching or *toggling* the adjacency relation for any pair $\{s, t\}$ of vertices from G such that s and t belong to two different sets of A, B, C. Figure 2 illustrates the pivot operation. The three sets are for this graph $A = \{r\}$, $B = \{t\}$, and $C = \{s\}$.

3 Graph Polynomials

We denote the family of all finite undirected graphs by $\mathcal{G}$. In the following we use the word *graph* as an abbreviation for *finite undirected graph*. Let p be a *graph property* (usually a graph invariant), which means that p is a mapping $p : \mathcal{G} \to \{\text{true, false}\}$. An *invariant graph property* p satisfies the relation $p(G) = p(H)$ for any two isomorphic graphs G and H. In the following, we will exclusively consider invariant graph properties so that we drop the word *invariant*. A graph property can be given as a *graph predicate* like 'is Hamiltonian,' 'is connected,' 'is planar,' or 'has a perfect matching.' Equivalently, we can define $\mathcal{G}_p$ to be the set of all finite graphs that satisfy p. A graph property p is *trivial* if its value is the same for all graphs, otherwise p is *nontrivial*.

We use in the following the *bracket notation*. Let p be any statement (or proposition), then

$$[p] = \begin{cases} 1, & \text{if } p \text{ is true,} \\ 0, & \text{if } p \text{ is false.} \end{cases}$$

Consequently, we have $\mathcal{G}_p = \{G \in \mathcal{G} : [p(G)] = 1\}$. We will restrict our attention to graph polynomials with a fixed number of variables. This excludes polynomials that uses individual variables for any vertex or edge of the graph.

Let G be a graph, $W \subseteq V(G)$, and $F \subseteq E(G)$. We denote by $G[W]$ the *induced subgraph* of G that has vertex set W and all edges of $E(G)$ that have both end vertices in W as its edge set. The *edge-induced subgraph* $G[F]$ has as vertex set the set of end vertices of edges from F, its edge set is F. Let F be an edge subset of G. The *spanning subgraph* $\langle F \rangle$ has vertex set $V(G)$ and edge set F.

Let r be a given positive integer and p a graph property. For $i = 1, \ldots, r$, let $k_i : \mathcal{G} \to \mathbb{N}$ be a function that assigns to any finite graph a nonnegative integer (usually a numerical graph invariant), for instance the number of vertices, edges, components, or spanning trees. A general definition of a graph polynomial is

$$f(G; x_1, \ldots, x_r) = \sum_{H \leq G} [p(H)] \prod_{i=1}^{r} x_i^{k_i(H)}. \tag{1}$$

The relation " $\leq$" can be read as "is subgraph of" or "is spanning subgraph of" or "is induced subgraph of," depending on the application that we intend. The definition given in Eq. (1) implies that we consider a graph polynomial as an ordinary generating function for graphs with a given property. There are other ways of defining classes of graph polynomials [31, 42].

The definition of a concrete graph polynomial often uses a modified version of Eq. (1) in one of the following ways:

$$f(G; x_1, \ldots, x_r) = \sum_{F \subseteq E(G)} [p(\langle F \rangle)] \prod_{i=1}^{r} x_i^{k_i(\langle F \rangle)}, \tag{2}$$

$$f(G; x_1, \ldots, x_r) = \sum_{W \subseteq V(G)} [p(G[W])] \prod_{i=1}^{r} x_i^{k_i(G[W])}, \tag{3}$$

$$f(G; x_1, \ldots, x_r) = \sum_{\pi \in \Pi(V(G))} \prod_{X \in \pi} [p(G[X])] \prod_{i=1}^{r} x_i^{k_i(G[X])}. \tag{4}$$

Here $\Pi(V(G))$ denotes the set of all partitions of the vertex set of a graph G. The range of the defining sums is in each case a set whose cardinality increases exponentially with the order and size of the graph. In physics this is usually referred to as the *state space* or *configuration space* of a model, for example in statistical mechanics. The art of computing a graph polynomial consists in a sophisticated *partitioning of the state space*, i.e. of the range of the above introduced sums.

In the following, we provide a list of graph polynomials that will be studied in this paper. The experienced reader will notice that sometimes different names or definitions of known graph polynomials are used in the sequel. The intention of this approach is to emphasize the character as generating functions and to unify the way in which graph polynomials are defined. Unless otherwise stated, G will always denote a graph.

We present each graph polynomial as a generating function, i.e. as a sum formula, and give a recurrence relation with respect to local graph operations. If a graph polynomial presented here is a *substitution instance* of another graph polynomial, then it inherits the recurrence relation from its host polynomial.

Note that we have two types of recurrence relations: Some are valid for every edge (vertex) of the graph while others behave different for normal edges, bridges, and loops. The reader will recognize them by the case distinctions given.

3.1 Spanning Subgraph Polynomial

Definition

$$\boxed{\mathcal{Z}(G; x, y) = \sum_{F \subseteq E(G)} x^{|F|} y^{k(\langle F \rangle)}.} \tag{5}$$

We call this polynomial *spanning subgraph polynomial* of G; it has been introduced in a slightly modified form as *dichromatic polynomial* by W. Tutte in [58, 59]. Nowadays it is called the *Tutte polynomial*. The spanning subgraph polynomial has been considered in the context of statistical mechanics by Fortuin and Kasteleyn [33]. They called it the *random cluster model*. Often it is denoted by $Z_G(q, w)$. A third version of the spanning subgraph polynomial is the *rank polynomial*. Nice introductions to these polynomials are given in [1, 16, 28, 35]. The following theorem has been shown in modified form in [58].

Recurrence Relation For any edge $e \in E(G)$:

$$\boxed{\mathcal{Z}(G; x, y) = \mathcal{Z}(G - e; x, y) + x\, \mathcal{Z}(G/e; x, y)} \tag{6}$$

The recurrence relation follows from Eq. (5), we obtain

$$\begin{aligned}
\mathcal{Z}(G; x, y) &= \sum_{F \subseteq E} x^{|F|} y^{k(\langle F \rangle)} \\
&= \sum_{F \subseteq E \setminus \{e\}} x^{|F|} y^{k(\langle F \rangle))} + \sum_{F : e \in F \subseteq E} x^{|F|} y^{k(\langle F \rangle))} \\
&= \mathcal{Z}(G - e; x, y) + x \sum_{F \subseteq E \setminus \{e\}} x^{|F|} y^{k(\langle F \cup \{e\} \rangle))} \\
&= \mathcal{Z}(G - e; x, y) + x\, \mathcal{Z}(G/e; x, y)
\end{aligned}$$

The last equality in this proof demonstrates the use of graph operations. If we know that the edge $e = \{u, v\}$ is contained in any spanning subgraph, then we conclude that the two end vertices u and v of e belong to the same component of $\langle F \cup \{e\} \rangle$. Consequently, we can merge the two vertices to *store this information* in the graph. This corresponds to a contraction of e. The correct edge count (in the exponent of x) is achieved by multiplying the sum with x. There are other ways of storing

information about included or excluded edges and/or vertices (about the *state* of the graph). Common methods are the usage of colors (integer weights) for vertices and edges or even more general edge weights from some ring (polynomials). By application of this approach, we can find *series* and *parallel reductions* for the Tutte polynomial.

However, it is neither possible nor necessary to explore all properties of the spanning subgraph polynomial in detail here, since it is arguably the most studied polynomial in graph theory with numerous publications on algebraic properties, computation, complexity, applications and generalizations, see [30]. We just present some graph polynomials that are well-known evaluations of the spanning subgraph polynomial. As a consequence, they obey the same recurrence relation.

3.2 Chromatic Polynomial

Definition

$$\boxed{P(G,x) = \sum_{F \subseteq E(G)} (-1)^{|F|} x^{k(\langle F \rangle)}} \tag{7}$$

Here we use an alternating sum as definition to be conform with the standard definition so that the $P(G,x)$ equals the number of proper vertex colorings of G with x colors for any $x \in \mathbb{N}$. The presentation of the chromatic polynomial given in Eq. (7) is a result of Whitney [60]. The comparison of Eq. (5) with Eq. (7) yields

$$P(G,x) = \mathcal{Z}(G;-1,x). \tag{8}$$

Using Eq. (6), we find the following relation.

Recurrence Relations

$$\boxed{P(G,x) = P(G-e,x) - P(G/e,x)} \tag{9}$$

for any edge e of G. Since the chromatic polynomial is invariant with respect to parallel replacement, we have also

$$\boxed{P(G,x) = P(G-e,x) - P(G /\!\!/ e,x).} \tag{10}$$

For a deeper introduction to chromatic polynomials, see [14, 15, 50, 58].

