A Dialogue with Yuri Gurevich about Mathematics, Computer Science and Life [*]

Cristian S. Calude,
Department of Computer Science
The University of Auckland
Auckland, New Zealand

1 A Dialogue with Yuri Gurevich about Mathematics, Computer Science and Life

Yuri Gurevich is well-known to the readers of this Bulletin. He is a Principal Researcher at Microsoft Research, where he founded a group on Foundations of Software Engineering, and a Professor Emeritus at the University of Michigan. His name is most closely associated with abstract state machines but he is known also for his work in logic, complexity theory and software engineering. The Gurevich-Harrington Forgetful Determinacy Theorem is a classical result in game theory. Yuri Gurevich is an ACM Fellow, a Guggenheim Fellow, and a member of Academia Europaea; he obtained honorary doctorates from Hasselt University in Belgium and Ural State University in Russia.

[*] Bulletin of the European Association for Theoretical Computer Science, No. 107, June 2012

Cristian Calude: Your background is in mathematics: MSc (1962), PhD (under P. G. Kontorovich, 1964) and Dr of Math (a post-PhD degree in Russia), all at Ural State University. Please reminisce about those years.

Yuri Gurevich: I grew up in Chelyabinsk, an industrial city in the Urals, Russia, and was in the first generation of my family to get systematic education. In 1957, after ten boring years in elementary + middle + high school, I enrolled in the local Polytechnic. I enjoyed student life, but I couldn't draw well, and I hated memorizing things. In the middle of the second year, one math prof advised me to transfer — and wrote a recommendation letter — to the Math Dept of the Ural State University in Ekaterinburg (called Sverdlovsk at the time), about 200 km to the north of Chelyabinsk. One of the Math Dept profs there examined me, and I joined the class of 1962, on the condition that I pass all the math exams taken during the last 1.5 years by my new classmates.

The Math Dept, formally the Dept of Mathematics and Mechanics, was demanding. Typically only a quarter of a class graduated after the five years of study. I did my first little research in classical analysis, with Prof. V.K. Ivanov, the best known Ekaterinburg mathematician. Ivanov was a good man but a busy one, the "prorector" of science. He advised me to go to computational math, because of its potential, or to join an active seminar. "You need interaction," he told me. Computational math seemed pedestrian to me at the time, and I joined the group-theory seminar of Prof. P.G. Kontorovich, the most active and competitive seminar in the Dept, with many enthusiastic participants and a list of open problems prominently posted on the wall. In my 1962 diploma thesis (article #1 at my website[1]) I solved the second problem on the problem list.

CC: P.G. Kontorovich, did he win a Nobel prize in economics?

YG: No, the Nobel prize winner was L.V. Kantorovich. But my Kontorovich was remarkable in his own way. He went from an orphanage to founding the Ekaterinburg algebra school that is active to this very day. His humor was legendary, and he knew seemingly all the languages. Once I found him reading some text and complaining that he understands the text but does not recognize the language it is written in. It turned out that the language was Esperanto, forbidden as a "product of bourgeois internationalism and cosmopolitanism" in the USSR.

[1] Here and below, references #*n* are to the Annotated Articles list at `http://research.microsoft.com/~gurevich/annotated.htm`

Maybe I can use this occasion to say a few words about Ural State University. Compared to other Soviet institutions, my alma mater (at least the hard sciences part of it) was a rare oasis of good will. Senior professors, like Ivanov and Kontorovich, created an atmosphere of decency. Even our philosophical seminars, a necessary fixture in Soviet universities, were different. Typically a philosophical seminar would be devoted to the study of the latest documents of the Central Committee of the Communist Party. The philosophical seminar of our Math Dept was devoted — surprise! — to philosophy, more exactly to the philosophical aspects of mathematics and mechanics. Later in my career, I spoke there about logic at the seminar.

But I am getting ahead of myself. Upon getting my university diploma, I wanted to do math research at a university or the Academy of Sciences which offered better conditions than any university. Conveniently the famous Steklov Math Institute of the Academy of Sciences opened a branch in Ekaterinburg and was hiring, and I applied there. But my chances were slim to none.

CC: Why? You probably were one of the best students or even the best student of your class.

