KSENIA TATARCHENKO*

Thinking Algorithmically: From Cold War Computer Science to the Socialist Information Culture

ABSTRACT

Cold War competition shaped the process of computerization in both East and West during the second half of the twentieth century. This article combines insights from Science and Technology Studies, which brought the analysis of Cold War techno-politics beyond the context of the nation-state, with approaches from Critical Algorithm Studies, to question the algorithm's role in the global "computer revolution." It traces the algorithm's trajectory across several geographical, political, and discursive spaces to argue that its mutable cultural valences made the algorithm a universalizing attribute for representing human-machine interactions across the ideological divide. It shows that discourses about the human capacity to devise algorithms, a practice central to computer programming, became a space for negotiating different versions of modern subjectivity. This article focuses on two related episodes to demonstrate how the notion of "algorithmic thinking" became explicitly associated with a range of politicized agendas, each claiming the algorithm's power. On one hand, the coupling of "algorithm" and "thinking" was used to describe a naturalized cognitive capacity shared among the members of the international scientific community and projected backward to the medieval scholar Al-Khwarizmi. On the other hand, the universal spread of "algorithmic thinking" became the educational goal of a late Soviet computer literacy campaign under the slogan of "Programming, the Second Literacy," a metaphor and a political vision conceived to bring about the Socialist "Information Age."

KEY WORDS: algorithm, Cold War, computer science, A. Ershov, Information Age, D. Knuth, programming, Soviet education

*Global Studies Institute, Geneva University, Sciences II, 30 Quai Ernest-Ansermet, CH–1211, Genève, Switzerland.

The following abbreviations are used: ACM, Association for Computing Machinery; AE, personal papers of Andrei Ershov, Ershov Archive; *CACM*, *Communications of ACM*; *JACM*, *Journal of ACM*; IFIP, International Federation for Information Processing.

Historical Studies in the Natural Sciences, Vol. 49, Number 2, pps. 194–225. ISSN 1939-1811, electronic ISSN 1939-182X. DOI: https://doi.org/10.1525/hsns.2019.49.2.194.

Men are all alike.
IN WHAT WAY
—ELIZA computer program conversation cited in Joseph Weizenbaum, *Computer Power and Human Reason: From Judgement to Calculation*, 1976

INTRODUCTION

Mind or machine, which trumps which? Debates about reasoning and self-perfection, automation and political order did not begin with Big Data. Early modern natural philosophers, who developed mechanical calculating machines, embraced mathematical practice for the bettering of moral character.[1] In 1950, the popularizers of computer technology wondered, "Can man build a superman?"[2] Today, non-profits casually proclaim their aim to foster digital modes of cognition in order to combat poverty and affirm the liberal political subject. "What we need to do is to create a different profile of a brain," asserted Rodrigo Arbodela of One Laptop per Child (OLPC), "a brain that is geared towards innovation, exploration, invention."[3] Meanwhile, the past is continuously recomposed to make sense of the present technological and political fractures: our media map the geopolitics of cybersecurity onto old political divisions, from Cold War to code war.[4] If techno-utopian visions did not begin with Big Data, the catchword, coupling visions of a better future with algorithmic powers of information processing, actualizes historical reflection about the relationship connecting minds, machines, and ideologies of the past century.

Today, algorithms are known to turn data into knowledge, and knowledge into value, including commodification and control over selves. Historians of science have countered teleological and deterministic claims associated with Big Data by tracing *long durée* genealogies of large-scale data collection and

1. See Matthew Jones, *The Good Life in the Scientific Revolution: Descartes, Pascal, Leibniz, and the Cultivation of Virtue* (Chicago: Chicago University Press, 2006).

2. *Time*, 23 Jan 1950, cover. Paul Edwards labeled the Cold War type of computer representations in American culture a "cyborg discourse": Paul Edwards, *The Closed World: Computers and the Politics of Discourse in Cold War America* (Cambridge, MA: MIT Press, 1996).