3.3 Flow Polynomial

Let G be a graph. For any two edge subsets A and B of G, we denote by $A \oplus B$ the symmetric difference of A and B. An *even subgraph* of G is a subgraph of G that has exclusively vertices of even degree. Let $\mathcal{C}(G) \subseteq 2^E$ the set of all even spanning subgraphs of G, where we define a subgraph by its edge set. Let $T = \langle F \rangle$ be a spanning forest of G, which implies that T is a spanning tree if G is connected. For any edge $e \in E \setminus F$, the graph $T + e$ contains a unique cycle C_e. It can be shown (see [35]) that the set $\{C_e \mid e \in E \setminus F\}$ forms a basis for the vector space $\mathcal{C}(G)$ (over $\mathbb{F}_2$). The vector space $\mathcal{C}(G)$ is called the *cycle space* of G. Its dimension is the *cycle rank* or the *cyclomatic number* of G, denoted by $\beta(G)$. We obtain

$$\beta(G) = |E(G)| - |V(G)| + k(G).$$

A cycle rank generating function for spanning subgraphs of a graph G is

$$\tilde{q}(G, x) = \sum_{F \subseteq E(G)} x^{\beta(\langle F \rangle)} \tag{11}$$

We slightly modify this definition to render the polynomial consistent with the spanning subgraph polynomial. By using an alternating sum in Eq. (11), we obtain the definition of the *flow polynomial*,

Definition

$$\boxed{q(G, x) = \sum_{F \subseteq E(G)} (-1)^{|E(G)|-|F|} x^{\beta(\langle F \rangle)}.} \tag{12}$$

We can easily verify that

$$q(G, x) = (-1)^{|E(G)|} x^{-|V(G)|} \mathcal{Z}(G; -x, x) \tag{13}$$

and

Recurrence Relation

$$\boxed{q(G, x) = \begin{cases} q(G/e, x) - q(G - e, x), & \text{if } e \text{ is not a loop,} \\ xq(G/e, x) - q(G - e, x), & \text{if } e \text{ is a loop.} \end{cases}} \tag{14}$$

The value $q(G, k)$ equals the number of nowhere-zero k-flows on G for any nonnegative integer k.

3.4 Connected Spanning Subgraph Polynomial

The *connected spanning subgraph polynomial* of a graph G is defined as follows.

Definition

$$\boxed{f(G,x) = \sum_{F \subseteq E(G)} [\langle F \rangle \text{ is connected}] x^{|F|}} \tag{15}$$

A slightly modified form of this polynomial is the well-known *reliability polynomial*. The polynomial $f(G,x)$ is just the coefficient of y^0 in $\mathcal{Z}(G;x,y)$:

$$f(G;x) = \left.\frac{\partial \mathcal{Z}(G;x,y)}{\partial y}\right|_{y=0} \tag{16}$$

We can easily verify the following relation.

Recurrence Relation

$$\boxed{f(G,x) = f(G-e,x) + x\, f(G/e,x)} \tag{17}$$

3.5 Euler Polynomial

An *Eulerian subgraph* of a graph $G = (V, E)$ is a spanning subgraph of G in which all vertices have even degree.

Definition The *Euler polynomial* of G is defined by

$$\boxed{\mathcal{E}(G,z) = \sum_{F \subseteq E} [\langle F \rangle \text{ is Eulerian}] z^{|F|}.} \tag{18}$$

Using the relation between Euler and Tutte polynomial, see [1], we find

$$\mathcal{E}(G,z) = (1-z)^{|E(G)|} 2^{-|V(G)|} \mathcal{Z}\left(G; \frac{2z}{1-z}, 2\right). \tag{19}$$

The following relation has been proved in [1].

Recurrence Relation

$$\boxed{\mathcal{E}(G,z) = \begin{cases} (1-z)\mathcal{E}(G-e,z) + z\mathcal{E}(G/e,z), & \text{if } e \text{ is not a loop,} \\ 1+z, & \text{if } e \text{ is a loop.} \end{cases}} \tag{20}$$

3.6 Clique Partition Polynomial

A *clique partition* of a graph G is a partition of $V(G)$ such that each block of π induces a clique (a complete subgraph) in G. We denote by $\Pi_{cl}(G)$ be the set of all clique partitions of G. The *clique partition polynomial* of a graph G, see [45], is defined by a sum ranging over all clique partitions.

Definition

$$\boxed{h(G,x) = \sum_{\pi \in \Pi_{cl}(G)} x^{|\pi|}.} \tag{21}$$

This polynomial is also called *adjoint polynomial*. It is closely related to the chromatic polynomial of G.

Theorem 1 *Let $G = (V, E)$ be a graph and e an edge of G. Then the clique partition polynomial of G satisfies the*

Recurrence Relation

$$\boxed{h(G,x) = h(G-e,x) + h(G \wr e, x).} \tag{22}$$

Proof Let $e = \{u, v\}$ be an edge of G. Let uv be the supervertex resulting from contraction of e. Each clique partition of G belongs to exactly one of the following two classes:

(1) The first class consists of clique partitions π with a block X containing both vertices u and v. All vertices of $X \setminus \{u, v\}$ are adjacent to both vertices u and v, otherwise X would not form a clique of G. Vertices that do satisfy this condition are linked by two parallel edges to the vertex uv in G/e. Consequently, those vertices are adjacent to uv in $G \wr e$. Identifying u and v yields a bijection between all clique partitions of this class and clique partitions of $G \wr e$.
(2) The second class comprises all those clique partitions having u and v in different blocks. These partitions are exactly the clique partitions of $G - e$.

□

If the edge e used in Theorem 1 is not contained in any triangle of G, then the statement of the theorem reduces to $h(G,x) = h(G-e,x) + x\,h(G \dagger e, x)$, [45].

3.7 Covered Components Polynomial

Let G be a graph. A *covered component* of G is a component that contains at least one edge. A component that is not covered is an *isolated vertex*. We denote by $c(G)$ the number of covered components and by $\mathrm{iso}(G)$ the number of isolated vertices

of G, which yields the equation $c(G) + \mathrm{iso}(G) = k(G)$. The *covered components polynomial* [57] is an alternative representation of the *edge elimination polynomial* introduced in [10]; it is defined by

Definition

$$\boxed{C(G; x, y, z) = \sum_{F \subseteq E} x^{k(\langle F \rangle)} y^{|F|} z^{c(\langle F \rangle)}.} \tag{23}$$

By comparison of this definition with the definition of the spanning subgraph polynomial, we find

$$\mathcal{Z}(G; x, y) = C(G; x, y, 1). \tag{24}$$

The covered components polynomial satisfies the three-term recurrence relation [57],

Recurrence Relation

$$\boxed{\begin{aligned} C(G; x, y, z) = C(G - e; x, y, z) + y\, C(G/e; x, y, z) \\ + (xyz - xy) C(G \dagger e; x, y, z). \end{aligned}} \tag{25}$$

3.8 Extended Cut Polynomial

Let $G = (V, E)$ be a graph of order n and size m. The *extended cut polynomial* of G is defined as follows.

Definition

$$\boxed{J(G; x, z) = \sum_{W \subseteq V} x^{|W|} z^{|\partial W| + |E(W)|}.} \tag{26}$$

Since this graph polynomial might be new to the majority of readers, we discuss it here in more detail.

Lemma 1 *Let G be a graph with k components $G_1 = (V_1, E_1), \ldots, G_k = (V_k, E_k)$. Then*

$$\boxed{J(G; x, z) = \prod_{i=1}^{k} J(G_i; x, z).} \tag{27}$$

Proof The definition of the extended cut polynomial yields

$$\begin{aligned} J(G;x,z) &= \sum_{W\subseteq V} x^{|W|}z^{|\partial W|+|E(W)|} \\ &= \sum_{W\subseteq V} x^{|W\cap V_1|+\ldots+|W\cap V_k|}z^{|\partial W\cap E_1|+|E(W)\cap E_1|+\ldots+|\partial W\cap E_k|+|E(W)\cap E_k|} \\ &= \sum_{W_1\subseteq V_1} x^{|W_1|}z^{|\partial W_1|+|E(W_1)|}\ \cdots\ \sum_{W_k\subseteq V_k} x^{|W_k|}z^{|\partial W_k|+|E(W_k)|} \\ &= J(G_1;x,z)\ \cdots\ J(G_k;x,z), \end{aligned}$$

□

Theorem 2 *Let $G=(V,E)$ be a graph and $e=\{u,v\}\in E$. Then we have the*

Recurrence Relation

$$\boxed{J(G;x,z)=z\,J(G-e;x,z)+(1-z)J(G\circ e;x,z).}$$

Let $\mathbf{r}=(r_v)_{v\in V}$ be a sequence of nonnegative integers. The extended cut polynomial of the graph $E_n^{\mathbf{r}}$ that is obtained from the empty graph with vertex set V by attaching r_v loops to each vertex $v\in V$ is

$$\prod_{v\in V}(1+x\,z^{r_v}).$$

These properties uniquely determine the polynomial $J(G;x,y)$ for any graph G.

