Yuri, Logic, and Computer Science

Andreas Blass[1], Nachum Dershowitz[2], and Wolfgang Reisig[3]

[1] Mathematics Department, University of Michigan
Ann Arbor, MI 48109–1043, U.S.A.
[2] School of Computer Science, Tel Aviv University Ramat Aviv 69978, Israel
[3] Humboldt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6,
10099 Berlin, Germany

Yuri Gurevich was born on May 7, 1940, in Nikolayev, Ukraine, which was a part of Soviet Union at the time. A year later, World War II reached the Soviet Union, and Yuri's father was assigned to work in a tank body factory near Stalingrad. So that's where Yuri spent the second year of his life, until the battle of Stalingrad forced the family, except for his father, to flee. Their home was destroyed by bombing only hours after they left. But fleeing involved crossing the burning Volga and then traveling in a vastly overcrowded train, in which many of the refugees died; in fact, Yuri was told later that he was the only survivor among children of his age. His mother decided that they had to leave the train, and the family lived for two years in Uzbekistan. In May 1944, the family reunited in Chelyabinsk, in the Ural Mountains, where the tank body factory had moved in the meantime, and that is where Yuri attended elementary and high school.

An anecdote from his school days (recorded in [123][4]) can serve as a premonition of the attention to resources that later flowered in Yuri's work on complexity theory. To prove some theorem about triangles, the teacher began with "Take another triangle such that ...." Yuri asked, "Where does another triangle come from? What if there are no more triangles?" (commenting later that shortages were common in those days). For the sake of completeness, we also record the teacher's answer, "Shut up."

After graduating from high school, Yuri spent three semesters at the Chelyabinsk Polytechnik. Because of a dissatisfaction with the high ratio of memorization to knowledge in the engineering program, Yuri left after a year and a half and enrolled in Ural State University to study mathematics.

Yuri obtained four academic degrees associated with Ural State University: his master's degree in 1962, his candidate's degree (equivalent to the Western Ph.D.) in 1964, his doctorate (similar to habilitation, but essentially guaranteeing an appointment as full professor) in 1968, and an honorary doctorate in 2005. At Ural State University, Yuri ran a flourishing logic seminar, and he founded a Mathematical Winter School that is still functioning today. It should also be noted that the four-year interval between the candidate's and doctor's degrees was unusually short.

[4] Numerical references are to the annotated bibliography in this volume; they match the numbering on Yuri's web site.

That four-year interval contained some very important non-academic events. Yuri and Zoe were married in March 1965, and their twin daughters, Hava and Naomi, were born in October 1966.

As mentioned above, once he had his doctorate (in the Russian sense), Yuri would ordinarily get a professorship, but a glance at his curriculum vitae shows that he overshot a bit, becoming not only a professor but chair of the Mathematics Department at the National Economy Institute in Sverdlovsk in 1969. This does not indicate a great enthusiasm for administrative work; in fact, though he's very good at administration, Yuri is not very fond of it. Nor does it indicate great interest in economics. What made this appointment extremely attractive is that it provided an apartment – a very important benefit in the Soviet Union, especially for a man with a young family.

Because of the political situation in the Soviet Union, Yuri and Zoe decided to move to Israel. That rather dangerous journey took them to Krasnodar and then to Tbilisi, the capital of beautiful and hospitable Georgia. Eventually they emigrated to Israel in October 1973.

During the period from his master's degree until his departure from the Soviet Union, Yuri established himself as a first-rate algebraist and logician. Already in his master's thesis [1], he solved an open problem in group theory, but his most important work from this period, at the interface of logic and algebra, concerned ordered abelian groups. His 1964 thesis for the candidate's degree [3] proved the decidability of the first-order theory of these groups; later he obtained decidability of the richer theory that includes quantification not only over elements but also over convex subgroups. This richer theory includes essentially all the questions that had attracted the attention of researchers in ordered abelian groups. It is fair to say that this work [19,25] of Yuri's subsumed that entire field.

In addition to this work on ordered abelian groups, Yuri made fundamental contributions to the decision problem for first-order logic. In particular, he completed [6,7,13] the analysis of decidability of classes of first-order formulas of the form "specified quantifier prefix and specified vocabulary." Yuri's work from that period contains other contributions, for example the undecidability of the first-order theory of lattice-ordered abelian groups [9], but there is also an important but non-technical contribution that must be mentioned here.

We quote part of a toast offered by Yuri's first Ph.D. student, Alexander Livchak, at a 2010 anniversary celebration of the Faculty of Mathematics of the Ural State University:

> Gurevich taught us to think freely. It was helpful that his specialty was logic – the science of proofs. He tried unobtrusively to impress upon us that the final judgment is ours and not that of the Central Committee of the Communist Party or that of Marx–Engels.
>
> It all started with a seminar on axiomatic set theory. The idea of a winter school was born there. The schedule of the Winter Math School included not only studies but also mandatory daily skiing and various entertainment activities. For example, Gurevich liked debates à la me-

> dieval scholastic disputes. He would volunteer to argue any ridiculous and obviously false thesis of our choice in order to demonstrate the art of arguing. Therein lay his secret "counter-revolutionary Zionist" (in the terminology of the time) plot: to teach us to argue, doubt, prove, refute. In general to teach us to think independently.

Yuri lived in Israel from 1973 to 1981, teaching at Ben-Gurion University of the Negev in Beer-Sheva, except for leaves of absence spent at Simon Fraser University in Vancouver, the Hebrew University of Jerusalem, and Bowling Green State University in Ohio. Very soon after his arrival in Israel, he impressed people by solving several problems posed by Saharon Shelah. This work and other results from his Israeli period concerned the monadic theory of linear orders, either in general or in the context of specific linear orders like the real line. One of these results is that, if the continuum hypothesis holds, the countability of subsets of the real line can be defined in monadic second-order logic. That work led to the first of numerous deep joint papers with Shelah. It also led to connections with the theory of ordered (non-abelian) groups, especially groups of automorphisms of linear orders.

Another major contribution from Yuri's Israeli period (although the paper [40] was prepared and published after Yuri was in the U.S.) is the Gurevich-Harrington theorem. This theorem concerns the existence of winning strategies in certain infinite games. In various contexts (mostly in topology) people had considered strategies that look only at the opponent's immediately previous move (rather than the whole history of the play) or a fixed number of previous moves. Yuri and Leo Harrington showed that, for many games, the winning player has a strategy that remembers, at any stage, only finitely much information from the past, though there is no bound on how long ago that information might have appeared. They used this result to greatly simplify the hardest part of Michael Rabin's proof of the decidability of the monadic theory of two successor functions.

Yuri spent the academic year 1981–82 as a visiting professor at Bowling Green State University in Ohio, which was at that time a major center of research on ordered groups; Andrew Glass and Charles Holland were on the faculty there. In addition to research on ordered groups, Yuri resumed thinking about computer science, which he had already been interested in even in the Soviet Union. He sought a computer science position in Israel or the U.S.

In 1982, Yuri accepted an appointment as professor of computer science[5] at the University of Michigan.[6] Yuri took his conversion to computer science

[5] Technically, he was professor of Computer and Communication Sciences since that was the name of the department. A subsequent reorganization put him and his fellow computer scientists into the engineering college, in the Computer Science and Engineering Division of the Department of Electrical Engineering and Computer Science. He now holds the title of Professor Emeritus of Electrical Engineering and Computer Science.

[6] An observation by Andreas Blass: The angriest I've ever seen Andrew Glass is when he talked about Bowling Green's failure to make a serious effort to keep Yuri.

seriously. He did not just write mathematics papers with a computer science facade. Although he finished various mathematical projects, he immediately began thinking deeply about computational issues and making significant contributions to his new field. Furthermore, with the enthusiasm of a new convert, he began the difficult project of trying to convert Andreas Blass to computer science. The project didn't entirely succeed; Blass still claims to be a set-theorist, but he certainly learned a great deal of computer science from Yuri.

Yuri's contributions to computer science span a vast part of that field, and we can mention only a few of them here. First, there are many results in complexity theory, but to appreciate this part of Yuri's work it is necessary to take into account the great variety of topics that fall under this heading. There is traditional complexity theory, largely directed toward the P vs. NP question but also covering many other aspects of polynomial-time computation. But there is also a strong connection to probabilistic issues, both in connection with the use of randomness in computation and in connection with average-case (as opposed to worst-case) complexity of algorithms. Yuri made important contributions in all these areas, including good explanations of Leonid Levin's theory of average-case complexity and natural complete problems for this theory. He also investigated far stricter resource bounds, including linear time, and in a joint paper [82] with Shelah showed that the notion of "linear times polylog time" is remarkably robust across different models of computation as long as one excludes ordinary Turing machines.

A second broad area of Yuri's research in computer science is connections between computation and logic, and this, too, spans several sub-areas. A particularly important contribution is Yuri's emphasis on the computational relevance of finite structures. Classical logic is greatly changed by restricting attention to finite structures, mainly because the compactness theorem, one of the chief traditional tools, becomes false in this context. Although there had certainly been earlier work on finite structures, Yuri's papers [60] and [74] led to a major increase of interest and activity in this field. Yuri also formulated in [74] the main open problem about the connection between logic and complexity, namely his conjecture that there is no logic that exactly captures polynomial time computability on unordered structures. (Part of the contribution here is making the conjecture precise by saying what should be meant by a "logic" in this context.)

Yuri's contributions to the interface between logic and computer science also include studies of Hoare logic and (motivated by better compatibility with Hoare logic) existential fixed-point logic. Another of Yuri's contributions is the introduction, in joint work with Erich Grädel [109], of metafinite model theory, in which models are primarily finite but are allowed to have a secondary, infinite part so that such operations as counting can be accommodated in a natural way.

We should also mention here Yuri's "Logic in Computer Science" column in the *Bulletin of the European Association for Theoretical Computer Science*. Here we find not only a great number of interesting columns written by Yuri himself and exploring the most diverse areas that could fit under the "logic and computer science" heading, often in the form of a Socratic dialogue with

his friend and disciple, "Quisani," but also columns that he solicited from other experts. The columns, later collected along with other *BEATCS* material in three books titled *Current Trends in Theoretical Computer Science*[7] make fascinating reading. They are not all just surveys either; Yuri sometimes used the column to present his new results. An outstanding example is [131], in which he proves that, if we could compute, in polynomial time, a complete isomorphism invariant for graphs, then we could also compute in polynomial time, from any graph as input, a standard representative (a canonical form) of its isomorphism class.

In terms of subsequent impact, Yuri's biggest achievement during his Michigan period was the invention of abstract state machines[8] (ASMs) and the ASM thesis. ASMs are an extraordinarily clean and general model of computation. Here "clean" means that ASMs have a simple, unambiguous semantics; "general" means that they can easily simulate a huge variety of computations, ranging from high-level algorithms down to hardware. Yuri proposed the "ASM thesis" that *every* algorithm can be faithfully represented, at its natural level of abstraction, by an ASM. Initial support for the thesis came from numerous case studies, in which Yuri, his students, and others gave ASM descriptions of a wide variety of software and hardware systems as well as abstract algortihms. Later, support came from rigorous proofs, but this is getting ahead of the story.

In 1998, Yuri joined Microsoft Research as a senior researcher, with the job of bringing ASMs into the real world of large-scale software development. The move from Michigan to Microsoft happened remarkably fast; the decision was made in August and involved getting a leave of absence from Michigan for the fall semester, which begins in early September. Fortunately, the relevant administrators at Michigan granted the leave, fully expecting that Yuri would soon return, especially because his job at Microsoft involved building a new research group from scratch. But Yuri handled his new administrative duties well, hiring several first-rate researchers, and he really enjoyed (and continues to enjoy) the knowledge that his work is having an impact on real computing. So, after two years on leave from Michigan, he officially retired. He has now been a professor emeritus for ten years.

Among Yuri's many contributions while at Microsoft, we describe just a few, of which several involve ASMs. Perhaps the most surprising is his discovery that, in certain contexts, the ASM thesis can actually be proved. In [141], Yuri presented some simple, natural postulates about sequential, non-interactive algorithms; argued that they are satisfied by anything that one would intuitively consider to be such an algorithm; and then proved that anything satisfying the postulates is equivalent, in a very strong sense, to an ASM (of a particular sort). That result has since been extended to parallel algorithms in [157-1,157-2], to interactive sequential algorithms in [166,170,171,176,182], and to a notion of equivalence that keeps track of exactly which part of the state is relevant at each step in [201].

---

[7] Edited by G. Rozenberg, A. Salomaa, and (for the last two) G. Paun; published by World Scientific in 1993, 2001, and 2004.

[8] Originally called "dynamic structures" and subsequently "evolving algebras."

A second, quite different, use of ASMs occurs in describing "choiceless polynomial time" computation. Here, the input to a computation is an unordered structure, and the computation is allowed to use parallelism and essentially arbitrary data structures, but it is not allowed to make arbitrary choices of elements (or, equivalently, to linearly order the input structure). This concept, originally introduced with quite a complicated definition by Shelah, turned out to be equivalent to computation by a rather standard sort of ASM over a structure consisting of the original input plus all hereditarily finite sets over it. The hereditarily finite sets capture the arbitrary data structures, and the ASMs take care of the rest of the computational issues. By itself, choiceless polynomial time is rather weak; it can't even count [120]. But when extended by counting, it is a surprisingly strong logic [150] and in fact is one of very few logics that might possibly capture polynomial-time computation on unordered structures (though that seems very unlikely).

Yet another use of ASMs is the proof of Church's thesis [188] on the basis of natural assumptions about computability.

Another aspect of Yuri's contribution to computer science is that he accurately assesses the quality of people's work and acts on the basis of that assessment. We omit the negative examples, to avoid unnecessary controversy, but describe one positive example, as seen by Blass. Ben Rossman, while taking a year off after finishing his undergraduate degree, solved an old problem of Yuri's and sent him the solution. Many people, getting a rather difficult-to-read manuscript, out of the blue, from someone with no credentials, would be inclined to ignore it, but Yuri read it carefully, decided it was correct, and told Blass about it enthusiastically. Not long afterward, Yuri, Rossman, and Blass were all at a LICS conference, and Rossman, who had met Blass a few months earlier at another conference but had not yet met Yuri, mentioned to Blass that he was looking for something interesting to do in the following summer. Blass immediately thought of Microsoft Research, but not of Yuri's group, which was at that time heavily engaged in very applied work, far from Rossman's theoretical interests. But there might be a possibility of an internship with Microsoft's Theory Group, so Blass suggested that Rossman check with Yuri about that possibility. Yuri promptly offered Rossman a visiting position in his group, and this unusual investment in theory paid off in several significant contributions by Rossman to the group's work [169,176,182].

Among Yuri's other recent technical contributions are work on efficient file transfer [183,190], on software testing [154,160,163,173], on security assessment [202], and on decentralized authorization [191,198,200].

There is a great deal more to be said about Yuri's work, in both mathematics and computer science, his generosity toward colleagues and students, and his amazing energy level. But this is being written while the rest of this volume is ready to go to the publisher, so we'll stop here. Some more information can be found in Jan Van den Bussche's contribution to this volume, and an indication of the community's admiration for Yuri can be inferred from the size of this volume.

Annotated List of Publications of Yuri Gurevich

The following list of publications and annotations is derived from Yuri Gurevich's website,[1]

`http://research.microsoft.com/en-us/um/people/gurevich/annotated.htm`.

Abbreviations:

BEATCS	= Bulletin of the European Association for Theoretical Computer Science
JSL	= Journal of Symbolic Logic
LNCS	= Lecture Notes in Computer Science
MSR-TR-Y-N	= Microsoft Research Technical Report number N of year Y
ACM ToCL	= ACM Transactions of Computation Logic
Doklady	= Doklady Akademii Nauk SSSR (Proceedings of the USSR Academy of Sciences)

0. Egon Börger, Erich Grädel, Yuri Gurevich: The Classical Decision Problem. Springer Verlag, Perspectives in Mathematical Logic, 1997. Second printing, Springer Verlag, 2001. Review in Journal of Logic, Language and Information 8:4 (1999), 478–481. Review in ACM SIGACT News 35:1 (March 2004), 4–7.

   The classical decision problem is (in its modern meaning) the problem of classifying fragments of first-order logic with respect to the decidability and complexity of the satisfiability problem as well as the satisfiability problem over finite domains. The results and methods employed are used in logic, computer science and artificial intelligence.

   The book gives the most complete and comprehensive treatment of the classical decision problem to date, and includes an annotated bibliography of 549 items. Much of the material is published for the first time in book form; this includes the classifiability theory, the classification of the so-called standard fragments, and the analysis of the reduction method. Many proofs have been simplified and there are many new results and proofs.

