1-2022

Russian logics and the culture of impossible: Part II: Reinterpreting algorithmic rationality

Ksenia TATARCHENKO
Singapore Management University, ktatarchenko@smu.edu.sg

Anya Yermakova

Liesbeth De Mol

Citation

TATARCHENKO, Ksenia, Yermakova, Anya, & De Mol, Liesbeth.(2022). Russian logics and the culture of impossible: Part II: Reinterpreting algorithmic rationality. *IEEE Annals of the History of Computing, 43(4)*, 57-69.

Available at: https://ink.library.smu.edu.sg/soss_research/3544

Published in IEEE Annals of the History of Computing (2021), 43 (4), 57-69. DOI: 10.1109/MAHC.2021.3135714

Russian Logics and the Culture of Impossible:

PART II, Reinterpreting Algorithmic Rationality[1]

Ksenia Tatarchenko
Anya Yermakova
Liesbeth De Mol

Keywords: algorithm, logic, cybernetics, Soviet Union, Cold War, Vladimir Uspensky

Part II: Reinterpreting Algorithmic Rationality

In this second part of the paper we study disciplinary, social, and pedagogical practices related to the algorithm across the late Soviet period to argue against a reductive, US-centric definition of Cold War "algorithmic rationality" as a diminishment of human agency.[2] Rather, we see the new technological development, computerization, in relation to the evolving intellectual traditions in logic and mathematics before the Revolution. The sweeping scope of *poznaniye* within the debates about Russian modernity helps explain its resilience as instantiated in the intellectual connections between the pre-revolutionary debates on the LEM and the post-war theories of algorithm. From this perspective, what mattered first was not always efficiency and utility, but the tinkering with mathematical objects as powerful methodological carriers for communicability between the abstract and the concrete. Although articulated in new discursive forms that Slava Gerovitch called "cyberspeak," the algorithm is akin to the LEM-defying logics and to expansions and discussions of the Jevons machine, in its functions as a nexus between the outer and inner human worlds and as a tool for exploring impossible scenarios, within and outside of the scope of "the mathematical."[3,4] Moreover, Methodology, the practical counterpart of *poznaniye*, sheds light on how algorithms came to be understood as objects that bring together human capacities towards action as much as abstraction, an understanding which was accompanied by a distinct approach to programming education and computer literacy.

Unlike the synthetic overview centered on ideas in Part I, here we choose to focus on the life and work of the logician and mathematician Vladimir Uspensky (1930-2018). This choice is motivated by both historiographic and pragmatic reasons. Synthetic narratives about the late USSR are skewed toward explaining its collapse, the teleology sustaining the misbelief that planned economy was incompatible with the Information Age.[5] We examine how the future-oriented ambitions of the late Soviet scientific intelligentsia transcend the so-called "computer gap," and what happened to their human-centric vision of the techno-political modernity after 1991. Uspensky's multivolume recollections provide significant insights into these questions.[6] Belonging to the generation at the heart of the intellectual and institutional

[1] This is a draft version of part 2 of a paper to be published in the special issue "Logic, programming and the shaping of computer science: National and Local Perspectives" of the *IEEE Annals for the History of Computing*

[2] See Introduction to this paper for a historiographic discussion on "algorithmic rationality."

[3] S. Gerovitch, *From Newspeak to Cyberspeak: A History of Soviet Cybernetics*, Cambridge, MA: MIT Press, 2002.

[4] See Part I for a discussion on the LEM and other logical-mathematical objects within the scope of *poznaniye*, or the process of knowledge-acquisition, where the epistemic focus is on gnoseological inseparability of subjectivity and objectivity, intuition and reason. See in particular the quote by N. Bugaev in Part I regarding mathematics' role in connecting the inner and outer worlds.

[5] The most influential account is in Manuel Castells, *End of Millennium. The Information Age: Economy, Society and Culture*, vol. III, Malden, MA; Oxford, UK: Blackwell, 1998, second edition, 2010.

[6] V. Uspensky, *Trudy po nematematike* [Works in Non-mathematics], 5 vols., Moscow: OGI, 2018.

consolidation of mathematical logic in the 1950s, he was particularly active in these developments. His role is best described not in terms of an administrative position -- the scientific secretary of the chair for mathematical logic for many decades, before taking the leadership in 1993 -- but as a node in multiple networks ranging from programming to, most notably, mathematical linguistics.[7] Finally, his humanistic vision of logic and mathematics came hand in hand with strong pedagogical commitments that, through his pupils and younger collaborators, we trace into contemporary reflections on programming and computer literacy in Russia and beyond.

Following Uspensky's career simultaneously reveals the continuity in the mediating role of logical investigation since the previous century as well as the new institutional and academic hierarchies that accompanied discipline formation. This part starts by considering the algorithm at the nexus of the disciplinary legitimacy exchanges from the 1950s to the 1980s as transpiring in the relationship that Uspensky maintained with Andrei Ershov (1931-1988), one of the first professionally trained programmers in the Soviet Union and an instigator of the 1985 computer education reform.[8] We will then offer a close reading of Uspensky's 1979 publication, which introduces Emil Post's formalism as a means to program with paper tools in schools, as a realization of an educational ideal of algorithmic mindedness shared among logicians and computer professionals. We also trace the persistent influences of the late Soviet approach to computer literacy in the post-1991 international setting, in particular under the UNESCO framework.

II. 1. Establishing Interdisciplinary Dynamics

A handwritten note left by the Soviet programmer Andrei Ershov on May 29, 1981, encapsulates the entanglement between personal and professional lives underlying the transformative vision of Vladimir Uspensky's "culture of the impossible." The note appears on the back side of an impassioned telegram received on the occasion of Ershov's 50th birthday:

> Unsuccessfully calling you the entire morning to wish you happy birthday and to express the feelings of affection, respect and admiration. Wishing you many long years of health and myself many years of collaboration with you. Lovingly, Uspensky.[9]

In this note addressed to his mother, Ershov also mentions Uspensky's professional identity -- a logician -- coupled with an epithet of "intelligentnyi," an adjective which simultaneously described a personal virtue of politeness and signaled a membership among Soviet intelligentsia. Ershov also added that the personal sympathy was mutual and revealed emotional support for Uspensky, who lost his spouse.[10] What's more, Ershov explained the development of their friendship: they first met at Moscow university in the 1950s, when he was a student, but the strong bond between them formed during the preparation for the 1979 international symposium on "Algorithms in Modern Mathematics," hosted by Ershov in Urgench, Uzbekistan.

Correspondence between the two scientists indeed demonstrates that personal ties came hand in hand with intellectual exchange. The 1979 international encounter mentioned above became a watershed moment and an intellectual catalyst for Uspensky. In the words of Uspensky's colleagues and pupils, Ershov's request to contribute a keynote on the theory of algorithms became "the cause to realize his own role as a synthetic thinker [osmyslitel']."[11] Uspensky took up the challenge of presenting an encompassing

[7] For a classical work focusing on personal networks across physics and biology, see: M. Adams, *Networks in Action: The Khrushchev Era, the Cold War, and the Transformation of Soviet Science*, Trondheim: Trondheim Studies on East European Cultures and Societies, 2000.