Proof Let L_r be a graph consisting of one vertex with r loops attached. Then we find

$$J(L_r;x,z)=1+xz^r \tag{28}$$

according to the definition of the extended cut polynomial. Lemma 1 gives together with Eq. (28)

$$J(E_n^{\mathbf{r}};x,z)=\prod_{v\in V}(1+x\,z^{r_v}).$$

Now we split the defining sum of the extended cut polynomial as follows:

$$\begin{aligned} J(G;x,z) &= \sum_{W\subseteq V} x^{|W|}z^{|\partial W|+|E(W)|} \\ &= \sum_{\substack{W\subseteq V\\ \{u,v\}\cap W\neq\emptyset}} x^{|W|}z^{|\partial W|+|E(W)|} + \sum_{W\subseteq V\setminus\{u,v\}} x^{|W|}z^{|\partial W|+|E(W)|} \end{aligned}$$

If $\{u, v\} \cap W \neq \emptyset$ then either $e \in \partial W$ or $e \in E(W)$, which yields

$$\sum_{\substack{W \subseteq V \\ \{u,v\} \cap W \neq \emptyset}} x^{|W|} z^{|\partial W| + |E(W)|} = z \sum_{\substack{W \subseteq V \\ \{u,v\} \cap W \neq \emptyset}} x^{|W|} z^{|\partial_{G-e} W| + |E_{G-e}(W)|}$$

and hence

$$\begin{aligned} J(G; x, z) &= z \sum_{W \subseteq V} x^{|W|} z^{|\partial_{G-e} W| + |E_{G-e}(W)|} - z \sum_{W \subseteq V \setminus \{u,v\}} x^{|W|} z^{|\partial W| + |E(W)|} \\ &\quad + \sum_{W \subseteq V \setminus \{u,v\}} x^{|W|} z^{|\partial W| + |E(W)|} \\ &= z J(G - e; x, z) - z \sum_{W \subseteq V \setminus \{u,v\}} x^{|W|} z^{|\partial W| + |E(W)|} \\ &\quad + \sum_{W \subseteq V \setminus \{u,v\}} x^{|W|} z^{|\partial W| + |E(W)|} \end{aligned}$$

For all $W \subseteq V \setminus \{u, v\}$, we have $\partial_G W = \partial_{G \circ e} W$ and $E_G(W) = E_{G \circ e}(W)$, which provides

$$\begin{aligned} J(G; x, z) &= z J(G - e; x, z) - z \sum_{W \subseteq V \setminus \{u,v\}} x^{|W|} z^{|\partial_{G \circ e} W| + |E_{G \circ e}(W)|} \\ &\quad + \sum_{W \subseteq V \setminus \{u,v\}} x^{|W|} z^{|\partial_{G \circ e} W| + |E_{G \circ e}(W)|} \\ &= z J(G - e; x, z) + (1 - z) J(G \circ e; x, z). \end{aligned}$$

□

Theorem 3 *The extended cut polynomial satisfies for each vertex $v \in V$ the*

Recurrence Relation

$$\boxed{J(G; x, z) = x z^{\deg v} J(G - v; x, z) + J(G \circ v; x, z).} \tag{29}$$

Note that the coefficient of $J(G - v; x, z)$ depends on the graph.

Proof Consider the polynomial

$$J(G; x, z) = \sum_{W \subseteq V} x^{|W|} z^{|\partial W| + |E(W)|}.$$

If a vertex $v \in V$ is included in W then all edges that are incident to v are counted with the exponent of z, which gives the factor $x z^{\deg v}$. Let ∂v be the set of edges of

G that have v as an end vertex. The polynomial J can be decomposed as follows:

$$\begin{aligned}\sum_{W\subseteq V} x^{|W|}z^{|\partial W|+|E(W)|} &= \sum_{W:v\in W\subseteq V} x^{|W|}z^{|\partial W|+|E(W)|} + \sum_{W\subseteq V\setminus\{v\}} x^{|W|}z^{|\partial W|+|E(W)|}\\ &= xz^{\deg v}\sum_{W:v\in W\subseteq V} x^{|W|-1}z^{|\partial W|+|E(W)|-\deg v}\\ &\quad+\sum_{W\subseteq V\setminus\{v\}} x^{|W|}z^{|\partial W|+|E(W)|}\\ &= xz^{\deg v}\sum_{W\subseteq V\setminus\{v\}} x^{|W|}z^{|\partial W\setminus\partial v|+|E(W)|-\deg v}\\ &\quad+\sum_{W\subseteq V\setminus\{v\}} x^{|W|}z^{|\partial W|+|E(W)|}\\ &= xz^{\deg v}J(G-v;x,z)+J(G\circ v;x,z)\end{aligned}$$

□

3.9 Edge Cover Polynomial

An *edge cover* of a graph G is an edge set F, $F\subseteq E(G)$, such that the set of all end vertices of edges in F is equal to $V(G)$. We denote by $\mathrm{Ends}(F)$ the set of all end vertices of edges from F.

Definition

$$\boxed{\Phi(G,x)=\sum_{F\subseteq E(G)}[\mathrm{Ends}(F)=V]x^{|F|}.} \tag{30}$$

The paper [3] provides an excellent introduction to edge cover polynomials. The edge cover polynomial satisfies the following recurrence relation for any edge $e=\{u,v\}$.

Recurrence Relation

$$\boxed{\begin{aligned}\Phi(G,x)&=(1+x)\Phi(G-e,x)+x\Phi(G-u,x)\\&\quad+x\Phi(G-v,x)+x\Phi(G\dagger e,x).\end{aligned}} \tag{31}$$

The proof is given in [3]. The edge cover polynomial of a graph can be calculated via inclusion–exclusion.

Lemma 2 ([3]) *The edge cover polynomial of a graph $G = (V, E)$ of size m satisfies*

$$\Phi(G, x) = \sum_{W \subseteq V} (-1)^{|W|} (1 + x)^{m - (|\partial W| + |E(W)|)}.$$

Theorem 4 *The edge cover polynomial of a graph G of size m with extended cut polynomial $J(G; x, z)$ is given by*

$$\Phi(G, x) = (1 + x)^m J\left(G; -1, \frac{1}{1 + x}\right).$$

Proof We substitute $x = -1$ and $z = \frac{1}{1+x}$ in the defining Eq. (26) of the extended cut polynomial, which yields

$$\sum_{W \subseteq V} (-1)^{|W|} \left(\frac{1}{1 + x}\right)^{|\partial W| + |E(W)|}.$$

Multiplication with $(1 + x)^m$ gives

$$\sum_{W \subseteq V} (-1)^{|W|} (1 + x)^{m - |\partial W| - |E(W)|}.$$

This is just the inclusion–exclusion representation of Φ shown in Lemma 2. □

Theorem 5 *Let G be a graph and v a vertex of G, then we have the*

Recurrence Relation

$$\boxed{\Phi(G, x) = \Phi(G \circ v, x) - \Phi(G - v, x).} \tag{32}$$

Proof Let m be the size of the graph G. We use Theorem 3,

$$J(G; x, z) = x z^{\deg v} J(G - v; x, z) + J(G \circ v; x, z).$$

Substitution of $x = -1$ and $z = \frac{1}{1+x}$ and multiplication with $(+x)^m$ yield

$$(1 + x)^m J\left(G; -1, \frac{1}{1 + x}\right) = (1 + x)^m J\left(G \circ v; -1, \frac{1}{1 + x}\right) - (1 + x)^{m - \deg v} J\left(G - v; -1, \frac{1}{1 + x}\right).$$

The graph $G \circ v$ has m edges, the graph $G - v$ has $m - \deg v$ edges. Consequently, the statement follows from Theorem 4. □

The following statement follows in the same way from Theorem 2.

Theorem 6 *The following relation is valid for any edge $e \in E(G)$,.*

Recurrence Relation

$$\boxed{\Phi(G, x) = \Phi(G - e, x) + x\Phi(G \circ e, x).} \tag{33}$$

The theorem is valid for loops too. However, some care is needed when the result of $G \circ e$ has an empty vertex set. In this case the edges still have to be processed. Let F_e be the set of all edges that are incident to an end vertex of e, except e itself. If the graph $G \circ e$ has an empty vertex set, then F_e is a set of loops. In this case, we define

$$\Phi(G \circ e, x) = (1 + x)^{|F_e|},$$

where each *free loop* contributes a factor $1 + x$ to the polynomial.

The edge cover polynomial is also an evaluation of the covered components polynomial.

Theorem 7 ([12]) *The edge cover polynomial can be obtained from the covered components polynomial of a graph by*

$$\Phi(G, y) = \lim_{x \to 0} C\left(G; x, y, \frac{1}{x}\right). \tag{34}$$

An almost immediate consequence of this theorem is the fact that the edge cover polynomial satisfies the following relation for any edge $e \in E(G)$.

Recurrence Relation

$$\boxed{\Phi(G, x) = \Phi(G - e, x) + x\Phi(G/e, x) + x\Phi(G \dagger e, x).} \tag{35}$$

3.10 Acyclic Polynomial and Feedback Polynomial

A *feedback vertex set* of a graph G is set $X \subseteq V(G)$ such that $G - X$ is an acyclic graph (a forest). The *acyclic polynomial* of a graph G is defined as follows.

Definition

$$\boxed{A(G, x) = \sum_{W \subseteq V(G)} [G[W] \text{ is acyclic}]\, x^{|W|}.} \tag{36}$$

We will call a vertex set $X \subseteq V(G)$ that induces an acyclic graph an *acyclic set*.