1. Yuri Gurevich: Groups covered by proper characteristic subgroups. Trans. of Ural University 4:1 (1963), 32–39 (Russian, Master Thesis)

2. Yuri Gurevich, Ali I. Kokorin: Universal equivalence of ordered abelian groups. Algebra and Logic 2:1 (1963), 37–39 (Russian)

   We prove that no universal first-order property distinguishes between any two ordered abelian groups.

3. Yuri Gurevich: Elementary properties of ordered abelian groups. Algebra and Logic 3:1 (1964), 5–39 (Russian, Ph.D. Thesis)

   We classify ordered abelian groups by first-order properties. Using that classification, we prove that the first-order theory of ordered abelian groups is decidable; this answers a question of Alfred Tarski.

[1] The editors thank Zoe Gurevich for her help.

3a. Yuri Gurevich: Elementary properties of ordered abelian groups. AMS Translations 46 (1965), 165–192

This is an English translation of [3].

4. Yuri Gurevich: Existential interpretation. Algebra and Logic 4:4 (1965), 71–85 (Russian)

We introduce a method of existential interpretation, and we use the method to prove the undecidability of fragments of the form $\exists^r\forall^*$ of various popular first-order theories.

5. Yuri Gurevich: On the decision problem for pure predicate logic. Doklady 166 (1966), 1032–1034 (Russian)

The $\forall\exists\forall\exists^*$ fragment of pure predicate logic with one binary and some number k of unary predicates is proven to be a conservative reduction class. Superseded by [6].

5a. Yuri Gurevich: On the decision problem for pure predicate logic. Soviet Mathematics 7 (1966), 217–219

This is an English translation of [5].

6. Yuri Gurevich: The decision problem for predicate logic. Doklady 168 (1966), 510–511 (Russian)

The $\forall\exists\forall\exists^*$ fragment of pure predicate logic with one binary and no unary predicates is a conservative reduction class and therefore undecidable for satisfiability and for finite satisfiability. This completes the solution of the classical decision problem for pure predicate logic: the prefix-vocabulary classes of pure predicate logic are fully classified into decidable and undecidable. See a more complete exposition in [7].

6a. Yuri Gurevich: The decision problem for predicate logic. Soviet Mathematics 7 (1966), 669–670

This is an English translation of [6].

7. Yuri Gurevich: Recognizing satisfiability of predicate formulas. Algebra and Logic 5:2 (1966), 25–35 (Russian)

This is a detailed exposition of the results announced in [6].

8. Yuri Gurevich: The word problem for some classes of semigroups. Algebra and Logic 5:2 (1966), 25–35 (Russian)

The word problem for finite semigroups is the following decision problem: given some number n of word pairs $(u_1, v_1), ..., (u_n, v_n)$ and an additional word pair (u_0, v_0), decide whether the n equations $u_1 = v_1, ..., u_n = v_n$ imply the additional equation $u_0 = v_0$ in all finite semigroups. We prove that the word problem for finite semigroups is undecidable. In fact, the undecidability result holds for a particular premise $E = (u_1 = v_1$ and ... and $u_n = v_n)$. Furthermore, this particular E can be chosen so that the following are recursively inseparable:
- $\{(u_0, v_0) :$ E implies $u_0 = v_0$ in every periodic semigroup$\}$,
- $\{(u_0, v_0) :$ E fails to imply $u_0 = v_0$ in some finite semigroup$\}$.

The paper contains some additional undecidability results.

9. Yuri Gurevich: Hereditary undecidability of the theory of lattice-ordered abelian groups. Algebra and Logic 6:1 (1967), 45–62 (Russian)

Delimiting the decidability result of [3] for linearly ordered abelian groups and answering Malcev's question, we prove the theorem in the title.

10. Yuri Gurevich: Lattice-ordered abelian groups and K-lineals. Doklady 175 (1967), 1213–1215 (Russian)

10a. Yuri Gurevich: Lattice-ordered abelian groups and K-lineals. Soviet Mathematics 8 (1967), 987–989

This is an English translation of [10].

11. Yuri Gurevich: A new decision procedure for the theory of ordered abelian groups. Algebra and Logic 6:5 (1967), 5–6 (Russian)

12. Yuri Gurevich: The decision problem for some algebraic theories. Doctor of Physico-Mathematical Sciences Thesis, Sverdlovsk, USSR, 1968 (Russian)

13. Yuri Gurevich: The decision problem for logic of predicates and operations. Algebra and Logic 8 (1969), 284–308 (Russian)

The article consists of two chapters. In the first part of the first chapter, the author rediscovers well-partial-orderings and well-quasi-orderings, which he calls *tight* partial orders and tight quasi-orders, and develops a theory of such orderings. (In this connection, it may be appropriate to point out Joseph B. Kruskal's article "The theory of well-quasi-ordering: A frequently discovered concept" in J. Comb. Theory A, vol. 13 (1972), 297–305.) To understand the idea behind the term "tight", think of a boot: you cannot move your foot far down or sidewise – only up. This is similar to tight partial orders where infinite sequences have no infinite descending subsequences, no infinite antichains, but always have infinite ascending subsequences.

In the second part of the first chapter, the author applies the theory of tight orders to prove a classifiability theorem for prefix-vocabulary classes of first-order logic. The main part of the classifiability theorem is that the partial order of prefix-vocabulary classes (ordered by inclusion) is tight. But there is an additional useful part of the classifiability theorem, about the form of the minimal classes outside a downward closed collection, e.g. the minimal classes that are undecidable in one way or another.

In the second chapter, the author completes the decision problem for (the prefix-vocabulary fragments of) pure logic of predicates and functions, though the treatment of the most difficult decidable class is deferred to [18]. In particular, the classes $[\forall^2,(0,1),(1)]$ and $[\forall^2,(1),(0,1)]$ are proved to be conservative reduction classes. (This abstract is written in January 2006.)

13a. Yuri Gurevich: The decision problem for logic of predicates and operations. Algebra and Logic 8 (1969), 160–174 (English)

This is an English translation of [13].

14. Yuri Gurevich: The decision problem for decision problems. Algebra and Logic 8 (1969), 640–642 (Russian)

Consider the collection D of first-order formulas α such that the first-order theory with axiom α is decidable. It is proven that D is neither r.e. nor co-r.e. (The second part had been known earlier.)

14a. Yuri Gurevich: The decision problem for decision problems. Algebra and Logic 8 (1969), 362–363 (English)

This is an English translation of [14].

15. Yuri Gurevich: Minsky machines and the $\forall\exists\forall\&\exists^*$ case of the decision problem. Trans. of Ural University 7:3 (1970), 77–83 (Russian)

An observation that Minsky machines may be more convenient than Turing machines for reduction purposes is illustrated by simplifying the proof from [7] that some $[\forall\exists\forall\&\exists^*,(k,1)]$ is a reduction class.

16. Yuri Gurevich, Igor O. Koriakov: A remark on Berger's paper on the domino problem. Siberian Mathematical Journal, 13 (1972), 459–463 (Russian)

    Berger proved that the decision problem for the unrestricted tiling problem (a.k.a. the unrestricted domino problem) is undecidable. We strengthen Berger's result. The following two collection of domino sets are recursively inseparable:
    (1) those that can tile the plane periodically (equivalently, can tile a torus) and
    (2) those that cannot tile the plane at all.
    It follows that the collection of domino sets that can tile a torus is undecidable.

16a. Yuri Gurevich, Igor O. Koriakov: A remark on Berger's paper on the domino problem. Siberian Mathematical Journal 13 (1972), 319–321 (English)

    This is an English translation of [16].

17. Yuri Gurevich, Tristan Turashvili: Strengthening a result of Suranyi. Bulletin of the Georgian Academy of Sciences 70 (1973), 289–292 (Russian)

18. Yuri Gurevich: Formulas with one universal quantifier. In: *Selected Questions of Algebra and Logic*, Volume dedicated to the memory of A.I. Malcev, Publishing house Nauka – Siberian Branch, Novosibirsk, (1973), 97–110 (Russian)

    The main result, announced in [9], is that the $\exists^*\forall\exists^*$ class of first-order logic with functions but without equality has the finite model property (and therefore is decidable for satisfiability and finite satisfiability). This result completes the solution in [9] for the classical decision problem for first-order logic with functions but without equality.

19. Yuri Gurevich: The decision problem for the expanded theory of ordered abelian groups. Soviet Institute of Scientific and Technical Information (VINITI), 6708:73 (1974), 1–31 (Russian)

20. Yuri Gurevich: The decision problem for first-order logic. Manuscript (1971), 124 pages (Russian)

    This was supposed to be a book (and eventually it became the core of the book [0]), but the publication of the original Russian book was aborted when the author left USSR. A German translation of the manuscript can be found in Universitätsbibliothek Dortmund (Ostsprachen-Übersetzungsdienst) and Technische Informationsbibliothek und Universitätsbibliothek Hannover.

21. Yuri Gurevich: The decision problem for standard classes. JSL 41 (1976), 460–464

    The classification of prefix-signature fragments of (first-order) predicate logic with equality, completed in [7], is extended to first-order logic with equality and functions. One case was solved (confirming a conjecture of this author) by Saharon Shelah.

22. Yuri Gurevich: Semi-conservative reduction. Archiv für Math. Logik und Grundlagenforschung 18 (1976), 23–25

23. I. Gertsbakh, Y. Gurevich: Constructing an optimal fleet for a transportation schedule. Transportation Science 11 (1977), 20–36

    A general method for constructing all optimal fleets is described.

24. Yuri Gurevich: Intuitionistic logic with strong negation. Studia Logica 36 (1977), 49–59

Classical logic is symmetric with respect to True and False but intuitionistic logic is not. We introduce and study a conservative extension of first-order intuitionistic logic that is symmetric with respect to True and False.

25. Yuri Gurevich: Expanded theory of ordered abelian groups. Annals of Mathematical Logic 12 (1977), 193–228

The first-order theory of ordered abelian groups was analyzed in [3]. However, algebraic results on ordered abelian groups in the literature usually cannot be stated in first-order logic. Typically they involve so-called convex subgroups. Here we introduce an expanded theory of ordered abelian groups that allows quantification over convex subgroups and expresses almost all relevant algebra. We classify ordered abelian groups by the properties expressible in the expanded theory, and we prove that the expanded theory of ordered abelian groups is decidable. Curiously, the decidability proof is simpler than that in [3]. Furthermore, the decision algorithm is primitive recursive.

26. Yuri Gurevich: Monadic theory of order and topology, I. Israel Journal of Mathematics 27 (1977), 299–319

We disprove two of Shelah's conjectures and prove some more results on the monadic theory of linearly orderings and topological spaces. In particular, if the Continuum Hypothesis holds then there exist monadic formulæ expressing the predicates "X is countable" and "X is meager" over the real line and over Cantor's Discontinuum.

27. Yuri Gurevich: Monadic theory of order and topology, II. Israel Journal of Mathematics 34 (1979), 45–71

Assuming the Continuum Hypothesis, we interpret the theory of (the cardinal of) the continuum with quantification over constructible (monadic, dyadic, etc.) predicates in the monadic (second-order) theory of real line, in the monadic theory of any other short non-modest chain, in the monadic topology of Cantor's Discontinuum and some other monadic theories. We exhibit monadic sentences defining the real line up to isomorphism under some set-theoretic assumptions. There are some other results.

28. Yuri Gurevich: Modest theory of short chains, I. JSL 44 (1979), 481–490

The composition (or decomposition) method of Feferman-Vaught is generalized and made much more applicable.

29. Yuri Gurevich, Saharon Shelah: Modest theory of short chains, II. JSL 44 (1979), 491–502

We analyze the monadic theory of the rational line and the theory of the real line with quantification over "small" subsets. The results are in some sense the best possible.

30. Yuri Gurevich: Two notes on formalized topology. Fundamenta Mathematicae 57 (1980), 145–148

31. Yuri Gurevich, W. C. Holland: Recognizing the real line. Transactions of American Math. Society 265 (1981), 527–534

We exhibit a first-order statement about the automorphism group of the real line that characterizes the real line among all homogeneous chains.

32. A. M. W. Glass, Yuri Gurevich, W. C. Holland, Saharon Shelah: Rigid homogeneous chains. Math. Proceedings of Cambridge Phil. Society 89 (1981), 7–17

33. A. M. W. Glass, Y. Gurevich, W. C. Holland, M. Jambu-Giraudet: Elementary theory of automorphism groups of doubly homogeneous chains. Springer Lecture Notes in Mathematics 859 (1981), 67–82

34. Yuri Gurevich: Crumbly spaces. Sixth International Congress for Logic, Methodology and Philosophy of Science (1979) North-Holland (1982), 179–191

    Answering a question of Henson, Jockush, Rubel and Takeuti, we prove that the rationals, the irrationals and the Cantor set are all elementarily equivalent as topological spaces.

35. S. O. Aanderaa, Egon Börger, Yuri Gurevich: Prefix classes of Krom formulas with identity. Archiv für Math. Logik und Grundlagenforschung 22 (1982), 43–49

36. Yuri Gurevich: Existential interpretation, II. Archiv für Math. Logik und Grundlagenforschung 22 (1982), 103–120

37. Yuri Gurevich, Saharon Shelah: Monadic theory of order and topology in ZFC. Annals of Mathematical Logic 23 (1982), 179–198

    In the 1975 Annals of Mathematics, Shelah interpreted true first-order arithmetic in the monadic theory of order under the assumption of the continuum hypothesis. The assumption is removed here.

38. I. Gertsbakh, Y. Gurevich: Homogeneous optimal fleet. Transportation Research 16B (1982), 459–470

39. Yuri Gurevich: A review of two books on the decision problem. Bulletin of the American Mathematical Society 7 (1982), 273–277

40. Yuri Gurevich, Leo Harrington: Automata, trees, and games. 14th Annual Symposium on Theory of Computing, ACM (1982), 60–65

    We prove a forgetful determinacy theorem saying that, for a wide class of infinitary games, one of the players has a winning strategy that is virtually memoryless: the player has to remember only boundedly many bits of information. We use forgetful determinacy to give a transparent proof of Rabin's celebrated result that the monadic second-order theory of the infinite tree is decidable.

41. Yuri Gurevich, H. R. Lewis: The inference problem for template dependencies. Information and Control 55 (1982), 69–79

    Answering a question of Jeffrey Ullman, we prove that the problem in the title is undecidable.

42. Andreas Blass, Yuri Gurevich: On the unique satisfiability problem. Information and Control 55 (1982), 80–88

    Papadimitriou and Yannakakis were interested whether Unique Sat is hard for $\{L - L' : L, L' \in NP\}$ when NP differs from co-NP (otherwise the answer is obvious). We show that this is true under one oracle and false under another.

43. E. M. Clarke, N. Francez, Y. Gurevich, P. Sistla: Can message buffers be characterized in linear temporal logic? Symposium on Principles of Distributed Computing, ACM (1982), 148–156

    In the case of unbounded buffers, the negative answer follows from a result in [28].

44. Yuri Gurevich: Decision problem for separated distributive lattices. JSL 48 (1983), 193–196

    It is well known that for all recursively enumerable sets X_1, X_2 there are disjoint recursively enumerable sets Y_1, Y_2 such that $Y_i \subseteq X_i$ and $(Y_1 \cup Y_2) = (X_1 \cup X_2)$. Alistair Lachlan called distributive lattices satisfying this property *separated*. He proved that the first-order theory of finite separated distributive lattices is decidable. We prove here that the first-order theory of all separated distributive lattices is undecidable.

45. Yuri Gurevich, Menachem Magidor, Saharon Shelah: The monadic theory of ω_2. JSL 48 (1983), 387–398

    In a series of papers, Büchi proved the decidability of the monadic (second-order) theory of ω_0, of all countable ordinals, of ω_1, and finally of all ordinals $< \omega_2$. Here, assuming the consistency of a weakly compact cardinal, we prove that, in different set-theoretic worlds, the monadic theory of ω_2 may be arbitrarily difficult (or easy).

46. Yuri Gurevich, Saharon Shelah: Interpreting second-order logic in the monadic theory of order. JSL 48 (1983), 816–828

    Under a weak set-theoretic assumption, we interpret full second-order logic in the monadic theory of order.

47. Yuri Gurevich, Saharon Shelah: Rabin's Uniformization Problem. JSL 48 (1983), 1105–1119

    The negative solution is given.

48. Yuri Gurevich, Saharon Shelah: Random models and the Gödel case of the decision problem. JSL 48 (1983), 1120–1124

    We replace Gödel's sophisticated combinatorial argument with a simple probabilistic one.

49. A. M. W. Glass, Yuri Gurevich: The word problem for lattice-ordered groups. Transactions of American Math. Society 280 (1983), 127–138

    The problem is proven to be undecidable.

