From Logic to Theoretical Computer Science – An Update[⋆]

Boris A. Trakhtenbrot

School of Computer Science, Tel Aviv University, Ramat Aviv 69978, Israel

1 Foreword

In October 1997, whilst touching up this text, exactly 50 years had past since I was accepted for graduate studies under P.S. Novikov. I started then to study and do research in logic and computability, which developed, as time will show, into research in Theoretical Computer Science (TCS).

After my emigration (in December 1980) from the Soviet Union (SU), I was encouraged by colleagues to experience the genre of memoirs. That is how [67,68] appeared, and more recently [72], conceived as contributions to the history of TCS in the SU. The present paper is intended as a more intimate perspective on my research and teaching experience. It is mainly an account of how my interests shifted from classical logic and computability to TCS, notably to Automata and Computational Complexity. Part of these reminiscences, recounting especially the scientific, ideological and human environment of those years (roughly 1945–67), were presented earlier at a Symposium (June 1991) on the occasion of my retirement. Occasionally, I will quote from [67,68,72], or will refer to them.

Before starting the main narrative I would like to recall some important circumstances which characterized those years.

First of all, the postwar period was a time of ground-breaking scientific developments in Computability, Information Theory, and Computers. That is widely known and needs no comment. The subjects were young and so were their founders. It is amazing that at that time the giants, Church, Kleene, Turing, and von Neumann, were only in their thirties and forties!

Now, about the specific background in the Soviet Union.

The genealogical tree of TCS in the SU contains three major branches, rooted in A.N. Kolmogorov (1903–87), A.A. Markov (1903–79) and P.S. Novikov (1901–75). In those troublesome times, these famous mathematicians also had the reputation of men with high moral and democratic principles. Their scientific interests, authority and philosophies influenced the development of mathematical logic, computability for several generations, and subsequently TCS in the SU.

Whereas Markov and Kolmogorov contributed directly to TCS, Novikov's involvement occurred through his strong influence on his disciples and collaborators. The most prominent of them – A.A. Lyapunov (1911–1973) – became a

[⋆] This chapter is an expanded and updated version of "From logic to theoretical computer science", which appeared in *People and Ideas in Theoretical Computer Science*, Cristian S. Calude, ed., Springer-Verlag, Singapore, pp. 314–342, 1998.

A. Avron et al. (Eds.): Trakhtenbrot/Festschrift, LNCS 4800, pp. 1–38, 2008.

Fig. 1. Three generations of scientists (from left to right): A. Slissenko, A.A. Markov, Jr., B.A. Kushner, B.A. Trakhtenbrot (from the archive of the Markov family)

widely recognized leader of "Theoretical Cybernetics" – the rubric which covered at that time most of what is considered today to belong to TCS.

As a matter of fact, for many offspring of those three branches, including myself, the perception of TCS was as of some kind of applied logic, whose conceptual sources belong to the theoretical core of mathematical logic. The affiliation with Logic was evident at the All-Union Mathematical Congress (Moscow, 1956), where Theoretical Cybernetics was included in the section on mathematical logic. Other examples: books on Automata [17,13] appeared in the series *Mathematical Logic and Foundations of Mathematics*; also, my first papers on Computational Complexity were published in Anatoly I. Maltsev's (1909-67) journal *Algebra and Logic.*

The early steps in TCS coincided with attacks of the official establishment on various scientific trends and their developers. In particular, Cybernetics was labeled a "pseudo-science", and Mathematical Logic – a "bourgeois idealistic distortion". That was the last stage of the Stalin era with persecution and victimization of "idealists", "cosmopolitans", etc. The survival and the long overdue recognition of Mathematical Logic and Cybernetics is in many respects indebted to Lyapunov, Markov, Novikov, Kolmogorov and S.A. Yanovskaya (1896–1966). But even after that, academic controversies often prompted such bureaucratic repression as the prevention of publications and the denial of degrees. Difficulties with publications also happened because of the exactingness and self-criticism of the authors and/or their mentors, or because the community was far from prepared to appreciate them. I told about that in [67] and [68].

Above, the emphasis was on the Soviet side; now, some remarks on the international context in which research in TCS was conducted in the SU.

The chronology of events reveals that quite a number of ideas and results in TCS appeared in the SU parallel to, independent of, and sometimes prior to, similar developments in the West. This parallelism is easy to explain by the fact that these were natural ideas occurring at the right time. In particular, that is how comprehensive theories of Automata and of Computational Complexity emerged in the 50s and 60s; I will elaborate on this subject in the next sections. But for a variety of reasons, even in those cases where identical or similar results were obtained independently, the initial motivation, the assessment of the results and their impact on the development and developers of TCS did not necessarily coincide. In particular, in the SU specific interest in complexity theory was aroused by discussions on the essence of brute force algorithms (*perebor* – in Russian). However, despite this difference in emphasis from the motivating concerns of the American researchers, after a few years these approaches virtually converged.

In the past, the priority of Russian and Soviet science was constantly propounded in Soviet official circles and media. This unrestrained boasting was cause for ironic comments in the West and for self-irony at home. But, as a matter of fact, the West was often unaware of developments in the SU, and some of them went almost entirely unnoticed. To some extent this was a consequence of the isolation imposed by language barriers and sociopolitical forces. In particular, travels abroad were a rare privilege, especially to the "capitalist" countries. My first trip abroad, for example, took place in 1967, but visits to the West became possible only in 1981 after my emigration to Israel.

Against this unfavorable background it is worth mentioning also the encouraging events and phenomena, which eased the isolation.

The International Mathematical Congress in Moscow (1966) was attended by the founders of our subject, namely, Church, Kleene, Curry, Tarski and other celebrities. It was an unforgettable and moving experience to have first-hand contact with these legendary characters. Later, Andrey P. Ershov (1931–1988) managed to organize a series of International Symposia on "Theoretical Programming", attended also by people from the West, and among them F. Bauer, E. Dijkstra, E. Engeler, C.A.R. Hoare, D. Knuth, J. McCarthy, R. Milner, M. Nivat, D. Park, M. Paterson, J. Schwartz, and A. van Wijngaarden. For many years, A. Meyer used to regularly send me proceedings of the main TCS symposia, a way to somehow compensate for the meetings my colleagues and myself were prevented from attending. This was part of our unusual and long-term contact by correspondence, which – after my emigration – switched to direct collaboration. All this reinforced our sense of belonging to the international TCS community.

2 Early Days

2.1 Brichevo

I was born in Brichevo, a village in Northern Bessarabia (now Moldova). Though my birth place has nothing to do with my career or with other events I am going to write about, let me begin with the following quotation:

> Brichevka, a Jewish agricultural settlement, founded in 1836. According to the general (1897) census of the population – 1644 inhabitants, 140 houses. . . (from vol. 5 of *The Jewish Encyclopedia*, St. Petersburg, 1912; translated from Russian).

Among the first settlers were Eli and Sarah Helman, the grandparents of my maternal grandfather. World War II brought about the collapse of Brichevo (or Brichevka). The great majority of the population did not manage to flee and were deported to the notorious Transnistria camps; only a small number survived and they dispersed over countries and continents. For years I used "Brichevo" as a reliable password: easy for me to remember, apparently impossible for outsiders to guess, and still a way to retain the memory of a vanished community.

After completing elementary school in Brichevo I attended high school in the neighboring towns of Belts and Soroka, where I was fortunate to have very good teachers of mathematics. My success in learning, and especially in mathematics, was echoed by the benevolence of the teachers and the indulgence of my fellow pupils. The latter was even more important to me, since it to some degree compensated for the discomfort and awkwardness caused by my poor vision.

2.2 Kishinev

In 1940 I enrolled in the Faculty of Physics and Mathematics of the newly-established Moldavian Pedagogical Institute in Kishinev. The curriculum covered a standard spectrum of teachers' training topics. In particular, mathematical courses presented basics in Calculus, Linear Algebra and Algebra of Polynomials, Analytical Geometry, Projective Geometry, Foundations of Geometry (including Lobachevski Geometry), Elements of Set Theory and Number Theory.

On June 22, 1941, Kishinev (in particular the close neighbourhood of our campus) was bombed by German air forces. In early July, I managed to escape from the burning city. Because of vision problems, I was released from military service and, after many mishaps, arrived as a refugee in Chkalov (now Orenburg) on the Ural River. Here, I enrolled in the local pedagogic institute. A year later we moved to Buguruslan in the Chkalov region, to where the Kishinev Institute was evacuated to in order to train personnel for the forthcoming return home as soon as our region would be liberated. Almost all the lecturers were former high school teachers – skilled people whose interests lay in the pedagogic aspects of mathematics and physics. (There were no recipients of academic degrees among them, but one of the instructors in the Chkalov institute bore the impressive name Platon Filosofov). Nikolai S. Titov, a former Ph.D. student of the Moscow University, who happened to flee to Buguruslan, lectured on Set Theory. I was deeply impressed by the beauty and novelty of this theory. Unfortunately, this was only a transient episode in those hard and anxious days. Actually, during the war years 1941–1944, my studies were irregular, being combined with employment in a felt boot factory, a storehouse and, finally, in the Kuybyshev-Buguruslan Gas Trust.

In August 1944 the institute was evacuated to Kishinev and I returned to my native regions for a position in the Belts college to train elementary school teachers. Only a year later did I take my final examinations and qualify as a high school mathematics teacher. That was my mathematical and professional background in September 1945 when (already at the age of $24\frac{1}{2}$) I decided to take a chance and seriously study mathematics.

2.3 Chernovtsy

I enrolled at the University of Chernovtsy (Ukraine) to achieve the equivalent of a master's degree in mathematics. In that first postwar year the university was involved in the difficult process of restoration. Since my prior education covered only some vague mathematical-pedagogical curriculum with examinations partially passed without having attended lectures, I did not know much to start off with. But there were only a few students and the enrollment policy of the administration was quite liberal. There were also only a few academic staff in our Faculty of Physics and Mathematics and soon I became associated with Alexander A. Bobrov, a prominent character on the general background. A.A. (b. 1912), who completed his Ph.D. thesis in 1938 under Kolmogorov, gave an original course in Probabilities. The distinguishing quality was not so much in the content of the course as in his style (completely new to me) of teaching and of involving the audience. A.A. did not seem to be strongly committed to his previously prepared lectures; during class he would try to examine new ideas and to improvise alternative proofs. As such trials did not always succeed, he would not hesitate to there and then loudly criticize himself and appeal to the audience for collaboration. This challenging style was even more striking in a seminar he held on Hausdorff's famous book on Set Theory, with the participation of both students and academic staff. Due to the "Bobrovian" atmosphere dominating the seminar, I started to relish the idea of research in this fascinating area. A.A. also helped me secure a job in the newly founded departmental scientific library. My primary task was to take stock of the heaps of books and journals extracted earlier from basements and temporary shelters, and to organize them into some bibliographical service. I remember reverently holding volumes of the *Journal fuer reine und angewandte Mathematik* with authentic papers and pictures of Weierstrass and other celebrities. As I later understood, the mathematical library was exclusively complete, and, as a matter of fact, disposed of all the important journals before WWII. As there was only a handful of graduate students it soon turned out that my library was not in much demand – in truth, for days there were no visitors at all; so most of the time I shared the roles of supplier and user of the library services. Through self study I mastered a significant amount of literature and reached some scientific maturity. I soon identified *Fundamenta Mathematicae* to be the journal closest to my interests in Descriptive Set Theory. All the volumes, starting with the first issue dated 1921, were on my table and I would greedily peruse them.