[8] For the notion of legitimacy echanges, see: Geoffrey Bowker, *Memory Practices in Sciences*, Cambridge, MA: MIT Press, 2005.

[9] Ershov Archive (EA): V. A. Uspensky to A. P. Ershov, 19.04.1981, f. 197. l. 18.

[10] The full version of the comments is quoted in A. Marchuk, ed., *A. P. Ershov — uchenyi i chelovek* [A. P. Ershov — *a scientists and a man*], Novosibirsk: Iz-vo SORAN, 2006, p. 488

[11] A. Shen, et al., "Vladimir Andreevich Uspensky (27.11.1930-27.06.2018)," *Russian Mathematical Surveys*, vol.

historical perspective on the theory of algorithms to an audience of the most distinguished practitioners and theoreticians, including one of the American doyens of mathematical logic, Stephen Kleene (1909-1994).[12] The way in which Uspensky described his concerns and solicited Ershov's input regarding how to engage the work of other participants in his keynote reveals a shared system of intellectual and social values. "I will try to incorporate any such guidance with maximum efforts," stressed Uspensky, "because I suppose that we diverge in neither our evaluation, nor in the understanding of the politeness decorum, nor in other such things."[13]

The blurring of personal and professional spheres as illustrated above is not an anecdotal or accidental feature of Uspensky-Ershov interactions, but, as we demonstrate in this section, a key to elucidating the dynamic links between mathematical logic, cybernetics, programming, and computer literacy in the later Soviet context. In other words, not only the intellectual content of transdisciplinary *poznaniye* but the very forms of intelligentsia sociability held on from pre-revolutionary times, the continuity that sheds light both on the late Soviet interdisciplinary interactions and the commitment to serving society expressed in a symbiotic relation between research and teaching agendas.[14]

Notwithstanding the affection demonstrated by the documents from the early 1980s, it was the academic hierarchies that structured early encounters of the two scientists in the 1950s. Although Ershov first met Uspensky while attending his seminar on algorithms and computability and they continued to intersect for several years in a number of Moscow's intellectual venues, their respective positions were not comparable. The nature of this difference illuminates disciplinary developments which framed a shared set of attitudes and pedagogical commitments toward algorithm as a way of thinking.

Definitive of Uspensky's identity as a mathematician was his role as a student of Andrei Kolmogorov, who determined Uspensky's intellectual profile for years. Kolmogorov suggested his area of research and supervised his 1952 diploma work titled "General Definition of Algorithmic Calculability and Algorithmic Reducibility."[15] The same theme was further developed in Uspensky's 1954 dissertation devoted to recursive functions and in his 1963 doctoral dissertation. In 1958, Uspensky co-authored with Kolmogorov the seminal paper on algorithms and would continue working on Kolmogorov's notion of complexity, unifying the theory of probability and the theory of algorithms, into the twenty-first century.[16] Uspensky's intellectual trajectory was deeply enmeshed with Kolmogorov's patronage networks and a disciplinary consolidation of mathematical logic in the context of the changing perception of cybernetics from a bourgeois pseudoscience to a meta-discipline at the service of communism.

Uspensky's early career helps in understanding the evolving status of mathematical logic in the 1950s. Both Uspensky's diploma and dissertation for the candidate of the mathematical sciences degree were conducted with Yanovskaya as chair in the history of mathematics. This was also the institutional location of the seminar on the theory of algorithms taught by Uspensky and attended by Ershov. As there were too few students attached to that non-prestigious section, however, most of Uspensky's teaching

74, no. 4, 2019, pp. 165-180, 170.

[12] Soviet mathematicians had a close knowledge of Kleene's famous *Introduction to Metamathematics* thanks to Uspensky's edition of its translation: S. Kleene, *Vvedenie v metamatematiku*, Moscow: Iz-vo inostrannoi literatury, 1957.

[13] EA: Uspensky to Ershov, 28.08.1979, f. 225, l. 464-465.

[14] One can argue that Uspensky and Ershov both embodied the pre-revolutionary intellectual ambitions of synthesizing various disciplines, as well as enacting a synthetic approach towards pedagogy and research. See Section 1 of Part I for more.

[15] The diploma thesis is available at http://lpcs.math.msu.su/~uspensky/bib/Uspensky_1952_Diploma.pdf. On the choice to work on recursive functions, see: "Moe uchinichestvo u Kolmogorova" [My studies with Kolmogorov], *Trudy po nematematike*, vol. 5, Moscow: OGI, 2018, pp. 116-152.

[16] A. N. Kolmogorov, V. A. Uspensky, "K opredeleniiu algoritma," *UMN* [Russian Mathematical Surveys], vol. 13, no. 4, 1958, pp. 3-28; the text was translated in English as "On the Definition of an Algorithm," *Amer. Math. Soc. Transl.*, vol. 29, no. 2, 1963, pp. 217-245. Also see: A. N. Kolmogrov, V. A. Uspensky, "Algoritmy i sluchainost'," *Teoriia veroiatnosti i ee primeneniia* [Probability theory and its applications], vol. 32, no. 2, 1987, pp. 425-455; the text was translated in English as "Algorithms and Randomness," both versions appearing in the same journal.

obligations upon finishing graduate school were at the chair for Analysis which was responsible for offering basic mathematics to the entire university. So when Uspensky took on a self-assigned mission to expand the methods of mathematics as tools of knowledge-making to other fields, such as by teaching mathematics to students of literature, his efforts reflected a search for intellectual and pedagogical opportunities. His linguistic interests first materialized in 1956, when he and the linguist V. V. Ivanov co-founded the seminar on mathematical linguistics at the department of Philology. These intellectual opportunities and the institutional support that they received in the late 1950s reflected the atmosphere of elation in relation to the changing status of cybernetics.[17]

1954, a year when Uspensky defended his dissertation and conducted the seminar on the theory of algorithm, seemed to be the moment of the official condemnation of cybernetics. It was defined as a pseudoscience and a form of metaphysical mechanism on the pages of the *Short Philosophical Dictionary*.[18] But the situation was drastically reversed already in 1955, when the existence of Soviet computers was announced abroad and at home. In the same year, several publications defended cybernetics as a body of theories including information theory, automatic control, and self organizing logical processes, naming mathematical logic among its tools. The campaign in defense of cybernetics was well prepared and included many members of the mathematics department where both Uspensky and Ershov were localized. Moreover, the drastic shift in public discourse on cybernetics around 1955 did not only result from a negative process of eliminating human-machine analogies, which underlined British and American cybernetic discourse and provoked harsh ideological criticism in the early 1950s. Most importantly for our analytical focus is that the cybernetic vogue came hand in hand with a forward-looking conceptualization of the algorithm as a computer program and as a site for developing particular kinds of human-machine interactions.