50. Yuri Gurevich: Critiquing a critique of Hoare's programming logics. Communications of ACM (May 1983), 385 (Tech. communication)

51. Yuri Gurevich: Algebras of feasible functions. 24th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, 1983, 210–214

    We prove that, under a natural interpretation over finite domains,
    (i) a function is primitive recursive if and only if it is logspace computable, and
    (ii) a function is general recursive if and only if it is polynomial time computable.

52. Yuri Gurevich, P. H. Schmitt: The theory of ordered abelian groups does not have the independence property. Trans. of American Math. Society 284 (1984), 171–182

53. Yuri Gurevich, H. R. Lewis: The word problem for cancellation semigroups with zero. JSL 49 (1984), 184–191

    In 1947, Post showed the word problem for semigroups to be undecidable. In 1950, Turing strengthened this result to cancellation semigroups, i.e. semigroups satisfying the cancellation property

(1) if $xy = xz$ or $yx = zx$ then $y = z$.

No semigroup with zero satisfies (1). The cancellation property for semigroups with zero and identity is

(2) if $xy = xz \neq 0$ or $yx = zx \neq 0$ then $y = z$.

The cancellation property for semigroups with zero but without identity is the conjunction of (2) and

(3) if $xy = x$ or $yx = x$ then $x = 0$.

Whether or not a semigroup with zero has an identity, we refer to it as a cancellation semigroup with zero if it satisfies the appropriate cancellation property. It is shown in [8] that the word problem for finite semigroups is undecidable. Here we show that the word problem is undecidable for finite cancellation semigroups with zero; this holds for semigroups with identity and also for semigroups without identity. (In fact, we prove a stronger effective inseparability result.) This provides the necessary mathematical foundation for [41]).

54. Yuri Gurevich, L. J. Stockmeyer, Uzi Vishkin: Solving NP-hard problems on graphs that are almost trees, and an application to facility location problems. Journal of the ACM 31 (1984), 459–473

Imagine that you need to put service stations (or MacDonald's restaurants) on roads in such a way that every resident is within, say, 10 miles of the nearest station. What is the minimal number of stations and how does one find an optimal placement? In general, the problem is NP hard; however in important special cases there are feasible solutions.

55. Andreas Blass, Yuri Gurevich: Equivalence relations, invariants, and normal forms. SIAM Journal on Computing 13 (1984), 682–689

For an equivalence relation E on the words in some finite alphabet, we consider the following four problems.

Recognition. Decide whether two words are equivalent.

Invariant. Calculate a function constant on precisely the equivalence classes.

Normal form. Calculate a particular member of an equivalence class, given an arbitrary member.

First member. Calculate the first member of an equivalence class, given an arbitrary member.

A solution for any of these problems yields solutions for all earlier ones in the list. We show that, for polynomial time recognizable E, the first member problem is always in the class Δ_2^P (solvable in polynomial time with an oracle for an NP set) and can be complete for this class even when the normal form problem is solvable in polynomial time. To distinguish between the other problems in the list, we construct an E whose invariant problem is not solvable in polynomial time with an oracle for E (although the first member problem is in $\mathrm{NP}^E \cap \mathrm{co\text{-}NP}^E$), and we construct an E whose normal form problem is not solvable in polynomial time with an oracle for a certain solution of its invariant problem.

56. Andreas Blass, Yuri Gurevich: Equivalence relations, invariants, and normal forms, II. Springer LNCS 171 (1984), 24–42

We consider the questions whether polynomial time solutions for the easier problems of the list for [55] yield NP solutions for the harder ones, or vice versa. We show that affirmative answers to several of these questions are equivalent to natural principles like NP = co-NP, (NP ∩ co-NP) = P, and the shrinking principle for NP sets. We supplement known oracles with enough new ones to show that

all questions considered have negative answers relative to some oracles. In other words, these questions cannot be answered affirmatively by means of relativizable polynomial-time Turing reductions. Finally, we show that the analogous questions in the framework where Borel sets play the role of polynomial time decidable sets have negative answers.

57. Yuri Gurevich, Saharon Shelah: The monadic theory and the 'next world'. Israel Journal of Mathematics 49 (1984), 55–68

    Let r be a Cohen real over a model V of ZFC. Then the second-order $V[r]$-theory of the integers (even the reals if V satisfies CH) is interpretable in the monadic V-theory of the real line. Contrast this with the result of [79].

58. W. D. Goldfarb, Yuri Gurevich, Saharon Shelah: A decidable subclass of the minimal Gödel case with identity. JSL Logic 49 (1984), 1253–1261

59. Yuri Gurevich, H. R. Lewis: A logic for constant depth circuits. Information and Control 61 (1984), 65–74

    We present an extension of first-order logic that captures precisely the computational complexity of (the uniform sequences of) constant-depth polynomial-time circuits.

60. Yuri Gurevich: Toward logic tailored for computational complexity. In: M. Richter et al. (eds.) Computation and Proof Theory, Springer Lecture Notes in Math. 1104 (1984), 175–216

    The pathos of this paper is that classical logic, developed to confront the infinite, is ill prepared to deal with finite structures, whereas finite structures, e.g. databases, are of so great importance in computer science. We show that famous theorems about first-order logic fail in the finite case, and discuss various alternatives to classical logic. The message has been heard.

60.5. Yuri Gurevich: Reconsidering Turing's thesis (toward more realistic semantics of programs). Technical report CRL-TR-36-84 University of Michigan, September 1984

    The earliest publication on the abstract state machine project.

61. J. P. Burgess, Yuri Gurevich: The decision problem for linear temporal logic. Notre Dame JSL 26 (1985), 115–128

    The main result is the decidability of the temporal theory of the real order.

62. Yuri Gurevich, Saharon Shelah: To the decision problem for branching time logic. In: P. Weingartner and G. Dold (eds.) Foundations of Logic and Linguistics: Problems and their Solutions, Plenum (1985), 181–198

63. Yuri Gurevich, Saharon Shelah: The decision problem for branching time logic. JSL 50 (1985), 668–681

    Define a tree to be any partial order satisfying the following requirement: the predecessors of any element x are linearly ordered, i.e. if ($y < x$ and $z < x$) then ($y < z$ or $y = z$ or $y > z$). The main result of the two papers [62,63] is the decidability of the theory of trees with additional unary predicates and quantification over nodes and branches. This gives the richest decidable temporal logic.

64. Yuri Gurevich: Monadic second-order theories. In: J. Barwise and S. Feferman (eds.) Model-Theoretical Logics, Springer-Verlag, Perspectives in Mathematical Logic (1985), 479–506

In this chapter we make a case for the monadic second-order logic (that is to say, for the extension of first-order logic allowing quantification over monadic predicates) as a good source of theories that are both expressive and manageable.

We illustrate two powerful decidability techniques here. One makes use of automata and games. The other is an offshoot of a composition theory where one composes models as well as their theories. Monadic second-order logic appears to be the most natural match for the composition theory.

Undecidability proofs must be thought out anew in this area; for, whereas true first-order arithmetic is reducible to the monadic theory of the real line R, it is nevertheless not interpretable in the monadic theory of R. A quite unusual undecidability method is another subject of this chapter.

In the last section we briefly review the history of the methods thus far developed and mention numerous results obtained using the methods.

64.5. Yuri Gurevich: A New Thesis. Abstracts, American Mathematical Society 6:4 (August 1985), p. 317, abstract 85T-68-203

The first announcement of the "new thesis", later known as the *Abstract State Machine thesis*.

65. Andreas Blass, Yuri Gurevich, D. Kozen: A zero-one law for logic with a fixed-point operator. Information and Control 67 (1985), 70–90

The zero-one law, known to hold for first-order logic but not for monadic or even existential monadic second-order logic, is generalized to the extension of first-order logic by the least (or iterative) fixed-point operator. We also show that the problem of deciding, for any π, whether it is almost-sure is complete for exponential time, if we consider only π's with a fixed finite vocabulary (or vocabularies of bounded arity) and complete for double-exponential time if π is unrestricted.

66. Andreas Blass, Yuri Gurevich: Henkin quantifiers and complete problems. Annals of Pure and Applied Logic 32 (1986), 1–16

We show that almost any non-linear quantifier, applied to quantifier-free first-order formulas, suffices to express an NP-complete predicate; the remaining non-linear quantifiers express exactly co-NL predicates (NL is Nondeterministic Log-space).

67. L. Denenberg, Y. Gurevich, S. Shelah: Definability by constant-depth polynomial-size circuits. Information and Control 70 (1986), 216–240

We investigate the expressive power of constant-depth polynomial-size circuit models. In particular, we construct a circuit model whose expressive power is precisely that of first-order logic.

68. Amnon Barak, Zvi Drezner, Yuri Gurevich: On the number of active nodes in a multicomputer system. Networks 16 (1986), 275– 282

Simple probabilistic algorithms enable each active node to find estimates of the fraction of active nodes in the system of n nodes (with a direct communication link between any two nodes) in time $o(n)$.

69. Yuri Gurevich: What does $O(n)$ mean? SIGACT NEWS 17:4 (1986), 61–63

70. Yuri Gurevich, Saharon Shelah: Fixed-point extensions of first-order logic. Annals of Pure and Applied Logic 32 (1986), 265–280

We prove that the three extensions of first-order logic by means of positive, monotone and inflationary inductions have the same expressive power in the case

of finite structures. An extended abstract of the above, in Proc. 26th Annual Symposium on Foundation of Computer Science, IEEE Computer Society Press (1985), 346–353, contains some additions.

71. Yuri Gurevich, Saharon Shelah: Expected computation time for Hamiltonian Path Problem. SIAM J. on Computing 16:3 (1987), 486–502

Let $G(n,p)$ be a random graph with n vertices and the edge probability p. We give an algorithm for Hamiltonian Path Problem whose expected run-time on $G(n,p)$ is $cn/p + o(n)$ for any fixed p. This is the best possible result for the case of fixed-edge probability. The expected run-time of a slightly modified version of the algorithm remains polynomial if $p = p(n) > n - c$ where c is positive and small.

The paper is based on a 1984 technical report.

72. Miklós Ajtai, Yuri Gurevich: Monotone versus positive. Journal of the ACM 34 (1987), 1004–1015

A number of famous theorems about first-order logic were disproved in [60] in the case of finite structures, but Lyndon's theorem on monotone vs. positive resisted the attack. It is defeated here. The counterexample gives a uniform sequence of constant-depth polynomial-size (functionally) monotone boolean circuits not equivalent to any (however nonuniform) sequence of constant-depth polynomial-size positive boolean circuits.

73. Andreas Blass, Yuri Gurevich: Existential fixed-point logic. In: E. Börger (ed.) Logic and complexity, Springer LNCS 270 (1987), 20–36

The purpose of this paper is to draw attention to existential fixed-point logic (EFPL). Among other things, we show the following.

- If a structure A satisfies an EFPL formula φ then A has a finite subset F such that every structure that coincides with A on F satisfies φ.
- Using EFPL instead of first-order logic removes the expressivity hypothesis in Cook's completeness theorem for Hoare logic.
- In the presence of a successor relation, EFPL captures polynomial time.

74. Yuri Gurevich: Logic and the Challenge of Computer Science. In: E. Börger (ed.) Current Trends in Theoretical Computer Science, Computer Science Press (1988), 1–57

The chapter consists of two quite different parts. The first part is a survey (including some new results) on finite model theory. One particular point deserves a special attention. In computer science, the standard computation model is the Turing machine whose inputs are strings; other algorithm inputs are supposed to be encoded with strings. However, in combinatorics, database theory, etc., one usually does not distinguish between isomorphic structures (graphs, databases, etc.). For example, a database query should provide information about the database rather than its implementation. In such cases, there is a problem with string presentation of input objects: there is no known, easily computable string encoding of isomorphism classes of structures. Is there a computation model whose machines do not distinguish between isomorphic structures and compute exactly PTIME properties? The question is intimately related to a question by Chandra and Harel in "Structure and complexity of relational queries", J. Comput. and System Sciences 25 (1982), 99–128. We formalize the question as the question whether there exists a logic that captures polynomial time (without presuming the presence of a linear order) and conjecture the negative answer. The

first part is based on lectures given at the 1984 Udine Summer School on Computation Theory and summarized in the technical report "Logic and the Challenge of Computer Science", CRL-TR-10-85, Sep. 1985, Computing Research Lab, University of Michigan, Ann Arbor, Michigan.

In the second part, we introduce a new computation model: evolving algebras (later renamed *abstract state machines*). This new approach to semantics of computations and, in particular, to semantics of programming languages emphasizes dynamic and resource-bounded aspects of computation. It is illustrated on the example of Pascal. The technical report mentioned above contained an earlier version of part 2. The final version was written in 1986.

75. Yuri Gurevich: Algorithms in the world of bounded resources. In: R. Herken (ed.) The universal Turing machine - a half-century story, Oxford University Press (1988), 407–416

    In the classical theory of algorithms, one addresses a computing agent with unbounded resources. We argue in favor of a more realistic theory of multiple addressees with limited resources.

76. Yuri Gurevich: Average case completeness. J. Computer and System Sciences 42:3 (June 1991), 346–398 (a special issue with selected papers of FOCS'87)

    We explain and advance Levin's theory of average case complexity. In particular, we exhibit the second natural average-case-complete problem and prove that deterministic reductions are inadequate.

77. Yuri Gurevich, Jim Morris: Algebraic operational semantics and Modula-2. CSL'87, 1st Workshop on Computer Science Logic, Springer LNCS 329 (1988), 81–101

    Jim Morris was a PhD student of Yuri Gurevich at the Electrical Engineering and Computer Science Department of the University of Michigan, the first PhD student working on the abstract state machine project. This is an extended abstract of Jim Morris's 1988 PhD thesis (with the same title) and the first example of the ASM semantics of a whole programming language.

78. Yuri Gurevich: On Kolmogorov machines and related issues. Originally in BEATCS 35 (June 1988), 71–82. Reprinted in: Current Trends in Theoretical Computer Science. World Scientific (1993), 225–234

    One contribution of the article was to formulate the Kolmogorov-Uspensky thesis. In "To the Definition of an Algorithm" [Uspekhi Mat. Nauk 13:4 (1958), 3–28 (Russian)], Kolmogorov and Uspensky wrote that they just wanted to comprehend the notions of computable functions and algorithms, and to convince themselves that there is no way to extend the notion of computable function. In fact, they did more than that. It seems that their thesis was this:

    > Every computation, performing only one restricted local action at a time, can be viewed as (not only being simulated by, but actually being) the computation of an appropriate KU machine (in the more general form).

    Uspensky agreed [J. Symb. Logic 57 (1992), p. 396]. Another contribution of the paper was a popularization of the following beautiful theorem of Leonid Levin.

    > Theorem. For every computable function $F(w) = x$ from binary strings to binary strings, there exists a KU algorithm A such that A conclusively inverts F and (Time of A on x) $= O$(Time of B on x) for every KU algorithm B that conclusively inverts F.

which had been virtually unknown, partially because it appeared (without a proof) in his article "Universal Search Problems" [Problems of Information Transmission 9:3 (1973), 265–266] which is hard to read.

79. Yuri Gurevich, Saharon Shelah: On the strength of the interpretation method. JSL 54:2 (1989), 305–323

    The interpretation method is the main tool in proving negative results related to logical theories. We examine the strength of the interpretation method and find a serious limitation. In one of our previous papers [57], we were able to reduce true arithmetic to the monadic theory of real line. Here we show that true arithmetic cannot be interpreted in the monadic theory of the real line. The reduction of [57] is not an interpretation.

80. Yuri Gurevich, Saharon Shelah: Time polynomial in input or output. JSL 54:3 (1989), 1083–1088

    There are simple algorithms with large outputs; it is misleading to measure the time complexity of such algorithms in terms of inputs only. In this connection, we introduce the class PIO of functions computable in time polynomial in the maximum of the size of input and the size of output, and some other similar classes. We observe that there is no notation system for any extension of the class of total functions computable on Turing machines in time linear in output and give a machine-independent definition of partial PIO functions.

81. Andreas Blass, Yuri Gurevich: On Matiyasevich's non-traditional approach to search problems. Information Processing Letters 32 (1989), 41–45

    Yuri Matijasevich, famous for completing the solution of Hilbert's tenth problem, suggested to use differential equations inspired by real phenomena in nature to solve the satisfiability problem for boolean formulas. The initial conditions are chosen at random and it is expected that, in the case of a satisfiable formula, the process, described by differential equations, converges quickly to an equilibrium which yields a satisfying assignment. A success of the program would establish NP=R. Attracted by the approach, we discover serious complications with it.