After considering some esoteric species of ordered sets, I turned to the study of delta-sigma operations, a topic promoted by Andrei N. Kolmogorov and also tackled in *Fundamenta.* At this stage, Bobrov decided that it was the right time to bring me together with the appropriate experts and why not with Kolmogorov himself! In the winter of 1946 Kolmogorov was expected to visit Boris V. Gnedenko (1912–1995) at the Lvov university. So far so good, except that at the last moment Kolmogorov canceled his visit. Gnedenko did his best to compensate for that annoying failure. He showed me exclusive consideration, invited me to lunch at his home and attentively inquired about all my circumstances. It was the first time that I had talked to a full professor and I felt somehow shy in his presence and in the splendour of his dwelling. B.V. listened to me patiently and, I guess, was impressed not as much by my achievements (which were quite modest, and after all, beyond the field of his main interests) but by my enthusiastic affection for Descriptive Set Theory. Anyway, he explained to me that for the time-being Kolmogorov had other research preferences and it would be very useful to contact Piotr S. Novikov and Alexei A. Lyapunov who, unlike Kolmogorov and other descendants of the famous Lusin set-theoretical school, were mostly still active in the field.

During this period I met Berta I. Rabinovich, who was to become my wife.

In the summer of 1946, I visited Moscow for the first time. Because it was vacation time and since no prior appointments had been set up, it was very difficult to get hold of people. Nevertheless, I managed to see Kolmogorov for a short while at the university and to give him my notes on delta-sigma operations. He was in a great hurry, so we agreed to meet again in a couple of weeks on my way back home; unfortunately this did not work out. Novikov was also unreachable, being somewhere in the countryside. I was more fortunate with A.A. Lyapunov, in whose house I spent a wonderful evening of scientific discussions alternated with tea-drinking with the whole family. A.A. easily came to know my case and presented me with a deeper picture of the Moscow set-theoretical community with a stress on the current research done by Novikov and by himself. He offered to inform Novikov in detail about my case and suggested that I visit Moscow at a more appropriate time for further discussions.

My second trip to Moscow was scheduled for May 1947 on the very eve of my graduation from the Chernovtsy University, when, beyond pure mathematics, the question of my forthcoming (if any) Ph.D. studies was on the agenda. All in all I had to stay in Moscow for at least a couple of weeks and that required appropriate logistics – a very nontrivial task at that time, in particular, because of the food rationing system and the troublesome train connections. Alas, at the first connection of the Lvov railroad station, local pickpockets managed to cut out the pocket with all my money. Despite this most regrettable incident, the trip ultimately turned out to be quite successful. The meetings with Novikov were very instructive and warm. And again, as in the case of the Lyapunovs, the atmosphere in the Novikov family was friendly and hospitable. Occasionally, Novikov's wife, Ludmila V. Keldysh (1904–1976), a prominent researcher in set theory in her own right, as well as A.A. Lyapunov, would also participate in the conversations. Counterbalancing my interests and efforts towards Descriptive Set Theory, Novikov called my attention to new developments I was not aware of in provincial Chernovtsy. He pointed to the path leading from a handful of hard set theoretical problems to modern concepts of mathematical logic and computability theory. He also offered his support and guidance should I agree to follow this path. I accepted Novikov's generous proposal, although with a sense of regret about my past dreams about Descriptive Set Theory.

Novikov held a permanent position at the Steklov Mathematical Institute of the USSR Academy. At that time, departments of mathematical logic did not yet exist in the USSR, but Novikov, together with Sofia A. Yanovskaya, had just started a research seminar "Mathematical Logic and Philosophical problems of Mathematics" in Moscow University, unofficially called The Bolshoy (great) Seminar. So, it was agreed that wherever other options might arise, Novikov would undertake my supervision and would do his best to overcome bureaucratic barriers.

3 Ph.D. Studies

In October 1947, I began my Ph.D. studies at the Kiev Mathematical Institute of the Ukrainian Academy of Sciences. The director of the institute, Mikhail A. Lavrentiev (1900–81), approved my petition to specialize in mathematical logic under P.S. Novikov and agreed to grant me long-term scientific visits to Moscow where I would stay with my advisor.

In Moscow, the Bolshoy seminar was then the main medium in which research and concomitant activities in that area were conducted. In particular, it was the forum where mathematical logicians from the first post-war generation (mostly students of P.S. Novikov, S.A. Yanovskaya and A.N. Kolmogorov) joined the community, reported on their ongoing research, and gained primary approval of their theses; and that is also what happened to me.

The atmosphere dominating the meetings of the seminar was democratic and informal. Everybody, including the students, felt and behaved at ease without strong regulations and formal respect for rank. I was happy to acquire these habits and later to promote them at my own seminars.

Actually the seminar was the successor of the first seminar in the USSR for mathematical logic, which was founded by Ivan I. Zhegalkin (1869–1947). After Zhegalkin's death it became affiliated with the Department of History of Mathematical Sciences of the Moscow University, whose founder and head was Yanovskaya. Its exceptional role in the development of mathematical logic in the USSR is a topic of its own and I will touch on it only very briefly.

The seminar usually engaged in a very broad spectrum of subjects from mathematical logic and its applications as well as from foundations and philosophy of mathematics. Here are some of the topics pursued by the senior participants:

Novikov – consistency of set-theoretical principles; Yanovskaya – philosophy of mathematics and Marx's manuscripts; Dmitri A. Bochvar (a prominent chemist in his main research area) – logic and set-theoretical paradoxes; Victor I. Shestakov (professor of physics) – application of logic to the synthesis and analysis of circuits.

Among the junior participants of the seminar I kept in close contact with the three Alexanders:

Alexander A. Zykov (1922–), also a Ph.D. student of Novikov, was at that time investigating the spectra of first order formulas. A.A. called my attention to Zhegalkin's decidability problem, which became the main topic of my Ph.D. thesis. He also initiated the correspondence, with me sending lengthy letters to Kiev with scientific Moscow news. This epistolary communication, followed later by correspondence with Kuznetsov and Sergey V. Yablonski (1925–1998). was a precious support in that remote time.

Alexander V. Kuznetsov (1927–87) was the secretary of the Bolshoy Seminar and conducted regular and accurate records of all meetings, discussions and problems. For years he was an invaluable source of information. For health reasons, A.V. did not even complete high school studies. As an autodidact in extremely difficult conditions, he became one of the most prominent soviet logicians. I had the good fortune to stay and to collaborate with him.

Alexander S. Esenin-Volpin (1924–) (the son of the famous Russian poet Sergey A. Esenin) was a Ph.D. student in topology under Pavel A. Alexandrov, but he early on became involved in logic and foundations of mathematics. A.S. became most widely known as an active fighter for human rights, and already in the late forties the KGB was keeping an eye on him. In the summer of 1949, we met in Chernovtsy, where he had secured a position after defending his thesis. Shortly thereafter he disappeared from Chernovtsy and we later learned that he had been deported to Karaganda (Kazakhstan). A couple of years later, I received a letter from him through his mother. I anxiously opened the letter, fearful of what I was about to learn. The very beginning of the letter was characteristic of Esenin-Volpin's eccentric character – "Dear Boris, let f be a function...".

During the years of my Ph.D. studies (1947–50), I actively (though not regularly) participated in the seminar meetings. Also the results which made up my thesis, "The decidability problem for finite classes and finiteness definitions in set theory" were discussed there. S. A. Yanovskaya offered the official support of the department in the future defense at the Kiev Institute of Mathematics; the other referees were A. N. Kolmogorov, A. A. Lyapunov and B. V. Gnedenko.

My thesis [43] included the finite version of Church's Theorem about the undecidability of first order logic: the problem of whether a first order formula is valid in all finite models is, like the general validity problem, undecidable, but in a technically different way. The novelty was in the formalization of the algorithm concept. Namely, I realized that, in addition to the process of formal inference, the effective process of (finite) model checking could also be used as a universal approach to the formalization of the algorithm concept. This observation anticipated my future concern with constructive processes on finite models.

Other results of the thesis which are seemingly less known, deal with the connection between deductive incompleteness and recursive inseparability.

In 1949 I proved the existence of pairs of recursively enumerable sets which are not separable by recursive sets. I subsequently learned that P.S. Novikov had already proved this, but, as usual, had not taken the trouble to publish what he considered to be quite a simple fact. (Note, that in 1951, Kleene who independently discovered this fact, published it as "a symmetric form of Godel's theorem".) In the thesis I showed that the recursive inseparability phenomenon implies that the means of any reasonably defined set theory are not enough to answer the question of whether two different finiteness definitions are equivalent. This incompleteness result was also announced in my short note [42] presented by A. N. Kolmogorov to the *Doklady*, but after Novikov's cool reaction to "inseparability", I refrained from explicitly mentioning that I had used these very techniques. Clearly, A. N. had forgotten that these techniques were in fact developed in the full text of my thesis and he later proposed the problem to his student Vladimir A. Uspenski (1930–) . Here is a quotation from *History of Mathematics* [40, p. 446]: "A. N. Kolmogorov pointed to the possible connection between the deductive incompleteness of some formal systems and the concept of recursive inseparability (investigated also by Trakhtenbrot). V. A. Uspenski established (1953) results, which confirm this idea. . . ."

Those early years were a period of fierce struggle for the legitimacy and survival of mathematical logic in the USSR. Therefore the broad scope of the agendas on the Bolshoy seminar was beneficial not only for the scientific contacts between representatives of different trends, but also, in the face of ideological attacks, to consolidate an effective defense line and to avoid isolation and discredit of mathematical logic. For us, the junior participants of the seminar, it was also a time when we watched the tactics our mentors adopted to face or to prevent ideological attacks. Their polemics were not free of abundant quotations from official sources, controlled self-criticism and violent attacks on real and imaginary rivals.

It was disturbing then (and even more painful now) to read S.A. Yanovskaya's notorious prefaces to the 1947–1948 translations of Hilbert and Ackermann's *Principles of Mathematical Logic* and Tarski's *Introduction to Logic and the Methodology of Deductive Sciences* in which Russell was blamed as a warmonger and Tarski, as a militant bourgeois. Alas, such were the rules of the game and S.A. was not alone in that game. I remember the hostile criticism of Tarski's book by A.N. Kolmogorov (apparently at a meeting of the Moscow Mathematical Society): "Translating Tarski was a mistake, but translating Hilbert was the correct decision," he concluded. This was an attempt to grant some satisfaction to the attacking philosophers in order to at least save the translation of Hilbert-Ackermann's book. I should also mention that S.A. was vulnerable – she was Jewish – a fact of which I was unaware for a long time. I learned about it in the summer of 1949 during Novikov's visit to Kiev. He told me then with indignation about official pressure on him "to dissociate from S.A. and other cosmopolitans".

However difficult the situation was, we – the students of that time – were not directly involved in the battle which we considered to be only a confrontation of titans. As it turned out this impression was wrong.

4 Toward TCS

In December 1950 after the defense of my thesis, I moved to Penza, about 700 km. SE of Moscow, for a position at the Belinski Pedagogical Institute.

At the beginning it was difficult for me to appropriately pattern my behaviour to the provincial atmosphere so different from the informal, democratic surroundings of the places I came from. These circumstances unfavorably influenced my relationships with some of the staff and students (in particular because of the constant pressure and quest for high marks). Because of this, though I like teaching, at the beginning, I did not derive satisfaction from it.[1] The situation was aggravated after a talk on mathematical logic I delivered to my fellow mathematicians. The aim of the talk entitled "The method of symbolic calculi in mathematics", was to explain the need and the use of exact definitions for the intuitive concepts "algorithm" and "deductive system". I was then accused of being "an idealist of Carnap-species". In that era of Stalin paranoia such accusations were extremely dangerous. At diverse stages of the ensuing developments, P.S. Novikov and A.A. Lyapunov (Steklov Mathematical Institute) and to some degree A.N. Kolmogorov and Alexander G. Kurosh (1908–1971) (Moscow Mathematical Society) were all involved in my defense, and S.A. Yanovskaya put my case on the agenda of the Bolshoy seminar. This story is told in [72].