But if the preoccupations of mathematical logic and programming intersected even before the public recognition of cybernetics, the theory of algorithms also became a point of contention. The cybernetic discourse facilitated the role of the algorithm as a circulating mathematical object, but interest in algorithms among mathematical logicians and programmers did not amount to sameness of professional practices and ideals, namely, the logician's ideal of generality and standardisation versus the programmer's ideal of a detailed and handy description.

Although today Alexei Lyapunov is best known for his public defense of cybernetics, his lasting intellectual legacy was sustained by the cohort of Moscow State University students, the first Soviet professional programmers, he trained, including Ershov. Lyapunov first formulated the theory of program schemata, which anachronistically could be described as a protoversion of a high-level language, when reading his pioneering course on "Principles of Programming" in 1952-53. Practical considerations regarding optimizing programs and mechanizing coding routines led Lyapunov to reconsider the notion of algorithm.[19] Whereas formalizations he was familiar with, such as Markov's normal algorithms, were oriented at elucidating computability, Lyapunov was training human operators for the new mathematical machines and was interested in operational aspects of algorithms. That labor considerations were paramount in relation to mathematical formalism is best illustrated with the issue of equivalent transformations of programs indispensable for programming programs (or compilers): to demonstrate equivalency one depended on a rigorous definition of program, its structure and functions. Working

[17] This seminar could be simultaneously read as a pragmatic move in time of the machine translation boom, as a project to revive the affinity between mathematics and linguistics spearheaded by Russian formalists, or as a personal passion rooted in family connections to Moscow's literary milieu. For a complementary perspective, see: V. V. Ivanov, "Iz proshlogo semiotiki, strukturnoi lingvistiki, i poetiki" [From the past of semiotics, structural linguistics, and poetics], *Ocherki istorii informatiki v Rossii* [Essays on the history of informatics in Russia], edited by D. A. Pospelov, and Ia. I. Fet, Novosibirsk: NITS SORAN, 1998, pp. 310–341.

[18] "Kibernetika," *Kratkii filosofskii slovar'* [Short Philosophical Dictionary], Moscow: Gos. iz-vo polit. lit., 1954, pp. 236-237.

[19] R. I. Podlovchenko, "A. A. Lyapunov i A. P. Ershov v teorii skhem programm i razvitii ee logicheskikh kontseptsii" [A. A. Lyapunov and A. P. Ershov on the theory of program schemata and its logical concepts], *A. P. Ershov — uchenyi i chelovek*, pp. 166-184.

within this larger agenda, in the early 1960s Ershov published a formalized definition of program as a universal operator algorithm realizing all computable functions.[20]

The tension in the approaches to the algorithm animated conversation between logicians and programmers of the late 1950s, such as those taking place at the 3rd All-Union Conference on Mathematics, where programming appeared under the section on mathematical logic. According to Trakhtenbrot, at this point, programming was generally understood as an applied theory of algorithms.[21] Encompassing the notion of program within the theory of algorithms brough advantages to both groups: on the one hand, it propelled the importance of mathematical logic, and, on the other hand, it validated a technical application. Yet, it came with a price, namely, institutional and personal hierarchies of the Soviet academic community.

These hierarchies come to light when comparing respective positions of a young programmer, such as Ershov, and a young logician -- Uspensky. Unlike exchanging ideas at seminars and conferences, the validation of work based on programming practice proved to be imbued with tensions that, in the case of Ershov, resulted in significant delays in obtaining an academic degree. The documents that trace an aborted attempt to obtain Markov's presence on Ershov's dissertation committee indicate that the prominent logician avoided the transfer of authority that his support would entail.[22] Significantly, the dissertation was prepared in 1958 as "Questions of the theory of algorithms in programming," the title signaling its theoretical ambitions, but was defended only in 1962 as "Operator algorithms" and benefited from the international exposure of Erhsov's English-language publications.[23] These disciplinary hierarchies were not limited to a single institutional setting but embodied in personal networks underlying a variety of professional activities.

Uspensky's identity as a mathematical logician is insufficient for understanding his successful navigation of the complex landscape of the Soviet scientific organization and the notoriously bewildering institutional alliances conducted in the name of cybernetics.[24] A few points of evidence are sufficient to indicate how Uspensky benefited from patronage networks centered on Kolmogorov. The young logician not only produced an overview of foreign literature on cybernetics for Kolmogorov's public speeches, but also directly participated in elaborating Soviet public discourse with an article on Norbert Wiener in the *Great Soviet Encyclopedia* alongside Kolmogorov's famous article on cybernetics.[25] As the editor of the

[20] A. A. Lyapunov, "O logicheskikh skhemakh programm" [On logical program schemata], *Problemy kibernetiki* [Problems of cybernetics], vol. 1, 1958, 46-74. The language is best known as Yanov program schemata under the name of one of Lyapunov's students, see: Iu. I. Yanov, "On the equivalence and transformation of program schemes," *Communications of the ACM*, vol. 1, no. 10, 1958, pp. 8-12. For Ershov's publications on operator algorithms, see: A. P. Ershov, "Operatornye algoritmy. 1. Osnovnye poniatiia" [Operator algorithms. 1. Basic concepts], *Problemy kibernetiki*, vol. 3, 1960, pp. 5-48; and "Operatornye algortmy. 2. Opisanie osnovnykh konstryktsii programmirovaniia" [Operator algorithms. 2. Descriptions of programming constructions], *Problemy kibernetiki*, vol.8, 1962, pp. 211-233.

[21] For recollections of such perceptions, see: B. Trankhtenbrot, "Pamiati A. P. Ershova" [In the memory of A. P. Ershov], *A. P. Ershov — uchenyi i chelovek*, pp. 343-352, 344. Also, see B. Trakhtenbrot, "From Logic to Theoretical Computer Science," *People and Ideas in Theoretical Computer Science*, edited by C. S. Calude, Singapore: Springer, 1998, pp. 314-341. Also see: *Trudy Tret'ego Vsesoiuzn. matemeticheskogo s'ezda, Moskva, iun' - iul' 1956* [Proceeding of the third all-Union Conference on mathematics, June-July 1956], Moscow: Iz-vo AN SSSR, 1956. Note Uspensky's participation in the editorial committee.

[22] For details, see the study of Ershov career: I. A. Kraineva, N. A. Cheremnykh, *Put' programmista* [Programmer's path], Novosibirsk: Nonparel', 2011.

[23] For additional context see: K. Tatarchenko, "'The Anatomy of an Encounter:' Transnational Mediation and Discipline Building in Cold War Computer Science," in *Communities of Computing: Computer Science and Society in the ACM*, edited by Tom Misa, New York: ACM Books and Morgan and Claypool, 2016, 199-227.

[24] For cybernetics and mathematical economics, see A. Leeds, "Dreams in Cybernetic Fugue: Cold War Technoscience, the Intelligentsia, and the Birth of Soviet Mathematical Economics," *Historical Studies in the Natural Sciences*, vol. 46, no. 5, 2016, pp. 633-668.