82. Yuri Gurevich, Saharon Shelah: Nearly linear time. Symposium on Logical Foundations of Computer Science in Pereslavl-Zalessky, USSR, Springer LNCS 363 (1989) 108–118

    The notion of linear time is very sensitive to the machine model. In this connection we introduce and study the class NLT of functions computable in nearly linear time $n \cdot (\log n)^{O(1)}$ on random access computers or any other "reasonable" machine model (with the standard multitape Turing machine model being "unreasonable" for that low complexity class). This gives a very robust approximation to the notion of linear time. In particular, we give a machine-independent definition of NLT and a natural problem complete for NLT.

83. Miklós Ajtai, Yuri Gurevich: Datalog vs First-Order Logic. J of Computer and System Sciences 49:3 (December 1994), 562–588 (Extended abstract in FOCS'89, 142–147)

    First-order logic and Datalog are two very important paradigms for relational-database query languages. How different are they from the point of view of expressive power? What can be expressed both in first-order logic and Datalog? It is easy to see that every existential positive first-order formula is expressible by a bounded Datalog query, and the other way round. Cosmadakis suggested

that there are no other properties expressible in first-order logic and in Datalog; in other words, no unbounded Datalog query is expressible in first-order logic. We prove the conjecture; that is our main theorem. It can be seen as a kind of compactness theorem for finite structures. In addition, we give counterexamples delimiting the main result.

84. Yuri Gurevich: Infinite Games. Originally in BEATCS (June 1989), 93–100. Reprinted in: Current Trends in Theoretical Computer Science. World Scientific (1993), 235–244

    Infinite games are widely used in mathematical logic. Recently infinite games were used in connection to concurrent computational processes that do not necessarily terminate. For example, operating system may be seen as playing a game "against" the disruptive forces of users. The classical question of the existence of winning strategies turns out to be of importance to practice. We explain a relevant part of the infinite game theory.

85. Yuri Gurevich: The Challenger-Solver game: Variations on the Theme of P=?NP. BEATCS (October 1989), 112–121. Reprinted in: Current Trends in Theoretical Computer Science. World Scientific (1993), 245–253

    The question $\mathrm{P}\stackrel{?}{=}\mathrm{NP}$ is the focal point of much research in theoretical computer science. But is it the right question? We find it biased toward the positive answer. It is conceivable that the negative answer is established without providing much evidence for the difficulty of NP problems in practical terms. We argue in favor of an alternative to $\mathrm{P}\stackrel{?}{=}\mathrm{NP}$ based on the average-case complexity.

86. Yuri Gurevich: Games people play. In: S. Mac Lane and D. Siefkes (eds.) Collected Works of J. Richard Büchi, Springer-Verlag (1990), 517–524

87. Yuri Gurevich, Saharon Shelah: Nondeterministic linear-time tasks may require substantially nonlinear deterministic time in the case of sublinear work space. Journal of the ACM 37:3 (1990), 674–687

    We develop a technique to prove time-space trade-offs and exhibit natural search problems (e.g. Log-size Clique Problem) that are solvable in linear time on polylog-space (and sometimes even log-space) nondeterministic Turing machine, but no deterministic machine (in a very general sense of this term) with sequential-access read-only input tape and work space n^σ solves the problem within time $n^{1+\tau}$ if $\sigma + 2\tau < \frac{1}{2}$.

88. Yuri Gurevich: Matrix Decomposition Problem is Complete for the Average Case. FOCS'90, 31st Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press (1990), 802–811

    The first algebraic average-case complete problem is presented. See [97] in this connection.

89. Yuri Gurevich, L. A. Moss: Algebraic Operational Semantics and Occam. CSL'89, 3rd Workshop on Computer Science Logic, Springer LNCS 440 (1990), 176–192

    We give evolving algebra semantics to the Occam programming language, generalizing in the process evolving algebras to the case of distributed concurrent computations.

    Later note: this was the first example of a distributed abstract state machine.

90. Yuri Gurevich: On Finite Model Theory. In: S. R. Buss et al. (eds.) Feasible Mathematics, (1990), 211–219

This is a little essay on finite model theory. Section 1 gives some counterexamples to classical theorems in the finite case. Section 2 gives a finite version of the classical compactness theorem. Section 3 announces two Gurevich-Shelah results. One is a new preservation theorem that implies that a first-order formula p preserved by any homomorphism from a finite structure into another finite structure is equivalent to a positive existential formula q. The other result is a lower bound result according to which a shortest q may be non-elementary longer than p.

A later note: Unfortunately, the proof of preservation theorem fell through – a unique such case in the history of the Gurevich-Shelah collaboration – and was later proved by Benjamin Rossman; see Proceedings of LICS 2005. Rossman also provided details for our lower bound proof.

91. Yuri Gurevich: On the Classical Decision Problem. Originally in BEATCS (October 1990), 140–150. Reprinted in: Current Trends in Theoretical Computer Science. World Scientific (1993), 254–265

92. Yuri Gurevich: Evolving Algebras: An Introductory Tutorial. Originally in BEATCS 43 (February 1991), 264–284. This slightly revised version appeared in: Current Trends in Theoretical Computer Science. World Scientific (1993), 266–292

    Computation models and specification methods seem to be worlds apart. The evolving algebra project is as an attempt to bridge the gap by improving on Turing's thesis. We seek more versatile machines able to simulate arbitrary algorithms, on their natural abstraction levels, in a direct and essentially coding-free way. The evolving algebra thesis asserts that evolving algebras are such versatile machines. Here sequential evolving algebras are defined and motivated. In addition, we sketch a speculative "proof" of the sequential evolving algebra thesis: Every sequential algorithm can be lock-step simulated by an appropriate sequential evolving algebra on the natural abstraction level of the algorithm.

93. Andreas Blass, Yuri Gurevich: On the Reduction Theory for Average-Case Complexity. CSL'90, 4th Workshop on Computer Science Logic, Springer LNCS 533 (1991) 17–30

    A function from instances of one problem to instances of another problem is a reduction if together with any admissible algorithm for the second problem it gives an admissible algorithm for the first problem. This is an example of a descriptive definition of reductions. We slightly simplify Levin's usable definition of deterministic average-case reductions and thus make it equivalent to the appropriate descriptive definition. Then we generalize this to randomized average-case reductions.

94. Yuri Gurevich: Average Case Complexity. ICALP'91, International Colloquium on Automata, Languages and Programming, Madrid, Springer LNCS 510 (1991), 615–628

    We motivate, justify and survey the average-case reduction theory.

95. Yuri Gurevich: Zero-One Laws. Originally in BEATCS 51 (February 1991), 90–106. Reprinted in: Current Trends in Theoretical Computer Science. World Scientific (1993), 293–309

96. Andreas Blass, Yuri Gurevich: Randomizing Reductions of Search Problems. SIAM J. on Computing 22:5 (1993), 949–975

    This is the journal version of an invited talk at FST&TCS'91, 11th Conference on Foundations of Software Technology and Theoretical Computer Science, New Delhi, India; see Springer LNCS 560 (1991), 10–24.

First, we clarify the notion of a (feasible) solution for a search problem and prove its robustness. Second, we give a general and usable notion of many-one randomizing reductions of search problems and prove that it has desirable properties. All reductions of search problems to search problems in the literature on average case complexity can be viewed as such many-one randomizing reductions. This includes those reductions in the literature that use iterations and therefore do not look many-one.

97. Andreas Blass, Yuri Gurevich: Matrix Transformation is Complete for the Average Case. SIAM J. on Computing 24:1 (1995), 3–29

    This is a full paper corresponding to the extended abstract [88] by the second author. We present the first algebraic problem complete for the average case under a natural probability distribution. The problem is this: Given a unimodular matrix X of integers, a set S of linear transformations of such unimodular matrices and a natural number n, decide if there is a product of at most n (not necessarily different) members of S that takes X to the identity matrix.

98. Yuri Gurevich, Jim Huggins: The Semantics of the C Programming Language. In E. Börger et al.(eds.) CSL'92 (Computer Science Logics), Springer LNCS 702 (1993), 274–308

    The method of successive refinement is used. The observation that C expressions do not contain statements gives rise to the first evolving algebra (ealgebra) which captures the command part of C; expressions are evaluated by an oracle. The second ealgebra implements the oracle under the assumptions that all the necessary declarations have been provided and user-defined functions are evaluated by another oracle. The third ealgebra handles declarations. Finally, the fourth ealgebra revises the combination of the first three by incorporating the stack discipline; it reflects all of C. (A later note: evolving algebras are now called abstract state machines.)

99. Thomas Eiter, Georg Gottlob, Yuri Gurevich: Curb Your Theory! A Circumscriptive Approach for Inclusive Interpretation of Disjunctive Information. In: R. Bajcsy, M. Kaufman (eds.) Proc. 13th Intern. Joint Conf. on AI (IJCAI'93) (1993), 634–639

    We introduce, study and analyze the complexity of a new nonmonotonic technique of common sense reasoning called curbing. Like circumscription, curbing is based on model minimality, but, unlike circumscription, it treats disjunction inclusively.

100. Yuri Gurevich: Feasible Functions. London Mathematical Society Newsletter 206 (June 1993), 6–7

     Some computer scientists, notably Steve Cook, identify feasibility with polynomial-time computability. We argue against that point of view. Polynomial-time computations may be infeasible, and feasible computations may be not polynomial time.

101. Yuri Gurevich: Logic in Computer Science. In: G. Rozenberg and A. Salomaa (eds.) Current Trends in Theoretical Computer Science, World Scientific Series in Computer Science 40 (1993), 223–394

102. Yuri Gurevich: The AMAST Phenomenon. Originally in BEATCS 51 (October 1993), 295–299. Reprinted in: Current Trends in Theoretical Computer Science. World Scientific (2001), 247–253

This humorous article incorporates a bit of serious criticism of algebraic and logic approaches to software problems.

103. Yuri Gurevich: Evolving Algebra 1993: Lipari Guide. Specification and Validation Methods, Oxford University Press (1995), 9–36

Computation models and specification methods seem to be worlds apart. The project on abstract state machines (a.k.a. evolving algebras) started as an attempt to bridge the gap by improving on Turing's thesis [92]. We sought more versatile machines which would be able to simulate arbitrary algorithms, on their natural abstraction levels, in a direct and essentially coding-free way. The ASM thesis asserts that ASMs are such versatile machines. The guide provided the definition of sequential and – for the first time – parallel and distributed ASMs. The denotational semantics of sequential and parallel ASMs is addressed in the Michigan guide [129].

104. Erich Grädel, Yuri Gurevich: Tailoring Recursion for Complexity. JSL 60:3 (September 1995), 952–969

Complexity classes are easily generalized to the case when inputs of an algorithm are finite ordered structures of a fixed vocabulary rather than strings. A logic L is said to capture (or to be tailored to) a complexity class C if a class of finite ordered structures of a fixed vocabulary belongs to C if and only if it is definable in L. Traditionally, complexity tailored logics are logics of relations. In his FOCS'83 paper, the second author showed that, on finite structures, the class of Logspace computable functions is captured by the primitive recursive calculus, and the class of PTIME computable functions is captured by the classical calculus of partially recursive functions. Here we continue that line of investigation and construct recursive calculi for various complexity classes of functions, in particular for (more challenging) nondeterministic classes, NLogspace and NPTIME.

105. Yuri Gurevich: Logic Activities in Europe. ACM SIGACT NEWS 25:2 (June 1994), 11–24

This is a critical analysis of European logic activities in computer science based on a Fall 1992 European tour sponsored by the Office of Naval Research.

106. Yuri Gurevich, Raghu Mani: Group Membership Protocol: Specification and Verification. Specification and Validation Methods, Oxford University Press (1995), 295–328

An interesting and useful group membership protocol of Flavio Christian involves timing constraints, and its correctness is not obvious. We construct a mathematical model of the protocol and verify the protocol (and notice that the assumptions about the environment may be somewhat weakened).

107. Egon Börger, Dean Rosenzweig, Yuri Gurevich: The Bakery Algorithm: Yet Another Specification and Verification. Specification and Validation Methods, Oxford University Press (1995), 231–243

The so-called bakery algorithm of Lamport is an ingenious and sophisticated distributed mutual-exclusion algorithm. First we construct a mathematical model A1 which reflects the algorithm very closely. Then we construct a more abstract model A2 where the agents do not interact and the information is provided by two oracles. We check that A2 is safe and fair provided that the oracles satisfy certain conditions. Finally we check that the implementation A1 of A2 satisfies the conditions and thus A1 is safe and fair.

108. Yuri Gurevich, Neil Immerman, Saharon Shelah: McColm's Conjecture. LICS 1994, Symp. on Logic in Computer Science, IEEE Computer Society Press (1994), 10–19

Gregory McColm conjectured that, over any class K of finite structures, all positive elementary inductions are bounded if every FOL + LFP formula is equivalent to a first-order formula over K. Here FOL + LFP is the extension of first-order logic with the least fixed point operator. Our main results are two model-theoretic constructions – one deterministic and one probabilistic – each of which refutes McColm's conjecture.

109. Erich Grädel, Yuri Gurevich: Metafinite Model Theory. Information and Computation 140:1 (1998), 26–81. Preliminary version in D. Leivant (ed.) Logic and Computational Complexity, Selected Papers, Springer LNCS 960 (1995), 313–366

Earlier, the second author criticized database theorists for admitting arbitrary structures as databases: databases are finite structures [60]. However, a closer investigation reveals that databases are not necessarily finite. For example, a query may manipulate numbers that do not even appear in the database, which shows that a numerical structure is somehow involved. It is true nevertheless that database structures are special. The phenomenon is not restricted to databases; for example, think about the natural structure to formalize the traveling salesman problem. To this end, we define metafinite structures. Typically such a structure consists of (i) a primary part, which is a finite structure, (ii) a secondary part, which is a (usually infinite) structure, e.g. arithmetic or the real line, and (iii) a set of "weight" functions from the first part into the second. Our logics do not allow quantification over the secondary part. We study definability issues and their relation to complexity. We discuss model-theoretic properties of metafinite structures, present results on descriptive complexity, and sketch some potential applications.

110. Andreas Blass, Yuri Gurevich: Evolving Algebras and Linear Time Hierarchy. In: B. Pehrson and I. Simon (eds.) IFIP 1994 World Computer Congress, Volume I: Technology and Foundations, North-Holland, Amsterdam, 383–390

A precursor of [118].

111. Yuri Gurevich, James K. Huggins: Evolving Algebras and Partial Evaluation. In: B. Pehrson and I. Simon (eds.) IFIP 1994 World Computing Congress, Volume 1: Technology and Foundations, Elsevier, Amsterdam, 587–592

The authors present an automated (and implemented) partial evaluator for sequential evolving algebras.

112. Yuri Gurevich: Evolving Algebras. In: B. Pehrson and I. Simon (eds.) IFIP 1994 World Computer Congress, Volume I: Technology and Foundations, Elsevier, Amsterdam, 423–427

The opening talk at the first workshop on evolving algebras. Sections: Introduction, The EA Thesis, Remarks, Future Work.

113. Yuri Gurevich, Saharon Shelah: On Rigid Structures. JSL 61:2 (June 1996), 549–562

This is related to the problem of defining linear order on finite structures. If a linear order is definable on a finite structure A, then A is rigid (which means that its only automorphism is the identity). There had been a suspicion that if K is the collection of all finite structures of a finitely axiomatizable class and if

every K structure is rigid, then K permits a relatively simple uniform definition of linear order. That happens not to be the case. The main result of the paper is a probabilistic construction of finite rigid graphs. Using that construction, we exhibit a finitely axiomatizable class of finite rigid structures (called *multipedes*) such that no $L^{\omega}_{\infty,\omega}$ sentence φ with counting quantifiers defines a linear order in all the structures. Furthermore, φ does not distinguish between a sufficiently large multipede M and a multipede M' obtained from M by moving a "shoe" to another foot of the same segment.

114. Yuri Gurevich: The Value, if any, of Decidability. Originally in BEATCS 55 (February 1995), 129–135. Reprinted in: Current Trends in Theoretical Computer Science, World Scientific (2001), 274–280

    A decidable problem can be as hard as an undecidable one for all practical purposes. So what is the value of a mere decidability result? That is the topic discussed in the paper.

115. Thomas Eiter, Georg Gottlob, Yuri Gurevich: Normal Forms for Second-Order Logic over Finite Structures and Classification of NP Optimization Problems. Annals of Pure and Applied Logic 78 (1996), 111–125

    We prove a new normal form for second-order formulas on finite structures and simplify the Kolaitis-Thakur hierarchy of NP optimization problems.