YG: I might have been but Steklov was *Judenfrei*. Even Ural State University had limitations. They accepted me only as a PhD student by correspondence, but they hired me also as a lecturer. It actually worked well for me. I taught about 20 hours a week and did my math. Today it sounds exhausting to me, but at the time I enjoyed it all and had time left to hang out with my dissident friends. I remember even feeling somewhat guilty for being paid to have fun.

CC: What does it mean "PhD student by correspondence"?

YG: This is for people who have regular jobs. They may correspond with the university by mail.

CC: How did you move to mathematical logic? Did you study it at Ural State University?

YG: No, mathematical logic wasn't taught there. In fact there were few math logicians in the whole USSR. Formal (as opposite to dialectical) logic had hard time in the USSR. However things were improving during the 1960s. Kleene's "Introduction to Metamathematics" was translated into Russian, and I got it as a birthday present in May 1962. I studied it and fell in love with logic. But what could an algebraist do in logic?

In the 1962–63 winter, a guest lecturer from Novosibirsk told us that a Polish student of Alfred Tarski, called Wanda Szmielew, proved the decidability of the first-order theory of abelian groups. A natural problem arose whether the first-order theory of ordered abelian groups is decidable. Szmielew and Tarski announced the decidability of that theory but then withdrew their claim. I worked on the problem. A big part of it was to understand when two ordered abelian groups have the same first-order properties. After a long chain of incremental advances, I proved that the theory is indeed decidable (#3). That became my PhD thesis which I defended in the spring of 1964 in Novosibirsk.

CC: Why Novosibirsk? Ural State University is not in Novosibirsk.

YG: By Soviet rules, you could defend your thesis in a science area X only at an institution with sufficient expertise in X. My choice was restricted to Moscow, Leningrad and Novosibirsk. Because of Maltsev's "Algebra and Logic" seminar, Novosibirsk was the best fit for me.

The 1964-65 academic year I was teaching at a new Krasnoyarsk State University in Siberia. By the way the word "State" in the names of Soviet universities meant simply "of the Soviet state". In the middle of that academic year I attended an algebraic winter school near Ekaterinburg. There I met a third-year Ural State University student Zoe, and I returned to Krasnoyarsk with a wife. We sought to move back to Ekaterinburg, and Ural State University accommodated us; the 1965-66 academic year I was already teaching there. My obsession with logic was contagious, and the logic seminar attracted the brightest students. During the winter breaks, we would rent a little house in the country to study but also to ski, play charades, etc.

CC: It sounds like scientific life in Soviet Union was similar to that in the West.

YG: It was similar, at least where hard sciences were concerned. But there were important differences. We were poorer. For example, Ural State University had no foreign currency, and western books and journals were not available in the library. More importantly, the totalitarian state was never far away. Here is an incident from one of those winter schools. One morning I woke up to much noise in another room, with none of my roommates in my room. I went to investigate. Two boys, surrounded by all the other students, were arguing whether there was state anti-Semitism in the USSR. Now all the eyes were upon me. What could I say? The safe lie of denial was out of the question, but publicly accusing the state of anti-Semitism was too dangerous, especially for a teacher. The chances were that there was an informant present. I spoke and spoke trying to humor my audience. I used whatever parables and jokes occurred to me leaving it up to the students to interpret things. Eventually passions subsided, and the attention deviated to other topics. And I remember wishing to be able to speak my mind.

But science and life interacted also independently of politics. Upon our return to Ekaterinburg, I had a bad motorcycle accident. In the hospital, they sewed me up but inadvertently infected me with hepatitis. As a result, I was quarantined for a month. No visitors were allowed in, and there were few books to read there. I used the time to think about the classical decision problem — classify infinite fragments of first-order predicate logic, given by restrictions on quantifier prefixes and the vocabulary, into decidable (for satisfiability) and undecidable. The problem attracted the attention of great logicians including Gödel, and there had been much progress in the early 1960s. If only one could prove that the $\forall\exists\forall\exists^*$ fragment with one binary relation is undecidable, the classification would be complete. The $\forall\exists\forall\exists^*$ problem was uniquely appropriate to my confinement. While the decision problem for ordered abelian groups required a long sustained effort and a long sequence of lemmas, each building upon the previous ones, the $\forall\exists\forall\exists^*$ problem seemed to require just a clever combinatorial trick. It was like jumping over a barrier. You give it a try and you fall, then another try and another fall, over and over again. Indeed, by the end of my quarantine, I got lucky and jumped over that barrier. The fame of the problem helped me to defend my Dr. of Math thesis later, in 1968.