[25] These definitions appeared in the additional volume of the encyclopedia: A. Kolmogorov, "Kibernetika" [Cybernetics], *BSE* [Great Soviet Encyclopedia], vol. 51, Moscow: BSE, 1958, pp. 149-151; and V. V. Ivanov, M. K.

mathematics section of the prestigious foreign literature press, Kolmogorov's influence was not limited to producing Soviet definitions.[26] In 1959, Uspensky acted as an editor of the translation of Ross Ashby's 1957 *An Introduction to Cybernetics*.[27] With a preface by Kolmogorov, this Russian language edition was among the highlights of the turn from anti-cybernetics campaign of the early 1950s to the cybernetic craze of the closing years of the decade.

In this context, it is not surprising to read in Uspensky's recollections of his close personal ties not to Ershov, but rather to Ershov's teacher. Uspensky initiated an inscription of the 1959 translated volume to Lyapunov as a "father of Soviet cybernetics," a dedication that reflected Uspensky's integration into cybernetic circles.[28] The integration took place formally — at the cybernetic council headed by admiral A. I. Berg, as well as informally — at the gatherings at Lyapunovs' flat reproducing the traditional forms of pre-revolutionary intelligentsia lifestyle. For instance, Uspensky recalled how Igor Poletaev, a military radio engineer and the author of the first Soviet popular account of cybernetics, demonstratively carried a "hidden" volume of Tsvetaeva's poetry in his briefcase as an intended "sign of belonging" [znak priobscchennosti].[29] The episode parallels those familiar from before the Revolution, such as Nikolai Vasiliev carrying Symbolist poetry.[30]

The intellectual and social hierarchy, illustrated by the differences between the early careers of Ershov and Uspensky, would not determine the relations between mathematical logic and programming in the long term, however. The new types of institutions, the Computer Centers, were created all over the country; with ever rising demand on new specialists, programming was becoming a mass profession entitled for its own body of theoretical knowledge. By the late 1960s, Ershov became the main Soviet spokesperson for the creation of theoretical programming, a new discipline closely connected to but independent from both cybernetics and mathematical logic. Ershov also liked to describe it as a Soviet equivalent to the American computer science.[31] Despite limited institutional successes, an official recognition of this agenda was Ershov's election to the Academy of Sciences as a corresponding member in 1970.[32] In this changing context, the role of the algorithm -- still central to the dialogue between programming and mathematical logic -- was to evolve.

This evolution was not limited to national space but took place on an international scene. In the early 1970s, Ershov depicted programmers' work as operating at the cutting edge of knowledge making, enabling the transformation of symbols into the real world (from word to action), and thus foundational for the dual power of programming to transform individuals and society. The distinct Russian and Soviet cultural dimensions shaping Ershov's professional ideas, that we can trace to the nineteenth century, did not diminish their attractiveness to his Western interlocutors.[33] On the contrary, well-versed in cultural idioms from both sides of the Iron Curtain, Ershov's messages struck a chord among visionary proponents of computer science such as Donald Knuth, the recipient of the 1974 ACM A.M. Turing Award for his

Polivanov, and V. A. Uspensky, "Wiener, Norbert," *BSE*, vol. 51, p. 59 (in Russian).

[26] Uspensky also acted as a mediator recommending foreign books and potential editors to Kolmogorov: EA: Uspensky to Ershov, 29.04.1959, f. 101, l. 264-5.

[27] W. Ashby, *An Introduction to Cybernetics* (New York: John Wiley & Sons, 1957). W. Ashby, *Vvedenie v kibernetiku*, Moscow: Iz-vo inostrannoi literatury, 1959.

[28] Uspensky, *Trudy po nematemetike*, p. 308.

[29] Ibid., p. 237, p. 246. Also see: I. A. Poletaev, *Signal* [Signal], Moscow: Sovetskoe radio, 1958. Compare with earlier social practices of Moscow mathematicians described in L. Graham and J.-M. Kantor, *Naming Infinity*.

[30] Vasiliev was known to carry poetry of Andrey Bely. See V. A. Bazhanov, N.A. *Vasil'ev i Ego Voobrazhaemaia Logika: Voskreshenie Odnoi Zabytoi Idei* [N.A. Vasil'ev and His Imaginary Logic: Revival of One Forgotten Idea] (Kanon+, Reabilitatsiia, 2009).

[31] A. P. Ershov, A. A. Lyapunov, "O formalizatsii poniatiia programma" [On formalization of the program as a concept], *Kibernetika* [Cybernetics], vol. 5, 1967, pp. 40-57. On Ershov as an international mediator, see Tatarchenko, "The Anatomy of an Encounter."

[32] EA: Uspensky to Ershov, 01.12.1970, f. 91, l. 212. Uspensky's note retraces the geography of their encounters.

[33] In particular, see: A. P. Ershov, "Aesthetics and the Human Factor in Programming," *Communications of the ACM*, vol. 15, no. 7, 1972, pp. 501-505.

major contributions to the analysis of algorithms and the design of programming languages. Together, Knuth and Ershov conceived of a gathering and a pilgrimage to the alleged birthplace of the medieval Arabic scholar who gave his name to "algorithm," Al Khwarizmi. In this meeting, held in 1979, the overview of the theory of algorithm and its applications was jointly presented by Uspensky and a young logician Aleksei Semenov (1950-), took center stage as a joint effort of speakers and organizers to offer perspectives from both disciplines.[34] These mutual efforts at an interdisciplinary dialogue grew into personal empathy as Uspensky and Erhov maintained a systematic exchange on many professional issues, from publication opportunities to issues of mathematical linguistics, to politics of degree attributions, to programming education.[35]

A programmatic document -- dating from 1983, four years after the 1979 conference -- articulates the interdependent relationship between computing and mathematical logic and demonstrates how an alliance based on personal trust and shared professional aspirations could be leveraged to influence scientific and technical policy. The report "Mathematical questions of the creation of new generation computers and the tasks of the mathematical training of cadres" appeared at the 1983 All-Union Meeting on System Programming under the signatures of A. A. Kolmogorov, V. A. Uspenskiy and A. L. Semenov.[36] As a member of the program committee, Ershov may well have been the mastermind behind the appearance of the report devoted to mathematical logic right after his own talk on the "Perspectives of Software Development Under the Twelfth Year Plan."[37] The format of the meeting, sponsored by the State Committee on Science and Technology, signaled its role as a platform for coordination between experts and policy makers. The resulting statements had the potential to influence organizational, not only intellectual, outcomes.[38]

In the report, the authors (most probably Uspensky and Semenov as Kolmogorov was already gravely ill) articulated mathematical questions of computing as a two-fold dynamic grounded in the nature of mathematical logic — a science of correlations between expressions and their semantics:

> By the act of realizing a program, the computer in fact determines its meaning, its semantics. In its turn, the structure of the computer has to be described in some language [of mathematical logic]. Therefore, the main subject of mathematical logic turns out to be equivalent to the defining relationship [between software and hardware] which is foundational to both the construction and the usage of the computer.[39]

This claim was supported on two rhetorical levels: historical and comparative.