116. Yuri Gurevich, James K. Huggins: The Railroad Crossing Problem:. An Experiment with Instantaneous Actions and Immediate Reactions. In: H. Kleine-Büning (ed.) Computer Science Logics, Selected papers from CSL'95, Springer LNCS 1092 (1996), 266–290

    We give an evolving algebra (= abstract state machine) solution for the well-known railroad crossing problem, and we use the occasion to experiment with computations where agents perform instantaneous actions in continuous time and some agents fire at the moment they are enabled.

117. Yuri Gurevich, James K. Huggins: Equivalence is in the Eye of the Beholder. Theoretical Computer Science 179:1–2 (1997), 353–380

    In a provocative paper "Processes are in the Eye of the Beholder" in the same issue of TCS (pp. 333–351), Lamport points out "the insubstantiality of processes" by proving the equivalence of two different decompositions of the same intuitive algorithm. More exactly, each of the two distributed algorithms is described by a formula in Lamport's favorite temporal logic and then the two formulas are proved equivalent. We point out that the equivalence of algorithms is itself in the eye of the beholder. In this connection, we analyze in what sense the two distributed algorithms are and are not equivalent. Our equivalence proof is direct and does not require formalizing algorithms as logic formulas.

118. Andreas Blass, Yuri Gurevich: The Linear Time Hierarchy Theorems for RAMs and Abstract State Machines. Springer J. of Universal Computer Science 3:4 (April 1997), 247–278

    Contrary to polynomial time, linear time badly depends on the computation model. In 1992, Neil Jones designed a couple of computation models where the linear-speed-up theorem fails and linear-time computable functions form a proper hierarchy. However, the linear time of Jones's models is too restrictive. We prove linear-time hierarchy theorems for random access machines and Gurevich's abstract state machines (formerly, evolving algebras). The latter generalization is

harder and more important because of the greater flexibility of the ASM model. One long-term goal of this line of research is to prove linear lower bounds for linear time problems.

119. Yuri Gurevich, Marc Spielmann: Recursive Abstract State Machines. Springer J. of Universal Computer Science 3:4 (April 1997), 233–246

    The abstract state machine (ASM) thesis, supported by numerous applications, asserts that ASMs express algorithms on their natural abstraction levels directly and essentially coding-free. The only objection raised to date has been that ASMs are iterative in their nature, whereas many algorithms are naturally recursive. There seems to be an inherent contradiction between (i) the ASM idea of explicit and comprehensive states, and (ii) higher level recursion with its hiding of the stack.

    But consider recursion more closely. When an algorithm A calls an algorithm B, a clone of B is created and this clone becomes a slave of A. This raises the idea of treating recursion as an implicitly multi-agent computation. Slave agents come and go, and the master/slave hierarchy serves as the stack.

    Building upon this idea, we suggest a definition of recursive ASMs. The implicit use of distributed computing has an important side benefit: it leads naturally to concurrent recursion. In addition, we reduce recursive ASMs to distributed ASMs. If desired, one can view recursive notation as mere abbreviation.

120. Andreas Blass, Yuri Gurevich, Saharon Shelah: Choiceless Polynomial Time. Annals of Pure and Applied Logic 100 (1999), 141–187

    The question "Is there a computation model whose machines do not distinguish between isomorphic structures and compute exactly polynomial time properties?" became a central question of finite model theory. One of us conjectured a negative answer [74]. A related question is what portion of PTIME can be naturally captured by a computation model. (Notice that we speak about computation whose inputs are arbitrary finite structures, e.g. graphs. In a special case of ordered structures, the desired computation model is that of PTIME-bounded Turing machines.) Our idea is to capture the portion of PTIME where algorithms are not allowed arbitrary choice but parallelism is allowed and, in some cases, implements choice. Our computation model is a PTIME version of abstract state machines. Our machines are able to PTIME simulate all other PTIME machines in the literature, and they are more programmer-friendly. A more difficult theorem shows that the computation model does not capture all PTIME.

121. Scott Dexter, Patrick Doyle, Yuri Gurevich: Gurevich Abstract State Machines and Schoenhage Storage Modification Machines. Springer J. of Universal Computer Science 3:4 (April 1997), 279–303

    We show that, in a strong sense, Schoenhage's storage modification machines are equivalent to unary basic abstract state machines without external functions. The unary restriction can be removed if the storage modification machines are equipped with a pairing function in an appropriate way.

122. Charles Wallace, Yuri Gurevich, Nandit Soparkar: A Formal Approach to Recovery in Transaction-Oriented Database Systems. Springer J. of Universal Computer Science 3:4 (April 1997), 320–340

    Failure resilience is an essential requirement for transaction-oriented database systems, yet there has been little effort to specify and verify techniques for failure recovery formally. The desire to improve performance has resulted in algorithms of

considerable sophistication, understood by few and prone to errors. In this paper, we show how the formal methodology of Gurevich's Abstract State Machines can elucidate recovery and provide formal rigor to the design of a recovery algorithm. In a series of refinements, we model recovery at several levels of abstraction, verifying the correctness of each model. This initial work indicates that our approach can be applied to more advanced recovery mechanisms.

123. Yuri Gurevich: Platonism, Constructivism, and Computer Proofs vs. Proofs by Hand. Originally in BEATCS 57 (October 1995), 145–166. A slightly revised version in: Current Trends in Theoretical Computer Science, World Scientific (2001), 281–302

    In one of Krylov's fables, a small dog, Moska, barks at the elephant who pays no attention whatsoever to Moska. This image comes to my mind when I think of constructive mathematics versus "classical" (that is mainstream) mathematics. In this article, we put a few words into the elephant's mouth. The idea to write such an article came to me in the summer of 1995 when I came across a fascinating 1917 bet between the constructivist Hermann Weyl and George Polya, a classical mathematician. An English translation of the bet (from German) is found in the article.

    Our main objection to the historical constructivism is that it has not been sufficiently constructive. The constructivists have been obsessed with computability and have not paid sufficient attention to the feasibility of algorithms. However, the constructivists' criticism of classical mathematics has a point. Instead of dismissing constructivism offhandedly, it makes sense to come up with a positive alternative, an antithesis to historical constructivism. We believe that we have found such an alternative. In fact, it is well known and very popular in computer science, namely, the principle of separating concerns.

    [Added in July 2006] The additional part on computer proofs vs. proofs by hand was a result of frustration that many computer scientists would not trust informal mathematical proofs, while many mathematicians would not trust computer proofs. I seemed obvious to me that, on the large scale, proving is not only hard but also is imperfect and has an engineering character. We need informal proofs and computer proofs and more, such as stratification, experimentation.

124. Natasha Alechina, Yuri Gurevich: Syntax vs. Semantics on Finite Structures. In: J. Mycielski et al. (eds.) Structures in Logic and Computer Science: A Selection of Essays in Honor of Andrzej Ehrenfeucht, Springer LNCS 1261 (1997), 14–33

    Logic preservation theorems often have the form of a syntax/semantics correspondence. For example, the Tarski-Łoś theorem asserts that a first-order sentence is preserved by extensions if and only if it is equivalent to an existential sentence. Many of these correspondences break when one restricts attention to finite models. In such a case, one may attempt to find a new semantical characterization of the old syntactical property or a new syntactical characterization of the old semantical property. The goal of this paper is to provoke such a study. In particular, we give a simple semantical characterization of existential formulas on finite structures.

125. Anatoli Degtyarev, Yuri Gurevich, Andrei Voronkov: Herbrand's Theorem and Equational Reasoning: Problems and Solutions. Originally in BEATCS 60 (Oct 1996), 78–95. Reprinted in: Current Trends in Theoretical Computer Science, World Scientific (2001), 303–326

The article (written in a popular form) explains that a number of different algorithmic problems related to Herbrand's theorem happen to be equivalent. Among these problems are the intuitionistic provability problem for the existential fragment of first-order logic with equality, the intuitionistic provability problem for the prenex fragment of first-order with equality, and the simultaneous rigid E-unification problem (SREU). The article explains an undecidability proof of SREU and decidability proofs for special cases. It contains an extensive bibliography on SREU.

126. Yuri Gurevich, Margus Veanes: Logic with Equality: Partisan Corroboration and Shifted Pairing. Information and Computation 152:2 (August 1999), 205–235

Herbrand's theorem plays a fundamental role in automated theorem proving methods based on tableaux. The crucial step in procedures based on such methods can be described as the corroboration (or Herbrand skeleton) problem: given a positive integer m and a quantifier-free formula, find a valid disjunction of m instantiations of the formula. In the presence of equality (which is the case in this paper), this problem was recently shown to be undecidable. The main contributions of this paper are two theorems. The Partisan Corroboration Theorem relates corroboration problems with different multiplicities. The Shifted Pairing Theorem is a finite tree-automata formalization of a technique for proving undecidability results through direct encodings of valid Turing machine computations. The theorems are used to explain and sharpen several recent undecidability results related to the corroboration problem, the simultaneous rigid E-unification problem and the prenex fragment of intuitionistic logic with equality.

127a. A. Degtyarev, Y. Gurevich, P. Narendran, M. Veanes, A. Voronkov: The Decidability of Simultaneous Rigid E-Unification with One Variable. RTA'98, 9th Conf. on Rewriting Techniques and Applications, Tsukuba, Japan, March 30 – April 1, 1998

The title problem is proved decidable and in fact EXPTIME-complete. Furthermore, the problem becomes PTIME-complete if the number of equations is bounded by any (positive) constant. It follows that the $\forall^*\exists\forall^*$ fragment of intuitionistic logic with equality is decidable, which contrasts with the undecidability of the EE fragment [126]. Notice that simultaneous rigid E-unification with two variables and only three rigid equations is undecidable [126].

127b. A. Degtyarev, Y. Gurevich, P. Narendran, M. Veanes, A. Voronkov: Decidability and Complexity of Simultaneous Rigid E-Unification with One Variable and Related Results. Theoretical Computer Science 243:1–2 (August 2000), 167–184

The journal version of [127a] containing also a decidability proof for the case of simultaneous rigid E-unification when each rigid equation either contains (at most) one variable or else has a ground left-hand side and the right-hand side of the form $x = y$ where x and y are variables.

128a. Yuri Gurevich, Andrei Voronkov: Monadic Simultaneous Rigid E-Unification and Related Problems. ICALP'97, 24th Intern. Colloquium on Automata, Languages and Programming, Springer LNCS 1256 (1997), 154–165

We study the monadic case of a decision problem known as simultaneous rigid E-unification. We show its equivalence to an extension of word equations. We prove decidability and complexity results for special cases of this problem.

128b. Yuri Gurevich, Andrei Voronkov: Monadic Simultaneous Rigid E-Unification. Theoretical Computer Science 222:1–2 (1999), 133–152

The journal version of [128a].

129. Yuri Gurevich: May 1997 Draft of the ASM Guide. Tech Report CSE-TR-336-97, EECS Dept, University of Michigan, 1997

The draft improves upon the ASM syntax (and appears here because it is used by the ASM community and it is not going to be published).

130. Yuri Gurevich, Alex Rabinovich: Definability and Undefinability with Real Order at the Background. JSL 65:2 (2000), 946–958

Let R be the real order, that is the set of real numbers together with the standard order of reals. Let I be the set of integer numbers, let Y range over subsets of I, let $P(I, X)$ be a monadic second-order formula about R, and let F be the collection of all subsets X of I such that $P(I, X)$ holds in R. Even though F is a collection of subsets of I, its definition may involve quantification over reals and over sets of reals. In that sense, F is defined with the background of real order. Is that background essential or not? Maybe there is a monadic second-order formula $Q(X)$ about I that defines F (so that F is the collection of all subsets X of I such that $Q(X)$ holds in I). We prove that this is indeed the case, for any monadic second-order formula $P(I, X)$. The claim remains true if the set I of integers is replaced above with any closed subset of R. The claim fails for some open subsets.

131. Yuri Gurevich: From Invariants to Canonization. Originally in BEATCS 63 (October 1997). Reprinted in: Current Trends in Theoretical Computer Science, World Scientific (2001), 327–331

We show that every polynomial-time full-invariant algorithm for graphs gives rise to a polynomial-time canonization algorithm for graphs.

132. Andreas Blass, Yuri Gurevich, Vladik Kreinovich, Luc Longpré: A Variation on the Zero-One Law. Information Processing Letters 67 (1998), 29–30

Given a decision problem P and a probability distribution over binary strings, do this: for each n, draw independently an instance $x(n)$ of P of length n. What is the probability that there is a polynomial time algorithm that solves all instances $x(n)$? The answer is: zero or one.

133. Erich Grädel, Yuri Gurevich, Colin Hirsch: The Complexity of Query Reliability. PODS'98, 1998 ACM Symposium on Principles of Database Systems

We study the reliability of queries on databases with uncertain information. It turns out that $\mathrm{FP}^{\#P}$ is the typical complexity class and that many results generalize to metafinite databases which allow one to use common SQL aggregate functions.

134. Thomas Eiter, Georg Gottlob, Yuri Gurevich: Existential Second-Order Logic over Strings. Journal of the ACM 47:1 (January 2000), 77–131

We study existential second-order logic over finite strings. For every prefix class C, we determine the complexity of the model checking problem restricted to C. In particular, we prove that, in the case of the Ackermann class, for every formula φ, there is a finite automaton A that solves the model checking problem for φ.

135. Andreas Blass, Yuri Gurevich: The Logic of Choice. JSL 65:3 (September 2000), 1264–1310

We study extensions of first-order logic with the choice construct (choose $x : \varphi(x)$). We prove some results about Hilbert's epsilon operator, but in the main part of the paper we consider the case when all choices are independent.

136. Yuri Gurevich: The Sequential ASM Thesis. Originally in BEATCS 67 (February 1999), 98–124. Reprinted in: Current Trends in Theoretical Computer Science, World Scientific (2001), 363–392

The thesis is that every sequential algorithm, on any level of abstraction, can be viewed as a sequential abstract state machine. (Abstract state machines, ASMs, used to be called evolving algebras.) The sequential ASM thesis and its extensions inspired diverse applications of ASMs. The early applications were driven, at least partially, by the desire to test the thesis. Different programming languages were the obvious challenges. (A programming language L can be viewed as an algorithm that runs a given L program on given data.) From there, applications of (not necessarily sequential) ASMs spread into many directions. So far, the accumulated experimental evidence seems to support the sequential thesis. There is also a speculative philosophical justification of the thesis. It was barely sketched in the literature, but it was discussed at much greater length in numerous lectures of mine. Here I attempt to write down some of those explanations. This article does not presuppose any familiarity with ASMs.

A later note: [141] is a much revised and polished journal version.

137. Giuseppe Del Castillo, Yuri Gurevich, Karl Stroetmann: Typed Abstract State Machines. Unfinished manuscript (1998).

This manuscript was never published. The work, done sporadically in 1996–98, was driven by the enthusiasm of Karl Stroetmann of Siemens. Eventually he was reassigned away from ASM applications, and the work stopped. The item wasn't removed from the list because some of its explorations may be useful. (An additional minor reason was to avoid changing the numbers of the subsequent items.)

138. Yuri Gurevich, Dean Rosenzweig: Partially Ordered Runs: a Case Study. In: Abstract State Machines: Theory and Applications, Springer LNCS 1912 (2000), 131–150

We look at some sources of insecurity and difficulty in reasoning about partially ordered runs of distributed abstract state machines, and propose some techniques to facilitate such reasoning. As a case study, we prove in detail correctness and deadlock–freedom for general partially ordered runs of distributed ASM models of Lamport's Bakery Algorithm.

139. Andreas Blass, Yuri Gurevich, Jan Van den Bussche: Abstract state machines and computationally complete query languages. Information and Computation 174:1 (2002), 20–36. An earlier version in: Abstract State Machines: Theory and Applications, Springer LNCS 1912 (2000), 22–33

Abstract state machines (ASMs) form a relatively new computation model holding the promise that they can simulate any computational system in lockstep. In particular, an instance of the ASM model has recently been introduced for computing queries to relational databases [120]. This model, to which we refer as the BGS model, provides a powerful query language in which all computable queries can be expressed. In this paper, we show that when one is only interested in polynomial-time computations, BGS is strictly more powerful than both QL and WHILE_NEW, two well-known computationally complete query languages. We then show that when a language such as WHILE_NEW is extended with a duplicate elimination mechanism, polynomial-time simulations between the language and BGS become possible.

140. Yuri Gurevich, Wolfram Schulte, Charles Wallace: Investigating Java Concurrency Using Abstract State Machines. In: Abstract State Machines: Theory and Applications, Springer LNCS 1912 (2000), 151–176

We present a mathematically precise, platform-independent model of Java concurrency using the Abstract State Machine method. We cover all aspects of Java threads and synchronization, gradually adding details to the model in a series of steps. We motivate and explain each concurrency feature, and point out subtleties, inconsistencies and ambiguities in the official, informal Java specification.

