The Logic in Computer Science Column

by

Yuri Gurevich

Computer Science & Engineering
University of Michigan, Ann Arbor, Michigan, USA
gurevich@umich.edu

THE UMBILICAL CORD OF FINITE MODEL THEORY

Yuri Gurevich

Computer Science & Engineering, University of Michigan

Abstract

Model theory was born and developed as a part of mathematical logic. It has various application domains but is not beholden to any of them. A priori, the research area known as finite model theory would be just a part of model theory but didn't turn out that way. There is one application domain — relational database management — that finite model theory had been beholden to during a substantial early period when databases provided the motivation and were the main application target for finite model theory.

Arguably, finite model theory was motivated even more by complexity theory. But the subject of this paper is how relational database theory influenced finite model theory.

This is NOT a scholarly history of the subject with proper credits to all participants. My original intent was to cover just the developments that I witnessed or participated in. The need to make the story coherent forced me to cover some additional developments.

1 Prelude

Q[1]: How come finite model theory is computer science while model theory is mathematics?

A[2]: Let me think. Model theory is a part of mathematical logic which was born and developed as a part of foundations of mathematics. Finite model theory (FMT) was developed much later. It did not exist as a separate research area before the 1970s. FMT was developed primarily by computer scientists. It was — and is — much influenced by complexity theory which is viewed traditionally as theoretical computer science (though mathematicians often view it as mathematics). Also, during a substantial early period FMT was much influenced by (relational) database theory (DBT), bona fide computer science. I doubt that FMT would have become a recognizable research area without that DBT link. In that sense, the DBT link is the umbilical cord of FMT.

Q: Tell me about that influence of DBT on FMT.

A: I'll try my best. But let me make two reservations. I am not a DBT expert and, more importantly, I am a wrong person to render a scholarly history of the subject complete with proper credits to all participants. It takes a special talent to give a scholarly account of a research field; I don't have such talent. I will tell you primarily about the developments that I witnessed or participated in.

[1] Quisani is my former student.

[2] The author

Q: One last question before you start your story. I presume that logicians studied finite models before the 1970s?

A: Yes, they did. Let me start with that logic prehistory of FMT.

2 Prehistory

Finite structures were relevant to humanity long before the notion of structure was introduced, in fact before the notion of theorem was introduced. Consider for example the fact that a finite set can be counted in any order, always giving the same result. The special cases of this fact — for 2-element sets, 3-element sets, etc. — must have been known very early. The Greeks probably knew the general fact before the notion of theorem was introduced.

While model theory was studied (e.g. by Skolem and Löwenheim) in the beginning of the 20th century, finite model theory arguably became a separate research area in the 1970s, and the term "finite model theory" was coined only toward the end of the century.

As the rest of this section shows, finite model theory was studied and used before it became a separate research area.

2.1 Boris A. Trakhtenbrot

In the paper [45] Boris Trakhtenbrot proved that finite satisfiability is not decidable. There is no algorithm that, given a first-order sentence φ, decides whether φ has a finite model.

It follows that finite validity is not recursively enumerable, that is the set of finitely valid (valid on all finite models) first-order sentences is not recursively enumerable. Indeed, let S be the set of finitely valid sentences and S' the complement of S. Sentences in S' have finite counterexamples. It follows that S' is recursively enumerable. If S were recursively enumerable as well, then the following (not very feasible) algorithm would decide, given a sentence φ, whether it is in S or in S'. Keep enumerating a list of S sentences and a list of S' sentences until φ occurs in one of the lists.

The failure of recursive enumerability is important. It means that there is no reasonable deductive system for finite validity. This contrasts with the situation for general validity (validity in arbitrary, not necessarily finite, structures) where there is such a deductive system.

In the paper [46] published in 1953, Trakhtenbrot strengthened his result. The set of finitely satisfiable sentences and the subset of unsatisfiable sentences are recursively inseparable: There is no recursive set that contains one of the two sets and is disjoint from the other.

Since FMT is typically considered as a part of theoretical computer science, it may be fitting to note that Trakhtenbrot was also a pioneer of computer science and had many good students (Figure 1 shows him with a few of them). See also [2] in this connection.