CC: What is the Dr. of Math degree for? The Russian system of academic degrees seems different from that in English-speaking countries.

YG: It is different. The first Russian postgraduate academic degree, an equivalent of PhD, is Candidate of Science, and the second is Doctor of Science. Here "Science" is a variable to be replaced with "Mathematics", "Physics", etc. The Dr. of Science degree was a big deal at the time. If you taught at a university, the degree was a necessary and, in practice, sufficient condition for getting a full professorship. All academic degrees in Russia were — and are — subject to approval by the Central Attestation Committee of Russia.

CC: This was and continue to be also the system in Romania: nowadays, this Committee includes also Romanians from diaspora.

YG: The system is supposed to impose some standards but of course it can be abused.

CC: Did you go to Novosibirsk to defend your Dr. of Math thesis.

YG: No, the atmosphere in Novosibirsk changed for the worse, and a "Jewish dissertation" had little chance there. My dissertation had also a large algebraic component and thus qualified as algebraic. I defended it in Ekaterinburg, and the degree was eventually approved by the Central Attestation Committee.

CC: Your scientific activity splits into three periods: Soviet (up to 1973), Israeli (1974–1981), and American (since 1982). Let's visit them in that order.

YG: During the Soviet period I worked primarily on two subjects. One was related to the classical decision problem. The complete classification mentioned above comprised nine minimal undecidable classes and three maximal decidable ones. I wanted to understand whether there was an a priori reason that the classification resulted in a finite table. It turned out that indeed there was a rather general reason. That encouraged me to work on the extensions of the classification to first-order logic with equality or function symbols or both. I made a good progress, and the Institute of Philosophy of the Russian Academy of Sciences asked me to write a book on the subject. I write too slow to produce a book, but I wrote a survey. It was withdrawn from publication upon our emigration from the USSR. Later the survey became the core of the 1997 Springer book "The Classical Decision Problem" by Egon Börger, Erich Grädel and myself.

The other subject was the decidability of algebraic theories. In particular, I continued my work on ordered abelian groups. It bothered me that theorems in the literature on the subject were not first-order; they were mostly in terms of so-called convex subgroups. I extended my analysis to the variant of the monadic second-order theory of ordered abelian groups where the set variables ranged over convex subgroups. Somewhat miraculously, the decision procedure not only survived but simplified. The extended theory (and its easy further extensions) accounted for virtually all theorems in the literature. My attempts to publish these results in the USSR were unsuccessful (which is a separate story) but I published them after my departure (#25).

I also did some applied work. In my later undergraduate years, I worked at the university computing center. Later I worked with the transportation industry on linking railway transportation to trucks. All that work influenced me and changed my attitude on pure vs. applied science. You may have heard about a mathematician working on a difficult four-legged table problem. He generalized the problem to n-legged tables and solved the cases $n \leq 2$, the case $n = \infty$ and the case of sufficiently large n. In the process he advanced his career but the original problem remained open. That's pure science ☺

CC: Now tell me about the Israeli period.

YG: That period started with a touch of drama, or comedy. The first few months we lived in Jerusalem and studied Hebrew. During my first trip to Hebrew University, I met a young logician, Saharon Shelah. "Do you have an open problem," he asked me. I told him my conjecture that the $\exists^*\forall\exists^*$ fragment of first-order logic with equality, one unary function and infinitely many unary relations is decidable for satisfiability. When I saw him again, a week or two later, he told me that he confirmed my conjecture. I smiled: "Tell me about it." He did. I could not follow his explanation, partially because my Hebrew was still insufficient and my English nonexistent, but I realized that he had all the intuition that led me to the conjecture and more. I was stunned. The first Israeli mathematician that I had a serious discussion with confirmed my conjecture. Maybe I should not seek a university position in Israel. I asked Shelah whether he had an open problem. He gave me his paper on the monadic second-order theory of the real line; it was submitted to Annals of Mathematics and had many open conjectures.