My health was undermined by permanent tension, fear and overwork (often more than 20 hours teaching weekly). It goes without saying that for about two years I was unable to dedicate enough time to research. It was in those circumstances that only the selfless care and support of my wife Berta saved me from collapse. I should also mention the beneficial and calming effect of the charming middle-Russian landscape which surrounded our dwelling. Cycling and skiing in the nearby forest compensated somewhat for our squalid housing. (Actually, until our move to Novosibirsk in 1961, we shared a communal flat, without water and heating facilities, with another family.)

But despite all those troubles I remember this period mainly for its happy ending. In the summer of 1992, forty years after this story took place, Berta and I revisited those regions. The visit to Penza was especially nostalgic. Most of the participants of those events had already passed away. Only the recollections and of course the beautiful landscape remained.

[1] Of course, I also had good students and one of them, Ilya Plamennov (1924–), was admitted through my recommendation to Ph.D. studies at the Moscow University. Later he became involved in classified research and was awarded the most prestigious Lenin Prize (1962).

Returning to the “Idealism” affair, the supportive messages I received from Moscow stressed the urgent need for a lucid exposition of the fundamentals of symbolic calculi and algorithms for a broad mathematical community. They insisted on the preparation of a survey paper on the topic, which “should be based on the positions of Marxism-Leninism and contain criticism of the foreign scientists-idealists”. There was also an appeal to me to undertake this work which would demonstrate my philosophical ideological loyalty. Nevertheless I did not feel competent to engage in work which covered both a mathematical subject and official philosophical demands. These demands were permanently growing and changing; they could bewilder people far more experienced than myself. So it seemed reasonable to postpone the project until more favorable circumstances would allow separation of logic from official philosophy. Indeed, such a change in attitude took place gradually, in particular due to the growing and exciting awareness of computers.

In 1956 the journal *Mathematics in School* published my tutorial paper “Algorithms and automated problem solving”. Its later revisions and extensions appeared as books which circulated widely in the USSR and abroad [50]. (Throughout the years I was flattered to learn from many people, including prominent logicians and computer scientists, that this tutorial monograph was their own first reading on the topic as students and it greatly impressed them.)

Meanwhile, I started a series of special courses and seminars over and above the official curriculum, for a group of strong students. These studies covered topics in logic, set theory and cybernetics, and were enthusiastically supported by the participants. Most of them were later employed in the Penza Computer Industry where Bashir I. Rameev, the designer of the “Ural” computers, was a prominent figure. Later, several moved with me to Novosibirsk. They all continued to attend the seminar after graduating from their studies. We would gather somewhere in the institute after a full day of work in Rameev’s laboratories (the opposite end of town), inspired and happy to find ourselves together. Here is a typical scene – a late winter’s evening, frosty and snowy, and we are closing our meeting. It is time to disperse into the lonely darkness, and Valentina Molchanova, a most devoted participant of our seminar, has still to cross the frozen river on her long walk home.

The publication of my tutorial on algorithms and the above-mentioned work with students increased my pedagogical visibility to such a degree that I was instructed by the Education Ministry, to compile the program of a course “Algorithms and Computers” for the pedagogical institutes. Moreover, the Ministry organized an all-Russian workshop in Penza, dedicated to this topic, with the participation of P.S. Novikov, A.I. Maltsev, and other important guests from Moscow.

In Penza there was a lack of scientific literature, not to mention normal contacts with well established scientific bodies. This obvious disadvantage was partially compensated by sporadic trips to Moscow for scientific contacts (and food supply), as well by correspondence with Kuznetsov, Sergey V. Yablonsky and Lyapunov.

I continued the work on recursive nonseparability and incompleteness of formal theories [44,48], started in the Ph.D. thesis. At the same time, I was attracted by Post's problem of whether all undecidable axiomatic systems are of the same degree of undecidability. This super-problem in Computability and Logic, with a specific flavour of descriptive set theory, was for a long time on the agenda of the Bolshoy Seminar. It inspired also my work on classification of recursive operators and reducibilities [45]. Later, A. V. Kuznetsov joined me and we extended the investigation to partial recursive operators in the Baire space [18]. These issues, reflected our growing interest in relativized algorithms (algorithms with oracles) and in set-descriptive aspects of computable operators, I worked then on a survey on this subject, but the (uncompleted) manuscript was never published. Nevertheless, the accumulated experience helped me later in the work on relativized computational complexity.

In 1956 Post's problem was solved independently by Albert A. Muchnik (1934–) – a young student of P. S. Novikov – and by the American Richard Friedberg. Their solutions were very similar and involved the invention of the priority method of computability theory. At that point it became clear to me that I had exhausted my efforts and ambitions in this area, and, that I am willing to switch to what nowadays would be classified as "Theoretical Computer Science". From the early 50's this research was enthusiastically promoted by A. A. Lyapunov and S. V. Yablonski under the general rubric "Theoretical Cybernetics"; it covered switching theory, minimization of boolean functions, coding, automata, program schemes, etc. Their seminars at the Moscow University attracted many students and scholars, and soon became important centers of research in these new and exciting topics. I was happy to join the cybernetics community through correspondence and trips to Moscow. The general atmosphere within this fresh and energetic community was very friendly, and I benefited much from it. Many "theoretical cybernetists" started with a background in Mathematical Logic, Computability and Descriptive Set Theory and were considerably influenced by these traditions. So, no wonder that, despite my new research interests in Switching and Automata Theory, I considered myself (as did many others) to be a logician. My formal "conversion" to Cybernetics happened on Jan. 9, 1960 when Sergey L. Sobolev (1908–1989) invited me to move to the Novosibirsk Akademgorodok and to join the cybernetics department of the new Mathematical Institute.

Topics in combinational complexity were largely developed by the Yablonski school, which attributed exceptional significance to asymptotic laws governing synthesis of optimal control systems. The impetus for these works was provided by Shannon's seminal work on synthesis of circuits. However, the results of S. V. Yablonski, Oleg B. Lupanov and their followers surpassed all that was done in the West at that time, as can be seen from Lupanov's survey [19]. But focusing on asymptotic evaluations caused the oversight of other problems for which estimates up to a constant factor are still important.

A *perebor* algorithm, or *perebor* for short, is Russian for what is called in English a "brute force" or "exhaustive search" method. Work on the synthesis and minimization of boolean functions led to the realization of the role of *perebor* as a trivial optimization algorithm, followed by Yablonski's hypothesis of its non-elimination. In 1959 he published a theorem which he considered proof of the hypothesis [77]. However the interpretation of the problem given in his results was not universally convincing – a presage of future controversies in the TCS community. I told this story in detail in [67], and will touch it briefly in the next section.

In the winter of 1954, I was asked to translate into Russian a paper by A. Burks and J. Wright, two authors I didn't know earlier. Unexpectedly, this episode strongly influenced my "Cybernetical" tastes and provided the impetus to research in Automata Theory. A curious detail is that in [7], the authors don't even mention the term "automaton", and focus on Logical Nets as a mathematical model of physical circuits. Afterwards. "Logical Nets" would also appear in the titles of my papers in Automata Theory, even though the emphasis was not so much on circuitry, as on operators, languages and logical specifications.

The use of propositional logic, promoted independently by V.I. Shestakov and C. Shannon, turned out to be fruitful for combinational synthesis, because it suffices to precisely specify the behaviour of memoryless circuits. However, for the expression of temporal constraints one needs other, appropriate, specification tools, which would allow to handle synthesis at two stages: At the first, behavioral stage, an automaton is deemed constructed once we have finite tables defining its next-state and output functions, or, equivalently, its canonical equations. This serves as raw material for the next stage, namely for structural synthesis, in which the actual structure (circuit) of the automaton is designed. (Note, that in [20] Kleene does not yet clearly differentiate between the stages of behavioural and structural synthesis.) After some exercises in structural synthesis I focused on behavioral synthesis and began to collaborate with Nathan E. Kobrinsky (1910–85), who at that time held a position in the Penza Polytechnical Institute. Our book "Introduction to the Theory of Finite Automata" [17] was conceived as a concord of pragmatics (N.E.'s contribution) and theory (summary of my results). The basic text was written in 1958, but the book was typeset in 1961, and distributed only in early 1962, when both of us had already left Penza.

5 Automata

5.1 Languages and Operators

The concept of a finite automaton has been in use since the 1930s to describe the growing automata now known as Turing machines. Paradoxically, though finite automata are conceptually simpler than Turing machines, they were not systematically studied until the Fifties, if we discount the early work of McCulloch and Pitts. A considerable part of the collection "Automata Studies [20] was already devoted to finite automata. Its prompt translation into Russian, marked the beginning of heightened interest by Soviet researchers in this field. In particular, the translation included a valuable appendix of Yuri T. Medvedev (one of the translators), which simplified and improved Kleene's results, and anticipated some of Rabin and Scott's techniques for nondeterministic automata.

As in the West, the initial period was characterized by absence of uniformity, confusion in terminology, and repetition of basically the same investigations with some slight variants. The subject appeared extremely attractive to many Soviet mathematicians, due to a fascination with automata terminology with which people associated their special personal expectations and interests. Automata professionals who came from other fields readily transferred their experience and expertise from algebra, mathematical logic, and even physiology to the theory of finite automata, or developed finite-automata exercises into approaches to other problems.

Kleene's regular expressions made evident that automata can be regarded as certain special algebraic systems, and that it is possible to study them from an algebraic point of view. The principal exponents of these ideas in the SU were Victor M. Glushkov (1923–1982) and his disciples, especially Alexander A. Letichevski, Vladimir N. Red'ko, Vladimir G. Bodnarchuk. They advocated also the use of regular expressions as a primary specification language for the synthesis of automata. Later, adherents of this trend in the SU and abroad developed a rich algebraic oriented theory of languages and automata (see [34]).

Counterbalancing this "algebra of languages" philosophy, I followed a "logic of operators" view on the subject, suggested by A. Burks and J. Wright. In [7] they focused on the input-output behaviour of logical nets, i.e. on operators that convert input words in output words of the same length, and infinite input sequences into infinite output sequences.[2] Apparently, they were the first to study infinite behaviour of automata with output, and to (implicitly) characterize input-output operators in terms of retrospection and memory. Furthermore, they considered Logical Nets as the basic form of interaction between input-output agents.

To summarize, Burks and Wright suggested the following ideas I adopted and developed in my further work on the subject:

1. Priority of semantical considerations over (premature) decisions concerning specification formalisms.
2. Relevance of infinite behaviour; hence, ω-sequences as an alternative to finite words.
3. The basic role of operators as an alternative to languages.

Accordingly to those ideas, I focused on two set-theoretical approaches to the characterization of favorite operators and ω-languages (i.e. sets of ω-sequences). The first is in terms of memory; hence, operators and languages with finite memory. The second one, follows the spirit of Descriptive Set Theory (DST), and selects operators and ω-languages by appropriate metrical properties and

[2] Compare this with D. Scott's argumentation in [38]: "The author (along with many other people) has come recently to the conclusion that the functions computed by the various machines are more important – or at least more basic – than the sets accepted by these devices. The sets are still interesting and useful, but the functions are needed to understand the sets. In fact by putting the functions first, the relationship between various classes of sets becomes much clearer. This is already done in recursive function theory and we shall see that the same plan carriers over the general theory."

set-theoretical operations. (Note, that the set of all ω-sequences over a given alphabet can be handled as a metrical space with suitably chosen metrics.)

My first reaction on the work of Burks and Wright was [49], submitted in 1956 even before the collection *Automata Studies* was available. A footnote added in proof mentions: “the author learned about Moore’s paper in [20], whose Russian translation is under print”.