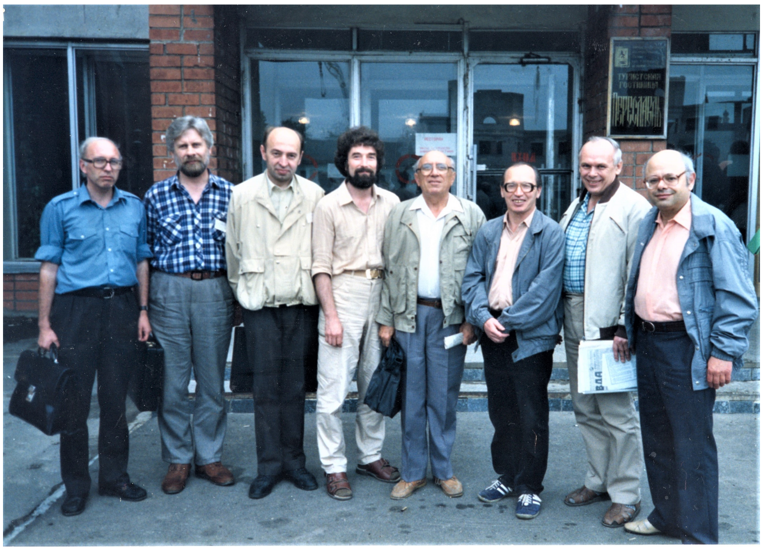

Figure 1: Boris Trakhtenbrot with some of his former students in 1992. Left to right: Vladimir Sazonov, Janis Barzdins, Rusins Freivalds, Alexandre Dikovsky, Boris Trakhtenbrot, Mars Valiev, Miroslav Kratko, and Valery Nepomnyaschy.

I first met Trakhtenbrot in the mid-1960s in the Novosibirsk Academgorodok [50] where he was running a lively seminar on the nascent computer science. In spite of the difference in age, we became friends. I used to come to the Akademgorodok from time to time to give a talk at the Algebra & Logic seminar run by academician Anatoly Maltsev. On the same occasions, I would attend Trakhtenbrot's seminar and talk to Trakhentbrot. I remember how he introduced time and space complexity (by means of so-called "signalizing functions") independently of his Western colleagues. I could have become a computer scientist much earlier than I did. But, at the time, I was moving from algebra to logic and it was logic that fascinated me.

2.2 Dana Scott, Patrick Suppes, and William W. Tait

(a) Dana S. Scott

(b) William W. Tait

(c) Patrick C. Suppes

It is easy to see that a purely universal sentence, that is a sentence of the form

$$\varphi = \forall x_1 \forall x_2 \ldots \forall x_n \Phi(x_1, x_2, \ldots, x_n)$$

where Φ is quantifier-free, is preserved by submodels: If $A \models \varphi$ and B is a submodel of A then $B \models \varphi$. According to the Łoś-Tarski theorem [10, Theorem 5.2.4], there is a converse: Any sentence that is preserved by substructures is equivalent to a universal sentence.

In the 1958 paper [41], Dana Scott and Patrick Suppes conjectured that the Łoś-Tarski theorem remains valid when only finite structures are taken into account. In the 1959 paper [42], William W. Tait disproved their conjecture. In retrospect, his counterexample is rather simple. Essentially it is a first-order sentence expressing that a linear order with a first element has also a last element. Every finite model has the property, and so the property is preserved by submodels of finite models. But an infinite model may violate the property in spite of being a submodel of one that has the property; the smallest example is the linear order ω of natural numbers as a submodel of $\omega + 1$.

But there was more in the Scott-Suppes article than the conjecture mentioned above. We will return to their paper later in this section.

2.3 Yuri Glebsky and his students

In the 1966 International Congress of Mathematicians in Moscow, Yuri Glebsky announced an unexpected zero-one law for first-order logic [23].

Consider a finite set V of relation symbols and restrict attention to models of vocabulary V with the universe of the form $\{1, 2, \ldots, n\}$. For each first-order sentence φ in vocabulary V, let $B(n)$ be the total number of n-element models and $A(n)$ the number of such models satisfying φ. The zero-one law states that, for every V and φ as above, $\lim_{n\to\infty} A(n) \big/ B(n)$ exists and is equal to either 0 or 1.

Figure 3: Yuri V. Glebsky

In 1969, Glebsky and three of his students — Dmitry I. Kogan, Mark I. Liogonky, and Vladimir A. Talanov — published a proof of the zero-one law [24]. In 1972, the paper was duly translated to English but wasn't noticed in the West. By chance, I knew paper [24]. This wasn't because I was so scholarly; that has never been my forte. But, in 1970, I served as an official opponent at the public defense of the PhD thesis of Liogonky. In the thesis, Liogonky proved that the 0-1 law remains valid if the structures are counted up to isomorphism

In 1972, Ron Fagin announced the 0-1 law, which he had discovered independently in the meantime [15]. He gave an elegant proof of the law in his thesis [16] and published it in [21]. A number of years later, I met Fagin and told him about Glebsky. Fagin was surprised, but he found the English translation of the 1969 paper, and the matter was settled.