The paper was full with original ideas, but it was difficult to read. It took me months just to understand the paper. After a year or so of hard work, I confirmed or refuted most of Shelah's conjectures. He was most kind; as he proofread his paper, he added footnotes announcing my results. The incident resulted in a fruitful collaboration with Shelah on monadic (second-order) theories. Survey #64 reflects a large initial segment of the results of the monadic project.

CC: Give me some flavor of that work.

YG: Shelah conjectured that countability is not definable in the monadic second-order theory MT($\mathcal{R}$) of the real line $\mathcal{R}$ with just the order relation (and no addition or multiplication). In this connection I thought of the known and unsuccessful attempts to define countability in measure-theoretic terms. Of course sets of Lebesgue measure zero can be uncountable, but also sets of universal measure zero (defined by Hausdorff) can be uncountable, and sets of strong measure zero (defined by Borel) can be uncountable under the continuum hypothesis. I expected the conjecture to be true but it turned out, somewhat surprisingly, that countability was definable in MT($\mathcal{R}$) under the continuum hypothesis. The construction built heavily on the methods developed by Shelah in his original paper.

One of the main results in Shelah's original paper was the undecidability of MT($\mathcal{R}$). The proof was a clever interpretation of first-order arithmetic in MT($\mathcal{R}$). In #57, Shelah and I interpreted second-order arithmetic in MT($\mathcal{R}$). Later, in apparent contradiction with these results, we discovered that first-order arithmetic, let alone second-order arithmetic, cannot be interpreted in MT($\mathcal{R}$) (#79). A closer examination of Shelah's original reduction revealed that it (and our generalization of it) went beyond the standard model-theoretic notion of interpretatibility. And there was an interesting connection to set theory. If W is a model of ZFC, let W' be the model of ZFC resulting from the extension of W with a Cohen real, a real number that does not exist in W. Paul Cohen discovered a technique, *forcing*, that allows one to do things like that. Think of W as the current set-theoretic world, and of W' as the next world. Our reduction in #57 was a reduction of the next-world second-order arithmetic to the current-world MT($\mathcal{R}$).

CC: Not too many mathematicians or computer scientists have a theorem bearing their name. Tell me about the Gurevich-Harrington Forgetful Determinacy Theorem and how did you arrive at it.

YG: The 1980–81 academic year was a logic year at Hebrew University. Both Leo Harrington and I were there and proved the theorem independently; we talked about that, and I volunteered to write the theorem up for publication. I do not know Leo's motivation. On my side, laziness played a role. In 1969, Michael Rabin used nondeterministic finite automata on infinite (colored) trees to prove the decidability of S2S, the monadic second-order theory of two successor relations. I understood the proof except for the complementation lemma according to which, for every tree automaton A, there is a complementary tree automaton that accepts exactly the trees that A doesn't. I kept thinking about the lemma but was reluctant to go through the difficult proof. And one day it occurred to me that it all, not only the complementation lemma but the whole paper of Rabin, was really about games. Things simplify (and become amenable to new useful generalizations) if you see them that way. For the games in question, the players can restrict themselves to "forgetful" strategies so that, at every point, the players need to remember only boundedly many bits about the history of the current play. Even finite automata are able to execute forgetful strategies; hence Rabin's result.

CC: Eventually you moved to the United States and to computer science. How did that happen?

YG: I had been contemplating more applied research already at the end of my Russian period but the Jerusalem logic seminar enthralled me. In spite of solving some high-profile logic problems, I was really a logic ignoramus. The seminar allowed me to learn cutting-edge logic developments. It was so much more efficient and so much more fun to learn things from seminar presentations than by reading papers. It was in Israel that I really became a logician, thanks to the logic seminar and joint work with Shelah. When the monadic project with Shelah began to wind down, I applied to computer science departments at some Israeli and US universities. All offers came from the US. I accepted a good offer from the University of Michigan, and in the summer of 1982 we moved to Ann Arbor, Michigan. There was another reason to choose the University of Michigan. Andreas Blass, the logician, was there, albeit in the Math Dept. Andreas and I have been actively collaborating ever since.

CC: Tell me about your work in finite model theory.

YG: Let me restrict myself to just one little story. At my first computer science conference, I heard a presentation by Moshe Vardi. He applied the interpolation theorem of first-order logic to relational databases viewed as first-order structures. I asked him whether his databases could be infinite, and he said yes. But naturally databases are finite of course. I looked into the issue. As I suspected, most classical theorems of first-order logic, including the interpolation theorem, fail in the finite case (#60). I had a sense of déjà vu. First-order logic wasn't right for ordered abelian groups, and it wasn't right for finite structures in the computer science context (#74).