The report's historical narrative was structured by Marxist dialectics, namely as an account of mutual influence of theory and practice, in which applications of computational technology shaped the theoretical development of mathematical logic, and the methods and apparatus of mathematical logic were, in turn, fruitfully used in the design of computational systems. Its key point was to avoid a narrow understanding of the relationship between computation and logic, conducive to a situation where only the most immediately practically relevant direction of theoretical research was to be supported. The comparative section gave the report a sense of urgency: it juxtaposed the Soviet and Western

[34] V. A. Uspensky, A. L. Semenov, "What Are the Gains of the Theory of Algorithms," *Algorithms in Modern Mathematics and Computer Science*, edited by A. Ershov and D. Knuth, Berlin, Heidelberg: Springer, 1981, pp. 100-234.

[35] The political aspect reached its apex by 1987, when Ershov used his connection with E. P. Velikhov, the powerful head of the branch for computing and automation, to leverage the support of the Academy of Sciences to back up collaborations with linguists. EA: A. P. Ershov to E. P. Velikhov, 09.03.1987, f. 455. l. 207-210.

[36] EA: A. A. Kolmogorov et al., "Matematicheskie voprosy sozdaniia EVM novogo pokoleniia i zadachi matematicheskii podgotovki kadrov," 15.11.83, f. 181, l. 349-352.

[37] EA: Rabochaia programma Vsesoiuznogo sovetschaniia po sistemam programmirovaniia EVM, 15-17.11.1983, f. 181, l. 334-35.

[38] EA: Proekt reshenia Vsesoiuznogo sovetschaniia po sistemam programmirovaniia EVM, 15-17.11.1983, f. 181, l. 333.

[39] EA: Kolmogorov et al., "Matematicheskie voprosy sozdaniia EVM novogo pokoleniia i zadachi matematicheskii podgotovki kadrov," 15. 11. 83, f. 181, l. 350.

developments in several subfields and highlighted the danger of decline from the dominant position that Soviet logic occupied in the recent past. The decline narrative led to pragmatic conclusions: to expand the research area of mathematical logic in the Academy of Sciences and in institutions of higher learning, and to increase the level and quality of training in mathematical logic of software and hardware developers. The overall preoccupation of the authors to upkeep the disciplinary vigor of mathematical logic by orienting policy measures related to programming was transparent and demonstrated that the arrow of interest in the alliance between the two communities, dating from the 1950s, had reversed.

Although invoking the interdependency of two fields and the authority of mathematical knowledge, the report did not resolve but sidelined the tension in the respective claims on the notion of the algorithm by the two communities. Instead, it made apparent the broader value system connecting logicians and programmers that was implicit already in the title of the report. It was a question of training, best understood if interpreted in light of the pre-revolutionary developments in Methodology. Training here should not be confused with a notion of acquiring a set of narrow skills, but seen instead as a process of formation of an ideal type of mind. The ideal embraced by Ershov, and in his broader professional community, was that of a computer specialist working at the edge of what is knowable, echoing the pre-revolutionary *poznaniye* discourse. The algorithmic thinking advocated by Ershov as the capacity to navigate between action and abstraction, in turn, reflected Uspensky's understanding of algorithms as mathematically undefinable. In other words, both scientists saw algorithms and programs as mathematical objects enabling mediation between the inner world of the mind and a purposeful transformation of the environment, much like Nikolai Bugaev's commentary about the role of mathematics back in the nineteenth century. This central place of algorithmic mindedness in both mathematical logic and computing reflected the intertwined evolution of two communities as demonstrated in the career trajectories of Ershov and Uspensky and forms a basis for analysing their profound engagement with pedagogy.

II. 2. Tracing Circulation: Teaching Programming through Algorithmics

Uspensky's theoretical work on algorithms, which aimed for a more general definition of algorithm than that provided by Post, Turing, Markov and Church, echoes the idea of logic as a means to reflect on the limit of knowledge acquisition which cannot be defined in advance.[40] But an even stronger continuity with the pre-revolutionary *poznaniye* is found in a shared vision of an individual mind as a site of improvement and in the fusion of research and pedagogical questions. This becomes clearer when looking into a booklet by Uspensky, published in 1979 which introduces Emil Post's formalism as a means of programming with paper tools for all children from preschool on.

The preface of Uspensky's *Post's Machine* opened with a direct indication of the book's goals and intended audience:

> This booklet is intended first of all for schoolchildren. [...] The book deals with a certain "toy" ("abstract" in scientific terms) computing machine – the so-called Post machine – in which calculations involve many important features inherent in the computations on real electronic computers.[41]

Expecting no knowledge of mathematics beyond arithmetic, Uspensky leveraged the playfulness implicit in the notion of the "toy" to engage his reader into the seriousness of mathematical abstraction and the realities of computerization. From the outset it is clear that the mathematician considered the abstract model to be immediately relevant to actual practices of programming. Programming, defined as "*preparing a program for the machine leading to the given goal*," is then represented as a general problem which does not change fundamentally when moving from abstract to real machines.[42]

[40] According to Kolmogorov and Uspensky, they failed in the attempt for a more general definition though. See: A. N. Kolmogorov, V. A. Uspensky, "On the Definition of an Algorithm.", footnote 1.

[41] V. A. Uspensky, *Post's Machine*, Mir Publishers: Little Mathematics Library, translated by R. Alavina, 1983, p. 7. The original Russian publication dates from 1979.

[42] Ibid., p. 23, emphasis ours.

Based on that supposition, Uspensky derives and analyses a number of quite basic programming features (e.g. the insight that different programs lead to the same goal; the idea of reusing existing programs; finding the shortest program; etc). By chapter 2, the reader is introduced to issues of equivalency, complexity, and program length. Chapter 3 describes the analysis of functioning programs as well as composition of new programs using the block diagrams. Chapter 4 returns to the question of similarities and differences between abstract and electronic machines to conclude that the Post machine is a simplified model of a real computer. This apparent pragmatic orientation of the text is, however, but one of multiple layers. Similarly to Trakhtenbrot's *Algorithms and Automatic Computing Machines*, an explicit engagement with younger audiences went hand-in-hand with an ambitious agenda combining educational, practical and theoretical concerns. *Post's Machine* contains an exposition of Uspensky's philosophy of learning and foundational reflections on the undefinedness of the notion of algorithm and certain impossible programs.

Addressed not to children but to educators, the "Methodological Notes" included in the text spell out a broader goal of training thinking habits. Uspensky argues that an introduction to the actual use of the Post machine as a computing device should be delayed until the execution of programs is grasped. He insists on the priority of training *poznaniye* itself from an early age:

> The capability of perceiving any system of concepts or any reasoning, in general (and regardless of) the purpose of the knowledge is obtained, i.e., before (and regardless of) any application seems one of the most important qualities which are trained by mathematical studies. Giving an idea of the goal you are after in presenting material [...] should not affect understanding which can and must proceed regardless of the goal. The ability to think formally is a special ability developing like every ability through training. This training can begin from an early age.[43]

In other words, one should first learn to think formally *with* the machine before moving to more concrete calculations and problems.