In 1977 Glebsky drowned in the Volga trying to save his son. His students admired him, and his untimely death devastated them [39].

2.4 Other relevant results

Q: To what extent is the collection of relevant "prehistoric" results above exhaustive?

A: It is definitely not exhaustive. There are many pre-1970 results that are relevant to some degree. For example, the recursive inseparability result of Trakhtenbrot provoked more subtle recursive inseparability results in algebra and logic. In the book [8], an appropriate reduction theory is presented, and it is proven that, for many natural classes *K* of first-order sentences, the set of finitely satisfiable *K* sentences and the set of unsatisfiable *K* sentences are recursively inseparable.

Also there is at least one very relevant result that is missing above. Commenting on a draft of this paper, Erich Grädel reminded me of the Büchi-Elgot-Trakhtenbrot Theorem [5, 11, 47], which he views as the first descriptive complexity result. The theorem states that a set of strings in any finite alphabet is accepted by a finite-state automaton if and only if it is definable in monadic second-order logic (MSO). It is certainly the first descriptive complexity result that I know.

3 Database theory

Edgar Frank "Ted" Codd wouldn't get into a model theory hall of fame. His famous theorem, known simply as Codd's Theorem [52], is good model theory, but it wasn't real news to logicians as we explain below. Yet he is a hero of our story. He invented the relational model for database management, which facilitated the birth of finite model theory as a separate research area.

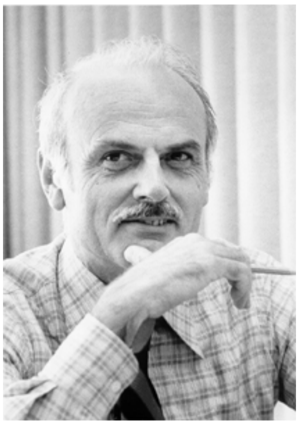

Figure 4: Edgar F. Codd

It may be hard for a theorist with no practical experience to appreciate Codd's contribution. Codd observed that databases can be adequately represented by (first-order) structures [13, 14]. But structures had originally been designed to represent data, or knowledge.

Q: Designed by whom?

A: By architects of what is now called model theory, starting from Löwenheim and Skolem, including Gödel and Tarski, especially Tarski; see [43] for example.

Q: Mathematical logic was developed within the framework of foundations of mathematics. Only mathematical data might have been of interest to logicians.

A: I don't know what you mean by mathematical data. Logicians certainly did not restrict attention to numerical data. Any data whatsoever may be involved in a mathematical problem and thus become "mathematical data" of interest to logicians.

Besides, logicians have been interested in problems beyond foundations of mathematics. The Scott-Suppes paper mentioned in §2.2 is on the foundations of theories of measurements. The authors foresaw relational databases:

> "From an abstract standpoint, a set of empirical data consists of a collection of relations between specified objects ... we treat sets of empirical data as being (finitary) relational systems" [41, §1]. ◁

In article [14], Codd defined a variable-free *relational algebra* and a *relational calculus*. The latter is a form of first-order logic, with propositional connectives and quantifiers, which is convenient for his purposes. He also proved a theorem, Codd's Theorem, establishing that relational algebra has essentially the expressive power of first-order logic.

Q: Why essentially and not literally?

A: Database queries should be safe in the appropriate technical sense [1, §5.3]. For example, no query should express the complement of a finite subset of an infinite domain. Codd's Theorem establishes that relational algebra has the expressive power of the safe fragment of first-order logic.

But Codd's Theorem wasn't real news for logicians either. An algebraic form of first-order logic, using so-called cylindric algebras, had been developed earlier by Tarski and his collaborators [12, 44, 32].

Q: If logicians knew all that, then what made Codd's work important?

A: In one word, engineering. There is a huge difference between treating sets of empirical data as relational systems and practical database management. Codd's approach was practical, and the result was a model of database management superior to all competitors. He received the Turing Award in 1981.

Q: What are the benefits of the relational database management?

A: One is simplicity; all information is represented by data values in addition to a few relation symbols. Another important benefit is clear and unambiguous semantics. Yet another important benefit is that the relational approach admits high level query languages.

Q: If we stick to theory, rather than practice, was there anything new in Codd's approach?

A: Yes, there was. Normally, in logic and in mathematics in general, structures are static. You study a particular structure, e.g. the field of real numbers, or a class of structures, e.g. commutative groups. But databases evolve and in that sense they are dynamic. For example, consider the salary database for some organization. New employees are hired, some employees leave the organization, some get raises, etc. To incorporate evolution, the language of databases should allow us to update databases in various ways.