Remark 1 The acyclic polynomial has been introduced in [13]. It is the ordinary generating function for acyclic sets (complements of feedback vertex sets) of a graph. Observe that the *acyclic polynomial* was introduced earlier in a different context to name the signed matching polynomial (matching defect polynomial) of a graph.

We can define a second polynomial in this context, the *feedback polynomial* of a graph G is the ordinary generating function for feedback vertex sets of G,

Definition

$$\boxed{F(G,x) = \sum_{W \subseteq V(G)} [G[V(G) \setminus W] \text{ is acyclic}]\, x^{|W|}.}$$

This definition implies

$$A(G,x) = x^n F\left(G, \frac{1}{x}\right). \tag{37}$$

Theorem 8 *Let G be a (not necessarily simple) graph and v a vertex of G. Then,*

Recurrence Relation

$$\boxed{F(G,x) = x\, F(G-v,x) + F(G /\!\!/ v,x).} \tag{38}$$

Proof Let $\mathcal{F}(G)$ be the set of all feedback vertex sets of G. Then we have

$$\begin{aligned} F(G,x) &= \sum_{W \in \mathcal{F}(G)} x^{|W|} \\ &= \sum_{\substack{W \in \mathcal{F}(G) \\ v \in W}} x^{|W|} + \sum_{\substack{W \in \mathcal{F}(G) \\ v \notin W}} x^{|W|} \\ &= \sum_{U \in \mathcal{F}(G-v)} x^{|U \cup \{v\}|} + \sum_{\substack{W \in \mathcal{F}(G) \\ v \notin W}} x^{|W|} \\ &= x \sum_{U \in \mathcal{F}(G-v)} x^{|U|} + \sum_{\substack{W \in \mathcal{F}(G) \\ v \notin W}} x^{|W|}. \end{aligned}$$

The first sum is just the feedback polynomial of $G - v$. In the second sum we count all feedback vertex sets that do not include v. If we remove v from G, then we have to store all cycles that contain v. This can be performed by replacing any path of length 2 that has v as its central vertex by a direct edge between the end vertices of the path. This corresponds exactly to the resolution $G /\!\!/ v$ of v. □

The following result is obtained by combining Eqs. (37) and (38).

Recurrence Relation

$$\boxed{A(G,x) = A(G-v,x) + x\,A(G /\!\!/ v,x).} \tag{39}$$

3.11 Subgraph Component Polynomial

Let G be a simple graph. The *subgraph component polynomial* of G has been defined in [56].

Definition

$$\boxed{Q(G;x,y) = \sum_{W\subseteq V(G)} x^{|W|}y^{k(G[W])}.} \tag{40}$$

This definition works pretty well for non-simple graphs too. However, the subgraph component polynomial does not provide any information about loops or parallel edges. Hence we will assume that all graphs considered here are simple.

Theorem 9 *Let $G = (V, E)$ be a graph and $v \in V$. Then,*

Recurrence Relation

$$\boxed{Q(G;x,y) = Q(G-v;x,y) + xQ(G/v;x,y) + x(y-1)Q(G-N[v];x,y).} \tag{41}$$

Proof From Eq. (40), we obtain

$$\begin{aligned}
Q(G;x,y) &= \sum_{W\subseteq V} x^{|W|}y^{k(G[W])} \\
&= \sum_{W\subseteq V\setminus\{v\}} x^{|W|}y^{k(G[W])} + \sum_{W:v\in W\subseteq V} x^{|W|}y^{k(G[W])} \\
&= Q(G-v;x,y) + x\sum_{W\subseteq V\setminus\{v\}} x^{|W|}y^{k(G[W\cup\{v\}])} \\
&= Q(G-v;x,y) + x\sum_{\substack{W\subseteq V\setminus\{v\}\\ W\cap N(v)\neq\emptyset}} x^{|W|}y^{k(G[W])} + xy\sum_{W\subseteq V\setminus N[v]} x^{|W|}y^{k(G[W])} \\
&= Q(G-v;x,y) + x\sum_{\substack{W\subseteq V\setminus\{v\}\\ W\cap N(v)\neq\emptyset}} x^{|W|}y^{k(G[W])} + xyQ(G-N[v];x,y) \\
&= Q(G-v;x,y) + xyQ(G-N[v];x,y)
\end{aligned}$$

$$+x\left(\sum_{W\subseteq V\setminus\{v\}} x^{|W|}y^{k(G[W])} - \sum_{W\subseteq V\setminus N[v]} x^{|W|}y^{k(G[W])}\right)$$
$$= Q(G-v;x,y) + xQ(G;G/v;x,y)$$
$$+x(y-1)Q(G-N[v];x,y).$$

For the last equality, we used vertex completion that simulates the paths traversing the vertex v. □

3.12 Cycle Polynomial

Let G be a simple graph. The *cycle polynomial* of G is given es follows.

Definition

$$\boxed{\hat{\sigma}(G;x,y) = \sum_{F\subseteq E(G)} [\forall v \in V(G) : \deg_{\langle F\rangle} v \in \{0,2\}]x^{|F|}y^{c(V,F)}.} \tag{42}$$

We call a graph whose components are either isolated vertices or cycles a *multicycle*. Hence we can consider the cycle polynomial of a graph G as the ordinary generating function for spanning multicycles of G.

Theorem 10 ([22, 23]) *Let G be a simple graph and $e = \{u, v\}$ an edge of G, then we have the following result.*

Recurrence Relation

$$\boxed{\begin{aligned}\hat{\sigma}(G;x,y) = \hat{\sigma}(G-e;x,y) + x\big[&\hat{\sigma}(G/e;x,y) - \hat{\sigma}(G-u;x,y)\\ &-\hat{\sigma}(G-v;x,y) + \hat{\sigma}(G\dagger e;x,y)\big].\end{aligned}} \tag{43}$$

3.13 Domination Polynomial

We denote by $D(G,x)$ the *domination polynomial* of a graph G.

Definition

$$\boxed{D(G,x) = \sum_{W\subseteq V} [N[W] = V]x^{|W|}.} \tag{44}$$

This polynomial has been introduced in [6], further results can be found in [4, 5, 39, 43, 44]. The domination polynomial satisfies a rather complex recursive relation, for a proof see [43]]. The following relation is valid for any edge $e = \{u, v\} \in E(G)$.

Recurrence Relation

$$\begin{aligned} D(G,x) = \frac{x}{x-1}\,[&D(G-e/u,x) + D(G-e/v,x) \\ &- D(G/u,x) - D(G/v,x) - D(G-N[u],x) - D(G-N[v],x) \\ &+ D(G-e-N[u],x) + D(G-e-N[v],x)]\,. \end{aligned} \tag{45}$$

We present this relation to show that recurrence relations can have a surprising number of terms and different local graph operations for some graph polynomials. So far all attempts to find a simpler recursive relation for the domination polynomial failed.

3.14 Generalized Domination Polynomial

Let G be a graph and $A, B \subseteq V(G)$. The *generalized domination polynomial* is defined as follows.

Definition

$$\hat{D}(G, A, B; x) = \sum_{W \subseteq A} [B \subseteq N[W]] x^{|W|}. \tag{46}$$

In contrast to the (classical) domination polynomial, the generalized domination polynomial restricts the choice of the dominating set to subsets of A and accepts on the other hand also *partial dominating sets* that cover at least all vertices of B. The polynomial $\hat{D}(G, A, B; x)$ might be considered as a polynomial for *three-colored graphs*. The three colors of the vertices correspond to the sets A, B and $V(G) \setminus (A \cup B)$.

The following properties of the generalized domination polynomial are easily verified:

1. The domination polynomial can be obtained from the generalized domination polynomial through

$$D(G,x) = \hat{D}(G, V(G), V(G); x).$$

2. If the set to be covered is empty, then all subsets of A are dominating,

$$\hat{D}(G, A, \emptyset; x) = (1 + x)^{|A|}.$$

3. If A is empty, then there is no partial dominating set,

$$\hat{D}(G, \emptyset, B; x) = 0^{|B|}.$$

4. If there is a vertex $v \in B$ with $N[v] \cap A = \emptyset$ then $\hat{D}(G, A, B; x) = 0$.
5. If v is a vertex of G with $v \notin A$ and $v \notin B$, then

$$\hat{D}(G, A, B; x) = \hat{D}(G - v, A, B; x).$$

Theorem 11 *Let G be a graph, $A, B \subseteq V(G)$ and $v \in A$. Then,*

Recurrence Relation

$$\boxed{\hat{D}(G, A, B; x) = \hat{D}(G, A - v, B; x) + x\,\hat{D}(G - v, A - v, B - N[v]; x).} \tag{47}$$

Proof Let $B, W \subseteq V(G)$ and $v \in V(G)$. The relation $B \subseteq N[W \cup \{v\}]$ implies $B - N[v] \subseteq N[W]$. Conversely, if $B - N[v] \subseteq N[W]$ then ew have also $B \subseteq N[W \cup \{v\}]$, which yields

$$[B \subseteq N[W \cup \{v\}]] = [B - N[v] \subseteq N[W]]. \tag{48}$$

We partition the sum of the defining Eq. (46), apply Eq. (48), use the fifth above given property and obtain

$$\begin{aligned}
\hat{D}(G, A, B; x) &= \sum_{W \subseteq A} [B \subseteq N[W]] x^{|W|} \\
&= \sum_{W \subseteq A - v} [B \subseteq N[W]] x^{|W|} + \sum_{W : v \in W \subseteq A} [B \subseteq N[W]] x^{|W|} \\
&= \hat{D}(G, A - v, B; x) + x \sum_{W : W \subseteq A - v} [B \subseteq N[W \cup \{v\}]] x^{|W|} \\
&= \hat{D}(G, A - v, B; x) + x \sum_{W : W \subseteq A - v} [B - N[v] \subseteq N[W]] x^{|W|} \\
&= \hat{D}(G, A - v, B; x) + x \hat{D}(G - v, A - v, B - N[v]; x).
\end{aligned}$$

□

Theorem 11 provides the foundation for a recursive algorithm for the calculation of the domination polynomial of a graph. The theorem also shows that the use

of colored graphs can offer the way to simpler recursive relations for graph polynomials. (Just compare the complexity of the recurrences given in Eq. (45) and Theorem 11.)