Later on, a realization came that real databases are not necessarily finite after all. For a simple example, consider a salary database of some organization. The organization may use a popular database-query language SQL to query the salary database. In addition to relational-algebra operations, SQL has so-called grouping and aggregation operations. This allows the organization to compute various statistics over the database, e.g. the average salary and the total salary expense of the organization. Note that the average salary may not occur in the database and, ignoring degenerate cases, the total salary surely does not occur. Thus the database gives us a function from the employees to numbers, say rational numbers, and has rational arithmetic in the background. In that sense, it is not truly finite. To formalize this phenomenon of finite foreground and infinite background, Erich Grädel and I introduced *metafinite* structures (#109). The metafinite phenomenon is not restricted to databases. The states of an algorithm often are metafinite. Most classical theorems of first-order logic, including the interpolation theorem, fail in the metafinite case.

CC: Finite model theory has intimate relations with computational complexity but your complexity work went beyond that.

YG: It did. In particular I worked on the average-case reduction theory pioneered by Leonid Levin. Consider NP complete problems equipped with probability distributions on the instances. Some such problems turn out to be easy on average but others remain complete even for the average case. Proving such average-case completeness results is difficult, and the reason is this. While the range of a worst-case reduction may consist of very esoteric and unrepresentative instances of the target problem, the range of an average-case reduction should be of non-negligible probability. A popular article #85 argues in favor of an alternative, based on the average-case complexity, to the P=?NP question. Consider a game between Challenger and Solver where Challenger repeatedly picks instances of a given NP problem (with a fixed probability distribution), and Solver solves them. The idea is to measure Solver's time in terms of Challenger's rather than in terms of the instance size. It may take a long time to produce hard instances.

CC: Tell me about your work on abstract state machines. In particular what motivated it?

YG: Right upon starting at Michigan, I volunteered to teach "Introduction to Computer Science with Pascal" to computer science majors. The Dept chair did not like the idea ("We hired you to teach theory.") but agreed that I teach the course once. Preparing that course was instructive. I had not realized how much I fell behind in programming technology. At Ural State University, I programmed on the naked machine (01 for addition, 02 for substraction, etc.), and Pascal seemed advanced. The troubling part was that Pascal wasn't sufficiently documented. The interpreter on my Macintosh and the compiler on the university mainframe often disagreed on whether a given program is legal. Which of them, if either, was right? What was I supposed to tell my 250 or so students? That was scary and brought home the problem of the semantics of programming languages.

In this connection, I studied denotational and algebraic semantics but found them wanting. It seemed infeasible to use them to specify the "dirty parts" of software. The celebrated declarativeness of denotational and algebraic specifications did not impress me. The advancers of the computer revolution weren't shy to program, specify and reason imperatively. There is a persistent confusion between declarative and high-level. Declarative specifications tend to be high-level, and executable specifications tend to involve unnecessary details. However I saw no reason why high-level specifications cannot be imperative and executable, amenable to testing and experimentation.

By Turing's thesis, every algorithm can be simulated by an appropriate Turing machine. Are Turing machines executable? In principle yes but of course this may be impractical. A bigger problem is that Turing machines work on the level of single bits. Are there more general state machines that specify algorithms on their natural abstraction level? Maybe that was too much to ask. But if yes then the reward would be high, for theory and practice. It would open a road to formalizing the notion of algorithms. On the practical level, it would enable us to specify software on whatever abstraction level is desired.

It was that line of thought that led me eventually to abstract state machines (ASMs). By the ASM thesis, every algorithm can be faithfully simulated by an ASM. We attempted to verify the thesis, which led to practical applications. There were also theoretical advances. The notion of sequential algorithms was formalized in #141; this formalization was used later by Nachum Dershowitz and myself to derive Turing's thesis from first principles (#188). Parallel and interactive algorithms were also formalized (#162).

CC: How did you get attracted to Microsoft?