Q: Did you know Codd?

A: No; I wish I did. Codd wrote his PhD thesis at the University of Michigan, but that was in the 1960s, long before I arrived there. But I heard plenty about Ted Codd from his advisor John Holland. John told me that Codd compromised his health trying to promote the relational approach at IBM. His fight at IBM is described in many places, even in the Wikipedia article [51].

4 The beginning

Q: After all these preliminaries are you ready to discuss FMT per se? If yes, start with the beginning.

A: I am ready to discuss FMT but starting with the beginning is a challenge. Until 1982, when I joined the University of Michigan and officially became a computer scientist, I was a mathematician and very different things were on my mind, primarily model theory. So, preparing for this discussion, I exchanged letters with experts. In particular, I posed two questions to Ron Fagin:

1. What do you count as the beginning of FMT? Is it your thesis?
2. Who coined the term “finite model theory”? And when?

“Although Trakhtenbrot’s Theorem was the first important result,” replied Ron on Nov. 9 2022, “[I] do feel that my Ph.D. thesis launched FMT as a field. Since I am biased, I called Moshe [Vardi] (whom I am cc-ing). He says he completely agrees that Trakhtenbrot had the first major result and that my thesis started FMT as a field.”

A couple of days later, Moshe sketched for me his view of the early history.

> “In the 1970s, there are three independent lines of work:
>
> a. 1970: Codd’s paper — start of relational DBT.
>
> b. 1973: Fagin’s dissertation — ‘Contributions to the model theory of finite structures’, where he proved $\Sigma^1_1 = \text{NP}$, the 0-1 Law, and that reachability is not definable in [existential] MSO.
>
> c. 1980: Immerman’s dissertation — ‘First order expressibility as a new complexity measure.’
>
> The work in the early 1980s built on these three legs, even though it was not explicitly FMT. For example, the 1982 papers by Immerman and by Vardi built on these legs, but did not emphasize the finiteness feature. Your papers in the 1980s raised the issue of finiteness and helped FMT become a well-defined field.” — Moshe Vardi Dec. 17 2022

Q: Does this sound reasonable to you?

A: Totally. Ron’s thesis [16] was truly pioneering; the results are published in five papers: [17, 18, 19, 20, 21]. I have already mentioned his elegant proof of the 0-1 law [21]. Another major result is that connectivity of undirected graphs is not expressible in existential monadic second-order logic [18]; since disconnectedness is expressible in the logic, the collection of expressible properties is not closed under complementation. The result known as Fagin’s Theorem asserts that the expressive power of existential second-order logic on finite structures is exactly that of the NP complexity class [17]. I mentioned above that Büchi-Elgot-Trakhtenbrot Theorem was the first descriptive complexity result. But Fagin’s Theorem started descriptive complexity theory in earnest. It wasn’t about finite automata but about one of the most important complexity classes, NP.

As far as relational database theory is concerned, the timing of Ron's thesis was perfect. Codd's 1970 and 1972 papers [13] and [14] respectively laid foundations for the relational model of database management, which provided motivation and application for the nascent finite model theory. In a different letter, of Dec. 17 2022, Ron wrote to me: "[W]hen I transferred from IBM Watson to IBM San Jose in 1975, Ted Codd became my mentor, which led me to write papers explicitly on relational data base theory."

In the 1970s and early 1980s there was much DBT work of relevance to FMT. I know several experts in DBT who are also experts in FMT — Serge Abiteboul, Phokion Kolaitis, Victor Vianu, and Moshe Vardi; they can tell you all about that. The book [1] is a good DBT textbook for logically inclined. One DBT article of influence on FMT that will play a role later in our story is the article [9] by Ashok Chandra and David Harel.

In 1980 Neil Immerman defended his PhD thesis [33]; the results were published in [34, 35]. I asked Neil what led him to FMT.

> "I was inspired by Fagin's characterization of NP and his proof that connectivity is not in monadic SOE [existential monadic second-order logic]," replied Neil, "I felt that lower bounds on full arity SO was going to be too hard, but I thought that first-order lower bounds would be more tractable. This was what inspired me to work in my thesis area developing what I called "First-Order Expressibility". Later Hartmanis [Neil's advisor] suggested, "Descriptive Complexity" as a more appealing name." — Neil Immerman Dec. 20 2022

In STOC 1982, Immerman and Vardi presented their versions of what became known as the Immerman-Vardi Theorem; they showed that, on ordered structures, the least fixed point extension of first order logic captures polynomial time, widely viewed as the most important complexity class [36, 48]. Immerman went on to become a champion, maybe I should say the champion, of descriptive complexity theory, as both a contributor and an evangelist; see his book [37] on the subject.