3.15 Matching Polynomial

A *matching* of a graph $G = (V, E)$ is an edge set $F \subseteq E$ such that no two edges of F have a common end vertex in G. A loop of G never belongs to a matching of G. The *matching polynomial*, see [31], of G is defined as follows.

Definition

$$\boxed{M(G, x) = \sum_{F \subseteq E} [F \text{ is matching}] x^{|F|}.} \tag{49}$$

We can easily prove the following relation by case distinction depending on whether a given edge belongs to a matching.

Recurrence Relation

$$\boxed{M(G, x) = M(G - e, x) + xM(G \dagger e, x).} \tag{50}$$

This relation is valid for any edge $e \in E(G)$; together with the initial value $M(E_n; x) = 1$ for the empty graph it uniquely defines the matching polynomial of any graph G.

In a similar way, we obtain the following result, where we assume that G has no loops, $v \in V(G)$ and $\Gamma(v)$ the set of edges that are incident to v in G.

Recurrence Relation

$$\boxed{M(G, x) = M(G - v, x) + x \sum_{e \in \Gamma(v)} M(G \dagger e, x).} \tag{51}$$

3.16 Independence Polynomial

A vertex subset $W \subseteq V$ of a graph $G = (V, E)$ is *independent* if no two vertices of W are adjacent in G. The *independence polynomial* $I(G, x)$ of a graph G has been introduced in [38].

Definition

$$\boxed{I(G,x) = \sum_{W \subseteq V} [G[W] \text{ is empty}] x^{|W|}.} \tag{52}$$

It satisfies the following recursive relation.

Recurrence Relations

$$\boxed{I(G,x) = I(G-v,x) + xI(G-N[v],x) \text{ for any } v \in V(G).} \tag{53}$$

In addition, we have the following result which is shown in [40]. For any edge $e = \{u, v\} \in E(G)$, we have

$$\boxed{I(G,x) = I(G-e,x) - x^2 I(G-N(u)-N(v),x).} \tag{54}$$

3.17 Interlace Polynomial

Let G be a simple graph and A its adjacency matrix which we consider now as a matrix over $\mathbb{F}_2$ (GF(2)). We will denote the rank of A by $\operatorname{rk} A$ or $\operatorname{rk} G$ and we call this number the *vertex rank* of G. (It is also called the *2-rank* of G.) For basic properties of the vertex rank, see [35], Chapter 8. We can also define the vertex rank by [55],

$$\operatorname{rk} G = 2|V(G)| - \log_2 \left(\sum_{W \subseteq V} \prod_{v \in V(G)} \left[1 + (-1)^{|N(v) \cap W|} \right] \right). \tag{55}$$

We consider here only the *two-variable interlace polynomial* defined in [8], because its definition is more natural in the sense of generating functions.

Definition

$$\boxed{r(G;x,y) = \sum_{W \subseteq V(G)} (x-1)^{\operatorname{rk} G[W]} (y-1)^{|W| - \operatorname{rk} G[W]}} \tag{56}$$

This polynomial essentially depends on the number of induced subgraphs of a graph G for which the empty set cannot be represented as a nontrivial symmetric difference of distinct open neighborhoods of vertices. For more background on interlace polynomials, see [2, 7, 25, 48]. The following result has been proved in [8]. It is satisfied for each edge $e = \{u, v\} \in E(G)$.

Recurrence Relation

$$\boxed{\begin{aligned} r(G;x,y) &= r(G-u;x,y) + r(G^{uv}-v;x,y) \\ &\quad + (x^2-2x)r(G^{uv}-u-v;x,y). \end{aligned}} \tag{57}$$

Remark 2 We have renamed the polynomial from q, as it is denoted in [8], to r in order to avoid a notational conflict with previously defined polynomials. The classic one-variable interlace polynomial is just an evaluation of the here presented two-variable polynomial. If we use vertex-labeled graphs (bicolored graphs), then we can find a even simpler recurrence for the interlace polynomial [55]. The edge pivot operation G^{uv} used in the recurrence relation also can be expressed in terms of *local vertex complementations*, see [20].

4 Comparing Graph polynomials

We call two nonisomorphic graphs G and H *Tutte-equivalent* if they have the same Tutte polynomial, i.e. $T(G;x,y) = T(H;x,y)$. In the same way we define *independence-equivalent*, *domination-equivalent*, *edge-cover-equivalent* graphs if they share the independence, domination or edge cover polynomial and similarly for all other above introduced graph polynomials.

For any graph polynomial f, we can define a relation $\sim_f$ on the set $\mathcal{G}$ of all finite undirected graphs by

$$G \sim_f H \Leftrightarrow f(G) = f(H).$$

Then the set $\mathcal{G}/\sim_f$ of equivalence classes of $\mathcal{G}$ under f defines a partition on $\mathcal{G}$. We say for two graph polynomials f and g that f is *stronger or equal* g, denoted by $g \preccurlyeq f$, if $\mathcal{G}/\sim_f$ is a refinement of $\mathcal{G}/\sim_g$. For possible ramifications of this order relation, see [47].

For some polynomials defined in this paper, we find the relations

$$\begin{aligned} P, q, f, \mathcal{E} &\preccurlyeq \mathcal{Z} \preccurlyeq C, \\ \Phi, I &\preccurlyeq J, \\ I &\preccurlyeq Q, \\ M &\preccurlyeq C. \end{aligned}$$

The first chain of inequalities follows from known properties of the Tutte polynomial and the results given in [10], the second relation is just Theorem 4, the third one is presented in [56], the last one in [11].

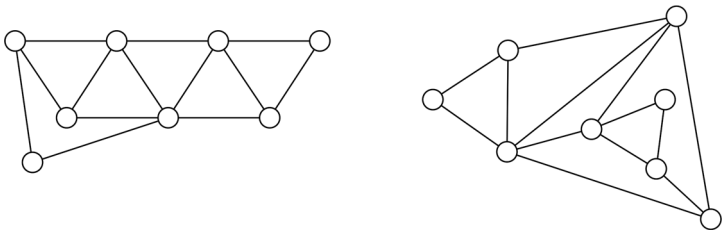


Fig. 3 Two nonisomorphic graphs with different domination polynomials that share the same independence, matching, and chromatic polynomial

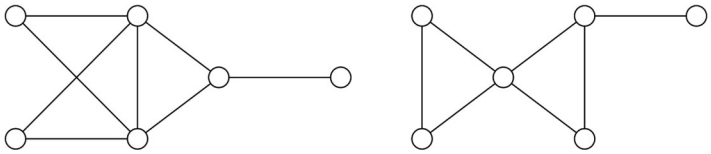


Fig. 4 Two nonisomorphic graphs with the same domination polynomial that have different independence, matching, and chromatic polynomials

However, these order relations are exceptional; most pairs of graph polynomials are incomparable. Figure 3 shows a pair of nonisomorphic graphs G and H with different domination polynomials that share the same independence, matching, and chromatic polynomial. The two graphs shown in Fig. 4 have the same domination polynomial, but different independence, matching, and chromatic polynomials.

We can show that the graph polynomials

$$\mathcal{Z}, J, A, Q, \hat{\sigma}, D, M, r$$

are pairwise incomparable. The proofs are given in the same way; we can give examples of graphs where one type of polynomials coincides whereas the two polynomials of the other type differ.

There is a second way of comparing graph polynomials. If we consider the recursions that a graph polynomial p satisfies, then we can identify a set p_{op} of local graph operations used in the recursion. The subgraph component polynomial Q uses the local graph operations vertex removal, neighborhood removal, and vertex completion or short the set $Q_{\mathrm{op}} = \{-v, /v, -N[v]\}$. For the edge cover polynomial Φ, we find four different sets $\Phi^1_{\mathrm{op}} = \{-e, -u, -v, \dagger e\}$, $\Phi^2_{\mathrm{op}} = \{-v, \circ v\}$, $\Phi^3_{\mathrm{op}} = \{-e, \circ e\}$ and $\Phi^4_{\mathrm{op}} = \{-e, /e, \dagger e\}$. We can define now an order relation for graph polynomials f and g by $f \trianglelefteq g \Leftrightarrow f_{\mathrm{op}} \subseteq g_{\mathrm{op}}$.

