# Model Theory in Computer Science: My Own Recurrent Themes

**Johann A. Makowsky**

**Technion–Israel Institute of Technology**
**Haifa, Israel**
janos@cs.technion.ac.il

**Abstract**

I review my own experiences in research and the management of science.

**1998 ACM Subject Classification** F.4.1 Mathematical Logic

**Keywords and phrases** Model theory, finite model theory, databases, graph invariants

**Digital Object Identifier** 10.4230/LIPIcs.CSL.2011.553

## Forty Years Ago

It was exactly forty years ago, that I got my ETH-Diploma in Mathematics and Physics, followed two years later by my PhD (Dr.sc.math.). Since then I have held appointments in Zurich, Warsaw, Stanford, Vancouver, Florence, Berlin and Haifa. Scientifically, I have travelled from Model Theory proper to applications thereof in Computer Science, and finally in Combinatorics, visiting the lands of Database Theory, Specification, Verification, Artificial Intelligence, Complexity and Algorithms. I was a founding member of EACSL, its vice-president, and finally its president till 2010. But at heart I remained a mathematician with a strong interest in computer science and its foundations.

In this *retiring president's address* I would like to sketch some of the recurrent ideas of my research, and some of the lessons I have learned in managing a scientific career, and managing science as an enterprise. I concentrate here on scientific details and leave more personal remarks for the lecture. Some other personal recollections can be found in [62].

## Model Theory: Categoricity and Finite Axiomatizability

My first attempt to tackle open problems was a consequence of reading M. Morley's fundamental paper on categoricity in power, [72] in the undergraduate seminar in mathematical logic at ETH Zurich, held by E. Specker and H. Läuchli, and regularly attended by the still very lucid octogenarian P. Bernays.

A first-order theory $T$ is categorical in some infinite cardinal $\kappa$ if $T$ has no finite models and all its models of size $\kappa$ are isomorphic. Morley asks, whether there is a finitely axiomatizable first-order theory $T$ which is $\kappa$-categorical for all $\kappa$, or for all uncountable $\kappa$. Attacking these questions required understanding of the structure theory of $\kappa$-categorical theories (stable theories, rank, degree, etc.) and some idea on how to prove or disprove finite axiomatizability. I made a thorough manual literature search in the library (no `scholar.google.com` was available then) about finite axiomatizability, from which I learned about Ehrenfeucht-Fraïssé games and ultraproducts, and other methods, but only the Ehrenfeucht-Fraïssé games seemed promising to me. With some ideas on how to approach Morley's question, I attended my first logic conference in 1970, where I received encouragement and a still unpublished preprint of


Computer Science Logic 2011 (CSL'11).
Editor: Marc Bezem; pp. 553–567
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

[3] from A. Lachlan. Upon my return I asked my supervisor, H. Läuchli, whether I could write my M.Sc. thesis about Morley's question, and he agreed. I managed to prove

▶ **Theorem 1** ([53]).
*(i) A first-order theory $T$ which is $\aleph_0$-categorical and strongly minimal (hence categorical in all infinite kappa) cannot be finitely axiomatizable.*
*(ii) There is a finitely axiomatizable complete first-order theory $T$ which is superstable.*

The second edition of [10] credits me[1] with partially solving two of Morley's problems. However, most of the problems listed in [72] were solved by S. Shelah, just the finite axiomatizability questions withstood his attacks. I soon realized that this was all I could prove using the available tools. Besides Ehrenfeucht-Fraïssé games I used non-periodic tilings of the plane, an idea which was suggested to me by my supervisor H. Läuchli.

The finite axiomatizability questions were finally solved by by M. Peretyatkin [76] (there is a finitely axiomatizable $\omega_1$-categorical theory), and by G. Cherlin, L. Harrington and A. Lachlan and by B. Zilber (there is no finitely axiomatizable theory categorical in all infinite powers), cf. [11], and [83]. G. Cherlin, L. Harrington and A. Lachlan use the classification theory of finite groups, and B. Zilber uses results on diophantine equations to overcome the difficulties I had not been able to overcome. Neither of these tools were available when I left the problem.

I presented my results on finite axiomatizability at the Logic Colloquium in Cambridge in 1971. There I met for the first time with S. Shelah, with whom I had corresponded before, and with W. Marek, who told me about student exchange programs between Switzerland and Poland. Both encounters had a major impact on my further scientific development.

▶ Lesson 1. Search and read the literature, even if it goes far back.

▶ Lesson 2. Go to conferences already as a student, but be properly prepared.

## Generalized Quantifiers

It was Wiktor Marek, who introduced me in 1971 to Lindström's Theorem. It had been rediscovered by Harvey Friedman, who gave it much publicity. What a great Theorem: Predicate Logic can be characterized, among *all the logics* as the only one which satisfies the Löwenheim-Skolem Property and one of the following: compactness or axiomatizability. Well, that's the way you might promote it, but then there are plenty of details, which make it less spectacular. And still, a new paradigm was found, which consisted in characterizing logics. I immediately studied Lindström's papers and all that was known about extensions of first-order logic and prepared a seminar talk about it. Later P. Lindström told me that his original motivation for the theorem had been to find a new application of Ehrenfeucht-Fraïssé games. I spent 1972 partially in Warsaw as an exchange student, cf. [62] working under the late A. Mostowski on generalized quantifiers.

There were two lines of studying extensions of first-order logic: (i) via generalized quantifiers, and (ii) via fragments of infinitary logics. What I tried to do was to find characterizations of logics using other properties than the Löwenheim-Skolem-Tarski Theorem and the Compactness Theorem. J. Barwise showed that the admissible fragments of $\mathcal{L}_{\omega_1,\omega}$ satisfy the Craig Interpolation Theorem. D. Scott and, independently before, E. Engeler,

[1] At the time joint papers were not common practice, and H. Läuchli let me publish the results under my name alone. In [10] credit is not extended to H. Läuchli, although I clearly stated his rôle in the results.