141. Yuri Gurevich: Sequential Abstract State Machines capture Sequential Algorithms. ACM ToCL 1:1 (July 2000), 77–111

What are sequential algorithms exactly? Our claim, known as the sequential ASM thesis, has been that, as far as behavior is concerned, sequential algorithms are exactly sequential abstract state machines: For every sequential algorithm A, there is a sequential abstract state machine B that is behaviorally identical to A. In particular, B simulates A step for step. In this paper we prove the sequential ASM thesis, so that it becomes a theorem. But how can one possibly prove a thesis? Here is what we do. We formulate three postulates satisfied by all sequential algorithms (and, in particular, by sequential abstract state machines). This leads to the following definition: a sequential algorithm is any object that satisfies the three postulates. At this point the thesis becomes a precise statement. And we prove the statement.

This is a non-dialog version of the dialog [136]. An intermediate version was published in MSR-TR-99-65

141a. Yuri Gurevich: Sequential Abstract State Machines capture Sequential Algorithms. Russian translation of [141], by P.G. Emelyanov. In: Marchuk A.G. (ed.) Formal Methods and Models of Informatics, System Informatics 9 (2004), 7–50, Siberian Branch of the Russian Academy of Sciencies

142. Andreas Blass, Yuri Gurevich: The Underlying Logic of Hoare Logic. Originally in BEATCS 70 (February 2000), 82–110. Reprinted in: Current Trends in Theoretical Computer Science, World Scientific (2001), 409–436

Formulas of Hoare logic are asserted programs $\varphi\ P\ \psi$ where P is a program and φ, ψ are assertions. The language of programs varies; in the 1980 survey by Krzysztof Apt, one finds the language of while programs and various extensions of it. But the assertions are traditionally expressed in first-order logic (or extensions of it). In that sense, first-order logic is the underlying logic of Hoare logic. We question the tradition and demonstrate, on the simple example of while programs, that alternative assertion logics have some advantages. For some natural assertion logics, the expressivity hypothesis in Cook's completeness theorem is automatically satisfied.

143. Andreas Blass, Yuri Gurevich: Background, Reserve, and Gandy Machines. In: P. Clote and H. Schwichtenberg (eds.) CSL'2000, Springer LNCS 1862 (2000), 1–17

Algorithms often need to increase their working space, and it may be convenient to pretend that the additional space was really there all along but was not previously used. In particular, abstract state machines have, by definition [103], an infinite reserve. Although the reserve is a naked set, it is often desirable to have some external structure over it. For example, in [120] every state was required to include all finite sets of its atoms, all finite sets of these, etc. In this connection,

we define the notion of a background class of structures. Such a class specifies the constructions (like finite sets or lists) available as "background" for algorithms.

The importation of reserve elements must be non-deterministic, since an algorithm has no way to distinguish one reserve element from another. But this sort of non-determinism is much more benign than general non-determinism. We capture this intuition with the notion of inessential non-determinism. Alternatively, one could insist on specifying a particular one of the available reserve elements to be imported. This is the approach used in [Robin Gandy, "Church's thesis and principles for mechanisms". In: J. Barwise et al. (eds.) The Kleene Symposium, North-Holland, 1980, 123–148]. The price of this insistence is that the specification cannot be algorithmic. We show how to turn a Gandy-style deterministic, non-algorithmic process into a non-deterministic algorithm of the sort described above, and we prove that Gandy's notion of "structural" for his processes corresponds to our notion of "inessential non-determinism."

144. Andreas Blass, Yuri Gurevich: Choiceless Polynomial Time Computation and the Zero-One Law. In: P. Clote and H. Schwichtenberg (eds.) CSL'2000, Springer LNCS 1862 (2000), 18–40

This paper is a sequel to [120], a commentary on [Saharon Shelah (#634) "Choiceless polynomial time logic: inability to express", same proceedings], and an abridged version of [149] that contains complete proofs of all the results presented here. The BGS model of computation was defined in [120] with the intention of modeling computation with arbitrary finite relational structures as inputs, with essentially arbitrary data structures, with parallelism, but without arbitrary choices. It was shown that choiceless polynomial time, the complexity class defined by BGS programs subject to a polynomial time bound, does not contain the parity problem. Subsequently, Shelah proved a zero-one law for choiceless-polynomial-time properties. A crucial difference from the earlier results is this: Almost all finite structures have no non-trivial automorphisms, so symmetry considerations cannot be applied to them. Shelah's proof therefore depends on a more subtle concept of partial symmetry.

After struggling for a while with Shelah's proof, we worked out a presentation which we hope will be helpful for others interested in Shelah's ideas. We also added some related results, indicating the need for certain aspects of the proof and clarifying some of the concepts involved in it. Unfortunately, this material is not yet fully written up. The part already written, however, exceeds the space available to us in the present volume. We therefore present here an abridged version of that paper and promise to make the complete version available soon.

145. Mike Barnett, Egon Börger, Yuri Gurevich, Wolfram Schulte, Margus Veanes: Using Abstract State Machines at Microsoft: A Case Study. In: P. Clote and H. Schwichtenberg (eds.) CSL'2000, Springer LNCS 1862 (2000), 367–379

Our goal is to provide a rigorous method, clear notation and convenient tool support for high-level system design and analysis. For this purpose we use abstract state machines (ASMs). Here we describe a particular case study: modeling a debugger of a stack based runtime environment. The study provides evidence for ASMs being a suitable tool for building *executable* models of software systems on various abstraction levels, with precise refinement relationships connecting the models. High level ASM models of proposed or existing programs can be used throughout the software development cycle. In particular, ASMs can be used to model inter-component behavior on any desired level of detail. This allows

one to specify application programming interfaces more precisely than it is done currently.

145.5. Colin Campbell, Yuri Gurevich: Table ASMs. In: Formal Methods and Tools for Computer Science, Eurocast 2001, eds. R. Moreno-Diaz and A. Quesada-Arencibia, Universidad de Las Palmas de Gran Canaria, Canary Islands, Spain (February 2001), 286–290

Ideally, a good specification becomes the basis for implementing, testing and documenting the system it defines. In practice, producing a good specification is hard. Formal methods have been shown to be helpful in strengthening the meaning of specifications, but despite their power, few development teams have successfully incorporated them into their software processes. This experience indicates that producing a usable formal method is also hard.

This paper is the story of how a particular theoretical result, namely the normal forms of Abstract State Machines, motivated a genuinely usable form of specification that we call ASM Tables. We offer it for two reasons. The first is that the result is interesting in and of itself and – it is to be hoped – useful to the reader. The second is that our result serves as a case study of a more general principle, namely, that in bringing rigorous methods into everyday practice, one should not follow the example of Procrustes: we find that it is indeed better to adapt the bed to the person than the other way round. We also offer a demonstration that an extremely restricted syntactical form can still contain sufficient expressive power to describe all sequential machines.

146. Andreas Blass, Yuri Gurevich: Inadequacy of Computable Loop Invariants. ACM ToCL 2:1 (January 2001), 1–11

Hoare logic is a widely recommended verification tool. There is, however, a problem of finding easily-checkable loop invariants; it is known that decidable assertions do not suffice to verify WHILE programs, even when the pre- and post-conditions are decidable. We show here a stronger result: decidable invariants do not suffice to verify single-loop programs. We also show that this problem arises even in extremely simple contexts. Let N be the structure consisting of the set of natural numbers together with the functions $S(x) = x + 1$, $D(x) = 2x$ and function $H(x)$ that is equal to $x/2$ rounded down. There is a single-loop program P using only three variables x, y, z such that the asserted program

$$x = y = z = 0 \ \{P\} \ \mathit{false}$$

is partially correct on N but any loop invariant $I(x, y, z)$ for this asserted program is undecidable.

147. Yuri Gurevich, Alex Rabinovich: Definability in Rationals with Real Order in the Background. Journal of Logic and Computation 12:1 (2002), 1–11

The paper deals with logically definable families of sets of rational numbers. In particular, we are interested whether the families definable over the real line with a unary predicate for the rationals are definable over the rational order alone. Let $\varphi(X, Y)$ and $\psi(Y)$ range over formulas in the first-order monadic language of order. Let Q be the set of rationals and F be the family of subsets J of Q such that $\varphi(Q, J)$ holds over the real line. The question arises whether, for every formula φ, the family F can be defined by means of a formula $\psi(Y)$ interpreted over the rational order. We answer the question negatively. The answer remains negative if the first-order logic is strengthened to weak monadic second-order logic. The

answer is positive for the restricted version of monadic second-order logic where set quantifiers range over open sets. The case of full monadic second-order logic remains open.

148. Andreas Blass, Yuri Gurevich: A New Zero-One Law and Strong Extension Axioms. Originally in BEATCS 72 (October 2000), 103–122. Reprinted in: Current Trends in Theoretical Computer Science, World Scientific (2004), 99–118

This article is a part of the continuing column on Logic in Computer Science. One of the previous articles in the column was devoted to the zero-one laws for a number of logics playing prominent role in finite model theory: first-order logic FO, the extension FO+LFP of first-order logic with the least fixed-point operator, and the infinitary logic where every formula uses finitely many variables [95]. Recently Shelah proved a new, powerful, and surprising zero-one law. His proof uses so-called strong extension axioms. Here we formulate Shelah's zero-one law and prove a few facts about these axioms. In the process we give a simple proof for a "large deviation" inequality à la Chernoff.

149. Andreas Blass, Yuri Gurevich: Strong Extension Axioms and Shelah's Zero-One Law for Choiceless Polynomial Time. JSL 68:1 (2003), 65–131

This paper developed from Shelah's proof of a zero-one law for the complexity class "choiceless polynomial time," defined by Shelah and the authors. We present a detailed proof of Shelah's result for graphs, and describe the extent of its generalizability to other sorts of structures. The extension axioms, which form the basis for earlier zero-one laws (for first-order logic, fixed-point logic, and finite-variable infinitary logic) are inadequate in the case of choiceless polynomial time; they must be replaced by what we call the strong extension axioms. We present an extensive discussion of these axioms and their role both in the zero-one law and in general. ([144] is an abridged version of this paper, and [148] is a popular version of this paper.)

150. Andreas Blass, Yuri Gurevich, Saharon Shelah: On Polynomial Time Computation Over Unordered Structures. JSL 67:3 (2002), 1093–1125

This paper is motivated by the question whether there exists a logic capturing polynomial time computation over unordered structures. We consider several algorithmic problems near the border of the known, logically defined complexity classes contained in polynomial time. We show that fixpoint logic plus counting is stronger than might be expected, in that it can express the existence of a complete matching in a bipartite graph. We revisit the known examples that separate polynomial time from fixpoint plus counting. We show that the examples in a paper of Cai, Fürer, and Immerman, when suitably padded, are in choiceless polynomial time yet not in fixpoint plus counting. Without padding, they remain in polynomial time but appear not to be in choiceless polynomial time plus counting. Similar results hold for the multipede examples of Gurevich and Shelah, except that their final version of multipedes is, in a sense, already suitably padded. Finally, we describe another plausible candidate, involving determinants, for the task of separating polynomial time from choiceless polynomial time plus counting.

150a. Andreas Blass, Yuri Gurevich: A Quick Update on the Open Problems in article [150] (December 2005).

151. Yuri Gurevich: Logician in the land of OS: Abstract State Machines at Microsoft. LICS 2001, IEEE Symp. on Logic in Computer Science, IEEE Computer Society (2001), 129–136

Analysis of foundational problems like "What is computation?" leads to a sketch of the paradigm of abstract state machines (ASMs). This is followed by a brief discussion on ASMs applications. Then we present some theoretical problems that bridge between the traditional LICS themes and abstract state machines.

152. Anuj Dawar, Yuri Gurevich: Fixed Point Logics. The Bulletin of Symbolic Logic 8:1 (2002), 65–88

    Fixed-point logics are extensions of first-order predicate logic with fixed point operators. A number of such logics arose in finite model theory but they are of interest to much larger audience, e.g. AI, and there is no reason why they should be restricted to finite models. We review results established in finite model theory, and consider the expressive power of fixed-point logics on infinite structures.

153. Uwe Glaesser, Yuri Gurevich, Margus Veanes: Universal Plug and Play Machine Models. MSR-TR -2001-59

    Recently, Microsoft took a lead in the development of a standard for peer-to-peer network connectivity of various intelligent appliances, wireless devices and PCs. It is called the Universal Plug and Play Device Architecture (UPnP). We construct a high-level Abstract State Machine (ASM) model for UPnP. The model is based on the ASM paradigm for distributed systems with real-time constraints and is executable in principle. For practical execution, we use AsmL, the Abstract state machine Language, developed at Microsoft Research and integrated with Visual Studio and COM. This gives us an AsmL model, a refined version of the ASM model. The third part of this project is a graphical user interface by means of which the runs of the AsmL model are controlled and inspected at various levels of detail as required for simulation and conformance testing, for example.

154. Wolfgang Grieskamp, Yuri Gurevich, Wolfram Schulte and Margus Veanes: Generating Finite State Machines from Abstract State Machines. ISSTA 2002, International Symposium on Software Testing and Analysis, ACM Software Engineering Notes 27:4 (2002), 112–122

    We give an algorithm that derives a finite state machine (FSM) from a given abstract state machine (ASM) specification. This allows us to integrate ASM specs with the existing tools for test-case generation from FSMs. ASM specs are executable, but have typically too many, often infinitely many, states. We group ASM states into finitely many hyperstates, which are the nodes of the FSM. The links of the FSM are induced by the ASM state transitions.

155. Yuri Gurevich, Wolfram Schulte, Margus Veanes: Toward Industrial Strength Abstract State Machines. MSR-TR-2001-98

    A powerful practical ASM language, called AsmL, is being developed in Microsoft Research by the group on Foundations of Software Engineering. AsmL extends the language of original ASMs in a number of directions. We describe some of these extensions.

156. Yuri Gurevich, Nikolai Tillmann: Partial Updates: Exploration. Springer J. of Universal Computer Science 7:11 (2001), 918–952

    The partial update problem for parallel abstract state machines has manifested itself in the cases of counters, sets and maps. We propose a solution of the problem that lends itself to an efficient implementation and covers the three cases mentioned above. There are other cases of the problem that require a more general framework.

157-1. Andreas Blass, Yuri Gurevich: Abstract State Machines Capture Parallel Algorithms. ACM ToCL 4:4 (October 2003), 578–651

We give an axiomatic description of parallel, synchronous algorithms. Our main result is that every such algorithm can be simulated, step for step, by an abstract state machine with a background that provides for multisets. See also [157-2].

157-2. Andreas Blass, Yuri Gurevich: Abstract State Machines Capture Parallel Algorithms: Correction and Extension. ACM ToCL 9:3 (June 2008), Article 19

We consider parallel algorithms working in sequential global time, for example circuits or parallel random access machines (PRAMs). Parallel abstract state machines (parallel ASMs) are such parallel algorithms, and the parallel ASM thesis asserts that every parallel algorithm is behaviorally equivalent to a parallel ASM. In an earlier paper [157-1], we axiomatized parallel algorithms, proved the ASM thesis and proved that every parallel ASM satisfies the axioms. It turned out that we were too timid in formulating the axioms; they did not allow a parallel algorithm to create components on the fly. This restriction did not hinder us from proving that the usual parallel models, like circuits or PRAMs or even alternating Turing machines, satisfy the postulates. But it resulted in an error in our attempt to prove that parallel ASMs always satisfy the postulates. To correct the error, we liberalize our axioms and allow on-the-fly creation of new parallel components. We believe that the improved axioms accurately express what parallel algorithms ought to be. We prove the parallel thesis for the new, corrected notion of parallel algorithms, and we check that parallel ASMs satisfy the new axioms.

158. Andreas Blass, Yuri Gurevich: Algorithms vs. Machines. Originally in BEATCS 77 (June 2002), 96–118. Reprinted in: Current Trends in Theoretical Computer Science, World Scientific (2004), 215–236

In a recent paper, the logician Yiannis Moschovakis argues that no state machine describes mergesort on its natural level of abstraction. We do just that. Our state machine is a recursive ASM.

159. Uwe Glaesser, Yuri Gurevich, Margus Veanes: Abstract Communication Model for Distributed Systems. IEEE Transactions on Software Engineering 30:7 (July 2004), 458–472

In some distributed and mobile communication models, a message disappears in one place and miraculously appears in another. In reality, of course, there are no miracles. A message goes from one network to another; it can be lost or corrupted in the process. Here we present a realistic but high-level communication model where abstract communicators represent various nets and subnets. The model was originally developed in the process of specifying a particular network architecture, namely the Universal Plug and Play architecture. But it is general. Our contention is that every message-based distributed system, properly abstracted, gives rise to a specialization of our abstract communication model. The purpose of the abstract communication model is not to design a new kind of network; rather it is to discover the common part of all message-based communication networks. The generality of the model has been confirmed by its successful reuse for very different distributed architectures. The model is based on distributed abstract state machines. It is implemented in the specification language AsmL and is being used for testing distributed systems.