The paper [49] deals with operators, and distinguishes between properties related to retrospection, which is nothing but a strong form of continuity, and those related to finite memory. In [55] a class of finite-memory ω-languages is defined which is proved to contain exactly those ω-languages, that are definable in second order monadic arithmetic. Independently Büchi found for them a characterization in terms of the famous “Büchi automata”.

In the paper [51], I started my main subject – synthesis of automata, developed later in the books [17] and [63].

6 Experiments and Formal Specifications

Usually, verbal descriptions are not appropriate for the specification of input-output automata. Here are two alternative approaches:

1. *Specification by examples.* This amounts to assembling a table which indicates for each input word x, belonging to some given set M, the corresponding output word z. Further, the synthesis of the automaton is conceived as an interpolation, based on that table. This approach was very popular among soviet practitioners, and suggested the idea of algorithms for automata-identification. Such an algorithm should comprise effective instructions as to:
   (a) What questions of the type “what is the output of the black box for input x?” should be asked?
   (b) How should the answers to these questions be used to ask other questions?
   (c) How to construct an automaton which is consistent with the results of the experiment?

   In his theory of experiments [20], Moore proved that the behavior of an automaton with k states can be identified (restored) by a multiple experiment of length $2k-1$. Independently, I established in [49] the same result, and used it in [17] to identify automata, with an a priori upper bound of memory. I conjectured also in [49] that the restorability degree of “almost” all automata is of order $\log k$, i.e. essentially smaller than $2k-1$. This conjecture was proved by Barzdins and Korshunov [63]. Barzdin also developed frequency identification algorithms [63], which produce correct results with a guaranteed frequency, even when there is no a priori upper bound of the memory. The complexity estimation for such algorithms relies on the proof of the $\log k$ conjecture. Later Barzdin and his collective in Riga significantly developed these ideas into a comprehensible theory of inductive learning.

2. *Formal Specifications.* The second approach, initiated by S.C. Kleene in [20], amounts to designing special specification formalisms, which suitably use logical connectives. However the use of only propositional connectives runs into difficulties, because they cannot express temporal relationships.
   Actually, Kleene's paper in [20] contains already some hints as to the advisability and possibility of using formulas of the predicate calculus as temporal specifications. Moreover, Church [10] attributes to Kleene the following:

   > *Characterization Problem:* Characterize regular events directly in terms of their expression in a formalized language of ordinary kind, such as the usual formulations of first or second order arithmetic.

6.1 Towards Logical Specifications

The years 1956–61 marked a turning point in the field, and Church reported about that at the 1962-International Mathematical Congress. Here is a quotation from [10]: "This is a summary of recent work in the application of mathematical logic to finite automata, and especially of mathematical logic beyond the propositional calculus."

Church's lecture provides a meticulous chronology of events (dated when possible up to months) and a benevolent comparison of his and his student J. Friedman's results with work done by Büchi, Elgot and myself. Nevertheless, in the surveyed period (1956–62) the flow of events was at times too fast and thus omission prone. That is why his conclusion: "all overlaps to some extent, though more in point of view and method than in specific content" needs some reexamination. Actually, the reference to Büchi's paper [6], as well as the discussion of my papers [51,53], were added only "in proof" to the revised edition of the lecture (1964). My other Russian papers [54,55] were still unknown to Church at that time.

Independently, I, myself [51], and somewhat later A. Church [9] developed languages based on the second order logic of monadic predicates with natural argument. Subsequently, another variant was published by R. Büchi [5].

In those works the following restrictions were assumed:

- Trakhtenbrot (1958) [51]: restricted first order quantification;
- Church (1959) [9]: no second order quantification;
- Büchi (1960) [5]: restriction to predicates that are true only on a finite set of natural numbers.

All these languages are particular cases of a single language, widely known now as S1S – Second Order Monadic Logic with One Successor, in which all the restrictions above are removed.

Various arguments can be given in favor of choosing one language or another, or developing a new language. Nevertheless, two requirements seem to be quite natural: The first one (expressiveness) represents the interest of the client, making easier for him the formulation of his intention. The second requirement reflects the viewpoint of the designer; there must be an (fairly simple (?)) algorithm for the synthesis problem in the language.

These two requirements are contradictory. The more comprehensive and expressive the language, the more universal and so more complex is the algorithm. Moreover, if the language is too comprehensive the required algorithm may not exist at all. It turned out that the choice of S1S supports the demand of expressiveness and still guarantees a synthesis algorithm. Indeed, one can show, that all other known specification formalisms can be embedded naturally into S1S. However, this process is in general irreversible.

6.2 Synthesis

Church's lecture focuses on four problems, namely: 1) simplification; 2) synthesis; 3) decision; 4) Kleene's Characterization Problem.[3]

Problem 2, better known as the Church-synthesis problem, amounts roughly to the following: Given a S1S-formula $A(x, y)$:

a) Does there exist an automaton M with input x and output y, whose behaviour satisfies $A(x, y)$?
b) If yes, construct such an automaton.

By solutions are understood algorithms that provide the correct answers and/or constructions.

Problem 4 presumes the invention of a logical formalism L (actually – a rich sublanguage of S1S), which expresses exactly the operators (or events) definable by finite automata, and is equipped with two translation algorithms: (i) from formulas to automata (Kleene-synthesis) and (ii) from automata to formulas (Kleene-analysis).

In accordance with the above classification, [17] deals with Kleene-synthesis and Kleene-analysis.

Actually, in [17] we used the following three formalisms to specify input-output operators:

1. At the highest level – formulas of S1S.
2. At the intermediate level – finite input-output automata represented by their canonical equations.
3. At the lower level – logical nets.

Note that in [17] regular expressions are not considered!

Correspondingly, we dealt there with both behavioral synthesis (from 1 to 2) and with structured synthesis (from 2 to 3).

Büchi was the first to use automata theory for logic and proved [6] that S1S is decidable. These achievements notwithstanding, the general Church-synthesis problem for specifications in full S1S remained open, not counting a few special classes of S1S-formulas, for which the problem was solved by Church and myself (see [8] and [54]). The game theoretic interpretation of Church-synthesis is due to McNaughton [22]. R. Büchi and L. Landweber used this interpretation to solve

[3] Of course there is also the problem of *efficiency*: estimate and improve the complexity of the algorithms and/or the succinctness of the results they provide.

the general Church-synthesis problem. Note, that the original proof in [22] was erroneous. Unfortunately, I did not detect this error, which was reproduced in the Russian edition of [63], and corrected later by L. Landweber in the English translation.

Part 1 of the book [63] constitutes a revised version of my lectures at Novosibirsk University during the spring semester of 1966. It summarizes the results of Church, Büchi-Landweber, McNaughton and myself, as explained above. Part 2, written by Barzdin, covers his results on automaton identification.

6.3 About the Trinity

The choice of the three formalisms in [17] is the result of two decisions. The first identifies three levels of specifications; one can refer to them respectively as the declarative, executable and interactive levels. The second chooses for each of these levels a favorite formalism. In [17] those were, respectively, S1S-Formulas, Automata and Logical Nets; these three are collectively called “The Trinity” in [70]. The first decision is more fundamental, and is recognizable also in computational paradigms beyond finite automata. The second decision is flexible even for finite automata; for example, the Trinity does not include Regular Expression (in [17], they are not even mentioned!) After Pnueli’s seminal work, Linear Temporal Logics (LTL) became very popular as a declarative formalism. But note that the various versions of LTL are in fact just the friendly syntactical sugar of S1S-fragments, and that the most extended one, called ETL, has the same expressive power as the whole S1S. In this sense, one can argue that S1S is the genuine temporal logic, and that the Trinity has a basic status. Moreover, recent computational paradigms are likely to revive interest in the original Trinity and its appropriate metamorphoses.

7 Complexity

7.1 Entering the Field

In 1960, I moved to the Akademgorodok, the Academic Center near Novosibirsk, where, through the initiative and guidance of Lyapunov, the Department of Theoretical Cybernetics was established within the Mathematical Institute.

I continued to work on automata theory, which I had begun at Penza, at first focusing mainly on the relationship between automata and logic, but also doing some work in structural synthesis [46,52,57]. At that time automata theory was quite popular, and that is what brought me my first Ph.D. students in Novosibirsk: M. Kratko, Y. Barzdin, V. Nepomnyashchy.

However, this initial interest was increasingly set aside in favor of computational complexity, an exciting fusion of combinatorial methods, inherited from switching theory, with the conceptual arsenal of the theory of algorithms. These ideas had occurred to me earlier in 1955 when I coined the term “signalizing function” which is now commonly known as “computational complexity measure”.

(But note that "signalizing" persisted for a long time in Russian complexity papers and in translations from Russian, puzzling English-speaking readers.) In [47] the question was about arithmetic functions f specified by recursive schemes R. I considered there the signalizing function that for a given scheme R and nonnegative x, returns the maximal integer used in the computation of $f(x)$ according to R. As it turned out, G.S. Tseytin (1936–), then a student of A.A. Markov at Leningrad University, began in 1956 to study time complexity of Markov's normal algorithms. He proved nontrivial lower and upper bounds for some concrete tasks, and discovered the existence of arbitrarily complex 0-1 valued functions (Rabin's 1960 results became available in the SU in 1963). Unfortunately, these seminal results were not published by Tseytin; later, they were reported briefly (and without proofs) by S.A. Yanovskaya in the survey [78].

Because of my former background, my interest in switching theory, automata, etc. never did mean a break with Mathematical Logic and Computability. In fact, the Sixties marked a return to those topics via research in complexity of computations.

I profited from the arrival of Janis M. Barzdins (1937–) and Rusins V. Freivalds (1942–) in Novosibirsk as my postgraduate students. These two, both graduates of the Latvian University in Riga, engaged actively and enthusiastically in the subject. Alexey V. Gladkiy (1928–) and his group in mathematical linguistics also became interested in complexity problems, concerning grammars and formal languages. Soon other people joined us, mainly students of the Novosibirsk University. My seminar "Algorithms and Automata" was the forum for the new complexity subjects, and often hosted visitors from other places. This is how research in computational complexity started in Novosibirsk; a new young generation arose, and I had the good fortune to work with these people over a lengthy period.

Subsequently, I joined forces with A.V. Gladkiy in a new department of our Mathematical Institute, officially called the Department of Automata Theory and Mathematical Linguistics. Its staff in different periods included our former students Mikhail L. Dekhtyar, (1946–), Mars K. Valiev (1943–), Vladimir Yu. Sazonov (1948–), Aleksey D. Korshunov (1936–), Alexander Ya. Dikovski (1945–), Miroslav I. Kratko (1936–) and Valeriy N. Agafonov (1943–1997).

The basic computer model we used was the Turing machine with a variety of complexity measures; for example, besides time and space, also the number of times the head of the machine changes its direction. Along with deterministic machines we considered also nondeterministic machines, machines with oracles, and probabilistic machines.

It is not surprising that we were attracted by the same problems as our colleagues in the West, notably – J. Hartmanis and R. Stearns. Independently and in parallel we worked out a series of similar concepts and techniques: complexity measures, crossing sequences, diagonalization, gaps, speed-up, relative complexity, to cite the most important ones.

Blum's machine-independent approach to complexity was new for us, and it aroused keen interest in our seminar. But, when later, at a meeting with Tseytin, I began telling him about Blum's work, he interrupted me almost at once and proceeded to set forth many basic definitions and theorems. As it turned out, he had realized it for some time already, but had never discussed the subject in public!

My "gap" theorem [61] was stimulated by Blum's theory. It illustrated a set of pathological time-bounding functions which need to be avoided in developing complexity theory. Meyer and McCreight's "Honesty Theorem" [21] showed how this can be done through the use of appropriate "honest" functions.