From the start, this ability to think formally is conceived as a performative one, akin to the turn of the century pedagogical experiments. Indeed, while the machine is imaginary, "existing only in our imagination," the students should work with it "as if existing." This is achieved by "performing" operations via its graphical representation, either "drawn in chalk on a blackboard or with the aid of paper tape and buttons or clips used as labels."[44] Crucially, this hands-on aspect of knowledge acquisition does not imply a need to have access to a particular computer -- formal thinking allows working with *any* concrete representation or materialization that enables the performative.

This operational or performative way of learning, namely, doing something without really knowing what you are doing, rooted in a basic incompleteness of *poznaniye* as a process, was not a unique feature of Uspensky. Rather, it belongs both to past pedagogical traditions, such as Shatunovsky's immersive lectures and public demonstrations of Jevons' machine, and to Uspensky's intellectual milieu. For instance, Uspensky's invocation of the power of the familiar is also found in Ershov's 1981 "Programming, the second literacy," the manifesto of the Soviet computer literacy campaign justifying its vision for universal programming skills based on the ubiquitous presence of algorithms, both in science and the real world.[45] Both use the same cultural reference to the character from Molière's *The Middle Class Gentleman*, Monsieur Jourdain, who, after 40 years discovered that he had always been speaking prose. Whether borrowed or found independently, the analogy to algorithms was obvious to the logician and the programmer alike. For Uspensky, it amounted to a realization that the language we have always been speaking in mathematics is that of algorithms.[46] Ershov, in turn, furthers the analogy to a more encompassing metaphor: "mankind [with the advent of computers] [...] has discovered that it lives in a

[43] Ibid., p. 21.

[44] Ibid., p. 19.

[45] A. Ershov, "Programming, the Second Literacy," in *Computer and Education: Proc. IFIP TC-3 3rd World Conference on Computer Education, WCCE 81*, Amsterdam: North Holland, 1981), 1–17; A. P. Ershov, *Programmirovanie – vtoraia gramostnot'*, Novosibirsk: VTS SO AN SSSR, 1981.

[46] Uspensky, *Post's Machine*, p. 68.

world of programs."[47]

Uspensky's conviction that Post's machine was well suited for such training of the mind, even among youngest children, was rooted intellectually in his understanding of algorithms, and socially in his life-long exposure to specialised mathematical education. In the post-war years, Uspensky attended a mathematics circle organized by one of Kolmogorov's students, E. V. Dynkin, and as the winner of the tenth Moscow's mathematics olympiad (1946) he was recommended to skip a grade, thus entering university a year earlier. His return to the mathematical circle as its leader and further collaboration with Dynkin resulted in early publications based on the circle's teaching materials.[48] It is in this context that one can understand the publication history of the *Post's Machine* as a reflection of Uspensky's experiments with teaching programming via algorithmics to school children and to students of the mechanics, mathematics, and philology departments of Moscow university since 1961-1962. Although the first edition of the book appeared in Russian in 1979, it was based on a series of articles published in 1967 in the magazine devoted to methodology of teaching mathematics, *Mathematics in School*. Experimenting with different forms of presentation to a wide range of Soviet youth from mathematically gifted schoolchildren to pupils who feared the subject allowed Uspensky to claim that Post's work was particularly suitable for a general "introduction to the theory of algorithms."[49]

Uspensky's choice to anchor his framework in a model of computability provided by Emil L. Post rather than by Turing was deliberate. Uspensky specified that, although this model is similar to but less known than Turing's original model, its greater simplicity is preferable for grasping the notion of algorithm. Besides the fact that Uspensky considered the model to be better than Turing's when it concerned finding the most general model for algorithms (because of the presence of a halting instruction in Post's model),[50] it is especially the formal simplicity of the model which made it more suited for school children. Amongst others, contrary to Turing's model, tape cells have only two states: blank or marked; and each "line" of a program can only have one instruction plus a goto, in contrast to Turing's machine which has print, move and transfer-of-control all in one operation. Simplicity being a notoriously slippery idea, Uspensky's motives become more transparent as the book advances. Once students are made familiar with the basic workings of the Post machine and have intuitively grasped its operations via its graphical representations, the teacher can move to the next stage and consider (abstract) programs that do something with which the student is more familiar, that is, simple arithmetic operations.

Rather than giving a whole series of examples, Uspensky sticks to one very basic example, that is, addition and, mostly, just unary addition. His analysis and elaboration of this apparently simple example illustrates how he integrates both theoretical and practical issues within one and the same text.[51] Indeed, exposing this close connection between abstract formalism and real computers, is one of the main goals behind Uspensky's detailed analysis of the problem of unary addition, showing how Post's machine can be applied to real programming.[52] But where the Post machine and the basic example of unary addition are employed, first of all, to teach some basic programming features alongside abstract thinking, it is that very same discourse that facilitates the reflection on the basic limitations and potentialities of the Post machine and, ultimately, algorithms. By introducing more and more general cases of unary addition to move to the addition of numbers, the logician ends up at the limits of the Post machine by considering the addition of an arbitrary quantity of numbers arbitrarily far apart.

That problem is an impossible program for the Post machine and so cannot be programmed on

[47] Ershov, "Programming, the Second Literacy," p. 5.

[48] Uspensky was the first freshman to run the mathematical circle. This work resulted in a popular science volume, E. B Dynkin and A. V. Uspensky, *Matematicheskie besedy* [Mathematical Conversations], Moscow: GITTL, 1952. It was translated in German as *Mathematische Unterhaltungen. Aufgaben¨uber das Mehrfarbenproblem, aus der Zahlentheorie und der Wahrscheinlichkeitsrechnun*g, Cologne: Aulis Verlag Deubner & Co.KG, 1979.

[49] Uspensky, *Post's Machine*, p. 8.

[50] On generality, see: A. N. Kolmogorov, V. A. Uspensky, "K opredeleniiu algoritma."

[51] Note the continuity to Shchukarev's integrated theory-practice description of the Jevons machine described in Part I.