160. Andreas Blass, Yuri Gurevich: Pairwise Testing. Originally in BEATCS 78 (October 2002), 100–132. Reprinted in: Current Trends in Theoretical Computer Science, World Scientific (2004), 237–266

We discuss the following problem, which arises in software testing. Given some independent parameters (of a program to be tested), each having a certain finite set of possible values, we intend to test the program by running it several times. For each test, we give the parameters some (intelligently chosen) values. We want to ensure that for each pair of distinct parameters, every pair of possible values is used in at least one of the tests. And we want to do this with as few tests as possible.

161. Yuri Gurevich, Nikolai Tillmann: Partial Updates. Theoretical Computer Science 336:2–3 (26 May 2005), 311–342. (A preliminary version in: Abstract State Machines 2003, Springer LNCS 2589 (2003), 57–86)

A datastructure instance, e.g. a set or file or record, may be modified independently by different parts of a computer system. The modifications may be nested. Such hierarchies of modifications need to be efficiently checked for consistency and integrated. This is the problem of partial updates in a nutshell. In our first paper on the subject [156], we developed an algebraic framework which allowed us to solve the partial update problem for some useful datastructures including counters, sets and maps. These solutions are used for the efficient implementation of concurrent data modifications in the specification language AsmL. The two main contributions of this paper are (i) a more general algebraic framework for partial updates and (ii) a solution of the partial update problem for sequences and labeled ordered trees.

162. Yuri Gurevich, Saharon Shelah: Spectra of Monadic Second-Order Formulas with One Unary Function. LICS 2003, 18th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society (2003), 291–300

We prove that the spectrum of any monadic second-order formula F with one unary function symbol (and no other function symbols) is eventually periodic, so that there exist natural numbers $p > 0$ (a period) and t (a p-threshold) such that if F has a model of cardinality $n > t$ then it has a model of cardinality $n + p$.

(In the web version, some additional proof details are provided because some readers asked for them.)

163. Mike Barnett, Wolfgang Grieskamp, Yuri Gurevich, Wolfram Schulte, Nikolai Tillmann, Margus Veanes: Scenario-oriented Modeling in AsmL and Its Instrumentation for Testing. In: 2nd International Workshop on Scenarios and State Machines: Models, Algorithms, and Tools, (2003) 8–14, held at ICSE 2003, International Conference on Software Engineering 2003

We present an approach for modeling use cases and scenarios in the Abstract state machine Language and discuss how to use such models for validation and verification purposes.

164. Andreas Blass, Yuri Gurevich: Algorithms: A Quest for Absolute Definitions. Originally in BEATCS 81 (October 2003), 195–225. Reprinted in: Current Trends in Theoretical Computer Science, World Scientific (2004), 283–311. Reprinted in: A. Olszewski et al. (eds.) Church's Thesis After 70 Years, Ontos Verlag (2006), 24–57

What is an algorithm? The interest in this foundational problem is not only theoretical; applications include specification, validation and verification of software and hardware systems. We describe the quest to understand and define the notion of algorithm. We start with the Church-Turing thesis and contrast

Church's and Turing's approaches, and we finish with some recent investigations.

165. Yuri Gurevich: Abstract State Machines: An Overview of the Project. In: D. Seipel and J. M. Turull-Torres (eds.) Foundations of Information and Knowledge Systems, Springer LNCS 2942 (2004), 6–13

    We quickly survey the ASM project, from its foundational roots to industrial applications.

166. Andreas Blass, Yuri Gurevich: Ordinary Interactive Small-Step Algorithms, I. ACM ToCL 7:2 (April 2006), 363–419. A preliminary version was published as MSR-TR-2004-16

    This is the first in a series of papers extending the Abstract State Machine Thesis – that arbitrary algorithms are behaviorally equivalent to abstract state machines – to algorithms that can interact with their environments during a step, rather than only between steps. In the present paper, we describe, by means of suitable postulates, those interactive algorithms that (1) proceed in discrete, global steps, (2) perform only a bounded amount of work in each step, (3) use only such information from the environment as can be regarded as answers to queries, and (4) never complete a step until all queries from that step have been answered. We indicate how a great many sorts of interaction meet these requirements. We also discuss in detail the structure of queries and replies and the appropriate definition of equivalence of algorithms. Finally, motivated by our considerations concerning queries, we discuss a generalization of first-order logic in which the arguments of function and relation symbols are not merely tuples of elements but orbits of such tuples under groups of permutations of the argument places.

167. Yuri Gurevich: Intra-Step Interaction. In: W. Zimmerman and B. Thalheim (eds.) Abstract State Machines 2004, Springer LNCS 3052 (2004), 1–5

    For a while it seemed possible to pretend that all interaction between an algorithm and its environment occurs inter-step, but not anymore. Andreas Blass, Benjamin Rossman and the speaker are extending the Small-Step Characterization Theorem (that asserts the validity of the sequential version of the ASM thesis) and the Wide-Step Characterization Theorem (that asserts the validity of the parallel version of the ASM thesis) to intra-step interacting algorithms.

    A later comment: This was my first talk on intra-step interactive algorithms. The intended audience was the ASM community. [174] is a later talk on this topic, and it is addressed to a general computer science audience.

168. Yuri Gurevich, Rostislav Yavorskiy: Observations on the Decidability of Transitions. In: W. Zimmerman and B. Thalheim (eds.) Abstract State Machines 2004, Springer LNCS 3052 (2004), 161–168

    Consider a multiple-agent transition system such that, for some basic types $T_1, ..., T_n$, the state of any agent can be represented as an element of the Cartesian product $T_1 \times \cdots \times T_n$. The system evolves by means of global steps. During such a step, new agents may be created and some existing agents may be updated or removed, but the total number of created, updated and removed agents is uniformly bounded. We show that, under appropriate conditions, there is an algorithm for deciding assume-guarantee properties of one-step computations. The result can be used for automatic invariant verification as well as for finite state approximation of the system in the context of test-case generation from AsmL specifications.

169. Yuri Gurevich, Benjamin Rossman, Wolfram Schulte: Semantic Essence of AsmL. Theoretical Computer Science 343:3 (17 October 2005), 370–412 Originally published as MSR-TR-2004-27

The Abstract state machine Language, AsmL, is a novel executable specification language based on the theory of Abstract State Machines. AsmL is object-oriented, provides high-level mathematical data-structures, and is built around the notion of synchronous updates and finite choice. AsmL is fully integrated into the .NET framework and Microsoft development tools. In this paper, we explain the design rationale of AsmL and provide static and dynamic semantics for a kernel of the language.

169a. Yuri Gurevich, Benjamin Rossman, Wolfram Schulte: Semantic Essence of AsmL: Extended Abstract. In: F. S. de Boer et al. (eds.) FMCO 2003, Formal Methods of Components and Objects, Springer LNCS 3188 (2004), 240–259

This is an extended abstract of article [169].

170. Andreas Blass, Yuri Gurevich: Ordinary Interactive Small-Step Algorithms, II. ACM ToCL 8:3 (July 2007), article 15. A preliminary version was published as a part of MSR-TR-2004-88

This is the second in a series of three papers extending the proof of the Abstract State Machine Thesis – that arbitrary algorithms are behaviorally equivalent to abstract state machines – to algorithms that can interact with their environments during a step rather than only between steps. The first paper is [166]. As in that paper, we are concerned here with ordinary, small-step, interactive algorithms. This means that the algorithms (1) proceed in discrete, global steps, (2) perform only a bounded amount of work in each step, (3) use only such information from the environment as can be regarded as answers to queries, and (4) never complete a step until all queries from that step have been answered. After reviewing the previous paper's formal description of such algorithms and the definition of behavioral equivalence, we define ordinary, interactive, small-step abstract state machines (ASM's). Except for very minor modifications, these are the machines commonly used in the ASM literature. We define their semantics in the framework of ordinary algorithms, and we show that they satisfy the postulates for these algorithms. This material lays the groundwork for the final paper in the series, in which we shall prove the Abstract State Machine Thesis for ordinary, intractive, small-step algorithms: All such algorithms are equivalent to ASMs.

171. Andreas Blass, Yuri Gurevich: Ordinary Interactive Small-Step Algorithms, III. ACM ToCL 8:3 (July 2007), article 16. A preliminary version was published as a part of MSR-TR-2004-88

This is the third in a series of three papers extending the proof of the Abstract State Machine Thesis – that arbitrary algorithms are behaviorally equivalent to abstract state machines – to algorithms that can interact with their environments during a step rather than only between steps. The first two papers are [166] and [170]. As in those papers, we are concerned here with ordinary, small-step, interactive algorithms. After reviewing the previous papers' definitions of such algorithms, of behavioral equivalence, and of abstract state machines (ASMs), we prove the main result: Every ordinary, interactive, small-step algorithm is behaviorally equivalent to an ASM. We also discuss some possible variations of and additions to the ASM semantics.

172. Andreas Blass, Yuri Gurevich: Why Sets? BEATCS 84 (October 2004). Revised and published as MSR-TR-2006-138; then reprinted in: A. Avron et al. (eds.) Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday, Springer LNCS 4800 (2008).

    Sets play a key role in foundations of mathematics. Why? To what extent is it an accident of history? Imagine that you have a chance to talk to mathematicians from a far away planet. Would their mathematics be set-based? What are the alternatives to the set-theoretic foundation of mathematics? Besides, set theory seems to play a significant role in computer science, in particular in database theory and formal methods. Is there a good justification for that? We discuss these and related issues.

173. Andreas Blass, Yuri Gurevich, Lev Nachmanson, Margus Veanes: Play to Test. MSR-TR-2005-04. FATES 2005, 5th International Workshop on Formal Approaches to Testing of Software, Edinburgh (July 2005)

    Testing tasks can be viewed (and organized!) as games against nature. We introduce and study reachability games. Such games are ubiquitous. A single industrial test suite may involve many instances of a reachability game. Hence the importance of optimal or near optimal strategies for reachability games. We find out when exactly optimal strategies exist for a given reachability game, and how to construct them.

174. Yuri Gurevich: Interactive Algorithms 2005 with Added Appendix. In: D. Goldin et al. (eds.) Interactive Computation: The New Paradigm, Springer-Verlag (2006), 165–182. Originally in: J. Jedrzejowicz and A. Szepietowski (eds.) Proceedings of MFCS 2005 Math Foundations of Computer Science (2005), Gdansk, Poland, Springer LNCS 3618 (2005), 26–38 (without the appendix)

    A sequential algorithm just follows its instructions and thus cannot make a nondeterministic choice all by itself, but it can be instructed to solicit outside help to make a choice. Similarly, an object-oriented program cannot create a new object all by itself; a create-a-new-object command solicits outside help. These are but two examples of intra-step interaction of an algorithm with its environment. Here we motivate and survey recent work on interactive algorithms within the Behavioral Computation Theory project.

175. Yuri Gurevich, Paul Schupp: Membership Problem for Modular Group. SIAM Journal on Computing 37:2 (2007), 425–459.

    The modular group plays an important role in many branches of mathematics. We show that the membership problem for the modular group is polynomial time in the worst case. We also show that the membership problem for a free group remains polynomial time when elements are written in a normal form with exponents.

176. Andreas Blass, Yuri Gurevich, Dean Rosenzweig, Benjamin Rossman: Interactive Small-Step Algorithms I: Axiomatization. Logical Methods in Computer Science 3:4 (2007), paper 3. A preliminary version appeared as MSR-TR-2006-170

    In earlier work, the Abstract State Machine Thesis – that arbitrary algorithms are behaviorally equivalent to abstract state machines – was established for several classes of algorithms, including ordinary, interactive, small-step algorithms. This was accomplished on the basis of axiomatizations of these classes of algorithms. Here we extend the axiomatization and, in a companion paper, the proof, to cover interactive small-step algorithms that are not necessarily ordinary. This

means that the algorithms (1) can complete a step without necessarily waiting for replies to all queries from that step and (2) can use not only the environment's replies but also the order in which the replies were received.

This is essentially part one of MSR-TR-2005-113. [182] is essentially the remainder of the technical report.

177. Yuri Gurevich, Tanya Yavorskaya: On Bounded Exploration and Bounded Nondeterminism. MSR-TR-2006-07

This report consists of two separate parts, essentially two oversized footnotes to [141]. In Chapter I, Yuri Gurevich and Tatiana Yavorskaya present and study a more abstract version of the bounded exploration postulate. In Chapter II, Tatiana Yavorskaya gives a complete form of the characterization, sketched in [141], of bounded-choice sequential algorithms.

178. Andreas Blass, Yuri Gurevich: Program Termination, and Well Partial Orderings. ACM ToCL 9:3 (July 2008)

The following known observation may be useful in establishing program termination: if a transitive relation R is covered by finitely many well-founded relations $U_1, \ldots, U_n$ then R is well-founded. A question arises how to bound the ordinal height $|R|$ of the relation R in terms of the ordinals $\alpha_i = |U_i|$. We introduce the notion of the stature $\|P\|$ of a well partial ordering P and show that $|R|$ less than or equal to the stature of the direct product $\alpha_1 \times \cdots \times \alpha_n$ and that this bound is tight. The notion of stature is of considerable independent interest. We define $\|P\|$ as the ordinal height of the forest of nonempty bad sequences of P, but it has many other natural and equivalent definitions. In particular, $\|P\|$ is the supremum, and in fact the maximum, of the lengths of linearizations of P. And the stature of the direct product $\alpha_1 \times \cdots \times \alpha_n$ is equal to the natural product of these ordinals.

179. Yuri Gurevich, Margus Veanes, Charles Wallace: Can Abstract State Machines Be Useful in Language Theory? Theoretical Computer Science 376 (2007) 17–29. Extended Abstract in DLT 2006, Developments in Language Theory, Springer LNCS 4036 (2006), 14–19

The abstract state machine (ASM) is a modern computation model. ASMs and ASM based tools are used in academia and industry, albeit in a modest scale. They allow one to give high-level operational semantics to computer artifacts and to write executable specifications of software and hardware at the desired abstraction level. In connection with the 2006 conference on Developments in Language Theory, we point out several ways that we believe abstract state machines can be useful to the DLT community.

180. Andreas Blass, Yuri Gurevich: A Note on Nested Words. MSR-TR-2006-139

For every regular language of nested words, the underlying strings form a context-free language, and every context-free language can be obtained in this way. Nested words and nested-word automata are generalized to motley words and motley-word automata. Every motley-word automation is equivalent to a deterministic one. For every regular language of motley words, the underlying strings form a finite intersection of context-free languages, and every finite intersection of context-free languages can be obtained in this way.

181. Yuri Gurevich: ASMs in the Classroom: Personal Experience. In: D. Bjørner and M. C. Henson (eds.) Logics of Specification Languages, Springer (2008), 599–602

We share our experience of using abstract state machines for teaching computation theory at the University of Michigan.

182. Andreas Blass, Yuri Gurevich, Dean Rosenzweig, Benjamin Rossman: Interactive Small-Step Algorithms II: Abstract State Machines and the Characterization Theorem. Logical Methods in Computer Science 3:4 (2007), paper 4. A preliminary version appeared as MSR-TR-2006-171

    In earlier work, the Abstract State Machine Thesis – that arbitrary algorithms are behaviorally equivalent to abstract state machines – was established for several classes of algorithms, including ordinary, interactive, small-step algorithms. This was accomplished on the basis of axiomatizations of these classes of algorithms. In a companion paper [176] the axiomatization was extended to cover interactive small-step algorithms that are not necessarily ordinary. This means that the algorithms (1) can complete a step without necessarily waiting for replies to all queries from that step and (2) can use not only the environment's replies but also the order in which the replies were received. In order to prove the thesis for algorithms of this generality, we extend here the definition of abstract state machines to incorporate explicit attention to the relative timing of replies and to the possible absence of replies. We prove the characterization theorem for extended ASMs with respect to general algorithms as axiomatized in [176].