[21, 22, 78] had shown that all countable structures over a countable vocabulary can be characterized up to isomorphism by a sentence in $\mathcal{L}_{\omega_1,\omega}$. G. Kreisel suggested his own criteria of choosing logics, [44]. In Fall 1972, E. Engeler and P. Bernays introduced me to G. Kreisel. I told him about my ideas, and he got very interested and encouraging. An intensive correspondence followed which lasted until he invited me in Fall 1973 to come to Stanford. He also provided me with a preprint of L. Tharp, published later as [81], without telling me that he was refereeing it. I naively used the material in my PhD thesis, trusting that it was given to me in good faith for use. I generalized Tharp's definitions and proved innocently theorems which may have been also on Tharp's mind.

▶ Lesson 3. Do not circulate papers you are supposed to treat confidentially.

Having worked on categorical first-order theories, I formulated and finally proved the following:

▶ **Theorem 2** ([52]). *Let $\mathcal{L}$ be a logic such that Craig's Interpolation Theorem holds for $\mathcal{L}$ and such that all countable structures over a countable vocabulary can be characterized up to isomorphism by a sentence in $\mathcal{L}$. Then $\mathcal{L}_{\omega_1,\omega}$ is a sublogic of $\mathcal{L}$.*

For this theorem I was inspiried by three papers by S. Feferman, [23, 24, 25], which among other things discuss abstract versions of the Feferman-Vaught Theorem, [26] which entered my toolbox already then.

In [52] I also announced characterization of the minimal fragments of $\mathcal{L}_{\omega_1,\omega}$ which satisfy the Sousline-Kleene version of the interpolation theorem and characterize one countable structure up to isomorphism. The same characterization was also announced by H. Friedman in the Notices of AMS. My proof used tools I had not yet properly mastered by then. However, H. Friedman's announcement brushed away my gut feelings, and made me believe that I had used the tools correctly. Only when I lectured in the Stanford Logic Seminar in fall 1973, J. Stavi showed me a counterexample to the theorem as stated in [52], and in Friedman's abstract. Together we fixed the theorem, which led to [67] and my prolonged collaboration with S. Shelah in abstract model theory.

▶ Lesson 4. Do not trust your own handwaving. Do trust your gut feelings when something is wrong with your proof.

Until 1984 much of my published work remained in abstract model theory. Abstract model theory had generated quite a bit of excitement in the logic community. This is witnessed in [5], where I contributed two chapters and co-authored one more, [57, 64, 56]. But after the publication of [5] the philosophically minded, including P. Lindström himself, lost interest, because, among other reasons, my theorems with S. Shelah had introduced large cardinals into the field, spoiling the hope for neat theorems.

S. Shelah taught me:

▶ Lesson 5. Never let aesthetics or ideology prevent you from proving a theorem!

But the reality of the mathematical community knows a law of diminishing return:

▶ Lesson 6. Not everybody who asks a mathematical question is willing to hear the answer if it requires too much time, energy or skill to understand it.

J. Stavi and S. Shelah invited me to Israel to work with them. Later romantic involvement strengthened the Israeli connection. Finally, I founded a family and stayed. But to find a job in Israel, I had to move to applications of logic in computer science.

## The Promised Land: Computer Science Logic

I was well aware that mathematical logic, especially model theory, had something to offer to theoretical computer science. I had attended the Specker-Strassen Seminar in Zurich in the early 70s, cf. [80], where we studied evolving complexity theory. It was finally E. Shamir in Jerusalem who gave me the crucial impulse to approach computer science successfully. In 1978 he arranged for a "blind date" with C. Beeri, who was struggling to find the right definition of database dependencies. He also told me to attend the ACM-STOC conference in 1979 in Atlanta, where I got acquainted with V. Pratt and his dynamic logic.

▶ Lesson 7. When you change fields, do not leave your old toolbox behind!

I tried to identify problems in theoretical computer science which could be tackled using model theoretic methods. I was looking for model theoretic characterizations of certain classes of syntactically defined formulas and for analogues of Lindström's Theorems. At the Logic Colloquium 1982 in Florence I was an invited speaker and I gave a talk on *Model theoretic issues in theoretical computer science: Relational Data Bases and Abstract Data Types*, [55]. Y. Gurevich discussed this paper with me at great length in the years 1982-84, and it inspired him to write his [34]. But I had written my paper for the wrong audience: The Logicians were not interested in Computer Science, and the first LiCS conference was held only in 1986. The first CSL conference was held in 1987, and EACSL was founded in 1992.

▶ Lesson 8. One can be too early and in the wrong place at the same time.

## The Fundamental Problem of Databases

My first published paper in theoretical computer science was an application of abstract model theory to dynamic logic, [48]. But my truly first result was in databases. In 1978 C. Beeri spent many hours trying to explain to me what J. Ullman had declared to be the *Fundamental Problem of Databases*. Imprecisely stated, it was the decision problem for database dependencies which at the time were meant to be generalizations of Functional Dependencies. Finally, C. Beeri accepted my suggestion, that database dependencies are to be identified with a certain subclass of universal-existential Horn formulas in purely relational first-order logic where satisfiability is restricted to finite relational structures. Actually, we defined the four classes of dependencies which later became known as FID (full implicational) and EID (embedded implicational) with the subclasses of equality generating and tuple generating dependencies. I showed C. Beeri, that in the case of EID's, the decision problem was undecidable, and suspected it to be well known. A quick consultation with M. Rabin confirmed my suspicion, although, as it turned out, M. Rabin was not quite right. My reduction used the word problem of finite semigroups. Rabin thought that this was known to be undecidable, and undecidable it was, but not well known. It had been proven by Y. Gurevich in 1966 and published only in Russian as [33]. Rabin did not know of it and confused it with some other well known decision problem. As a result of Rabin's remarks I turned my back to databases and looked for other topics. I did not realize then that, even if I could solve the technical problem, I still did not understand why solving it was important for databases. C. Beeri continued to work on this with his PhD student M. Vardi, and much of what we discussed together was further elaborated in Vardi's thesis. I tried my luck, without success, in improving Galil's lower bound for the worst case run time of the Davis-Putnam procedure, [27]. I also tried my luck, again without success, in understanding the complexity of computing the permanent. I returned to both of these topics much later.