Q: What about your contributions that Vardi mentioned?

A: I'll speak more about that in the next section. Here let me mention the very first result: Under a natural interpretation over finite domains, a function is primitive recursive if and only if it is logspace computable, and it is general recursive if and only if it is polynomial time computable [27].

Q: Did Fagin reply to your second question?

A: Yes. He wrote: "I don't know (and neither does Moshe) who coined the term 'finite model theory'. Moshe says he will look into that." Moshe wrote to us the same day and surprised me. "The first paper that I found that used FMT in its title is [30]. Ron followed up with [22]. So it seems that Yuri deserves the credit for coining the term[3]."

Q: Did you coin the term?

A: Not consciously. Contrary to Moshe and Ron, I did not belong to the database community. I came from model theory, and the term "finite model theory" suggests itself to model theorists working with finite models. And it is possible of course that the term was used earlier though not in the titles of papers.

[3] I took the liberty of replacing URLs with references to the bibliography

5 The importance of being finite

Q: How did you arrive to relational databases?

A: At FOCS 1982, Moshe Vardi presented "On decomposition of relational databases" where he made essential use of Beth's Definability Theorem [49]. I worried that the theorem may not survive the restriction to finite structures and I asked him whether his databases may be infinite. He said yes.

Q: Infinite databases! This is crazy.

A: And a little expensive.

Q: Right.

A: I thought that first-order logic may not be the best fit for databases.

Q: What gave you this idea?

A: My prior experience with ordered abelian groups (OAGs). In my PhD thesis [25], I proved the decidability of the first-order theory of OAGs, solving a problem of Tarski. But the OAG algebra is rarely first-order. So I extended the decidability to the monadic second-order theory of OAGs where the set variables range over convex subgroups [26]. The expanded OAG theory essentially subsumed the OAG literature of the time while the decidability proof became simpler. Obviously first-order logic wasn't the best fit for OAGs.

Returning home after the FOCS conference, I constructed counterexamples for some famous classical logic theorems, including Beth's Definability, in the finite case, that is the case where only finite structures are taken into account [28]. In the process, I discovered that William Tait had given the counterexample for the Łoś-Tarski theorem described in §2.2.

Q: Was there a classical theorem that you expected to fail in the finite case but couldn't find a counterexample?

A: Yes, Lyndon's theorem [38]. It states that a first-order sentence $\varphi(P)$ is monotone in P if and only if it is equivalent to a sentence $\psi(P)$ where P has only positive occurrences. Eventually Miki Ajtai and I proved that it fails in the finite case [4].

Q: Why did Codd strive to achieve the expressivity of first-order logic.

A: That was rather natural, I guess. First-order logic is relatively expressive. A great many relational queries are expressible in first-order logic. Besides, first-order logic is the logic of standard logic textbooks. What else was there? Propositional logic is way too weak while second-order logic is way too expressive.

Q: But first-order logic was developed in the framework of foundations of mathematics, I understand, where infinite structures are indispensable.

A: This is true. Also, in the foundational framework, it is important that first-order logic admits an adequate deductive system, where a sentence is provable if and only it is valid. In fact, first-order logic was introduced as a deductive system. It was obvious that all provable sentences are valid. Gödel's completeness theorem established that all valid sentences are provable. As we saw in §2.1, finite validity is not recursively enumerable. Thus there is no logic calculus where a sentence is provable if and only it is valid on all finite structures.

Q: In reality, had first-order expressibility proved insufficient?

A: Yes. There is no first-order query (or even existential monadic second-order query) that, given an arbitrary relation R returns the transitive closure of R [18], and transitive closure is needed in database management [53].

Q: If first-order expressivity isn't sufficient, what can you do? Maybe you can carve out an appropriate fragment of second-order logic.

A: Yes, but it is more convenient to think in different terms, like transitive closure or least fixed point operators [3].

Q: But what guides you?

A: Ah, this is the key point. The answer is computational complexity.

Q: What's the connection exactly?

A: This is a long story. Let me refer you to the paper [28].

6 Metafinite structures

Q: You said that you didn't coin the term "finite model theory" consciously. Have you ever consciously coined a term?

A: Sure, more than once. One of those terms is relevant to our current conversation: metafinite.

Q: Hmm, you can't be half pregnant or for that matter half infinite. You don't claim that databases are metafinite, I presume, as you claimed a minute ago that they are finite.

A: I do claim that databases are metafinite. Let me explain. A database contains only finitely many records (tuples in relations) at any given moment. In that sense it is perfectly finite. It is also dynamic as we saw in §3. But implicitly databases often involve static infinite structures.