There are other interesting questions about order relations between graph polynomials which are beyond the scope of this paper, see [47].

5 Vertex-labeled Graphs

The aim of this section is to show that the introduction of graphs with vertex labels (or colored vertices) can reduce the number of terms and local graph operations needed in recurrence relations for graph polynomials.

The computation of graph polynomials can be (in some cases) simplified by the introduction of *vertex labels*. A vertex label is a map $\ell : V(G) \to L$ that assigns a label from a label set L to each vertex. In many application L will be a finite set of integers sometimes referred to as *colors*. The generalized domination polynomial introduced in Sect. 3.14 can be considered as a graph polynomial defined for vertex-labeled graphs. The label set is in this case $L = \{\alpha, \beta, \gamma\}$. Instead of a polynomial $\hat{D}(G, A, B; x)$ that uses four arguments we have now a polynomial $\check{D}(G_L, x)$ that takes a labeled graph G_L and a variable x as arguments. The vertex labels are defined by

$$\ell(v) := \begin{cases} \alpha, & \text{if } v \in A, \\ \beta, & \text{if } v \in B, \\ \gamma, & \text{if } v \in V(G) \setminus (A \cup B). \end{cases}$$

Consequently, the vertex labels provide just another way *of storing the information* about an *assumption* (or *condition*) for the further computation, in this case the membership of a vertex to a set of dominating vertices, dominated vertices, or to the set of vertices that are neither dominating nor dominated.

We can also use vertex labels to describe a more efficient method for the computation of the covered components polynomial. For this purpose, we use a label set wit just two labels, say

$$L = \{\text{black}, \text{white}\}.$$

We denote the set of all black vertices by B. We denote a labeled graph here by (G, B) or just by G if the vertex labels are known from the context. A component of a labeled graph (G, B) that contains a vertex from B is always considered being a covered component. In drawings of graphs, we present vertices belonging to B by blackened circles. In the following, we write just $C(G)$ instead of $C(G; x, y, z)$.

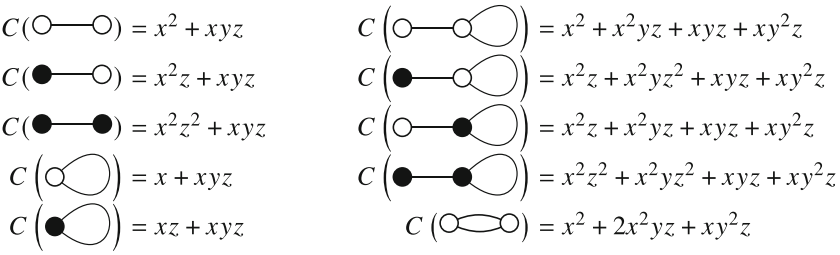
A single vertex (a complete graph K_1) has the covered components polynomial

$$C(\bigcirc) = xy. \tag{58}$$

If the vertex of the graph K_1 is labeled, then the result changes to

$$C(\bullet) = xyz. \tag{59}$$

First we collect the covered components polynomials for some small labeled graphs:



If the set of labeled vertices is empty, then we obtain the covered components polynomial for (unlabeled) graphs:

$$C((G, \emptyset)) = C(G) \tag{60}$$

If all vertices of a graph are labeled, then the covered components polynomial is equivalent to the spanning subgraph polynomial and hence to the Tutte polynomial:

$$C((G, V(G)); x, y, z) = \mathcal{Z}(G; y, xz). \tag{61}$$

Theorem 12 *Let (G, B) be a labeled graph and $e = \{u, v\}$ an edge of G. The covered components polynomial satisfies*

$$C((G, B)) = C((G - e, B)) + y\, C((G/e, B \cup \{w\})),$$

where w is the vertex obtained from merging u and v. This recurrence equation together with Eq. (60) and the initial conditions presented in Eqs. (58) and (59) determine the covered components polynomial uniquely.

Proof The covered components polynomial of a labeled graph (G, B) is given by

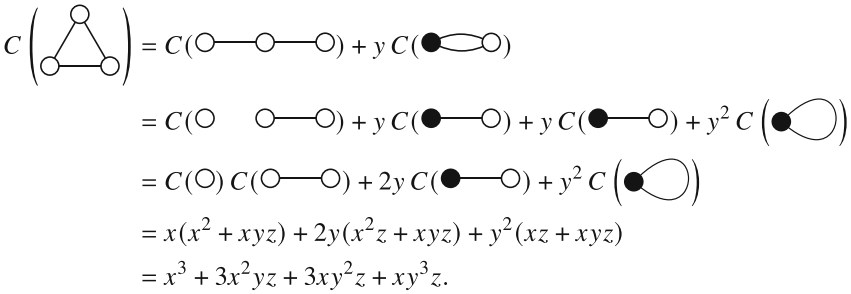
$$C((G, B)) = \sum_{F \subseteq E(G)} x^{k(V,F)} y^{[|F|]} z^{c(V,F)},$$

where $c(V, F)$ is now the sum of the number of covered components and the number of labeled isolated vertices of (V, F). We obtain

$$\begin{aligned} C((G, B)) &= \sum_{F \subseteq E(G) \setminus \{e\}} x^{k(V,F)} y^{[|F|]} z^{c(V,F)} + \sum_{F : e \in F \subseteq E(G)} x^{k(V,F)} y^{[|F|]} z^{c(V,F)} \\ &= C((G - e, B)) + y \sum_{F \subseteq E(G)} x^{k(V,F \cup \{e\})} y^{[|F|]} z^{c(V,F \cup \{e\})} \\ &= C((G - e, B)) + y\, C((G/e, B \cup \{w\})). \end{aligned}$$

□

As an example, we calculate the covered components polynomial of a complete graph K_3:



For the last but one step we used our table of results for small graphs.

The comparison of Eq. (25) and Theorem 12 shows that the introduction of vertex-labeled graphs leads to a reduction of the number of terms in the recursion. The third local graph operation, edge extraction, is no longer needed.

6 Parallel and Series Reductions via Multivariate Extensions

In this section, we introduce parallel and series reductions. These have been known for a long time in electrical networks, random cluster models and network reliability [26, 37, 51]. The derivation of series and parallel reductions for the Tutte polynomial is given in [52]. A *series–parallel graph* or short *sp-graph* is a graph G that can be reduced to an empty graph by a sequence of the following elementary operations:

- Contraction of a bridge in G.
- Replacement of a pair of parallel edges by single one.
- Replacement of a path P_3 with an internal vertex of degree 2 by an edge joining the end vertices of the path.

The three operations in this list can also be considered as local graph operations. We have not included them in the list presented in Sect. 2, because they do not apply to all vertices or edges of the graph. The operations are only applicable if additional conditions are met: an edge is bridge, there are parallel edges, there is a vertex of degree 2.

A characterization of sp-graph by matroids has been presented in [21]. Many computational NP-hard problems for graphs can be efficiently solved in the class of sp-graphs, [41, 51, 53]. A series-parallel graph has no minor isomorphic to K_4 and it is complete characterized by this property, see [27]. The efficient calculation of the spanning subgraph polynomial in sp-graphs can be performed through series and parallel reductions. The required information is stored in form of *edge weights*. Let

G be a graph and $w : E(G) \to K[\mathbf{x}]$ a weight function that assigns a polynomial with coefficients in a field K to each edge of G.

Remark 3 We will use not only polynomials, but also rational functions as edge weights. This could be avoided by multiplying by a polynomial that is the common denominator of all fractions. However, this approach complicates the presentation, we will use fractions of polynomials here. The final result is still a polynomial.

We write w_e for the weight of an edge e. Initially, we set $w_e = x$ for all $e \in E$. We will identify the edge set of G with the set of the first $|E(G)|$ natural numbers, i.e. $E(G) = \{1, \ldots, m\}$. Then the weight function is given by a vector $\mathbf{w} = (w_1, \ldots, w_m)$. The vector $\mathbf{w}_{-k} = (w_1, \ldots, w_{k-1}, w_{k+1}, \ldots, w_m)$ is obtained from $\mathbf{w}$ by deleting its k-th component. We define a multivariate graph polynomial recursively by

$$\hat{Z}(G; \mathbf{w}, y) = \begin{cases} y^n, \text{ if } G = E_n, \\ \hat{Z}(G - k; \mathbf{w}_{-k}, y) + w_k \, \hat{Z}(G/k; \mathbf{w}_{-k}, y) \text{ otherwise.} \end{cases} \tag{62}$$

Equation (62) implies that for any loop $e \in E$ the relation

$$\hat{Z}(G; \mathbf{w}, y) = (1 + w_e)\hat{Z}(G/e; \mathbf{w}_{-e}, y) \tag{63}$$

is satisfied. Comparing this definition with Eq. (6), we find

$$\mathcal{Z}(G; x, y) = \hat{Z}(G; (x, x, \ldots, x), y). \tag{64}$$

The multivariate version of the original definition of the spanning subgraph polynomial, presented in Eq. (5), is

$$\hat{Z}(G; \mathbf{w}, y) = \sum_{F \subseteq E} y^{k((V,F))} \prod_{e \in F} w_e. \tag{65}$$