3. Rodrigo Arbodela, "Children: A Mission, Not a Market," Tedx Rio Conference presentation, Feb 2011, quoted in Anita Say Chan, *Networking Peripheries: Technological Futures and the Myth of Digital Universalism* (Cambridge, MA: MIT Press, 2013), 200–01.

4. For an example, see Emma Emeozor, "From Cold War to 'Code War'," *The Sun*, 9 Jan 2018 at http://sunnewsonline.com/from-cold-war-to-code-war/ (accessed 15 Jun 2018).

processing throughout modern scientific, state, and corporate institutions before and after the advent of the computer.[5] The contemporary power of computer algorithms and the shifts in forms of governance associated with them are discussed under the banner of "data and society" and the consolidating field of algorithm studies.[6] But neither body of scholarship identified the contradictory historical roles assigned to the algorithm as a tool of computing for both humans and machines during the second half of the twentieth century.

Between the arguments for continuity across the modern knowledge regimes and those focusing on ongoing changes entailed by computerization, sits a largely understudied body of expertise that emerged after World War II, better known under its American name of computer science. Even less understood is how the circulation of knowledge within this international community enabled multiple versions of the Information Society.

This essay reconstructs the historical meanings of the notion "algorithmic thinking" to highlight the multiple politicized functions of the concept of algorithm and to contribute to scholarly exploration into geographies and cultural landscapes of digital universalism and utopianism.[7] The meanings of "algorithmic thinking" draw these two categories together. Tracing the algorithm's trajectories across geographical, political, and discursive spaces, I argue that its mutable cultural valences made the algorithm a universalizing attribute for representing human-machine interactions across the Cold War ideological divide. I show that discourses about human capacity to devise algorithms, a practice central to computer programming, became a site where different versions of modern subjectivity were confronted and negotiated.[8]

5. Elena Aronova, Christine von Oertzen, and David Sepkoski, eds., *Data Histories*, *Osiris* 32 (Chicago: Chicago University Press, 2017).

6. For a snapshot of the intellectual orientation of the field, see the reading list compiled by The Social Media Collective at https://socialmediacollective.org/reading-lists/critical-algorithm-studies/ (accessed 15 Jun 2018).

7. Fred Turner, *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism* (Chicago: Chicago University Press, 2006); Anita Say Chan, *Networking Peripheries* (ref. 3).

8. Modernity is an important category for Soviet historiography. A reference on Socialist modernity in the 1930s is Stephen Kotkin, "Modern Times: The Soviet Union and the Interwar Conjuncture," *Kritika* 2, no. 1 (2001): 111–64. A suggestion for a different approach to describe the postwar Soviet modernity can be found in Anna Krylova, "Soviet Modernity: Stephen Kotkin and the Bolshevik Predicament," *Contemporary European History* 23, no. 2 (2014): 167–92. For a methodological discussion of Soviet modernity in a transnational perspective, see Michael David-Fox, *Crossing Borders: Modernity, Ideology, and Culture in Russia and the Soviet Union*

First, I trace the rise of the algorithm as a central category for the new discipline of computer science and for the process of international community building during the 1960s and the 1970s. Drawing on my previous, in-depth studies of transnational encounters in computing, I emphasize that this new scientific community was bound not only by the communality of their intellectual agenda and interpersonal networks, but also by a discourse about the specific cognitive element of their work. This community claimed to possess a common, natural, and apolitical "algorithmic" way of thinking and supported such claims by tracing the origins of the word "algorithm" to the alleged birthplace of the medieval scholar Al-Khwarizmi.[9] "I think that we computer scientists of the 20th century, in particular the algorithmic community," mused the Austrian computer pioneer and the leader of the international community, Hans Zemanek, "have quite a lot to learn from Al Khwarezmi's method and success."[10] The personal networks among such like-minded individuals of the international algorithmic community sustained systematic coordination efforts across the ideological divide and were resilient enough to withstand multiple geopolitical crises. Yet belonging to the international professional community did not diminish the scientists' allegiances to their home societies' versions of modernity.