[52] Uspensky, *Post's Machine*, p. 23.

the Post machine or in general, at least, if one accepts what Uspensky calls "Post's proposal" and what is also known as Post's thesis: stated roughly, it is the idea that any general problem that can be computed in a finite number of steps can also be computed by the Post machine, or, to put it in Uspensky's phrasing, that there can be no algorithm that is not solvable by Post's machine. Uspensky then anchors the unprovability of this proposal, in the undefinability of the notion of algorithm itself: "This 'working hypothesis' … cannot be proved, at least with the idea underlying the word 'proof' common in mathematics, and not at all because it is untrue, but because the notion of algorithm involved in it is not 'mathematically' defined."[53] It is here that mathematics faces its own limitations in that the substantiation of Post's "working hypothesis" develops according to a path that Uspensky believes to be "more traditional for a naturalist than for a mathematician."[54] This understanding of algorithm is consistent and independent of whether Uspensky's audience included those with school curriculum background only or the founders of the field, such as during the 1979 conference. In Ugrench, Uspensky also described the algorithm as undefinable mathematically: unlike other mathematical concepts, it has a semantic meaning which is imperative and operative -- an algorithm needs to be performed.[55]

While it is hardly surprising to find a mathematician promoting a mathematical approach to programming, Uspensky's idea of teaching basic concepts in programming to children via a theoretical machine demonstrates how he integrated two apparently disconnected discourses. On the one hand, issues of computer literacy and, on the other, theoretical models of computability, or, in Uspensky's interpretation, algorithms, belonged together. Matters of the theory of algorithms thus did not respect the established disciplinary borders of the Soviet academy and subverted intellectual hierarchy typically overlooking children's minds, such as naturalized in the US-centered Cold War historiography.

The case of Uspensky is in perfect resonance with the Soviet and pre-Soviet contexts we have sketched in both parts of this paper and with a cultural construction of logic and mathematical logic as an essential capability for human development, not an exclusive academic occupation. This vision, enacted through systematic engagements in pedagogy among Russian and Soviet scholars, also found a policy realization in the computer education reforms spearheaded by Ershov in the mid-1980s. The goals of the reform were to teach algorithmic thinking as shaped by an imagination of the programmer as a person who had trained their cognitive abilities to distinguish and work around algorithmically unsolvable problems. Similarly, Uspensky's ambition of expanding the methods of mathematics to other fields, including to the human sciences, as well as to inculcate formal thinking early, from childhood on, was anchored in his conception of mathematics being a part of humanities. That is, mathematics is a way of cultivating one's own mind through (abstract) thinking, or in the words of a Soviet structural linguist, "[delineating] between the definable and indefinable, [and] between the deductive and the inductive"[56]

This Soviet approach to teaching programming gains from being put in relation to the "constructivist" approach to computational thinking one finds in the U.S. at around the same time.[57] Whereas logic and mathematics were mostly confined to academic computer science, the most important work on literacy at that time is connected to the name of Seymour Papert, who left South Africa for Cambridge and Geneva, where he became a protege of Jean Piaget, before moving to MIT to collaborate with Marvin Minsky in 1963. Papert applied Piaget's constructivist theory of learning in which one starts from the concrete and "objects-to-think-with" to acquire more abstract skills.[58] These ideas led to the

[53] Uspensky, *Post's Machine*, p. 70.

[54] Uspensky, *Post's Machine*, p. 71.

[55] V. A. Uspensky, A. L. Semenov, "What Are the Gains of the Theory of Algorithms," p. 100.

[56] In their obituary, a group of Russian mathematicians observed that Uspensky "regarded mathematics as a humanities discipline rather than a natural science." S. I. Adian et al., "Vladimir Andreevich Uspensky (27/11/1930–27/6/2018)," *Russia Mathcal Surveys*, vol. 74, 2019, p. 735-753, p. 749. D. M. Segal quoted in Gerovitch, *From Newspeak to Cyberspeak*, p. 228.

[57] Not to be confused with constructive mathematics of Markov Jr. For a critical and historical account of what has come to be known as "computational thinking," see: P. J. Denning, and M. Tedre, *Computational thinking*, MIT Press, 2019.

[58] One counterexample is Edmund C. Berkeley who wanted to popularize Boolean logic with a device known as

development of the visual programming environment Logo with the turtle being the main object-to-think with. Today, popular educational languages like Scratch are still anchored in that (visual) approach. Uspensky's way of teaching programming via the Post machine can be seen as a variant of that approach: in this light, the object-to-think with becomes an imaginary machine which enhances both formal and procedural thinking. It is thus not surprising that Papert's ideas found a fertile ground in the late Soviet Union and post-Socialist spaces of the 1990s. Papert himself actively contributed to these interactions with frequent travels to Russia and to Eastern European countries.

Aleksei Semenov, a figure who connects several main protagonists of this article, was also a key mediator in this international dialogue. A collaborator of Uspensky on the theory of algorithms in 1979, Semenov became Papert's major Russian interlocutor in the aftermath of the collapse.[59] This was not an unexpected turn of events. By the mid-1980s, Semenov worked with Ershov, who oversaw the creation of the first Soviet textbook for the new class to begin in the fall 1985. Produced by a collective of authors, the textbook was particularly influenced by Semenov and one of Uspensky's students, Alexander Shen, both having strong connections to and teaching experience at Moscow's mathematical schools. Semenov followed this line of work, accepting a post of the vice-director at the special collective "School," created under the auspices of the Soviet Academy of Sciences and the Ministry of Enlightenment to promote experimental teaching technologies. After 1991, the organization transformed into a non-state firm promoting Logo materials in the post-Soviet spaces.[60]

Beyond the organizational and social continuity, the influence of the Russian logical tradition corresponds to Uspensky's more general ideal of cultivating minds through the mathematical culture of the impossible. From this perspective, a more recent work by Semenov on Information and Communication Technology (ICT) in education for the division for higher education of UNESCO, is particularly interesting. UNESCO's ideal of an "Education for all" was pledged by all countries present at the World Education Forum in Dakar in 2000. It is in that context that Semenov authored a number of books, including *Information and Communication Technologies in Schools.* Rather than reusing Ershov's notion of "second literacy," Semenov speaks of a new literacy in these terms:

> The new literacy – the system of basic linguistic, logico-computational, and communicative skills and competencies, needed to deal with internal and external technology – is a latchkey that opens the doors of subsequent stages of organized teaching and learning. An introduction of ICT in schools gives students an impetus to learn, unlocking many doors of perception and cognition.[61]

In other words, the new literacy is a system of basic competencies that becomes an entry to training minds, a position that we saw propagated in *Post's Machine*. Although in the 2000s Semenov is addressing an audience familiar with the modern concept of "cognition" rather than "poznaniye," his words echo several generations of Russian-speaking educators believing in children's capacity to master the fundamental concepts early on and valuing this capacity as shared and universal. Like *poznaniye*, the absorption of foundations of computing is achieved through a performative immersion:

> Here is one possible way to introduce computer mathematics into elementary school. We start with basic notions, not trying to give them an exact definition – neither logically nor philosophically correct – but instead describing them in intuitive terms. These notions are introduced to students in the form of visual (graphical) and palpable (manipulative) examples. In that way, a general (non-verbalized) understanding arises in a student's head due to the inherent

Simple Simon. For more details, see: B. Longo, *Edmund Berkeley and the social responsibility of computer professionals*, ACM Books, 2015.

[59] A. L. Semenov, "Simor Papert i my: konstruktsionism -- obrazovatel'naia filosofia XXI veka" [Seymour Papert and us: Constructivism -- an educational philosophy for the twenty-first century], *Voprosy obrazovaniia* [Questions of education], vol. 1, 2017, pp. 269–294.