Parallel Reduction Let G be a graph $u, v \in V(G)$ and $F_{uv} = \{e_1, \ldots, e_r\}$ a set of r ($r > 1$) parallel edges between u and v. We denote the weighted spanning subgraph polynomial of G just by $\hat{Z}(G)$ without the additional arguments. Our goal is to replace the r edges by a single one with a new weight in such a way that the spanning subgraph polynomial remains invariant. We denote the replacement edge by a and the new reduced graph by G'. The graph obtained from G by removing all r parallel edges is denoted by $G_{u|v}$; the graph G_{uv} denotes the graph that is obtained from $G_{u|v}$ by merging the two vertices u and v. From Eq. 62, we conclude

$$\hat{Z}(G') = \hat{Z}(G_{u|v}) + w_a \, \hat{Z}(G_{uv}). \tag{66}$$

We consider now the original graph G. If we contract at least one of the r parallel edges and remove all other parallel edges, then we obtain the graph G_{uv}. If remove

all r parallel edges, then we obtain $G_{u|v}$, which yields

$$\hat{Z}(G) = \hat{Z}(G_{u|v}) + \left[\prod_{e \in F_{uv}} (1 + w_e) - 1 \right] \hat{Z}(G_{uv}). \tag{67}$$

Comparing Eqs. (66) and (67) shows that

$$w_a = \prod_{e \in F_{uv}} (1 + w_e) - 1 \tag{68}$$

is the correct weight for the replacement edge a.

Series Reduction Let z be a vertex of degree 2 in G. The two neighbors of z are denoted by u and v. There are exactly two edges, denoted by $e = \{u, z\}$ and $f = \{v, z\}$, incident to z in G. We replace z and its incident edges by a new edge $a = \{u, v\}$, which results in a reduced graph G'. The replacement graph G' obviously satisfies the Eq. (66). From Eq. (62), we conclude

$$\hat{Z}(G) = w_e w_f \, \hat{Z}(G_{uv}) + (w_e + w_f + y) \hat{Z}(G_{u|v}). \tag{69}$$

Equating Eqs. (66) and (69) yields

$$w_a \, \hat{Z}(G_{uv}) + \hat{Z}(G_{u|v}) = w_e w_f \, \hat{Z}(G_{uv}) + (w_e + w_f + y) \hat{Z}(G_{u|v}).$$

The only unknown parameter in this equation is the new edge weight w_a. The solution has to be independent of the two polynomials $\hat{Z}(G_{uv})$ and $\hat{Z}(G_{u|v})$, which results in two equations, $w_a = w_e w_f$ and $1 = w_e + w_f + y$. The last equation violates the independent choice of the edge weights. We can overcome this problem with a weaker requirement to our reduction. We introduce a reduction factor h such that $\hat{Z}(G) = h \, \hat{Z}(G')$ can be used as reduction approach, which yields after substitution of Eqs. (66) and (69) the equality

$$h \, w_a \, \hat{Z}(G_{uv}) + h \, \hat{Z}(G_{u|v}) = w_e w_f \, \hat{Z}(G_{uv}) + (w_e + w_f + y) \hat{Z}(G_{u|v}).$$

By comparing the coefficients of the polynomials $\hat{Z}(G_{uv})$ and $\hat{Z}(G_{u|v})$, we obtain the desired edge weight,

$$w_a = \frac{w_e w_f}{w_e + w_f + y}, \tag{70}$$

as the solution of a system of linear equations.

7 Related Topics and Open Problems

We have shown that local graph operations used in reductions and recursions are powerful tools for computing graph polynomials. In some cases, it may be useful to work with edge- or vertex-weighted graphs to reduce the number of terms of a recursion or to obtain new reductions.

Problem 1 The list of local graph operations presented in Sect. 2 is not complete. What other local graph operations are useful for graph polynomials? What do we gain if we allow to modify the graph within a certain distance from a given vertex?

Problem 2 Sometimes relations between graph polynomials are "hidden." For example, we can derive a polynomial of a given graph from a graph polynomial of its complement or its line graph. What other relations between known graph polynomials can we discover in this way?

Universality of Graph Polynomials There are some graph polynomials which have the nice property of being a *universal polynomial* of graphs. A graph polynomial p is universal if any other graph polynomial that satisfies the same recursive relation as p (and perhaps some additional conditions) can be obtained from p by variable substitution and multiplication by some factor. The spanning subgraph (Tutte) polynomial, the covered components (edge elimination) polynomial, the subgraph component polynomial, and the extended cut polynomial are universal [9, 12, 25, 28, 55, 56]. An open question is to find a general strategy for obtaining a list of properties that guarantees universality, such as satisfying an edge deletion-contraction relation and being multiplicative with respect to components.

Problem 3 What other local graph operations lead to universal graph polynomials?

Recursive Relations and Complexity Since the computation of each graph polynomial presented in this paper is an NP-hard problem (often #P-complete, [36]), the algorithmic application of recurrence relations for the computation of the polynomial cannot provide an efficient way to its computation. However, often we can get *better exponential-time algorithms* by a proper selection of the edges and/or vertices used for local graph operations, together with a suitable set of initial graphs (terminal conditions for the recursion). In this way, we obtain for the chromatic or reliability polynomial an algorithm that needs $t(G)$ (number of spanning trees) instead of 2^m steps. For the independence polynomial, we can find an algorithm with time complexity $O(1.4^n)$ instead of $O(2^n)$ [54]. Reducing complexity becomes a real challenge when many different local graph operations, reductions and other methods are combined.

Problem 4 How can we find better exponential-time algorithms for graph polynomials?

Universal Versus Special Reductions The local graph operations introduced in Sect. 2 and applied in Sect. 3 are *universal* in the sense that they can be used for *any*

vertex or edge of the graph. For vertices or edges with special properties, additional reductions may be possible. Consider, as an example, a *dominating vertex* (a vertex that is adjacent to any other vertex). It is not a hard exercise to prove that for the domination, independence, and chromatic polynomial the following relations are valid for any dominating vertex v:

$$D(G, x) = x(1 + x)^{n-1} + D(G - v, x),$$
$$I(G, x) = x + I(G - v, x),$$
$$P(G, x) = xP(G - v, x - 1).$$

Further reductions can be obtained for vertices of small degree, simplicial vertices, bridges or loops. Reductions for special vertices and/or edges often result in polynomial-time algorithms for graph polynomials of special graph classes like threshold graphs, co-graphs, sp-graphs or chordal graphs.

Problem 5 An interesting open question is what other local graph operations can be used to derive reductions for graph polynomials? In which case and for which class of graphs can we obtain a polynomial-time algorithm?

Problem 6 When we use several different local graph operations in an algorithm, the computational complexity can depend on the order of the operations. How can we find the optimum order of the local graph operations?

Sum Formulae and Recursive Definitions All above listed graph polynomials can be defined by a sum ranging over vertex subsets, edge subsets, or partitions of the vertex set. Alternatively, we can define each graph polynomial by a recurrence relation together with initial conditions. It seems natural that we can use the recursive definition to compute the graph polynomial by a *state distinction* or by a *decomposition tree*, so that we end up in a summation process that yields a sum formula (not necessarily over edge or vertex subsets, but more general *states*). For more details, see [32].

Starting from a definition of a graph polynomial as a generating function (a sum formula), it seems to be in general a more difficult task to split the sum in way that allows an interpretation of the resulting polynomials by some local graph operations. In some cases, we can at least show that a certain type of linear recurrence relation *does not exist*. For an example, see [43].

Problem 7 Is there a general way to find suitable local graph operations and a recursive relation for a graph polynomial with a given subset representation?

Toplogical Properties of Graphs Extracting of topological properties of graphs, like planarity, linkless embeddability, and genus, from a graph polynomial seems to be a widely open problem. Most attempts to deal with topological properties focus on generalizing graphs to embedded graphs, ribbon graphs, delta matroids or other more complex objects [18, 24, 29]. However, there seems to be no fundamental

obstacle to a polynomial defined on undirected graphs containing the information necessary to derive all the topological properties of interest.

Problem 8 Can we find local graph operations that allow a topological interpretation? Can we find a graph polynomial with a fixed number of variables that provides interesting topological information? Is there a way to extract more topological properties of graphs from existing graph polynomials?

Problem 9 This problem is related to the previous problem. Let $\mathcal{H}$ be a given finite set of (small) graphs. Can we define a graph polynomial $f(G)$ that gives for each $H \in \mathcal{H}$ the number of occurrences of H as a minor of G?

Acknowledgments The author would like to thank János (Johann A. Makowsky) for his hospitality, support and many interesting discussions in Haifa, Zurich, Dagstuhl and some other places. I am also very grateful to the reviewer of the first draft of this paper for her careful reading and many valuable suggestions for improving the exposition.