In 1967, I published a set of lecture notes [61] for a course "Complexity of Algorithms and Computations" that I had given in Novosibirsk. The notes contained an exposition of results of Blum and Hartmanis-Stearns, based on their published papers, as well as results of our Novosibirsk group: my "gap" theorem, Barzdin's crossing sequences techniques [2], and my other results reported on our seminar [58,59,60].

I sent a copy of these notes to M. Blum (by then at Berkeley). Further I am quoting Albert Meyer [24]:

> Blum passed on a copy of the Trakhtenbrot notes to me around 1970 when I was at MIT since I knew of a graduate student who was interested in translating them. His work was not very satisfactory, but then Filloti came to MIT to work as a post-doc with me and did a respectable job. By this time the notes began to seem dated to me (about five years old in 1972!) and I decided that they needed to be revised and updated. This youthful misjudgment doomed the project since I was too impatient and perfectionist to complete the revision myself, and the final editing of the translation was never completed.

In the academic year 1970–71, V.N. Agafonov continued my 1967-course, and published the lecture notes [1] as Part 2 of "Complexity of Algorithms and Computations". But, unlike Part 1, which focused on complexity of computations (measured by functions) Part 2 was dedicated to descriptive complexity of algorithms (measured by numbers). It contained a valuable exposition of the literature around bounded Kolmogorov complexity and pseudo-randomness, including contributions of Barzdin and of Valery himself.

7.2 Towards Applications

In the SU it was fully in the tradition of Algorithm Theory to handle applications of two kinds: (i) Proving or disproving decidability for concrete problems, (ii)Algorithmic interpretation of mathematical concepts (for example – along the line of constructive analysis in the Markov School). So, it seemed natural to look for similar applications in the complexity setting.

The attitude of the "classical" cybernetics people (notably, Yablonski) to the introduction of the theory of algorithms into complexity affairs was quite negative. The main argument they used was that the theory of algorithms is essentially a theory of diagonalization, and is therefore alien to the complexity area that requires combinatorial constructive solutions. And indeed, except some simple lower bounds supported by techniques of crossing sequences, all our early results rested on the same kind of "diagonalization" with priorities, as in classical computability theory.

But whereas in Algebra and Logic there were already known natural examples for undecidability phenomena which were earlier analyzed in the classical theory, no natural examples of provable complexity phenomena were known. This asymmetry was echoed by those who scoffed at the emptiness of the diagonal techniques with respect to applications of complexity theory. In particular, they distrusted the potential role of algorithm based complexity in the explanation of *perebor* phenomena, and insisted on this view even after Kolmogorov's new approach to complexity of finite objects.

In the summer of 1963, during a visit by A.N. Kolmogorov to the Novosibirsk University, I learned more about his new approach to complexity and the development of the concepts of information and randomness by means of the theory of algorithms. In the early cybernetics period, it was already clear that the essence of problems of minimization of boolean functions was not in the particular models of switching circuits under consideration. Any other natural class of 'schemes', and ultimately any natural coding of finite objects (say, finite texts) could be expected to exhibit similar phenomena, and, in particular, those related to *perebor*. But, unlike former pure combinatorial approaches, the discovery by Kolmogorov (1965), and independently by Solomonoff (1964) and Chaitin (1966), of optimal coding for finite objects occurred in the framework of algorithm and recursive function theory. (Note that another related approach was developed by A.A. Markov (1964) and V. Kuzmin (1965).)

7.3 Algorithms and Randomness

I became interested in the correlation between these two paradigms back in the Fifties, when P.S. Novikov, called my attention to algorithmic simulation of randomness in the spirit of von Mises-Church strategies. Ever since, I have returned to this topic at different times and for various reasons, including the controversies around *perebor*. Since many algorithmic problems encounter essential difficulties (non existence of algorithms or non existence of feasible ones), the natural tendency is to use devices that may produce errors in certain cases. The only requirements are that the probability or frequency of the errors does not exceed some acceptable level and that the procedures are feasible. In the framework of this general idea, two approaches seemed to deserve attention: probabilistic algorithms and frequential algorithms.

In the academic year 1969–70, I gave a course "Algorithms and Randomness" which covered these two approaches, as well as algorithmic modeling of of Mizes-Church randomness.

The essential features of a frequential algorithm M are generally as follows.

1. M is deterministic, but each time it is applied, it inputs a whole suitable sequence of inputs instead of an individual one, and then inputs the corresponding sequence of outputs.

2. The frequency of the correct outputs must exceed a given level.

The idea of frequency computations is easily generalized to frequency enumerations, frequency reductions, etc.

I learned about a particular such model from a survey by McNaughton (1961), and soon realized that as in the probabilistic case, it is impossible to compute functions that are not computable in the usual sense [56].

Hence, the following questions [65]:

1. Is it possible to compute some functions by means of probabilistic or frequential algorithms with less computational complexity than that of deterministic algorithms?
2. What reasonable sorts of problems (not necessarily computation of functions) can be solved more efficiently by probabilistic or frequential algorithms than by deterministic ones? Do problems exist that are solvable by probabilistic or frequential algorithms but not by deterministic algorithms?

These problems were investigated in depth by Barzdin [3], Freivald [12] and their students.

7.4 Relativized Complexity

Computations with oracles are a well established topic in the Theory of Algorithms, especially since Post's classical results and the solution of his famous problem by Muchnik and Friedberg. So it seemed to me quite natural to look how such issues might be carried to the complexity setting [62]. At this point I should mention that Meyer's confession, about the translation of my lecture notes, points only on a transient episode in our long-time contacts. Let me quote Albert again [24]:

> Repeatedly and independently our choices of scientific subareas, even particular problems, and in one instance even the solution to a problem, were the same. The similarity of our tastes and techniques was so striking that it seemed at times there was a clairvoyant connection between us. Our relationship first came about through informal channels – communications and drafts circulated among researchers, lecture notes, etc. These various links compensated for the language barrier and the scarcity of Soviet representation at international conferences. Through these means there developed the unusual experience of discovering an intellectual counterpart, tackling identical research topics, despite residing on the opposite side of the globe.... Today ... we find ourselves collaborating firsthand in an entirely different area of Theoretical Computer Science than complexity theory to which we were led by independent decisions reflecting our shared theoretical tastes.

As to computations with oracles, we both were attracted by the question:to what extent can be simplified a computation by bringing in an oracle, and how accurately can the reduction of complexity be controlled depending on the choice of the oracle? This was the start point for a series of works of our students (mainly M. Dekhtyar, M. Valiev in Novosibirsk and N. Lynch at MIT) with similar results of two types: about oracles which do help (including the estimation of the help) and oracles which cannot help. The further development of the subject by Meyer and M. Fischer ended with a genuine complexity-theoretic analog to the famous Friedberg-Muchnik theorem. It reflects the intuitive idea that problems might take the same long time to solve but for different reasons! Namely:

> *There exist nontrivial pairs of (decidable!) sets, such that neither member of a pair helps the other be computed more quickly.*

Independently of Meyer and Fisher, and using actually the same techniques, I obtained an improvement of this theorem. That happened in the frame of my efforts to use relative algorithms and complexity in order to formalize intuitions about mutual independence of tasks and about *perebor*.

7.5 Formalizing Intuitions

Autoreducibility. When handling relativized computations it is sometimes reasonable to analyze the effect of restricted access to the oracle. In particular, this is the case with the algorithmic definition of "collectives", i.e. of random sequences in the sense of von Mises-Church. This definition relies on the use of "selection strategies", which are relative algorithms with restricted access to oracles. A similar situation arises with the intuition about mutual independence of individual instances which make up a general problem [64]. Consider, for example, a first order theory T. It may well happen that there is no algorithm, which, for an arbitrary given formula $\mathcal{A}$, decides whether $\mathcal{A}$ is provable or not in T. However, there is a trivial procedure W which reduces the question about $\mathcal{A}$ to similar questions for other formulas; W just inquires about the status of the formula $(\neg(\neg\mathcal{A}))$. The procedure W is an example of what may be called *autoreduction*. Now, assume that the problem is decidable for the theory T, and hence the correct answers can be computed directly (without autoreduction). It still might happen that one cannot manage without very complex computations, whereas the autoreduction above is simple.

A guess strategy is a machine M with oracle, satisfying the condition: for every oracle G and natural number n, the machine M, having been started with n as input, never addresses the oracle with the question "$n \in G$?" (although it may put any question "$\nu \in G$?" for $\nu \neq n$). A set G is called *autoreducible* if it possesses an autoreduction, i.e. a guess strategy which, having been supplied with the oracle G, computes the value $G(n)$ for every n. Otherwise G is *non-autoreducible*, which should indicate that the individual queries "$n \in G$?" are mutual independent.

It turned out that:

- (i) The class of non-autoreducible sequences is essentially broader than the class of random sequences.

- (ii) There are effectively solvable mass problems M of arbitrary complexity with the following property: autoreductions of M are not essentially less complex than their unconditional computations.

Understanding *Perebor*. Disputes about *perebor*, stirred by Yablonski's paper [77], had a certain influence on the development, and developers of complexity theory in the Soviet Union. By and large, reflections on *perebor* spurred my interest in computational complexity and influenced my choice of special topics, concerning the role of sparse sets, immunity, oracles, frequency algorithms, probabilistic algorithms, etc. I told this story in detail in [67]; below I reproduce a small fragment from [67].

The development of computational complexity created a favorable background for alternative approaches to the *perebor* topics: the inevitability of *perebor* should mean the nonexistence of algorithms that are essentially more efficient. My first attempt was to explain the plausibility of *perebor* phenomena related to the "frequential Yablonski-effect"; it was based on space complexity considerations. Already at this stage it became clear that space complexity was too rough and that time complexity was to be used. Meanwhile I began to feel that another interpretation of *perebor* was worth considering, namely, that the essence of *perebor* seemed to be in the complexity of interaction with a "checking mechanism", as opposed to the checking itself. This could be be formalized in terms of oracle machines or reduction algorithms as follows. Given a total function f that maps binary strings into binary strings, consider Turing machines, to compute f, that are equipped with the oracle G that delivers (at no cost!) the correct answers to queries "$f(x) = y$?" (x, y may vary, but f is always the same function). Among them is a suitable machine $\mathrm{M}_{\mathrm{perebor}}$ that computes $f(x)$ by subsequently addressing the oracle with the queries

$$f(x) = B(0)?, f(x) = B(1)?, \dots , f(x) = B(i)? \dots$$

where $B(i)$ is the i-th binary string in lexicographical order. Hence, in the computation of the string $f(x)$ the number of steps spent by $\mathrm{M}_{\mathrm{perebor}}$ is that represented by the string $f(x)$. I conjectured in 1966 that for a broad spectrum of functions f, no oracle machine M can perform the computation essentially faster. As for the "graph predicates" $G(x, y)$ =def $f(x) = y$, it was conjectured that they would not be too difficult to compute. From this viewpoint, the inevitability of *perebor* could be explained in terms of the computational complexity of the reduction process. The conjecture was proved by M. I. Dekhtyar in his master's thesis (1969) for different versions of what "essentially faster" should mean. Using modern terminology, one can say that Dekhtyar's construction implicitly provides the proof of the relativized version of the NP $\neq$ P conjecture. For the first time, this version was explicitly announced by Baker, Gill and Soloway (1975) together with the relativized version of the NP = P conjecture. Their intention was to give some evidence to the possibility that neither NP = P nor NP $\neq$ P is provable in common formalized systems. As to my conjecture, it had nothing to do with the ambitious hopes to prove the independence of the

NP = P conjecture. As a matter of fact, I then believed (and to some extent do so even now) that the essence of *perebor* can be explained through the complexity of relative computations based on searching through the sequence of all binary strings. Hence, being confident that the true problem was being considered (and not its relativization!), I had no stimulus to look for models in which *perebor* could be eliminated.