183. Dan Teodosiu, Nikolaj Bjørner, Yuri Gurevich, Mark Manasse, Joe Porkka: Optimizing File Replication over Limited-Bandwidth Networks using Remote Differential Compression. MSR-TR-2006-157

    Remote Differential Compression (RDC) protocols can efficiently update files over a limited-bandwidth network when two sites have roughly similar files; no site needs to know the content of another's files a priori. We present a heuristic approach to identify and transfer the file differences that is based on finding similar files, subdividing the files into chunks, and comparing chunk signatures. Our work significantly improves upon previous protocols such as LBFS and RSYNC in three ways. Firstly, we present a novel algorithm to efficiently find the client files that are the most similar to a given server file. Our algorithm requires 96 bits of metadata per file, independent of file size, and thus allows us to keep the metadata in memory and eliminate the need for expensive disk seeks. Secondly, we show that RDC can be applied recursively to signatures to reduce the transfer cost for large files. Thirdly, we describe new ways to subdivide files into chunks that identify file differences more accurately. We have implemented our approach in DFSR, a state-based multimaster file replication service shipping as part of Windows Server 2003 R2. Our experimental results show that similarity detection produces results comparable to LBFS while incurring a much smaller overhead for maintaining the metadata. Recursive signature transfer further increases replication efficiency by up to several orders of magnitude.

184. Martin Grohe, Yuri Gurevich, Dirk Leinders, Nicole Schweikardt, Jerzy Tyszkiewicz, Jan Van den Bussche: Database Query Processing Using Finite Cursor Machines. Theory of Computing Systems 44:4 (April 2009), 533–560. An earlier version appeared in: ICDT 2007, International Conference on Database Theory, Springer LNCS 4353 (2007), 284–298

    We introduce a new abstract model of database query processing, finite cursor machines, that incorporates certain data streaming aspects. The model describes quite faithfully what happens in so-called "one-pass" and "two-pass query processing". Technically, the model is described in the framework of abstract state

machines. Our main results are upper and lower bounds for processing relational algebra queries in this model, specifically, queries of the semijoin fragment of the relational algebra.

185. Andreas Blass, Yuri Gurevich: Zero-One Laws: Thesauri and Parametric Conditions. BEATCS 91 (February 2007), 125–144. Reprinted in: A. Gupta et al. (eds.) Logic at the Crossroads: An Interdisciplinary View, Allied Publishers Pvt. Ltd., New Delhi (2007), 187–206

    The zero-one law for first-order properties of finite structures and its proof via extension axioms were first obtained in the context of arbitrary finite structures for a fixed finite vocabulary. But it was soon observed that the result and the proof continue to work for structures subject to certain restrictions. Examples include undirected graphs, tournaments, and pure simplicial complexes. We discuss two ways of formalizing these extensions, Oberschelp's parametric conditions (Springer Lecture Notes in Mathematics 969, 1982) and our thesauri of [149]. We show that, if we restrict thesauri by requiring their probability distributions to be uniform, then they and parametric conditions are equivalent. Nevertheless, some situations admit more natural descriptions in terms of thesauri, and the thesaurus point of view suggests some possible extensions of the theory.

186. Andreas Blass, Yuri Gurevich: Background of Computation. BEATCS, 92 (June 2007)

    In a computational process, certain entities (for example, sets or arrays) and operations on them may be automatically available, for example by being provided by the programming language. We define background classes to formalize this idea, and we study some of their basic properties. The present notion of background class is more general than the one we introduced in an earlier paper [143], and it thereby corrects one of the examples in that paper. The greater generality requires a non-trivial notion of equivalence of background classes, which we explain and use. Roughly speaking, a background class assigns to each set (of atoms) a structure (for example, of sets or arrays or combinations of these and similar entities), and it assigns to each embedding of one set of atoms into another a standard embedding between the associated background structures. We discuss several, frequently useful, properties that background classes may have, for example that each element of a background structure depends (in some sense) on only finitely many atoms, or that there are explicit operations by which all elements of background structures can be produced from atoms.

187. Robert H. Gilman, Yuri Gurevich, Alexei Miasnikov: A Geometric Zero-One Law. JSL 74:3 (September 2009)

    Each relational structure X has an associated Gaifman graph, which endows X with the properties of a graph. If x is an element of X, let $B_n(x)$ be the ball of radius n around x. Suppose that X is infinite, connected and of bounded degree. A first-order sentence s in the language of X is almost surely true (resp. a.s. false) for finite substructures of X if for every x in X, the fraction of substructures of $B_n(x)$ satisfying s approaches 1 (resp. 0) as n approaches infinity. Suppose further that, for every finite substructure, X has a disjoint isomorphic substructure. Then every s is a.s. true or a.s. false for finite substructures of X. This is one form of the geometric zero-one law. We formulate it also in a form that does not mention the ambient infinite structure. In addition, we investigate various questions related to the geometric zero-one law.

188. Nachum Dershowitz, Yuri Gurevich: A natural axiomatization of computability and proof of Church's Thesis. Bulletin of Symbolic Logic 14:3 (September 2008), 299–350. An earlier version was published as MSR-TR-2007-85

Church's Thesis asserts that the only numeric functions that can be calculated by effective means are the recursive ones, which are the same, extensionally, as the Turing-computable numeric functions. The Abstract State Machine Theorem states that every classical algorithm is behaviorally equivalent to an abstract state machine. This theorem presupposes three natural postulates about algorithmic computation. Here, we show that augmenting those postulates with an additional requirement regarding basic operations gives a natural axiomatization of computability and a proof of Church's Thesis, as Gödel and others suggested may be possible. In a similar way, but with a different set of basic operations, one can prove Turing's Thesis, characterizing the effective string functions, and – in particular – the effectively-computable functions on string representations of numbers.

188a. Yuri Gurevich: Proving Church's Thesis. CSR 2007, Computer Science – Theory and Applications, 2nd International Symposium on Computer Science in Russia, Springer LNCS 4649 (2007), 1–3

This is an extended abstract of the opening talk of CSR 2007. It is based on [188].

189. Yuri Gurevich, Dirk Leinders, Jan Van den Bussche: A Theory of Stream Queries. DBPL 2007, 11th International Symposium on Database Programming Languages, Springer LNCS 4797 (2007), 153–168

Data streams are modeled as infinite or finite sequences of data elements coming from an arbitrary but fixed universe. The universe can have various built-in functions and predicates. Stream queries are modeled as functions from streams to streams. Both timed and untimed settings are considered. Issues investigated include abstract definitions of computability of stream queries; the connection between abstract computability, continuity, monotonicity, and non-blocking operators; and bounded memory computability of stream queries using abstract state machines (ASMs).

190. Nikolaj Bjørner, Andreas Blass, Yuri Gurevich: Content-Dependent Chunking for Differential Compression, the Local Maximum Approach. Journal of Computer and System Sciences 76:3–4 (May–June 2010), 154–203. Originally published as MSR-TR-2007-109

When a file is to be transmitted from a sender to a recipient and when the latter already has a file somewhat similar to it, remote differential compression seeks to determine the similarities interactively so as to transmit only the part of the new file not already in the recipient's old file. Content-dependent chunking means that the sender and recipient chop their files into chunks, with the cutpoints determined by some internal features of the files, so that when segments of the two files agree (possibly in different locations within the files), the cutpoints in such segments tend to be in corresponding locations, and so the chunks agree. By exchanging hash values of the chunks, the sender and recipient can determine which chunks of the new file are absent from the old one and thus need to be transmitted.

We propose two new algorithms for content-dependent chunking, and we compare their behavior, on random files, with each other and with previously used

algorithms. One of our algorithms, the local maximum chunking method, has been implemented and found to work better in practice than previously used algorithms.

Theoretical comparisons between the various algorithms can be based on several criteria, most of which seek to formalize the idea that chunks should be neither too small (so that hashing and sending hash values become inefficient) nor too large (so that agreements of entire chunks become unlikely). We propose a new criterion, called the slack of a chunking method, which seeks to measure how much of an interval of agreement between two files is wasted because it lies in chunks that don't agree.

Finally, we show how to efficiently find the cutpoints for local maximum chunking.

191. Yuri Gurevich, Itay Neeman: DKAL: Distributed-Knowledge Authorization Language. MSR-TR-2008-09. First appeared as MSR-TR-2007-116

DKAL is an expressive declarative authorization language based on existential fixed-point logic. It is considerably more expressive than existing languages in the literature, and yet feasible. Our query algorithm is within the same bounds of computational complexity as, e.g., that of SecPAL. DKAL's distinguishing features include

- explicit handling of knowledge and information,
- targeted communication that is beneficial with respect to confidentiality, security, and liability protection,
- the flexible use and nesting of functions, which in particular allows principals to quote (to other principals) whatever has been said to them,
- flexible built-in rules for expressing and delegating trust,
- information order that contributes to succinctness.

191a. Yuri Gurevich, Itay Neeman: DKAL: Distributed-Knowledge Authorization Language. CSF 2008, 21st IEEE Computer Security Foundations Symposium, 149–162

This is an extended abstract of [191]. DKAL is a new declarative authorization language for distributed systems. It is based on existential fixed-point logic and is considerably more expressive than existing authorization languages in the literature. Yet its query algorithm is within the same bounds of computational complexity as, e.g., that of SecPAL. DKAL's communication is targeted, which is beneficial for security and for liability protection. DKAL enables flexible use of functions; in particular, principals can quote (to other principals) whatever has been said to them. DKAL strengthens the trust delegation mechanism of SecPAL. A novel information order contributes to succinctness. DKAL introduces a semantic safety condition that guarantees the termination of the query algorithm.

192. Andreas Blass, Nachum Dershowitz, Yuri Gurevich: When Are Two Algorithms the Same? Bulletin of Symbolic Logic 15:2 (2009), 145–168. An earlier version was published as MSR-TR-2008-20

People usually regard algorithms as more abstract than the programs that implement them. The natural way to formalize this idea is that algorithms are equivalence classes of programs with respect to a suitable equivalence relation. We argue that no such equivalence relation exists.

193. Andreas Blass, Yuri Gurevich: Two Forms of One Useful Logic:. Existential Fixed Point Logic and Liberal Datalog, BEATCS 95 (June 2008), 164–182

A natural liberalization of Datalog is used in the Distributed Knowledge Authorization Language (DKAL). We show that the expressive power of this liberal Datalog is that of existential fixed-point logic. The exposition is self-contained.

194. Andreas Blass, Yuri Gurevich: One Useful Logic That Defines Its Own Truth. MFCS 2008, 33rd International Symposium on Mathematical Foundations of Computer Science, Springer LNCS 5162 (2008), 1–15

Existential fixed point logic (EFPL) is a natural fit for some applications, and the purpose of this talk is to attract attention to EFPL. The logic is also interesting in its own right as it has attractive properties. One of those properties is rather unusual: truth of formulas can be defined (given appropriate syntactic apparatus) in the logic. We mentioned that property elsewhere, and we use this opportunity to provide the proof.

195. Nikolaj Bjørner, Andreas Blass, Yuri Gurevich, Madan Musuvathi: Modular difference logic is hard. MSR-TR-2008-140

In connection with machine arithmetic, we are interested in systems of constraints of the form $x + k \leq y + l$. Over integers, the satisfiability problem for such systems is polynomial time. The problem becomes NP complete if we restrict attention to the residues for a fixed modulus N.

196. Andreas Blass, Yuri Gurevich: Persistent Queries in the Behavioral Theory of Algorithms. ACM ToCL, to appear. An earlier version appeared as MSR-TR-2008-150

We propose a syntax and semantics for interactive abstract state machines to deal with the following situation. A query is issued during a certain step, but the step ends before any reply is received. Later, a reply arrives, and later yet the algorithm makes use of this reply. By a persistent query, we mean a query for which a late reply might be used. Syntactically, our proposal involves issuing, along with a persistent query, a location where a late reply is to be stored. Semantically, it involves only a minor modification of the existing theory of interactive small-step abstract state machines.

197. Yuri Gurevich, Arnab Roy: Operational Semantics for DKAL: Application and Analysis. TrustBus 2009, 6th International Conference on Trust, Privacy and Security in Digital Business, Springer LNCS 5695 (2009),149–158

DKAL is a new authorization language based on existential fixed-point logic and more expressive than existing authorization languages in the literature. We present some lessons learned during the first practical application of DKAL and some improvements that we made to DKAL as a result. We develop operational semantics for DKAL and present some complexity results related to the operational semantics.

198. Yuri Gurevich, Itay Neeman: Infon Logic: the Propositional Case. ACM ToCL, to appear. The ToCL version is a correction and slight extension of the version called The Infon Logic published in BEATCS 98 (June 2009), 150–178

Infons are statements viewed as containers of information (rather then representations of truth values). In the context of access control, the logic of infons is a conservative extension of logic known as constructive or intuitionistic. Distributed Knowledge Authorization Language uses additional unary connectives “p said”

and "p implied" where p ranges over principals. Here we investigate infon logic and a narrow but useful primal fragment of it. In both cases, we develop the model theory and analyze the derivability problem: Does the given query follow from the given hypotheses? Our more involved technical results are on primal infon logic. We construct an algorithm for the multiple derivability problem: Which of the given queries follow from the given hypotheses? Given a bound on the quotation depth of the hypotheses, the algorithm works in linear time. We quickly discuss the significance of this result for access control.

199. Nikolaj Bjørner, Yuri Gurevich, Wolfram Schulte, and Margus Veanes: Symbolic Bounded Model Checking of Abstract State Machines. International Journal of Software and Informatics 3:2–3 (June/September 2009), 149–170

Abstract State Machines (ASMs) allow us to model system behaviors at any desired level of abstraction, including levels with rich data types, such as sets or sequences. The availability of high-level data types allows us to represent state elements abstractly and faithfully at the same time. AsmL is a rich ASM-based specification and programming language. In this paper we look at symbolic analysis of model programs written in AsmL with a background T of linear arithmetic, sets, tuples, and maps. We first provide a rigorous account of the update semantics of AsmL in terms of background T, and we formulate the problem of bounded path exploration of model programs, or the problem of Bounded Model Program Checking (BMPC), as a satisfiability modulo T problem. Then we investigate the boundaries of decidable and undecidable cases for BMPC. In a general setting, BMPC is shown to be highly undecidable (Σ^1_1-complete); restricted to finite sets, the problem remains RE-hard (Σ^0_1-hard). On the other hand, BMPC is shown to be decidable for a class of basic model programs that are common in practice. We apply Satisfiability Modulo Theories (SMT) tools to BMPC. The recent SMT advances allow us to directly analyze specifications using sets and maps with specialized decision procedures for expressive fragments of these theories. Our approach is extensible; background theories need in fact only be partially solved by the SMT solver; we use simulation of ASMs to support additional theories that are beyond the scope of available decision procedures.

200. Yuri Gurevich, Itay Neeman: DKAL 2 – A Simplified and Improved Authorization Language. MSR-TR-2009-11

Knowledge and information are central notions in DKAL, a logic based authorization language for decentralized systems, the most expressive among such languages in the literature. Pieces of information are called infons. Here we present DKAL 2, a surprisingly simpler version of the language that expresses new important scenarios (in addition to the old ones) and that is built around a natural logic of infons. Trust became definable, and its properties, postulated earlier as DKAL house rules, are now *proved*. In fact, none of the house rules postulated earlier is now needed. We identify also a most practical fragment of DKAL where the query derivation problem is solved in *linear time*.

201. Andreas Blass, Nachum Dershowitz, Yuri Gurevich: Exact Exploration and Hanging Algorithms. CSL 2010, 19th EACSL Annual Conference on Computer Science Logic (August 2010), to appear

Recent analysis of sequential algorithms resulted in their axiomatization and in a representation theorem stating that, for any sequential algorithm, there is an abstract state machine (ASM) with the same states, initial states and state

transitions. That analysis, however, abstracted from details of intra-step computation, and the ASM, produced in the proof of the representation theorem, may and often does explore parts of the state unexplored by the algorithm. We refine the analysis, the axiomatization and the representation theorem. Emulating a step of the given algorithm, the ASM, produced in the proof of the new representation theorem, explores exactly the part of the state explored by the algorithm. That frugality pays off when state exploration is costly. The algorithm may be a high-level specification, and a simple function call on the abstraction level of the algorithm may hide expensive interaction with the environment. Furthermore, the original analysis presumed that state functions are total. Now we allow state functions, including equality, to be partial so that a function call may cause the algorithm as well as the ASM to hang. Since the emulating ASM does not make any superfluous function calls, it hangs only if the algorithm does.

202. Andreas Blass, Yuri Gurevich, Efim Hudis: The Tower-of-Babel Problem, and Security Assessment Sharing. MSR-TR-2010-57. BEATCS 101 (June 2010), to appear

The tower-of-Babel problem is rather general: How to enable a collaboration among experts speaking different languages? A computer security version of the tower-of-Babel problem is rather important. A recent Microsoft solution for that security problem, called Security Assessment Sharing, is based on this idea: A tiny common language goes a long way. We construct simple mathematical models showing that the idea is sound.