YG: I was convinced that the ASM approach was more practical than other formal methods but all methods work on small examples, and my attempts to find an industrial partner were unsuccessful. In the summer of 1998, I visited Microsoft Research (MSR), in Redmond, WA, by an invitation of their crypto group. On that occasion I volunteered an ASM lecture. The lecture went rather well. There were many good questions. One of the MSR directors, Jim Kajiya, asked particularly astute and pointed questions. He said that he was surprised to see a formal specification method that seemed scalable. He proposed me to start a new MSR group on foundations of software engineering, and I jumped at the opportunity. The atmosphere and conditions at MSR are great, and the geographical area is spectacular. But what attracted me most was of course the opportunity to apply ASMs.

CC: Did it work? Could you apply ASMs at Microsoft?

YG: It was tough. I was lucky to hire the right people, and we built a tool, Spec Explorer, that facilitated writing executable specifications and playing with them. In particular, one could test the conformance between a spec and implementation. Spec Explorer was kept compatible with the Microsoft technology stack which consumed a lot of time and effort. The tech transfer was the biggest challenge. It is relatively easy to "sell" an incremental improvement to product groups. But Spec Explorer required learning and training, and product groups are busy. For a while we had only a few courageous groups here and there using Spec Explorer with our help. At a certain point, the European Union required from Microsoft high-level executable specifications of numerous communication protocols. The Windows division took over Spec Explorer and used it extensively and successfully.

CC: How applied is your work at Microsoft now? Do you use some theoretical results you proved as a "blue-sky researcher"?

YG: When Spec Explorer left MSR, I spent a couple of years catching up with theoretical work but then I returned to applications. Microsoft is an engineering place, and you catch the bug and want to influence technology. From time to time, I do internal consulting, developing efficient algorithms for various purposes. But my main current occupation is with Distributed Knowledge Authorization Language (DKAL). With the advent of cloud computing, a policy-management problem arises. In a brick-and-mortar setting, many policies may be unwritten. Clerks learn them from their colleagues. If they don't know a policy, they know whom to ask. In the cloud, the clerks disappear. The policies have to be handled automatically. The most challenging aspect is how to handle the interaction of policies, especially in federated scenarios where there is no central authority. DKAL was created to deal with such problems. The DKAL project has a large logic component so my logic expertise is useful.

CC: If you could dream about the year 3012, which result or concept would you like to see still "alive"?

YG: Hmm. "It's tough to make predictions, especially about the future," said Yogi Berra, the famous American baseball player and a philosopher of a kind. We live in quickly changing times. In the computer industry, long-term refers to just a few years ahead. It is an interesting question to what extent the future is predictable, even probabilistically. Let me just express the hope that the humanity will survive till 3012 and that the scientific method will survive as well. It may seem that the second is obvious given the first, but it is not necessarily so. Lucio Russo convincingly argues in "The Forgotten Revolution: How Science Was Born in 300 BC and Why it Had to Be Reborn" (2004) that the scientific method was not invented but reinvented by Galileo, Newton and their contemporaries, that science was discovered in the Hellenistic period and then was forgotten.

CC: How do you see the relevance of theoretical computer science for the computer technology?

YG: Theory made weighty contributions to computer technology. Think of Alan Turing, John von Neumann, modern cryptography. The search technology that made Google rich is based on clever algorithms. One important theoretical contribution is for some reason less known to theorists than it deserves; I searched for it in vain in computation theory books. It is the 1965 discovery of LR(k) languages by Donald Knuth: "A language can be generated by an LR(k) grammar if and only if it is context-free and deterministic, if and only if it can be generated by an LR(1) grammar." LR(k) grammars can be parsed in time essentially proportional to the length of string, and their discovery revolutionized compiler construction.

But it is hard to influence computer technology by advancing theory, especially if the result is a non-incremental change in technology. "Nothing is more difficult than to introduce a new order," writes Niccolo Machiavelli in *The Prince*, "Because the innovator has for enemies all those who have done well under the old conditions and lukewarm defenders in those who may do well under the new." I lifted this quotation from a 2006 book "The Change Function: Why Some Technologies Take Off and Others Crash and Burn." The author, Pip Coburn, argues that the chances of adoption of a new disruptive technology is given by

$$\frac{\text{pain of the crises}}{\text{pain of adoption}}$$

To achieve successful technology transfer starting from just a theoretical advance is harder yet (though one may get lucky).

CC: Many thanks.