To the Perebor account [67] it is worth adding the following quotations from my correspondence with Mike Sipser (Feb. 1992):

> S. You write that Yablonski was aware of *perebor* in the early 50's, and that he even conjectured that *perebor* is inevitable for some problems in 1953–54. But the earliest published work of Yablonski that you cite is 1959. Is there a written publication which documents Yablonski's awareness of these issues at the earlier time? This seems to be an important issue, at least from the point of establishing who was the first to consider the problem of eliminating brute force search. Right now the earliest document I have is Godel's 1956 letter to Von-Neumann.
>
> T. I cannot remember about any publication before 1959 which documents Yablonski's awareness of these issues but I strongly testify and confirm that (a quotation follows from my paper [59]): *"Already in 1954 Yablonski conjectured that the solution of this problem is in essence impossible without complex algorithms of the kind of perebor searching through all the versions. . . ."* He persistently advocated this conjecture on public meetings (seminars and symposia).
>
> S. Second, is it even clear that Yablonski really understands what we presently mean by eliminating brute force search? He claimed to have proven that it could not be eliminated in some cases back in 1959. So there must be some confusion.
>
> T. That is indeed the main point I am discussing in Section 1 of my *perebor* paper [67]. The conclusion there is that there is no direct connection between Yablonski's result and what we presently mean by eliminating *perebor*. Hence the long year controversy with Yablonski.
>
> S. I'd appreciate your thoughts on how to handle Yablonski's contribution to the subject.
>
> T. I would mention three circumstances:
> - 1. In Yablonski's conjecture the notion of *perebor* was a bit vague and did not anticipate any specific formalization of the idea of *complexity*. Nevertheless (and may be just due to this fact) it stimulated the investigation of *different approaches* to such a formalization, at least in the USSR.
> - 2. Yablonski pointed from the very beginning on very attractive candidates for the status of problems which need essentially *perebor*. See Sect. 1 of [67], where synthesis of circuits is considered in this context.

> – 3. Finally, he made the point that for his candidates the disaster caused by *perebor* might be avoided through the use of probabilistic methods.
>
> ... Let me mention that as an alternative to Yablonski's approach I advocated the idea of complexity of computations with oracles. In these terms, I formulated a conjecture which presently could be interpreted as the relativised version of P not equal NP. This conjecture was proved by my student M. Dekhtyar [11].

Turning Points. The controversies around *perebor* were exacerbated by the emergence of the new approach to complexity of algorithms and computations. And it was precisely this approach which was relevant for the genuine advance in the investigation of *perebor* in the seminal works of Leonid Levin in the SU and the Americans, Steven Cook and Richard Karp.

The discovery of NP-complete problems gave evidence to the importance of the Theory of Computational Complexity. Soon another prominent result strengthened this perception. In 1972, A. Meyer and Stockmeyer (see [23]) found the first genuine natural examples of inherently complex computable problems. This discovery was particularly important for me because the example came from the area of automata theory and logic in which I had been involved for a long time. Clearly, for the adherents of the algorithmic approach to complexity, including myself, these developments confirmed the correctness of their views on the subject and the worthwhileness of their own efforts in the past. However the time had also come for new research decisions.

In the 70s, certain trends began to develop, which ultimately resulted in fundamental changes. Part of my group (Agafonov, Lomazova, Sazonov, Valiev) and other participants of our seminar became increasingly interested and involved in the investigation of the theory of programming. On the other side, people previously engaged in complexity started to lose interest in this subject.

Against this background, our relationship improved with the Department of Theoretical Programming, headed in the Computer Center by Andrei P. Ershov (1931–1988) – one of the most prominent leaders of programming in the SU. Many of his collaborators and students participated in our seminar. Quoting V.E. Kotov: “In years of stagnation Ershov managed to create around himself a healthy political and humane situation, completely different than that outside; it made our life and work easier.” I benefited from his liberal, benevolent, position; then it was very important. Unfortunately, it did not work against the devising of a hostile official mathematical establishment (I told this story in [75]).

In 1979, I came to the difficult decision about emigration to Israel. We departed in December 1980 with the traumatic prospect of separation (in those times most likely for ever) with relatives, friends, colleagues, students. Most of the remaining staff members left the Mathematics Institute, and our department fell apart.

Since January 1, 1981, I am affiliated with Tel Aviv University. But this is another story!

8 Epilogue

For a long time, I was not actively involved in automata and computational complexity, being absorbed in other topics. During that period both areas underwent impressive development, which is beyond the subject of this account.

My entry into the field happened at an early stage, when formation of concepts and asking the right questions had high priority, at least as far as solving well established problems. This is also reflected in my exposition above, in which the emphasis was rather on the conceptual framework in the area. Some of those concepts and models occurred in very specific contexts, or were driven by curiosity rather than by visible applications. Did they anticipate problems beyond their first motivation? I would like to conclude with some remarks about this.

The first is connected to the new and very attractive paradigms of Timed Automata and Hybrid Systems (HS). Nowadays, despite significant achievements, the area is still dominated by an explosion of models, concepts and ad hoc notation, a reminder of the situation in Automata Theory in the Fifties. However, I believe that the classical conceptual framework can still help to elucidate the intuitions underlying the new paradigms, and to avoid reinvention of existing ideas [31].

One way to do so is to start with two separate and orthogonal extensions of the basic model of a finite automaton M. The first one is by interconnecting M with an oracle N, which is also an automaton, but, in general, with an infinite set of states. Typically, think about a logical net over components M and N, with subsequent hiding of N. Whatever M can do while using N is called its relativization with respect to this oracle. The other extension is with continuous time (instead of discrete time, as in the classical case) but without oracles.

For each of these extensions, considered apart, it becomes easier to clarify how (if any) and to what extent the heritage of classical Automata Theory can be adapted.

Appropriate combinations of the two might facilitate the adaptation of classical heritage, whenever it makes sense.

The next remark is about a resurgence of interest in autoreducibility and frequency computations.

It was instructive to learn that the idea of restricting access to oracles, now underlies several concepts, which are in fact randomized and/or time bounded versions of autoreducibility: coherence, checkability, self-reducibility, etc. Most of these concepts were identified independently of (though later than) my original autoreducibility, and have occupied a special place in connection with program checking and secure protocols (see [4] for details and references).

On the other hand, the idea of frequency computation was extended to bounded query computations and parallel learning. Also, interesting relationships were discovered between autoreducibility, frequency computations and various other concepts.

My final remark is about the continuous conceptual succession since my youthful exercises in Descriptive Set Theory (DST), which should be clear from the previous exposition. In particular, it is quite evident that computational complexity is inspired by computability. But the succession can be traced back even to DST; just keep in mind the ideas which lead from classification of sets and functions to classification of what is computable, and ultimately to hierarchies within computational complexity.

9 Addendum – November 2007

These sketchy notes address three additional research topics I have pursued since the 1970s. Theory of Programming is the last project I was involved in before leaving the SU. I developed it further while in Tel Aviv University through collaboration with Albert Meyer and Joseph Halpern [76] at MIT. Research on the next two topics, Concurrency and Continuous-Time Paradigms, started in Tel Aviv, where I managed to attract to these subjects some of my colleagues and students. In particular, Alex Rabinovich became my main collaborator and coauthor.

These notes are mainly compiled from non-technical parts of texts adopted from previous publications. I am aware of the imperfection of such undertakings. Unfortunately, at this moment, I have no possibility to provide a more comprehensive and lucid account and to give tribute to those who shared the efforts with me.

9.1 Theory of Programming

Introductory Remarks. In the 70s a broad spectrum of topics was on the agenda of the SU community and, in particular of our group. Investigation (and even the primary definitions) of program schemes started long ago in the SU with the works of Lyapunov, Yanov, Ershov and others. Other developments, such as comparative schematology, program logic, as well as verification and specification of programs came mainly from the West.

On the other hand, our interests were significantly inspired by Algol-68, the Scott-Strachey theory of denotational semantics [39] and the importance of lambda-calculus for the theory of programming (see the essay [69]).

Functional languages. The thesis that lambda-calculus underlies programming languages. was first extensively argued by Peter Landin (the author of ISWIM) in his 1965–1966 papers. As a major innovation to lambda-calculus, ISWIM includes the additional binding mechanisms through the let and letrec constructs, which permit the statement of declarations (definitions) in a convenient programming style. Scott's language LCF [37] is ISWIM enriched with fixed point operators and conditionals; PCF is that part to which an appropriate arithmetical signature is added. Whereas lambda-calculus is usually recognized as a sequential language, Scott raised the idea to enrich PCF with parallelism facilities, say with the parallel function OR. Hence, the distinction between sequential and parallel functions and the comparative power of parallel functions suggested themselves.

Imperative languages. However, this was a purely functional approach not assuming the imperative features of real programming languages like FORTRAN, Algol, and Pascal. These languages allow constructs that are alien to the spirit of lambda-calculus, e.g. assignments and **goto**'s. Such constructs, directly inherited from the von Neumann computer architecture, were nicknamed "dirty features" by adepts of pure functional programming. Especially **goto**'s were long ago recognized as a troublesome control mechanism by the pioneers of structured programming (Dijkstra's "**goto**'s considered harmful"). Note that John Backus, the designer of FORTRAN, later joined the criticism of "dirty features". He called for the liberation of programming from the von Neumann style and outlined a new functional language. Hence, the question: to what extent and in what form can Landin's thesis be adopted to imperative languages? That is what I became interested in.

Presented below is a summary of research performed by our group.

Work on Program Schemes, Program Logic, and Specification and Verification of Programs

- V. N. Agafonov:
  - Syntactical analysis in compilation.
  - Typology (semantics) of programming languages and verification of programs.
  - Specification of programs.
- I. A. Lomazova:
  - Inductive conditions in Hoare's logic for programs with loops (independent of Apt, Bergstra, Tucker).
  - Semantics and complete Hoare-type algorithmic logic for programs with **goto**.
- B. A. Trakhtenbrot:
  - Universality of classes of program schemes.
  - Recursive program schemes and computable functionals.
  - Relaxation rules and completeness of algorithmic logic.
- M. K. Valiev:
  - Axiomatization and decision complexity for variants of PDL (independent of Halpern).

Work on LCF-PCF

- V. Yu. Sazonov [35,36]: All the results below were obtained independently (and even prior) to G. Plotkin's "LCF as a programming language" [26] and related works on semantics of type-free lambda calculus by M. Hyland and C. Wadsworth.

  - Precise characterization of the expressive power of the programming part of Scott's language LCF (currently known as PCF). This was given in terms of computational strategies – a precursor of currently widely used game semantics.

- Exact correspondence between operational and denotational semantics in terms of computational strategies (in both typed and type-free settings).
- Characterization of degrees of parallelism in computations.

Work on Algol-Like Languages [66,76,14]
My view on the essence of Algol-like languages was crystallized in discussions with V. Sazonov. The never-published technical report [66] is my first account on the subject. As I learnt later, J. C. Reynolds promoted similar ideas on the essence of what he called "Idealized Algol" [32,33].

The main idea is to face the impediments of the original Algol through the design of an appropriate Algol-like language (Idealized Algol) that can be explained in the lambda-calculus core and supported with denotational semantics. Here are some of the principles that characterize this class of languages:

– Exclude **goto**'s but preserve assignments.
– The language is fully typed.
– Higher order procedures of all finite types are allowed.
– There is a clear distinction between locations and storable values (integers, say).
– Blocks with local storage, and sharing (aliasing) are allowed.