[60] For additional context on the post-Soviet period and the entrepreneurship of Russian coders and IT specialists in a global setting, see: V. Lepinay and M. Biagioli, *From Russia with Code: Programming Migrations in Post-Soviet Times*, Durham, NC: Duke University Press, 2019.

[61] A. Semenov, *Information and communication technologies in schools. A handbook for teachers or how ICT can create new open learning environments*, UNESCO, 2005, p. 145.

mechanics of cognition through direct perception and acting.[62]

The earlier undefinedness of certain mathematical notions is now embraced in a more general way: even when working with notions that *can* be defined (formally or otherwise) this is not what is required to train the younger mind. Instead, it is the performative, through what Papert called an object-to-think-with, that enables the mind to transform itself and to “unlock” its innate possibilities.

The movement of ideas and practices related to algorithmic thinking across time and space, as well as the cultural specificity of the role of logic in shaping the definition of the computer literacy in the late Soviet and post-Soviet times, brings forward several more general remarks, turning this time to a self-reflection on our own methods and on the history of computing.

Conclusions: Next to the West

In the two parts of this article, spanning from the mid-nineteenth century to the early 2000s, we look beyond the received narrative of the Soviet technological failure in computing. We reconstruct a resilient disposition to focus on the mind throughout different periods of Russian and Soviet history as expressed in an evolving scientific vocabulary. These longstanding developments are crucial for understanding how mathematical logicians and programmers came to emphasize the impossibility to define the algorithm mathematically. Instead, they envisioned a social transformation enabled not by the access to the machine but by actualizing individual minds thanks to inculcating algorithmic mindedness from early childhood. We refer to this humanistic vision as the culture of the impossible.

Despite the radical rupture of the revolution, we see important continuities linking the disciplinary rise of mathematical logic in the 1950s to the agendas inherited from the nineteenth century *poznaniye* and its practical facet known as Methodology. From the intellectual links connecting challenges to the LEM and the algorithm at the border of what is knowable, to the idea that research and education belong together, we locate these continuities in social mechanisms that historians of Russian science have labeled “intelligentsia science.” We demonstrate the enduring power of these tightly knit social groups, overriding private and professional spheres’ distinction, by attending to personal and career trajectories of Vladimir Uspensky and Andrei Ershov, and in lesser detail to that of Sofiya Yanovskaya. As many recollections make clear, the legitimacy exchanges among mathematical logic, cybernetics, and theoretical programming reproduced familiar mechanisms of intelligentsia’s integration based on personal trust. Ultimately, the strength of the influence of mathematical logic on the non-instrumental understanding of computer literacy in the late Soviet Union was rooted in a shared understanding of the algorithm and the program as mathematical objects bridging the inner and outer worlds, which was entirely continuous with how logico-mathematical objects were regarded before the Revolution. This epistemic communality between logicans and programmers in the Soviet Union was far from accidental; it was a result of decades-long intertwined institutional consolidation enabled by the expressive strength of cybernetic discourse as well as by the obscured yet sustained pre-revolutionary intellectual commitments.

Therefore, the Cold War context underexplains these intellectual and social developments culminating in the ideal of algorithmic mindedness. Unlike the algorithmic rationality preoccupied with the apocalyptic scenario of nuclear destruction, the focus on inner minds entailed scenarios of social transformation and global peace with the advent of universal computer literacy. Characteristically, both at the turn of the twentieth and twenty-first centuries, ideas, scholars-educators and pedagogical materials were not contained by a West versus East framework but traveled, were translated, and transformed. Both Jevons’s machine and Papert’s Logo diffused practices predicated on the vision of a human capacity to learn and to activate the mind. In this light, we do not only want to restore the Russian and Soviet history of logic and programming in their national specificity. We demonstrate that there is a constant and

[62] Ibid., p. 205.

ongoing dialogue with developments in the West, which is traced through individuals and through collective practices, rather than through ideological, othering divides. This dialogue offers perspective on the relational aspects of the post-Cold War globalization developments in the West, and is especially useful for future directions related to the contemporary debates on computational thinking.

It is common to analyze the history of computing, logic and mathematics in terms of oppositions such as: the abstract versus the concrete; the practical versus the theoretical; the formal versus the informal; and the deductive versus the inductive. However, the materials discussed above make obvious that such binary structures are to be questioned. The Soviet vision of computer literacy, addressed here through the lens of Uspensky's *Post's Machine*, is an indication of how these oppositions vanish. In this text, formal thinking is accompanied by the performative and by objects-to-think with, as school children are unknowingly introduced to foundational mathematical objects through the "performing of programs" while playing with buttons and drawing representations of the "toy" Post machine. This particular text captures what is representative of numerous Soviet initiatives that did not strive to teach a narrow technical skill but rather aimed to demonstrate the possibilities and boundaries of the mind itself. Placing Uspensky's *Post's Machine* in a much broader tradition of *poznaniye*, helps to see its pragmatic dimension, which is inconceivable from an epistemology of binarism, responsible for the oppositional categories that came to define much of Western valuation criteria for good reasoning.

Attention to a pervasive presence of values and attitudes inherent to *poznaniye* encourages another reconsideration relevant to our own field: the methodological opposition between history of technology and history of science. Troubling the assumptions underlying this binarism suggests that history of computing can, and should, belong to both, as was called for by Michael Mahoney in these pages back in 1988.[63] We believe that the long-ignored case spanning Soviet, pre-Soviet, and post-Soviet spaces presents such a union and propels the methodological opportunities for transdisciplinary engagements. These include media-theoretical methods such as those that consider mathematical tools of machine-less programming as a form of interface.[64]

The ability to simultaneously think formally and practically -- was the essence of Methodology and the studies of *poznaniye* before the Revolution, it was the take-away from Uspensky's "Methodological notes," and it was the point of Papert's object-to-think-with. This is not an exclusively non-Western narrative, as the literature, the actors, and the objects had clear relations to the Western world. This is a narrative of how historical evidence well-intelligible but not reducible to Western analytical categories can be re-synthesized, re-organized, in a way that is not arguing against the West but rather presenting a semantic structure that doesn't need to be validated by the Western framework.[65] We've shown that "literacy" in computing is not a category with a single definition, but rather one that is shaped by the background assumptions and views on things as diverse as the role of embodied cognition in learning and in machine-building, performativity as a central method for communication, what is understood to be "logical," and the validity of the binary notion of truth. With that, we have suggested that literacy can and perhaps should be expanded to the ability to critically and methodically approach the impossible, the elusive, the unverbalized, the performative, and the transrational.

[63] M. S. Mahoney, "The History of Computing in the History of Technology," *IEEE Ann. Hist. Comput.*, vol. 10, no. 2, 1988, pp. 113-125.

[64] We thank Vladimir Lukin for this idea.

[65] For an example from Global South, see: Anita Chan, *Networking Peripheries: Technological Futures and the Myth of Digital Universalism*, Cambridge, MA: MIT Press, 2013.