In spring 1980 I was guest of V. Pratt at MIT. Visiting Princeton, I attended a colloquium lecture where Ullman's *Fundamental Problem of Databases* was mentioned again as the most important open problem in database theory. I told the speaker that I had solved it a year before, and he mentioned rumors that A. Chandra and H. Lewis also just solved it. Back at MIT, I showed C. Papadimitriou my proof and asked him about the rumor that A. Chandra and H. Lewis had obtained the same result. He confirmed and was kind enough to arrange that this coincidence would result in two joint papers, [9, 8].

▶ **Theorem 3** (A. Chandra, H. Lewis and JAM, [9, 8]).
*(i)* *The decision problem for* embedded implicational dependencies *is undecidable.*
*(ii)* *The decision problem for* full implicational dependencies *decidable and complete in exponential time, even for the typed case.*

The distinction between typed and untyped dependencies seemed to me cosmetic but was considered important to the database community. Typed here means that we look at many-sorted finite structures where sorts correspond to attributes. The undecidability of *typed* EID remained in our paper open. 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.

▶ Lesson 9. Do not get discouraged when you are told without proper references that your result is well known.

▶ Lesson 10. Not every problem which looks easy from where you stand is easy for others approaching the problem from a different angle.

▶ Lesson 11. The fact that you can solve technical problems in other people's domain, does not make you an expert in this domain.

## Program Correctness and Termination

I have three papers dealing somehow with logic and program termination. One is an application of abstract model theory to dynamic logic, [48], one is a completeness theorem for a proof rule for fair termination, [32], and one contains weak second-order characterizations of various program verification systems, [51]. [32] is a good example of the previous lesson: I was able to provide a proof of a theorem formulated by N. Francez and O. Grumberg on fair termination, see [32], without grasping the essence of the problem. I really understood the problem only after reading the paper by D. Lehmann, A. Pnueli and J. Stavi, [45], which presented a different approach to fair termination.

More significantly, there was also a cultural problem: Discussing the problem with J. Stavi, he suggested that all this was a trivial application of J. Shoenfield's Tree Lemma, [79]. We went through this idea together and indeed came to the conclusion, that technically there was not much new, and that unwinding trees in special cases would be a good topic for PhD, or rather MSc students. We both grossly underestimated the gap between a logician who had studied all of Shoenfield's book [79], and a computer scientist who was interested in a particular application. The gap is not only technical, but also on the levels of abstraction. I tried to find graduate students to explore uses of the Tree Lemma, I tried also to collaborate with my colleagues, but I failed to bridge this gap. They could not believe that such a general lemma would help them, and they were not willing to spend the time to learn what appeared to them exotically abstract. One day, in 1981, I told D. Harel about my failure in recruiting partners for this project. I told him about my discussions with J. Stavi, and I sketched to him to use of the Tree Lemma. But David was busy with other research projects. Nevertheless, somehow our discussion made him think about all this, which led to [35]. He finally published a beautiful journal paper, [36] in which he acknowledges our suggestion.

▶ Lesson 12. Do not underestimate the amount of work and ingenuity sometimes needed in applying a clean abstract theorem to a concrete problem.

▶ Lesson 13. It turns out that it is sometimes easier to reinvent the wheel for special applications.

More recently, while applying logic to graph polynomials, the Tree Lemma appeared again: first in my work with B. Godlin and E. Katz, [28], and then, while trying to turn the abstract result of [28] into a concrete result, in my work with I. Averbouch and B. Godlin to [2].

## The Fundamental Problem of Database Design

The big problem of database design is the *choice of the basic relations* and the development of a *restructuring technology.* It does not matter whether we are in the Entity-Relationship model or the Relational Model of Database. There are many attempts to formulate criteria for a good choice of basic relations, some of them heuristical, some of them with a solid body of techniques, theorems and algorithms. Normal form theory is widely taught and popular because it lends itself readily to exam problems. But the last word in the design of databases has not been said. Many databases were designed fifty years ago, have become old-fashioned and have to be converted into new designs without loss of information while preserving the underlying constraints. I had three excellent students in Databases, V. Markowitz (PhD), U. Rotics (MSc), and E. Ravve (PhD). My most quoted paper in databases, [70], needed six years to get published, because the then editor of the IEEE Transactions of Software Engineering lost the paper. Only when he was replaced, the new editor hastened to publish it without sending us a referee report. I had never attended a database conference until I was an invited speaker at the ER-conference in 1996. V. Markowitz was better known than me, and at this conference many greeted me with "Ah, you were the supervisor of V. Markowitz". I naively thought that the impact of one's work was a function solely of one's results. This may still be true in the very long run, and the way my work is quoted in monographs may attest to this, but it is definitely wrong in the short run.

▶ Lesson 14. Unfortunately, you have to promote yourself by personally reporting about your work. People only read results of which they have heard before.

## Horn Formulas

Horn formulas are well-known in Model Theory because they are preserved under various product constructions, cf. [10]. Product constructions are important for the algebraists (universal and other) but rarely occur in Computer Science. I encountered Horn formulas in Computer Science (not under this name) first in my discussions with C. Beeri about database dependencies. I also encountered them in the algebraic specification of data types, [47], a then very promising field of research which did not bring the results its proponents hoped for, [29, 30, 20]. And then Horn formulas started to play a central rôle in Logic Programming and rule-based reasoning. Trained as a model theorist, I started to ask myself *why Horn formulas matter in Computer Science.* In Spring 1982 I taught a course where we discussed the satisfiability problem for Horn clauses. I showed to my students that it was solvable in polynomial (cubic) time and asked the students to come up with an algorithm which runs in less than cubic time. Oded Goldreich, then a graduate student, gave an $O(n \lg n)$-algorithm. This inspired Alon Itai and me to design a linear time algorithm. We did not think then that it this was such an important or particularly difficult result, so we did not rush into publication. A preprint was circulated in May 1982, [38], containing the result, but the main thrust of the paper was in proposing and analyzing a complexity measure for logic programming based on unit resolution and unification steps.