Actually, these languages preserve as much as possible of the rich expressive power of the original Algol, and have sufficient structure to yield a rich algebra and proof theory. Moreover, the denotational semantics of a program is provided by a two-step process:

1. A purely syntactic translation of the program into an ISWIM expression. This step provides the "true" ISWIM syntax for the program – a worthy alternative for the Algol-jargon which came down through history.
2. Assignment of semantics to ISWIM in the standard way, assuming an adequate choice of domains, which is consistent with the underlying intuition.

Thus, programs simply inherit their semantics from the ISWIM-terms into which they are translated. In this way, procedures are entirely explained at a purely functional level – independent of the interpretation of program constructs – by continuous models for lambda-calculus. However, the usual (complete partial order) models are not adequate to model local storage allocation. New domains of store models are offered to solve this problem and partial correctness theory over store models is developed.

9.2 Concurrency

Introductory Remarks. Before the emergence of modern theory of concurrency, logical nets presented the main model of concurrency in automata theory (see Sect. 5). The interaction of components in a logical net is often called synchronous (we would prefer "simultaneous") interaction. Our concern below is about asynchronous nets and we call them simply *nets*.

Nets are widely used in the theory of concurrency. One evident reason for that is the convenience of visualizing the communication structure of systems, as it comes to light for example in Petri nets or in data flow nets. On the other hand, as emphasized by Pratt, nets seem to cover almost any situation which involves "sharing" or communication. For an engineer it could mean sharing of component terminals by connecting them electrically; a mathematician can consider sharing of variables in a system of equations.

Modularity reflects the Frege Principle: any two expressions $expr_1$ and $expr_2$ that have the same meaning (semantics) can be replaced by each other in every appropriate context without changing the meaning of the overall expression. In a conventional syntax with signature Σ (call it "textual" as opposed to the graphical syntax of nets), a complex piece of syntax *expr* may be uniquely decomposed into simpler sub-pieces: $expr = op(expr_1, \ldots, expr_k)$, where *op* is in Σ. Typically, a denotational semantics is formulated in such a compositional style and hence supports modularity. However, often one starts with an operational semantics that lacks a compositional structure. Then a standard way to prove modularity is to discover a compositional semantics which is equivalent to the operational one.

In net models of concurrency, a syntax is provided by some specific class *NN* of labeled graphs called nets, and a semantics is usually defined globally either in an operational style through appropriate firing (enabling) rules, as in the case of Petri nets, or as a solution of a system of equations. However, the notion of context makes sense for nets and therefore modularity of nets may be defined and investigated.

For a long time, despite the rich and suggestive information of the Petri pictorial approach, the question of compositionality or modularity had not been raised. For elementary Petri nets modularity was established by Antoni Mazurkiewicz. Namely, he discovered a compositional semantics which is equivalent to the original "token game" semantics. This compositional approach to Petri nets provided the initial stimulus to our work.

Challenges that we Addressed:

- Unification of concepts and approaches for the numerous models and semantics proposed for nets [27].
- The "causal (called also true) vs. interleaving semantics" dilemma for Petri nets.
- Modularity issues for nets (especially, for data flow nets).
- The close subject of compositional proof systems for nets ([71]).

Below we consider in more detail our work on causality and modularity issues. Petri nets and data flow nets are the fundamental paradigms we focused on.

Discerning Causality. The question here is: to what extent is the causal (partial order) semantics relevant for concurrent computational problems?

It appears that usually one starts with a common interleaving semantics sem_1 for the system under consideration. Yet, at a later stage, a causal semantics sem_2

is chosen which is consistent with sem_1 in a natural sense. In [16], we studied possible choices of causal semantics for C/E Petri nets. Modularity arguments are used to show that there is a unique such semantics satisfying some simple and necessary modularity and consistency conditions. Moreover, this is true also for other formalisms, such as Milner's CCS and Hoare's CSP. However, in other situations the proper choice of causal semantics still has to be justified by extra conditions. This is illustrated for P/T nets, i.e. Petri nets with bounded capacities (a model that reflects distribution of resources).

Data Flow Nets [28,29]. Unlike other types of nets, here the main concern is the I(nput)/O(utput) behavior of a system (under appropriate topological restrictions). Kahn's Principle states that for special deterministic agents (Kahn automata) the I/O behavior of a net can be obtained from the I/O behaviors of its components as the solution of an appropriate system of equations. This implies I/O compositionality (and hence also modularity) for such nets. In an attempt to generalize Kahn's result to the nondeterministic agent merge, Brock and Ackermann observed that I/O modularity fails. This is the so called Brock-Ackermann anomaly. Hence, two problems are raised by the pioneering works of Kahn and of Brock-Ackermann:

1. Why do data flow nets behave compositionally for Kahn-automata but not for Brock-Ackermann automata?
2. What are the fundamental limits for the applicability of Kahn's Principle?

We carried out an extensive development of dataflow semantics which provides very precise answers to these questions. We showed that modularity may fail even for components with functional I/O behavior. We gave a characterization of functional agents ("smoothness") which is both sufficient and necessary to support modularity and Kahn's Principle. Moreover, the class SMOOTH of these automata is the unique largest I/O-modular class of automata with functional behavior. Any extension of the class of Kahn automata by a component with "ambiguous" I/O-behavior – not necessarily merge – spoils modularity [29,28]. Analyzing the possible deviations from Kahn's Principle we identified two kinds of anomalies. The "meagerness" anomaly may occur even for functional agents and not only for ambiguous ones, as sometimes is misunderstood the original Brock-Ackermann example. On the other hand, the "ambiguity"-anomaly is indeed rooted in the semantics of nondeterminism. These results were credited as an essential progress towards an "ultimate Kahn's Principle".

Nets of Relations. A conceptual and technical novelty we started is the idea to consider semantics of nets of relations [30]. We identified observable relations and nets of observable relations as appropriate tools for the investigation of data flow nets over nondeterministic agents. We showed that the main source of the Brock-Ackermann anomaly is in the semantics of nets of relations. If one considers nets over a subclass of observable relations, it may happen that the semantics over such nets is modular.

9.3 Continuous-Time Paradigms

In Sect. 5 we briefly discussed the status of, and the relationship between, three basic formalism of automata theory: automata, logic and circuits. These are collectively called there "The Trinity". As emphasized in the position paper [70], more recent developments in the theory of real-time systems put forward the task of lifting the classical trinity to continuous time. Some of the work in this direction is summarized in a special issue of Fundamenta Informaticae ([73]). Below I quote the contents of, and the preface to this issue.

The Papers in [73]:

[HR] Hirshfeld, Y., and Rabinovich, A.: Logics for Real Time: Decidability and Complexity [15].

[S] Slissenko, A.: A Logic Framework for Verification of Timed Algorithms [41].

[T] Trakhtenbrot, B. A.: Understanding Automata Theory in the Continuous Time Setting [74].

[PRD] Pardo, D., Rabinovich, A., and Trakhtenbrot, B. A.: Synchronous Circuits over Continuous Time: Feedback Reliability and Completeness [25].

From the Preface of the Editor: In the last 15 years, research in Computer Science has involved many paradigms in which continuous time appears whether in a pure way or in cooperation with discrete time. In particular, this is evident in subjects concerning Automata, Circuits and Logic. This issue of FI consists of four papers dedicated to such subjects. The papers will be referred to below as [HR], [S], [PRT], [T] (the initials of the authors). [HR] and [S] are about continuous-time logics, whereas in [T] and [PRT] the focus is on automata and circuits (logical nets).

In [HR] the concern is about Monadic Logics of Order (MLO) and their relationship to Temporal Logics. For discrete time, decidability of Second Order Monadic Logic (SOML) follows from the connection between SOML and automata theory. It is well known that SOML covers numerous temporal logics (TL). Ultimately, these TL may be considered as syntactically sugared versions of SOML fragments. In [HR] further facts of this kind are established for continuous time. However, in this case, instead of traditional automata-theoretic techniques one needs to use properly general theorems from logic.

The logic framework developed in [S] is based on First Order Timed Logic (FOTL), that allows functions and predicates with more than one argument; moreover, it allows also some arithmetic. This makes the logic expressive enough to represent, more or less directly, continuous-time properties of distributed algorithms. But, on the other hand, it makes the logic undecidable. The fundamental observation that, nevertheless, permits the efficient use this logic for verification is as follows: the underlying theories of continuous time (e.g. the theory of real addition, Tarski algebra, etc.) are decidable or have much better complexity than the corresponding theories of discrete time. Interesting decidable classes of the verification problem are based on appropriate properties of FOTL.

The companion papers [T] and [PRT] draw their initial motivation from the literature on Hybrid Automata, Circuits and related control problems. Concrete problems about circuits (feedback reliability, completeness) and control (sample-and-hold architectures in continuous time) are also the subject of [T] and [PRT]. Yet, a more general contribution is the development of a conceptual framework that allows one to highlight the genuine distinctions and similarities between the discrete-time and continuous-time tracks.

There is a growing feeling in the community that the literature on these subjects, as well on the related logics, is plagued by a Babel of models, constructs and formalisms with an amazing discord of terminology and notation. Further models and formalisms are engendered, and it is not clear where to stop. Hence, appeals like:

> look back to sort out what has been accomplished and what needs to be done... by surveying logic-based and automata-based real-time formalisms and putting them into a perspective. (R. Alur and T. Henzinger).

> ... isolate the right concepts, ... formulate the right models, and discard many others, that do not capture the reality we want to understand....
> (J. Hartmanis).

[HR], [T] and [PRT] are strongly committed to the analysis in depth of the various continuous-time paradigms and to their robust conceptual integration in mainstream Automata Theory and Logic. These papers come to the following conclusions about some misconceptions in the previously suggested models and logics:

1. The standard continuous-time model for logic was ignored as a yardstick; instead, different kinds of sequences of continuous-time bits were used. This may have been an attempt to pursue the connection with automata theory since automata were traditionally associated with sequences. This was the main cause in the rejection of the classical model. It complicated the subsequent research. The choice of the temporal logic became an arbitrary decision [HR].
2. Input/output behavior of automata was ignored in favor of generating devices. Functions (in particular, input/output behavior of automata) are more fundamental than sets (say, languages accepted by automata). Accordingly, circuits and feedback should be defined explicitly as generic concepts. Surprisingly, it has been left unobserved in the literature that some flaws in the conceptual decisions concerning continuous time are identifiable already at the level of discrete-automata theory [T].