Q: We considered a salary database in §3. Does it involve an infinite structure?

A: Yes, a query to a salary database may return a number, say the average salary, that does not appear in the database. Where does this new number come from? From the infinite structure of rational numbers.

Q: That involvement of an infinite structure, what is it mathematically?

A: A good question. Erich Grädel and I worked on it [31]. In our formalization, a metafinite structure consists of

1. a primary part, which is a finite structure,
2. a secondary part, which is a (usually infinite) structure, e.g. the arithmetic of real numbers, and
3. "weight" functions from the primary part to the secondary.

Q: I guess I see how the salary database looks as a metafinite structure. The primary part is the set of employees, the secondary part is rational arithmetic, and the unique weight function assigns employees their salaries. But how do you compute the average salary?

A: In this example, the secondary part would be an extended rational arithmetic, where the arithmetical operations are applied to finite multisets. In particular, one can obtain the total salary of all employees and the number of employees.

Q: How is this related to finite model theory?

A: Quantification is allowed over the primary part only.

We investigated the theory of metafinite structures, with Erich doing most of the mathematical work. We found out that many results of finite model theory survive, sometimes in more than one form, in metafinite model theory.

Q: Do you view metafinite model theory as a separate research area?

A: No. In my view, it belongs to the research area known as finite model theory.

7 Genericity

Q: I presume that tailoring logics to complexity, at least to polynomial time complexity, has been achieved.

A: Not really. A complication arose. The traditional model of computation used to define complexity classes is the Turing machine model. Inputs for Turing machines are strings. Your algorithm may work with graphs or other structures, but you have to encode your input structure as a string to produce input for a Turing machine. We need a more general computation model that works directly with structures, and replacing your input structure with an isomorphic one should not affect the computation.

Q: This looks to me like a generalization that only mathematicians can come with. What on earth has it to do with query languages?

A: In database theory parlance, query languages should be *generic* which means that they should abstract from the physical layer of databases. In this connection, let us call more general computation models that work directly with structures *generic*; also let us call polynomial time defined according to a generic computation model *generic*.

Q: What does it mean to abstract from the physical layer?

A: For simplicity, consider a database that holds just one relation. That relation is a set of tuples. There is no order on the set. Accordingly, a query asking for the first tuple is meaningless. On the other hand, in a memory of a computer hosting the database the tuples are stored in some order.

Q: Fine, take advantage of that order and increase the expressivity of your query language. Specifically, the first-tuple query will return a particular tuple.

A: But the order is accidental. It varies from one database host to another. The first-tuple query may return different tuples on different hosts.

In [9], Chandra and Harel raised an important question. Fix some standard encoding of structures by binary strings and call a Turing machine M *generic* if the set of structures S such that M accepts the standard string encoding of S is closed under isomorphisms. Chandra and Harel asked whether the set of generic Turing machines is recursive. If the answer is positive, then the good Turing machines constitute a generic computation model.

In [29], I conjectured that the answer to the Chandra-Harel question is negative and that no reasonable (satisfying some minimal conditions) logic captures generic polynomial time on structures. The question remains open.

Actually, the situation is a bit strange. In [6, 7] a logic is defined — it is called BGS (after the authors) — such that every BGS definable property of structures is generic polynomial time computable and we don't know any generic polynomial time property that is not BGS definable.

Q: Do you believe that BGS captures generic polynomial time?

A: No, most probably it does not. In [40], Ben Rossman came close to exhibiting a set of structures that is computable in generic polynomial time but is not BGS definable. He constructed a generic polynomial time computable function that (a) given a finite vector space, constructs its dual and (b) is not BGS definable.

8 Postlude

Q: Have you now covered all DBT/FMT developments that you witnessed or participated in?

A: No. One issue is logic programming languages, specifically Prolog and Datalog. That issue is involved and richly deserves a separate conversation.

Q: Did finite model theory cut the umbilical cord to database management?

A: Yes. "DBT was a huge motivator for FMT," wrote Moshe Vardi to me on Nov. 24 2022, "but then FMT moved from PODS [the Symposium on Principles of Database Systems, the premier international conference on the theoretical aspects of database systems] to LICS [ACM/IEEE Symposium on Logic in Computer Science], which is a conference about mathematical foundations."

Q: Is finite model theory still a separate research area?

A: Research areas are social groups to an extent. My impression is that finite model theory becomes less of a separate research area in spite of strong internal social/communal connections. But I am not the right person to answer this question as I had moved on to other research areas quite a while ago.