▶ **Theorem 4** (A. Itai and JAM, [38]).
*Propositional HornSat is solvable in linear time.*

The paper was finally published only in 1987, [39], because the referees disagreed and gave contradictory recommendations: one wanted more details in the motivation and background material, while the other recommended cutting it. As a result, the paper was reworked, and most of the credit for the linear time algorithm went to W. Dowling and J. Gallier, [19].

▶ Lesson 15. What may look as an exercise to you may still be an important result for others.

My answer to why Horn formulas matter in Computer Science may be found in [49], and my early advocating of the use of model theoretic methods in Computer Science in [55] and in [58].

## Finite Model Theory

It took me a while to really grasp why the restriction to finite models in databases was such an important issue. In [54, 68] I tried to characterize database dependencies using preservation theorems. I knew from my previous work that the interpolation theorems of first-order logic fail when we look at finite models only. But I did not realize then, and had to learn it from Y. Gurevich that the classical preservation theorems also fail. A notable exception of these failures is B. Rossman's Homomorphism Preservation Theorem, [77]. It remains open, whether K. Compton's Preservation Theorem for classes of structures closed under disjoint union and taking of components, [12], has an analogue for finite structures.

I was aware of Fagin's characterization of **NP** using existential second-orderlogic, but only when M. Vardi and N. Immermann proved their characterization of **P**, I realized that they had actually proved some kind of Lindström Theorem in terms of Complexity Theory.

It took me a while to understand in depth that two ideas of early Model Theory, mostly neglected in logic monographs before 1985 with the exception of D. Monk's [71], would be pervasive in applications of Model Theory to Computer Science and Combinatorics: Ehrenfeucht-Fraïssé games and Feferman-Vaught-type Theorems, cf. [60].

My own dabblings in Finite Model Theory were first concerned, with moderate success, in explaining generalized quantifiers in terms of oracle computations and in trying to capture relativized complexity classes by using suitably chosen generalized quantifiers, cf. [66]. However, the use of oracles in low complexity classes depends subtly on the exact way oracles are accessed, cf. [7], and my treatment of the subject remained sketchy.

## Monadic Second-Order Logic

Monadic Second-Order Logic over arbitrary structures is much stronger than First-Order Logic, and its semantics inherits problems of set theory. In contrast to this, over finite structures, Monadic Second-Order Logic seems natural and manageable. In 1995 Bruno

Courcelle visited the Technion and his visit was the beginning of an intense collaboration. B. Courcelle's book with J. Engelfriet, [13], gives a full account of the use of Monadic Second-Order Logic via a language theoretic approach. My own work in this direction started with Y. Pnueli, A. Pnueli's nephew, with whom I proved a hierarchy theorem for Second-Order Logic over finite structures: Let $AA_{m,n}$ be the class of properties of structures definable in Second-Order Logic with $m$ alternations of second-order quantifiers using relation variables of arity at most $n$.

▶ **Theorem 5** (JAM and Y. Pnueli, [65]). *The hierarchy formed by $AA_{m,n}$ is strict.*

In this period I had three PhD students: U. Rotics, E. Ravve and G. Kogan, working with me in three different directions: With E. Ravve I tried to find applications of Feferman-Vaught-like theorems to system verification, a project which gave limited success due to complexity limitations, and the fact that we did not work out a real life example. U. Rotics, my former student in Databases, approached me, after working for several years in industry, with ideas on how to generalize tree-width of graphs. Finally, G. Kogan , a new immigrant from the former Soviet Union, came to my colleague M. Kaminski and me with ideas on how to compute permanents of special classes of matrices, see [41]. Unfortunately, he was not able to complete the necessary non-mathematical requirements and failed to turn his excellent work into an orderly PhD. With U. Rotics we discovered independently the notion of clique-width introduced by B. Courcelle, J. Engelfriet and G. Rozenberg in [14] and further developed in [18], which led to [15, 16, 17].

▶ **Theorem 6** (B. Courcelle, JAM and U. Rotics). *Let $CW(k)$ be the class of graphs $G$ of clique-width at most $k$, and $\Phi$ denote a decision problem, optimization problem or counting problem, or even a graph polynomial, which is definable in Monadic Second-Order Logic. Then $\Phi$ can be solved on graphs in $CK(k)$ is polynomial time.*

When we proved this, we had to assume that the graph $G$ was given together with its parse-tree witnessing its clique-width. However, this assumtion can be eliminated using results of R. Seymour and S. Oung [75, 74].

## Graph Polynomials and Knot Theory

G. Kogan inspired me to apply the techniques developed in [17] to the computation of permanents. Let $M$ be an $(n \times n)$-matrix over some field $\mathbb{F}$. $M$ is orthogonal over $\mathbb{F}$ if $MM^t = I$. $M$ has rank at most $k$ over $\mathbb{F}$ if it has at most $k$ rows (columns) which are linearily independent. $M$ has tree-width at most $k$ if the graph $G_M = ([n], E_M)$ has tree-width at most $k$, where $(i, j) \in E_M$ iff $m_{ij} \neq 0$.