References

1. Aigner, M.: A Course in Enumeration. Springer-Verlag, Berlin (2007)
2. Aigner, M., Van der Holst, H.: Interlace polynomials. Linear Algebra and its Applications, vol. 377, pp. 11–30. Elsevier (2004)
3. Akbari, S., Oboudi, M.R.: On the edge cover polynomial of a graph. Eur. J. Comb. **34**(2), 297–321 (2013)
4. Akbari, S., Alikhani, S., Oboudi, M.R., Peng, Y.-H.: On the zeros of domination polynomial of a graph. Comb. Graphs **531**, 109–115 (2010)
5. Alikhani, S., Peng, Y.-H.: Introduction to domination polynomial of a graph. Preprint (2009). arXiv:0905.2251
6. Arocha, J.L., Llano, B.: Mean value for the matching and dominating polynomials. Discussiones Mathematicae Graph Theory **20**, 57–69 (2000)
7. Arratia, R., Bollobás, B., Sorkin, G.B.: The Interlace Polynomial: A New Graph Polynomial. IBM Thomas J. Watson Research Division, New York (2000)
8. Arratia, R., Bollobás, B., Sorkin, G.B.: A two-variable interlace polynomial. Combinatorica **24**, 567–584 (2004)
9. Averbouch, I.: Completeness and Universality Properties of Graph Invariants and Graph Polynomials. Ph.D. Thesis. Technion - Israel Institute of Technology, Haifa, Israel, 2011
10. Averbouch, I., Godlin, B., Makowsky, J.A.: A most general edge elimination polynomial. International Workshop on Graph-Theoretic Concepts in Computer Science, pp. 31–42. Springer, Berlin (2008)
11. Averbouch, I., Godlin, B., Makowsky, J.A.: An extension of the bivariate chromatic polynomial. Eur. J. Comb. **31**(1), 1–17 (2010)
12. Averbouch, I., Kotek, T., Makowsky, J.A., Ravve, E.: The universal edge elimination polynomial and the dichromatic polynomial. Electronic Notes Discrete Math. **38**, 77–82 (2011)
13. Barton, C., Brown, J.I., Pike, D.A.: Acyclic polynomials of graphs. Aust. J. Comb. **82**(2), 146–181 (2022)
14. Biggs, N.: Algebraic Graph Theory, vol. 67. Cambridge University Press, Cambridge (1993)
15. Birkhoff, G.D.: A determinant formula for the number of ways of coloring a map. Ann. Math. **14**(1/4), 42–46 (1912)
16. Bollobás, B.: Modern Graph Theory, vol. 184. Springer, Berlin (2013)

17. Bollobás, B., Riordan, O.: Polychromatic polynomials. Discrete Math. **219**(1–3), 1–7 (2000)
18. Bollobás, B., Riordan, O.: A polynomial invariant of graphs on orientable surfaces. Proc. Lond. Math. Soc. **83**(3), 513–531 (2001)
19. Bondy, J.A., Murty, U.S.R.: Graph Theory. Springer, Berlin (2008)
20. Bouchet, A.: Graph polynomials derived from Tutte–Martin polynomials. Discrete Math. **302**(1–3), 32–38 (2005)
21. Brylawski, T.H.: A combinatorial model for series-parallel networks. Trans. Am. Math. Soc. **154**, 1–22 (1971)
22. Chen, X.: Digraph Polynomials for Counting Cycles and Paths. Preprint (2017). arXiv:1712.00686
23. Chen, X.: Polynomials for Counting of Cycles and Paths. Bachelor Thesis. Mittweida University of Applied Sciences, 2017
24. Chun, C., Moffatt, I., Noble, S.D., Rueckriemen, R.: Matroids, delta-matroids and embedded graphs. J. Comb. Theory Series A **167**, 7–59 (2019)
25. Courcelle, B.: A multivariate interlace polynomial and its computation for graphs of bounded clique-width. Electron. J. Comb. R69–R69 (2008)
26. Doyle, P.G., Laurie Snell, J.: Random Walks and Electric Networks, vol. 22. American Mathematical Society, Providence (1984)
27. Duffin, R.J.: Topology of series-parallel networks. J. Math. Anal. Appl. **10**(2), 303–318 (1965)
28. Ellis-Monaghan, J.A., Merino, C.: Graph polynomials and their applications I: The Tutte polynomial. In: Structural Analysis of Complex Networks, pp. 219–255. Springer, Berlin (2011)
29. Ellis-Monaghan, J.A., Moffatt, I.: Graphs on Surfaces: Dualities, Polynomials, and Knots, vol. 84. Springer, Berlin (2013)
30. Ellis-Monaghan, J.A., Moffatt, I.: Handbook of the Tutte Polynomial and Related Topics. CRC Press, Boca Raton (2022)
31. Farrell, E.J.: An introduction to matching polynomials. J. Comb. Theory Series B **27**(1), 75–86 (1979)
32. Fischer, E., Makowsky, J.A.: Linear recurrence relations for graph polynomials. Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday, pp. 266–279 (2008)
33. Fortuin, C.M., Kasteleyn, P.W.: On the random-cluster model: I. Introduction and relation to other models. Physica **57**(4), 536–564 (1972)
34. Godlin, B., Katz, E., Makowsky, J.A.: Graph polynomials: From recursive definitions to subset expansion formulas. J. Logic Comput **22**(2), 237–265 (2012)
35. Godsil, C., Royle, G.F.: Algebraic Graph Theory, vol. 207. Springer Science & Business Media, Berlin (2013)
36. Goodall, A., Hermann, M., Kotek, T., Makowsky, J.A., Noble, S.D.: On the complexity of generalized chromatic polynomials. Adv. Appl. Math. **94**, 71–102 (2018)
37. Grimmett, G.: The Random-cluster Model, vol. 333. Springer, Berlin (2006)
38. Gutman, I., Harary, F.: Generalizations of the matching polynomial. Utilitas Math **24**(1), 97–106 (1983)
39. Heinrich, I., Tittmann, P.: Neighborhood and domination polynomials of graphs. Graphs Comb. **34**(6), 1203–1216 (2018)
40. Hoede, C., Li, X.: Clique polynomials and independent set polynomials of graphs. Discrete Math. **125**(1–3), 219–228 (1994)
41. Kikuno, T., Yoshida, N., Kakuda, Y.: A linear algorithm for the domination number of a series-parallel graph. Discrete Appl. Math. **5**(3), 299–311 (1983)
42. Kotek, T., Makowsky, J.A., Rave, E.V.: On sequences of polynomials arising from graph invariants. Eur. J. Comb. **67**, 181–198 (2018)
43. Kotek, T., Preen, J., Simon, F., Tittmann, P., Trinks, M.: Recurrence relations and splitting formulas for the domination polynomial. Electron. J. Comb. **19**(3), P47 (2012)
44. Kotek, T., Preen, J., Tittmann, P.: Subset-sum representations of domination polynomials. Graphs Comb. **30**(3), 647–660 (2014)

45. Liu, R.-Y., Zhao, L.-C.: A new method for proving chromatic uniqueness of graphs. Discrete Math. **171**(1–3), 169–177 (1997)
46. Makowsky, J.A.: From a zoo to a zoology: towards a general theory of graph polynomials. Theory Comput. Syst. **43**(3–4), 542–562 (2008)
47. Makowsky, J.A., Ravve, E.V., Kotek, T.: A logician's view of graph polynomials. Ann. Pure Appl. Logic **170**(9), 1030–1069 (2019)
48. Morse, A.: The interlace polynomial. Graph Polynomials. Chapman and Hall/CRC, Boca Raton (2016)
49. Negami, S.: Polynomial invariants of graphs. Trans. Am. Math. Soc. **299**(2), 601–622 (1987)
50. Read, R.C.: An introduction to chromatic polynomials. J. Comb. Theory **4**(1), 52–71 (1968)
51. Satyanarayana, A., Wood, R.K.: A linear-time algorithm for computing K-terminal reliability in series-parallel networks. SIAM J. Comput. **14**(4), 818–832 (1985)
52. Sokal, A.D., Webb, B.S.: The multivariate Tutte polynomial (alias Potts model). Surveys Comb. **327**(2005), 173 (2005)
53. Takamizawa, K., Nishizeki, T., Saito, N.: Linear-time computability of combinatorial problems on series-parallel graphs. J. ACM **29**(3), 623–641 (1982)
54. Tarjan, R.E., Trojanowski, A.E.: Finding a maximum independent set. SIAM J. Comput. **6**(3), 537–546 (1977)
55. Tittmann, P.: Graph Polynomials: The Eternal Book. ResearchGate, Berlin (2024)
56. Tittmann, P., Averbouch, I., Makowsky, J.A.: The enumeration of vertex induced subgraphs with respect to the number of components. Eur. J. Comb. **32**(7), 954–974 (2011)
57. Trinks, M.: The covered components polynomial: a new representation of the edge elimination polynomial. Electron. J. Comb. **19**(#50) (2012)
58. Tutte, W.T.: A contribution to the theory of chromatic polynomials. Can. J. Math. **6**, 80–91 (1954)
59. Tutte, W.T.: Graph Theory as I Have Known it. Clarendon Press, Oxford (1998)
60. Whitney, H.: A logical expansion in mathematics. Bull. Am. Math. Soc. **38**(8), 572–579 (1932)