Acknowledgement

As I worked on this little paper, I exchanged letters with Anuj Dawar, Ron Fagin, Erich Grädel, Neil Immerman, Mark Liogonky, Ben Rossman, and Moshe Vardi. Their timely replies were most helpful. And I much benefited from discussing all aspects of this paper with Andreas Blass. Finally, the photo of Boris Trakhtenbrot with his students comes courtesy of the Trakhtenbrot family archives, the photos of Dana Scott and William Tait come courtesy of Scott and Tait respectively, the photo of Patrick Suppes is from the Guggenheim Foundation, and the photos of Yuri Glebsky and Edgar Codd are from Wikipedia.

References

[1] Serge Abiteboul, Richard Hall, and Victor Vianu, “Foundations of databases,” Addison-Wesley 1995

[2] Arnon Avron, Nachum Dershowitz, and Alexander Rabinovich, “Boris A. Trakhtenbrot: Academic Genealogy and Publications,” in “Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday,” *Lecture Notes in Computer Science* 4800 46–57 Springer 2008

[3] Alfred V. Aho and Jeffrey D. Ullman, “Universality of data retrieval languages,” *ACM Symposium on Principles of Programming Languages* 6 110–117 1979

[4] Miklos Ajtai and Yuri Gurevich, “Monotone versus positive,” *Journal of Association for Computing Machinery* 34 1004–1015 1987

[5] J. Richard Büchi, “Weak second order arithmetic and finite automata,” *Zeitschrift für mathematische Logik and Grundlagen der Mathematik* (now *Mathematical Logic Quarterly)* 6 66–92 1960

[6] Andreas Blass, Yuri Gurevich, and Saharon Shelah, “Choiceless polynomial time,” *Annals of Pure and Applied Logic* 100 141–187 1999

[7] Andreas Blass, Yuri Gurevich, and Saharon Shelah, “On polynomial time computation over unordered structures,” *Journal of Symbolic Logic* 67:3 1093–1125 2002

[8] Egon Börger, Erich Grädel, and Yuri Gurevich, “The classical decision problem,” Springer 1997

[9] Ashok Chandra and David Harel, “Structure and complexity of relational queries,” *Journal of Computer and System Sciences* 25 99–128 1982

[10] Chen Chung Chang and H. Jerome Keisler, “Model theory,” 3rd edition, Elsevier 1990

[11] Calvin C. Elgot, “Decision problems of finite automata design and related arithmetics,” *Transactions of the American Mathematical Society* 98 21–52 1961

[12] Louise H. Chin and Alfred Tarski, “Remarks on projective algebras,” *Bulletin of the American Mathematical Society* 54:1 80–81 1948

[13] Edgar F. Codd, “A relational model of data for large shared data banks,” *Communications of the ACM* 13:6 377–387 1970

[14] ———, “Relational completeness of database sublanguages,” in R. Rustin(editor), “Courant Computer Science Symposium 6: Data Base Systems,” 65–98 Prentice-Hall 1972

[15] Ronald Fagin, “Probabilities on finite models,” *Notices of American Mathematical Society* A714 Abstract 72T-E90 1972

[16] ———, “Contributions to the model theory of finite structures,” University of California Berkeley June 1973

[17] ———, “Generalized first-order spectra and polynomial-time recognizable sets,” in Richard M Karp (editor), “Complexity of Computation,” *SIAM–AMS Proceedings* 7 43–73 American Mathematical Society 1974

[18] ———, “Monadic generalized spectra,” *Zeitschrift für mathematische Logik and Grundlagen der Mathematik* (now *Mathematical Logic Quarterly)* 21 89–96 1975

[19] ———, “A two-cardinal characterization of double spectra,” ibid. p. 121–122

[20] ———, “A spectrum hierarchy,” ibid. p. 123–134

[21] ———, “Probabilities on finite models,” *Journal of Symbolic Logic* 41:1 50–58 1976

[22] ———, “Finite-model theory: A personal perspective," *Theoretical Computer Science* 116 3–31 1993

[23] Yuri V. Glebsky, “Quantitative estimates of the satisfiability of predicate formulas,” *Abstracts of short scientific reports of the 1966 International Congress of Mathematicians* Secton 1 p. 32 Moscow 1966 (in Russian)

[24] Yuri V. Glebsky, Dmitry I. Kogan, Mark I. Liogonky, Vladimir A. Talanov, “Range and degree of satisfiability of formulas in the retricted predicate calculus,” *Cybernetics (now Cybernetics and Systems Analysis)* 5:2 142–154, Russian version 1969, English translation 1972

[25] Yuri Gurevich, “Elementary properties of ordered abelian groups,” *Algebra and Logic* 3:1 5–39 1964 (in Russian), *American Mathematical Society Translations* 46 165–192 1965