▶ **Theorem 7.** *Let $\mathbb{F}$ be any field.*

***A. Barvinok [4]*** *If $M$ has rank at most $k$,* $\mathrm{per}(M)$ *can be computed in polynomial time, where the constants depend on $k$.*

***JAM, 1997, cf. [17]*** *If $M$ has tree-width at most $k$,* $\mathrm{per}(M)$ *can be computed in polynomial time, where the constants depend on $k$.*

***G. Kogan, [41]*** *If $\mathbb{F}$ has characteristic* 3*, and $M$ is orthogonal over $\mathbb{F}$,* $\mathrm{per}(M)$ *can be computed in polynomial time.*

I clearly felt that my result on permanents had little to do with permanents and I was looking for other applications of the techniques developed in [17]. While on sabbatical in Zurich, I met V. Turaev and told him about my result. He suggested I should try to apply these techniques to the Jones polynomial in Knot Theory. I spent a year learning Knot Theory, especially knot polynomials and the Tutte polynomial which led to [59, 46, 61, 6].

A knot diagram is a planar signed graph. Its size is the number of crossings. If the knot diagram is alternating, it can be represented by an unsigned graph. The tree-width of signed graph is the same as the tree-width of the underlying unsigned graph. F. Jaeger, [40], showed that the Jones polynomial of an alternating knot diagram is essentially the Tutte polynomial.

▶ **Theorem 8.** *Let $G$ be a graph and $D$ be a knot diagram.*

***A. Andrzejak [1], S. Noble [73]*** *If $G$ is of tree-width at most $k$ the Tutte polynomial can be computed in polynomial time and is FPT (fixed parameter tractable). The same holds for the Jones polynomial of alternating knots.*

***JAM, [59, 61]*** *If $D$ is a (not necessarily alternating) knot diagram of tree-width at most $k$ the Jones polynomial can be computed in polynomial time and is FPT (fixed parameter tractable).*

This led me to study more graph polynomials with the ambitious goal of developing a general framework in which graph polynomials can be compared, cf. [63].

▶ Lesson 16. Do not restrict your supervising of PhD students to your own predefined topics.

## Back to categoricity

In 2005 the CSL conference was held in Oxford. It was the first CSL conference after I was elected president of EACSL. B. Zilber, an old friend from the times I worked in Model Theory on the finite axiomatizability of categorical theories, was now professor of Mathematical Logic in Oxford. Using the occasion, I went to see B. Zilber. While I was explaining to him my work on graph polynomials, he noticed that my examples of graph polynomials occurred as size functions of finite approximations in totally categorical theories. I could not believe what he told me! Instead of attending the CSL lectures we started to explore this further, and indeed, it worked. We began to work out a general theory of graph polynomials using model theoretic methods which resulted in the papers [69, 42, 43].

My scientific trip which started in Model Theory went far afield, to Databases, Logic Programming, Algorithmics and Complexity, only to return me to my origins. Now I work in applications of Model Theory to Finite Combinatorics. In January 2009 M. Grohe and I have organized a special session, *Model Theoretic Methods in Finite Combinatorics*, at the Joint AMS-ASL meeting in Washington, D.C. The book [31] is the result of this special session.

## Giving Credit

My narrative mentions several time the issue of giving credit to others. Clearly, it is not possible to remember precisely and all the time who or what inspired us to get our research results. Our memory is not reliable and conversations with colleagues which do not affect our work immediately tend to be forgotten. I have sinned on these accounts, and so have most of us. A bit of concerted introspection, however, helps a lot in avoiding careless omissions.

▶ Lesson 17. One cannot be careful enough in giving credit.

## Logicians and Computer Scientists

Anthropologists study humans and their *cultural systems* consisting of people sharing a purpose and certain tools and values. Cultural systems are well defined objects of study which were adapted by R.L. Wilder in [82] to the study of the evolution of mathematical concepts. Professional organisations, both of the logicians and the emerging community of computer scientists, had difficulties in acknowledging the relevance of Logic to Computer Science from the point of view of their respective disciplines.

On a personal level, when I was hired in 1980 by the Technion in Haifa, I was met with great suspicion, in spite of or because of the rôle logic played in Israel. The three founding fathers of Computer Science in Israel, M. Rabin, E. Shamir and S. Even, all understood the relevance of logic to Computer Science, but those coming from the culture of Electrical Engineering did not. The establishment of LiCS as an IEEE conference came as a complete surprise for them.

On a more global scale, in these early years neither the Association of Symbolic Logic, nor its German (rather German language) counterpart the DVMLG, showed genuine interest in the new partnership of Logic and Computer Science. Both the LiCS and CSL conferences were founded in 1986 and 1987 respectively because they had to create their own research community based on this partnership. In the last twenty or so years this partnership has thrived and spawned many new subcommunities, some of them concerned with foundational questions, but many using logical tools for genuine computer engineering disciplines much like calculus and differential equations are used in traditional engineering disciplines.

My own involvement in scientific organizations was within EACSL, LiCS and the German Logic Association (DVMLG). I served as vice-president and president of EACSL from 2002 until 2010. I also served on the board of DVMLG in the same period. I had myself several goals set to be realized during my terms:

- To strengthen the cooperation between LiCS and EACSL;
- To increase the visibility of EACSL;
- To increase the control of the scientists over their publication media;
- To further acceptance of Logic for Computer Science also as part of traditional Logic;
- To strengthen European activities of DVMLG.

I am happy to say that a good part of this agenda was realized. During my time

- the Ackermann Award was created;
- Cooperation between LiCS and EACSL was firmly established;
- The CSL proceedings moved finally from Springer to LIPIcs;
- The journal *Mathematical Logic Quarterly* became formally affiliated with DVMLG;
- Logic in Computer Science is now well represented on the board of DVMLG;
- Cooperation between EACSL and CiE (Computing in Europe) and KGS (Kurt Gödel Society) are well established;
- DVMLG held its first joint meeting with the Polish Logic Society in 2010.