References

1. Agafonov, V.N.: Complexity of algorithms and computations (part 2). Lecture Notes, p. 146. Novosibirsk State University (1975)
2. Barzdin, J.M.: Complexity of the recognition of the symmetry predicate in Turing machines. Problemy Kibernetiki 15, 245–248 (1965)
3. Ja Barzdin, M.: On computability on probabilistic machines. DAN SSSR 189, 699–702 (1969)
4. Beigel, R., Feigenbaum, J.: On being incoherent without being very hard. Computational Complexity 2, 1–17 (1992)
5. Büchi, J.R.: Weak second-order arithmetic and finite automata. Z. Math. Logik Grundlagen Math. 6, 66–92 (1960)
6. Büchi, J.R.: On a decision method in restricted second order arithmetic. In: Proc. of the 1960 Intl. Cong. on Logic, Philosophy and Methodology of Sciences, pp. 1–11. Stanford Univ. Press (1962)
7. Burks, A., Wright, J.: Theory of logical nets. Proc. IRE 41(4) (1953)
8. Church, A.: Applications of recursive arithmetic to the problem of circuit synthesis. In: Summaries of the Summer Institute of Symbolic Logic, vol. 1, pp. 3–50. Cornell Univ., Ithaca, NY (1957)
9. Church, A.: Application of recursive arithmetic to the theory of computers and automata, notes, summer conference course. In: Advanced Theory of the Logical Design of Digital Computers, pp. 1–68. University of Michigan (1959)
10. Church, A.: Logic, arithmetic and automata. In: Proceedings of Intl. Congress of Mathematicians, pp. 23–35 (1962)
11. Dekhtyar, M.I.: The impossibility of eliminating complete search in computing functions from their graphs. In: DAN SSSR 189, pp. 748–751 (1969)
12. Freivald, R.V.: Fast probabilistic algorithms. LNCS, vol. 74, pp. 57–69 (1979)
13. Glushkov, V.M.: Synthesis of digital automata, Fizmatgiz, Moscow (1962)
14. Halpern, J., Meyer, A., Trakhtenbrot, B.A.: The semantics of local storage, or what makes the free list free (preliminary report). In: Conference Record of the XI ACM Symposium on Principles of Programming Languages (POPL), pp. 245–257 (1984)
15. Hirshfeld, Y., Rabinovich., A.: Logics for real time: decidability and complexity. Fundamenta Informatica 62(1), 1–28 (2004)
16. Hirshfeld, Y., Rabinovich, A., Trakhtenbrot, B.A.: Discerning causality in interleaving behavior. In: Proceedings of Logic at Botic 1989, Pereslavl Zalessky, USSR (July 1989)
17. Kobrinski, N.E., Trakhtenbrot, B.A.: Introduction to the Theory of Finite Automata, Fizmatgis, Moscow, pp. 1–404 (1962), English translation. In: Studies in Logic and the Foundations of Mathematics, North-Holland(1965)
18. Kuznetsov, A.V., Trakhtenbrot, B.A.: Investigation of partial recursive operators by techniques of Baire spaces. Doklady AN SSR 105 6, 896–900 (1955)
19. Lupanov, O.B.: An approach to systems synthesis – a local coding principle. Problems of Cybernetics 14, 31–110 (1965)
20. McCarthy, J., Shannon, C. (eds.): Automata Studies, Princeton (1956)
21. McCreight, E.M., Meyer, A.R.: Classes of computable functions defined by bounds on computation. In: Proc. of 1st STOC, pp. 79–88 (1969)
22. McNaughton, R.: Finite-state infinite games. In: Project MAC Rep., September 1965, MIT, Cambridge (1965)
23. Meyer, A.R.: Weak monadic second order theory of successor is not elementary recursive. In: Proj. MAC, MIT, Cambridge (1973)

24. Meyer, A.R.: unpublished memo; see the second chapter of this collection
25. Pardo, D., Rabinovich, A., Trakhtenbrot, B.A.: Synchronous circuits over continuous time: feedback reliability and completeness. Fundamenta Informatica 62(1), 123–137 (2004)
26. Plotkin, G.: LCF considered as a programming language. Theoretical Comp. Science 5, 223–257 (1977)
27. Rabinovich, A., Trakhtenbrot, B.A.: Behavior structures and nets of processes. Fundamenta Informaticae 11, 357–403 (1988)
28. Rabinovich, A., Trakhtenbrot, B.A.: Nets of processes and data-flow. In: de Bakker, J.W., de Roever, W.-P., Rozenberg, G. (eds.) Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency. LNCS, vol. 354, pp. 574–602. Springer, Heidelberg (1989)
29. Rabinovich, A., Trakhtenbrot, B.A.: Nets and data-flow interpreters. In: Proceedings of LICS (1989)
30. Rabinovich, A., Trakhtenbrot, B.A.: Communication among relations. In: Proceedings of the 17th Colloquium on Automata, Languages and Programming, Warwick, England (1990)
31. Rabinovich, A., Trakhtenbrot, B.A.: From finite automata toward hybrid systems. In: Chlebus, B.S., Czaja, L. (eds.) FCT 1997. LNCS, vol. 1279, pp. 411–422. Springer, Heidelberg (1997)
32. Reynolds, J.C.: Idealized Algol and its specification logic, Technical Report, Syracuse University pp. 1-81 (1981)
33. Reynolds, J.C.: The essence of Algol. In: de Bakker, van Vliet (eds.) International Symposium on on Algorithmic Languages, North-Holland, pp. 345–372 (1981)
34. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages. pp. I–III, Springer, Berlin (1997)
35. Sazonov, V.Yu., Expressibility of functions in Scott's LCF language, Algebra i Logika 15, 308–320 (in Russian); 192–206 in English edition, 1976.
36. Sazonov, V.Y.: Functionals computable in series and in parallel. Siberian Math. Journal 17(3), 648–672 (1976) (in Russion); 498–516 in English edition
37. Scott, D.A.: Type-theoretical alternative to CUCH, OWHY, ISWIM. Theoretical Computer Science 121, 411–440 (1993) Reprint of a manuscript, Oxford University (1969)
38. Scott, D.: Some definitional suggestions in automata theory.J. of Computer and Syst. Sci., 187–212 (1967)
39. Scott, D., Strachey, C.: Toward a mathematical semantics of computer languages. In: Proceedings of a Symposium on Computer and Automata, New York (1971)
40. Shtokalo, I.Z. (ed.): History of Mathematics, Kiev (in Russian, 1970)
41. Slissenko, A.: A logic framework for verification of timed algorithms. Fundamenta Informaticae 62(1) (August 2004)
42. Trakhtenbrot, B.A.: The impossibility of an algorithm for the decidability problem on finite classes. Doklady AN SSR 70(4), 569–572 (1950)
43. Trakhtenbrot, B.A.: Decidability problems for finite classes and definitions of finite sets. Ph.D. Thesis, Math. Inst. of the Ukrainian Academy of Sciences, Kiev (1950)
44. Trakhtenbrot, B.A.: On recursive separability. Doklady AN SSR 88(6), 953–956 (1953)
45. Trakhtenbrot, B.A.: Tabular representation of recursive operators. Doklady AN SSR 101(4), 417–420 (1955)
46. Trakhtenbrot, B.A.: The synthesis of non-repetitive schemas. Doklady AN SSR 103(6), 973–976 (1955)

47. Trakhtenbrot, B.A.: Signalizing functions and tabular operators. Uchionnye Zapiski Penzenskogo Pedinstituta (Transactions of the Penza Pedagogoical Institute) 4, 75–87 (1956)
48. Trakhtenbrot, B.A.: On the definition of finite set and the deductive incompleteness of set theory. Izvestia AN SSR 20, 569–582 (1956)
49. Trakhtenbrot, B.A.: On operators, realizable by logical nets. Doklady AN SSR 112(6), 1005–1006 (1957)
50. Trakhtenbrot, B.A.: Algorithms and Computing Machines, Gostechizdat (1957) second edition by Fizmatgiz, 1960; English translation in the series: Topics in Mathematics, D.C. Heath and Company, Boston, pp. 1–101 (1963)
51. Trakhtenbrot, B.A.: The synthesis of logical nets whose operators are described in terms of monadic predicates. Doklady AN SSR 118(4), 646–649 (1958)
52. Trakhtenbrot, B.A.: The asymptotic estimate of the logical nets with memory. Doklady AN SSR 127(2), 281–284 (1959)
53. Trakhtenbrot, B.A.: Some constructions in the monadic predicate calculus. Doklady AN SSR 138(2), 320–321 (1961)
54. Trakhtenbrot, B.A.: Finite automata and the monadic predicate calculus. Doklady AN SSR 140(2), 326–329 (1961)
55. Trakhtenbrot, B.A.: Finite automata and the monadic predicate calculus. Siberian Math. Journal 3(1), 103–131 (1962)
56. Trakhtenbrot, B.A.: On the frequency computation of recursive functions. Algebra i Logika, Novosibirsk 1(1), 25–32 (1963)
57. Trakhtenbrot, B.A.: On the complexity of schemas that realize many-parametric families of operators. Problemy Kibernetiki 12, 99–112 (1964)
58. Trakhtenbrot, B.A.: Turing Computations with Logarithmic Delay. Algebra i Logika, Novosibirsk 3(4), 33–48 (1964)
59. Trakhtenbrot, B.A.: Optimal computations and the frequency phenomena of Yablonski. Algebra i Logika, Novosibirsk 4(5), 79–93 (1965)
60. Trakhtenbrot, B.A.: On normalized signalizing functions for Turing computations. Algebra i Logika, Novosibirsk 5(6), 61–70 (1966)
61. Trakhtenbrot, B.A.: The Complexity of Algorithms and Computations, Lecture Notes, ed. by Novosibirsk University, pp. 1–258 (1967)
62. Trakhtenbrot, B.A.: On the complexity of the mutual-reduction algorithms in the construction of Novikov and Boone. Algebra i Logika 8, 50–71 (1969)
63. Trakhtenbrot, B.A., Barzdin, J.M.: Finite Automata (Behavior and Synthesis), Nauka, Moscow, pp. 1–400 (1970) English translation in: Fundamental Studies in Computer Science 1, North-Holland (1973)
64. Trakhtenbrot, B.: On autoreducibility. Doklady AN SSR 192(6), 1224–1227 (1970)
65. Trakhtenbrot, B.A.: Notes on the complexity of probabilistic machine computations. In: Theory of Algorithms and Mathematical Logic, ed. by the Computing Center of the Academy of Sciences, pp. 159–176 (1974)
66. Trakhtenbrot, B.A.: On denotational semantics and axiomatization of partial correctness for languages with procedures as parameters and with aliasing (extended abstract), Technical Report, Tel Aviv University, p. 20 (August 1981)
67. Trakhtenbrot, B.A.: A survey of Russian approaches to perebor (brute-force search) algorithms. Annals of the History of Computing 6(4), 384–400 (1984)
68. Trakhtenbrot, B.A.: Selected Developments in Soviet Mathematical Cybernetics, Monograph Series, sponsored by Delphic Associates, Washington, XIV + 122 pages (1985)

69. Trakhtenbrot, B.A.: Comparing the Church and Turing approaches: two prophetical messages. In: The Turing Universal Machine – A Half Century Survey, pp. 603–630. Oxford University Press, Oxford (1988)
70. Trakhtenbrot, B.A.: Origins and metamorphoses of the Trinity: logic, nets, automata. In: Proc. of the 10th IEEE Symposium on LICS, San Diego (1995)
71. Trakhtenbrot, B.A.: On the power of compositional proofs. Fundamenta Informaticae 30(1), 83–95 (1997)
72. Trakhtenbrot, B.A.: In memory of S.A. Yanovskaya (1896–1966) on the centenary of her birth. Modern Logic 7(2), 160–187 (1997)
73. Trakhtenbrot, B.A.(ed.): Special Issue on Continuous-Time Paradigms in Logic and Automata. Fundamenta Informaticae 62(1) (August 2004)
74. Trakhtenbrot, B.A.: Understanding Basic Automata Theory in the Continuous Time Setting. Fundamenta Informaticae 62(1), 69–121 (2004)
75. Trakhtenbrot, B.A.: In memory of Andrei P. Ershov. In: Ershov, a Scientist and Human Being (in Russian), Publishing House of the Russian Academy of Sciences in Novosibirsk (2006)
76. Trakhtenbrot, B.A., Halpern, J., Meyer, A.R.: From denotational to operational and axiomatic semantics for Algol-like languages (an overview). In: Clarke, E., Kozen, D. (eds.) Logic of Programs 1983. LNCS, vol. 164, pp. 474–500. Springer, Heidelberg (1984)
77. Yablonski, S.V.: Algorithmic difficulties in the synthesis of minimal contact networks (in Russian), Problems of Cybernetics, vol. 2, Moscow (1959)
78. Yanovskaya, S.: Mathematical logic and fundamentals of mathematics. In: Mathematics in the USSR for 40 Years, Moscow, Fizmatgiz, pp. 13–120 (1959)