[26] ———, “Expanded theory of ordered abelian groups,” *Annals of Mathematical Logic* 12 193–228 1977

[27] ———, “Algebras of feasible functions,” in FOCS 1982, *IEEE Symposium on Foundations of Computer Science* 24 210–214 1982

[28] ———, “Toward logic tailored for computational complexity,” in M. Richter et al. (editors), “Computation and Proof Theory,” *Springer Lecture Notes in Mathematics* 1104 175–216 1984

[29] ———, “Logic and the challenge of computer science,” in E. Börger (editor), “Current trends in theoretical computer sciene” 1–57, Computer Science Press 1988

[30] ———, “On finite model theory,” in S.R. Buss and P.J. Scott (editors), “Feasible Mathematics” 211–219 Birkhäuser 1990

[31] ———, “Metafinite model theory,” *Information and computation* 140:1 26–81 1998

[32] Tomasz Imieliński and Witold Lipski, Jr., “The relational model of data and cylindric algebras ,” *Journal of Computer and System Sciences* 28 80–102 1984

[33] Neil Immerman, “First order expressibility as a new complexity measure,” PhD thesis Cornell 1980

[34] ———, “Number of quantifiers is better than number of tape cell,” *Journal of Computer and System Sciences* 22 384–406 1981

[35] ———, “Upper and lower bounds for first order expressibility,” ibid 25 76–98 1982

[36] ———, “Relational Queries Computable in Polynomial Time,” in STOC 1982, *Symposium on Theory of Computation* 14 147–152 1982

[37] ———, “Descriptive complexity,” Springer 1999

[38] Roger C. Lyndon, “An interpolation theorem in the predicate calculus,” *Pacific Journal of Mathematics* 9 155–164 1959

[39] Mark I. Liogonky and Vladimir A. Talanov, “On the 0-1 law, discovered by Yuri V. Glebsky, and related results obtained in the Mathematical Logic and Algebra Department of the Nizhny Novgorod State University,” *Mathematics in Higher Education* 12 93–102 2014 (in Russian), there seems to be no English translation

[40] Benjamin Rossman, “Choiceless computation and symmetry,” in A. Blass, N. Dershowitz, and W. Reisig (editors), “Fields of logic and computation: Essays dedicated to Yuri Gurevich on the occasion of his 70th birthday,” *Springer Lecture Notes in Computer Science* 6300 565–580 2010

[41] Dana Scott and Patrick Suppes, “Foundational aspects of theories of measurement,” *Journal of Symbolic Logic* 23:2 113–128 1958

[42] William W. Tait, “A Counterexample to a Conjecture of Scott and Suppes”, *Journal of Symbolic Logic* 24:1 15–16 1959

[43] Alfred Tarski, “Der Wahrheitsbegriff in den formalisierten Sprachen,” *Studia Philosophica* I 261–405 1935. English Translation in “Logic, Semantics and Metamathematics” 152–279 Oxford University Press 1956

[44] Alfred Tarski and Frederick B. Thompson, “Some general properties of cylindric algebras,” *Bulletin of the American Mathematical Society* 58:1 p. 65 1952

[45] Boris A. Trakhtenbrot, “The impossibility of an algorithm for the decidability problem on finite classes,” *Proceedings of the USSR Academy of Sciences* 70:4 569–572 1950 (in Russian), *American Mathematical Society Translations* Series 2 23 1–6 1963

[46] ———, “On recursive separability,” *Proceedings of the USSR Academy of Sciences* 88 953—956 1953 in Russian. No English translation is available, but there is an informative review by Andrzej Mostowski in *Journal of Symbolic Logic* 19:1 p. 60 1954

[47] ———, “Finite automata and the logic of one-place predicates,” *Siberian Mathematical Journal* 3 103–131 1962 (in Russian), *American Mathematical Society Translations* Series 2 59 23—55 1966

[48] Moshe Vardi, “Complexity of relational query languages,” in STOC 1982, *Symposium on Theory of Computation* 14 137–146 1982

[49] ———, “On decomposition of relational databases,” in FOCS 1982, *IEEE Symposium on Foundations of Computer Science* 23 176–185 1982

[50] Wikipedia contributors, “Akademgorodok,”
`https://en.wikipedia.org/w/index.php?&oldid=1122095807`

[51] ———, “Edgar F. Codd,”
`https://en.wikipedia.org/w/index.php?&oldid=1125454029`

[52] ———, “Codd’s theorem,”
`https://en.wikipedia.org/w/index.php?&oldid=1068162856`

[53] Moshe M. Zloof, “Query-by-example: a database language,”
IBM Systems Journal 16 324–343 1977