Today the importance of logic for computer science seems to be well recognized and the Turing Awards of M. Rabin and D. Scott (Automata Theory), A. Pnueli (Temporal Logic) and E. Clark, A. Emerson and J. Sifakis (Model Checking) for the development of verification tools testify this. I said, "seems to be well recognized". The truth is that today many Computer Science Departments in which logic-based courses were a compulsory part of the curriculum tend to abandon teaching these courses. My suspicion is that this is partly due to the fact that we have not adapted the syllabi of our courses to the true needs of Computer Science. We tend to teach logic still in the tradition of the book by D. Hilbert and W. Ackermann [37] telling the same old stories about Hilbert's program, the paradoxes, and how K. Gödel put an end to the misguided hopes to use logic as the ultimate foundation of mathematics. I have detailed my thoughts about this in [50].

The standard course Linear Algebra evolved in the 1950s as an answer to physicists' needs providing them with the mathematical tools for quantum mechanics. We have yet to design a convincing logic course as an answer to the true needs of computer scientists and engineers. But we have to do this fast and in a concerted effort before it is too late and all our achievements are turned into well-used but ill-understood tools of the trade. I personally hope that the Turing Centenary will serve on an international scale as a reminder to the scientific public at large that we logicians still have something to offer to advance Computer (Computing) Science still further.

▶ Lesson 18. In your basic courses, show what one can do, and show its limitations, but do not speak mostly about the dashed hopes of the past.

## Future Challenges

The scientific community at large, and we logicians in particular, face several challenges. Technological and economic changes lead to radical changes in research and teaching. Both are threatened by short-range commercial interests and the effects of mass production in education and research. Our traditional models of producing young scientists and engineers, and of producing and evaluating research do not scale. What proved itself over the centuries in small elitist communities fails to function at the current scale of scientific and technological activities. Knowledge used to be the source of enlightenment and emancipation. Therefore it was meant to be shared by large parts of mankind. Knowledge is the basis for being largely autonomous individuals. Today knowledge tends to be delegated to the CLOUD, and access to the CLOUD will be controlled by few. Education increasingly emphasizes the ability to merely use techniques rather than to understand them thoroughly. The slow disappearence of logic-based courses is only a symptom. Delegating knowledge to the CLOUD endangers our freedom and maturity.

> Enlightenment is man's emergence from his self-imposed immaturity. Immaturity is the inability to use one's understanding without guidance from another. This immaturity is self-imposed when its cause lies not in lack of understanding, but in lack of resolve and courage to use it without guidance from another. Sapere Aude! [dare to know] "Have courage to use your own understanding!"–that is the motto of enlightenment.
>
> E. Kant, "An Answer to the Question: What is Enlightenment?" (1784)

> I do not know whether we will ever reach mature adulthood. Many things in our experience convince us that the historical event of the Enlightenment did not make us mature adults, and we have not reached that stage yet. However, it seems to me that a meaning can be attributed to that critical interrogation on the present and on ourselves which Kant formulated by reflecting on the Enlightenment. It seems to me that Kant's reflection is even a way of philosophizing that has not been without its importance or effectiveness during the last two centuries. The critical ontology of ourselves has to be considered not, certainly, as a theory, a doctrine, nor even as a permanent body of knowledge that is accumulating; it has to be conceived as an attitude, an ethos, a philosophical life in which the critique of what we are is at one and the same time the historical analysis of the limits that are imposed on us and an experiment with the possibility of going beyond them.
>
> M. Foucault "What is Enlightenment ?" ("Qu'est-ce que les Lumières?"),
> in Rabinow (P.), ed., The Foucault Reader, Pantheon Books, 1984, pp. 32-50.

Being an "autonomous individual" an utopia, but striving to approximate being one is still a noble task.

> Even in teaching mathematics we can at least attempt to teach 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 to understand.
>
> Roger Godement, Cours d'Algèbre, Hermann, Paris 1966 (my translation)

## References


1 A. Andrzejak. An algorithm for the Tutte polynomials of graphs of bounded treewidth. *Discrete Mathematics*, 190:39–54, 1998.

2 I. Averbouch, B. Godlin, and J.A. Makowsky. An extension of the bivariate chromatic polynomial. *European Journal of Combinatorics*, 31.1:1–17, 2010.

3 J.T. Baldwin and A.H. Lachlan. On strongly minimal sets. *J. Symbolic Logic*, 36(1):79–96, 1971.

4 A.I. Barvinok. Two algorithmic results for the traveling salesman problem. *Mathematics of Operations Research*, 21:65–84, 1996.

5 J. Barwise and S. Feferman, editors. *Model-Theoretic Logics*. Perspectives in Mathematical Logic. Springer Verlag, 1985.

6 M. Bläser, H. Dell, and J.A. Makowsky. Algebraic point-to-point reductions for the colored Tutte polynomial. Preprint, 2007.

7 J.F. Buss. Alternations and space–bounded computations. *Journal of Computer and System Sciences*, 36:351–378, 1988.

8 A. Chandra, H. Lewis, and J.A. Makowsky. Embedded implicational dependencies and their implication problem. In *ACM Symposium on the Theory of Computing 1981*, pages 342–354. ACM, 1981.

9 A. K. Chandra, H. R. Lewis, and J. A. Makowsky. Embedded implicational dependencies and their inference problem. In *XP1 Workshop on Database Theory*, 1980.

10 C.C. Chang and H.J. Keisler. *Model Theory*. Studies in Logic, vol 73. North–Holland, 3rd edition, 1990.

11 G. Cherlin, L. Harrington, and A. Lachlan. $\aleph_0$-categorical $\aleph_0$-stable structures. *Annals of Pure and Applied Logic*, 18:227–270, 1980.

12 K. J. Compton. Some useful preservation theorems. *J. Symb. Log.*, 48(2):427–440, 1983.

13 B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-order Logic, a Language Theoretic Approach*. Cambridge University Press, 2012, in press.

14 B. Courcelle, J. Engelfriet, and G. Rozenberg. Handle-rewriting hypergraph grammars. *J. Comput. System Sci.*, 46:218–270, 1993.

15 B. Courcelle, J.A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graph of bounded clique width, extended abstract. In J. Hromkovic and O. Sykora, editors, *Graph Theoretic Concepts in Computer Science, 24th International Workshop, WG'98*, volume 1517 of *Lecture Notes in Computer Science*, pages 1–16. Springer Verlag, 1998.

16 B. Courcelle, J.A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33.2:125–150, 2000.

17 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*, 108(1-2):23–52, 2001.

18 B. Courcelle and S. Olariu. Upper bounds to the clique–width of graphs. *Discrete Applied Mathematics*, 101:77–114, 2000.

19 W. F. Dowling and J. H. Gallier. Linear-time algorithms for testing the satisfiability of propositional horn formulae. *J. Log. Program.*, 1(3):267–284, 1984.

20 H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1: Equations und Initial Semantics*, volume 6 of *Monographs in Theoretical Computer Science. An EATCS Series*. Springer, 1985.

21 E. Engeler. A characterization of theories with isomorphic denumerable models. *Not. Amer. Math. Soc.*, 6:161, 1959. Abstract.

22 Erwin Engeler. *Untersuchungen zur Modelltheorie*. PhD thesis, Department of Mathematics, ETH-Zurich, Switzerland, 1958.

23 S. Feferman. Applications of many–sorted interpolation theorems. In L. Henkin et al., editor, *Proceedings of the Tarski Symposium*, volume 25 of *Proceedings of Symposia in Pure Mathematics*, pages 205–223. American Mathematical Society, 1974.

24 S. Feferman. Two notes on abstract model theory, I: Properties invariant on the range of definable relations between strcutures. *Fundamenta Mathematicae*, 82:153–165, 1974.

25 S. Feferman. Two notes on abstract model theory, II: Languages for which the set of valid sentences is s.i.i.d. *Fundamenta Mathematicae*, 89:153–165, 1975.

26 S. Feferman and R. Vaught. The first order properties of algebraic systems. *Fundamenta Mathematicae*, 47:57–103, 1959.

27 Z. Galil. On the complexity of regular resolution and the davis-putnam procedure. *Theor. Comput. Sci.*, 4(1):23–46, 1977.

28 B. Godlin, E. Katz, and J.A. Makowsky. Graph polynomials: From recursive definitions to subset expansion formulas. *Journal of Logic and Computation*, in press:xx–yy, 2011. DOI: 10.1093/logcom/exq006.

29 J. Goguen, J.W. Thatcher, E.G. Wagner, and J.B. Wright. Initial algebra semantics and continuous algebras. *Journal of ACM*, 24:68–95, 1977.

30 J. A. Goguen and R. M. Burstall. Introducing institutions. In *Logic of Programs*, pages 221–256, 1983.

31 M. Grohe and J.A. Makowsky, editors. *Model Theoretic Methods in Finite Combinatorics*, volume 558 of *Contemporary Mathematics*. American Mathematical Society, 2011. in press.

32 O. Grumberg, N. Francez, J.A. Makowsky, and W.P. de Roever. A proof rule for fair termination of guarded commands. *Information and Control*, 66.1-2:83–102, 1985.

33 Y. Gurevich. The word problem for some classes of semigroups (russian). *Algebra and Logic*, 5(2):25–35, 1966.

34 Y. Gurevich. Logic and the challenge of computer science. In E. Börger, editor, *Trends in Theoretical Computer Science*, Principles of Computer Science Series, chapter 1. Computer Science Press, 1988.

35 D. Harel. A general result on infinite trees and its applications (preliminary report). In *STOC*, pages 418–427, 1984.

36 D. Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *J. ACM*, 33(1):224–248, 1986.

37 D. Hilbert and W. Ackermann. *Principles of Mathematical Logic*. Chelsea Publishing Company, 1950.

38 A. Itai and J.A. Makowsky. On the complexity of Herbrand's theorem. Technical Report No. 243, Departmnet of Computer Science, Technion–Israel Institute of Technology, Haifa, Israel, 1982.

39 A. Itai and J.A. Makowsky. Unification as a complexity measure for logic programming. *Journal of Logic Programming*, 4(2):105–117, 1987.

40 F. Jaeger. Tutte polynomials and link polynomials. *Proceedings of the American Mathematical Society*, 103:647–654, 1988.

41 G.P. Kogan. Computing the permanent over fields of characteristic 3: Where and why it becomes difficult. In *FOCS'96*, pages 108–114. IEEE, 1996.

42 T. Kotek, J.A. Makowsky, and B. Zilber. On counting generalized colorings. In *Computer Science Logic, CSL'08*, volume 5213 of *Lecture Notes in Computer Science*, page 339–353, 2008.

43 T. Kotek, J.A. Makowsky, and B. Zilber. On counting generalized colorings. In M. Grohe and J.A. Makowsky, editors, *Model Theoretic Methods in Finite Combinatorics*, volume 558 of *Contemporary Mathematics*, pages xx–yy. American Mathematical Society, 2011. in press.

44 G. Kreisel. Choice of infinitary languages by means of definability criteria; generalized recursion theory. In J. Barwise, editor, *The Syntax and Semantics of Infinitary Languages*, volume 72 of *Springer Lecture Notes in Mathematics*, pages 139–151. Springer, 1968.

45 D. J. Lehmann, A. Pnueli, and J. Stavi. Impartiality, justice and fairness: The ethics of concurrent termination. In *ICALP*, pages 264–277, 1981.

46 M. Lotz and J.A. Makowsky. On the algebraic complexity of some families of coloured Tutte polynomials. *Advances in Applied Mathematics*, 32(1-2):327–349, 2004.

47 B. Mahr and J.A. Makowsky. Characterizing specification languages which admit initial semantics. *Theoretical Computer Science*, 31:49–60, 1984.

48 J. A. Makowsky. Measuring the expressive power of dynamic logics: An application of abstract model theory. In *ICALP*, pages 409–421, 1980.

49 J. A. Makowsky. Why Horn formulas matter for computer science: Initial structures and generic examples. *Journal of Computer and System Sciences*, 34.2/3:266–292, 1987.

50 J. A. Makowsky. From Hilbert's program to a logic tool box. *Ann. Math. Artif. Intell.*, 53(1-4):225–250, 2008.

51 J. A. Makowsky and I. Sain. Weak second order characterizations of various program verification systems. *Theoretical Computer Science*, 66:299–321, 1989.

52 J.A. Makowsky. Langages engendrés à partir des formules de Scott. *C.R. hebd. Acad. Sc. Paris*, 276:1585–1587, 1973.

53 J.A. Makowsky. On some conjectures connected with complete sentences. *Fund. Math.*, 81:193–202, 1974.

54 J.A. Makowsky. Characterizing database dependencies. In *ICALP'81*, volume 115 of *Lecture Notes in Computer Science*, pages 86–97. Springer Verlag, 1981.

55 J.A. Makowsky. Model theoretic issues in theoretical computer science, part I: Relational databases and abstract data types. In G. Lolli et al., editor, *Logic Colloquium '82*, Studies in Logic, pages 303–343. North Holland, 1984.

56 J.A. Makowsky. Abstract embedding relations. In J. Barwise and S. Feferman, editors, *Model-Theoretic Logics*, Perspectives in Mathematical Logic, chapter 20. Springer Verlag, 1985.

57 J.A. Makowsky. Compactness, embeddings and definability. In J. Barwise and S. Feferman, editors, *Model-Theoretic Logics*, Perspectives in Mathematical Logic, chapter 18. Springer Verlag, 1985.

58 J.A. Makowsky. Model theory and computer science: An appetizer. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 1, chapter I.6. Oxford University Press, 1992.

59 J.A. Makowsky. Colored Tutte polynomials and Kauffman brackets on graphs of bounded tree width. In *Proceedings of the 12th Symposium on Discrete Algorithms*, pages 487–495. SIAM, 2001.

60 J.A. Makowsky. Algorithmic uses of the Feferman-Vaught theorem. *Annals of Pure and Applied Logic*, 126.1-3:159–213, 2004.

61 J.A. Makowsky. Colored Tutte polynomials and Kauffman brackets on graphs of bounded tree width. *Disc. Appl. Math.*, 145(2):276–290, 2005.

**62** J.A. Makowsky. Encounters with A. Mostowski. In W. Marek and M. Srebrny, editors, *70 years of Foundational Studies: In memoriam of Andrzej Mostowski*, page In press. IOS Press, 2008.

**63** J.A. Makowsky. From a zoo to a zoology: Towards a general theory of graph polynomials. *Theory of Computing Systems*, 43:542–562, 2008.

**64** J.A. Makowsky and D. Mundici. Abstract equivalence relations. In J. Barwise and S. Feferman, editors, *Model-Theoretic Logics*, Perspectives in Mathematical Logic, chapter 19. Springer Verlag, 1985.

**65** J.A. Makowsky and Y. Pnueli. Arity vs. alternation in second order logic. *Annals of Pure and Applied Logic*, 78(2):189–202, 1996.

**66** J.A. Makowsky and Y.B. Pnueli. Oracles and quantifiers. In *CSL'93*, volume 832 of *Lecture Notes in Computer Science*, pages 189–222. Springer, 1994.

**67** J.A. Makowsky, S. Shelah, and J. Stavi. $\Delta$–logics and generalized quantifiers. *Annals of Mathematical Logic*, 10:155–192, 1976.

**68** J.A. Makowsky and M. Vardi. On the expressive power of data dependencies. *Acta Informatica*, 23.3:231–244, 1986.

**69** J.A. Makowsky and B. Zilber. Polynomial invariants of graphs and totally categorical theories. MODNET Preprint No. 21, `http://www.logique.jussieu.fr/modnet/Publications/Preprint%20 server`, 2006.

**70** V. M. Markowitz and J. A. Makowsky. Identifying extended entity-relationship object structures in relational schemas. *IEEE Trans. Software Eng.*, 16(8):777–790, 1990.

**71** J.D. Monk. *Mathematical Logic.* Graduate Texts in Mathematics. Springer Verlag, 1976.

**72** M. Morley. Categoricity in power. *Trans. Amer. Math. Soc.*, 114:514–38, 1965.

**73** S. D. Noble. Evaluating the Tutte polynomial for graphs of bounded tree-width. *Combinatorics, Probability, and Computing*, 7:307–321, 1998.

**74** S. Oum. Approximating rank-width and clique-width quickly. In *Graph Theoretic Concepts in Computer Science, WG 2005*, volume 3787 of *Lecture Notes in Computer Science*, pages 49–58, 2005.

**75** S. Oum and P. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Ser. B*, 96(4):514–528, 2006.

**76** M.G. Peretyat'kin. An example of an $\omega_1$-categorical complete finitely axiomatizable theory. *Algebra and Logic*, 19:202–229, 1980.

**77** B. Rossman. Homomorphism preservation theorems. *J. ACM*, 55(3), 2008.

**78** D. Scott. Logic with denumerably long formulas and finite strings of quantifiers. In *Theory of Models*, Studies in Logic, pages 329–341, 1965.

**79** J. Shoenfield. *Mathematical Logic.* Addison-Wesley Series in Logic. Addison-Wesley, 1967.

**80** E. Specker and V. Strassen, editors. *Komplexität von Entscheidungsproblemen*, volume 43 of *Lecture Notes in Computer Science.* Springer Verlag, 1976.

**81** L.H. Tharp. Continuity and elementary logic. *J. Symb. Log.*, 39(4):700–716, 1974.

**82** R.L. Wilder. *Mathematics as a Cultural System.* Pergamon Press, 1981.

**83** B. Zilber. *Uncountably Categorical Theories*, volume 117 of *Transl.of Math.Monographs.* Amercan Mathematical Society, Providence, R.I., 1993.