In the second part, I turn to the less know Soviet vision for the Information Society, centered on the notion of the algorithm and enabling human agency through the technical skill of programming. I sketch how the prominent Soviet member of the international community, Andrei Ershov, appropriated the idea about the specific "algorithmic thinking" associated with professional practices of programming into the social-engineering agenda guiding the Soviet educational reforms of the late 1980s. This Soviet programming expert used the ideas gathered from his Western networks to develop a research program in programming for children at his home institution, the Akademgorodok Computer Center. He eventually became the main proponent of the country-wide reform, operating important patronage networks that led to the reform's

(Pittsburgh: University of Pittsburgh, 2015). For the most recent debates, see the special forum: "Sporia o Modernosti," *NLO* 140, no. 4 (2016), especially, Michael David-Fox, "Russian–Soviet Modernity: None, Shared, Alternative, or Entangled?," 19–44.

9. See A. P. Ershov and D. E. Knuth, eds., *Algorithms in Modern Mathematics and Computer Science: Proceedings, Urgench, Uzbek SSR, September 16–22, 1979* (Berlin; New York: Springer-Verlag, 1981).

10. H. Zemanek, "DIXIT ALGORIZMI. His Background, his Personality, his Work, and his Influence," in Ershov and Knuth, *Algorithms* (ref. 9), 1–81, on 81.

implementation in September 1985. Andrei Ershov's actions were guided by the belief that "exercising to control the computer, the human develops his capacity to control himself."[11] I analyze Ershov's texts to demonstrate that the reform and its compulsory classes on "The Basics of Information Technology" aspired to bring the new Soviet generation to the Socialist version of the Information Age, the age of "Information Culture"; the Socialist education system would actualize the natural capacity of human beings to "algorithmic thinking" in order to transform them into new Soviet political subjects, new "new men," mastering the thought, the word, and the action.[12]

To fully articulate what is at stake in bringing the two overlapping but contradictory notions of "algorithmic thinking" together, we have to recognize the strengths and the limits of the transnational historiography of Cold War science and technology and the place of computing histories within it—this task is reserved for the Conclusion of this essay. Both historical and critical studies of the notion of algorithm agree that its power is mutable.[13] After World War II, "algorithm" became explicitly associated with a range of diverging politicized claims about human capacities for procedural thinking. Hence, the Cold War trajectories of the algorithm function as a magnifying lens attesting to its changing social and cultural valences. "In many recent studies," observed Hunter Heyck and David Kaiser when summarizing the historiography of Cold War science of the 2000s, "science is technoscience, technology is technopolitics, and both are instruments for the reconstruction of self and society."[14] This article is about how algorithm partook in the collapse and reconstruction

11. A. Ershov, "Komp'iuterizatsiia shkoly i matematicheskoe obrazovanie," plenary speech at the ICME, Budapest, August, 1988, f. 344/ l. 220–59, on 221. Citations are according to the pdf file at http://ershov.iis.nsk.su/archive/eaindex.asp?did=3444, on 2.

12. Central to the Soviet revolutionary ideology was the fashioning of a new type of man, strong and creative. For an intellectual history tracing the official and unofficial roots of the concept, see Bernice Glatzer Rosenthal, *Nietzsche and Soviet Culture: Ally and Adversary* (Cambridge: Cambridge University Press, 1994). Compare with Slava Gerovitch's analysis of cosmonauts as "new Soviet men": "'New Soviet Man' Inside Machine: Human Engineering, Spacecraft Design, and the Construction of Communism," in *Osiris* 22 (Chicago: Chicago University Press, 2007), 135–57.

13. For emphasis on mutability conjugated with the notions of imagination, creativity, and play, see Martine Mahnke and Emma Uprichard, "Algorithming the Algorithm," in *Society of the Query Reader: Reflections on Web Search*, ed. René König and Miriam Rasch (Amsterdam: Institute of Network Cultures, 2014). Also, see Solon Barocas, Sophie Hood, and Malte Ziewitz, "Governing Algorithms: A Provocation Piece" (29 Mar 2013), http://dx.doi.org/10.2139/ssrn.2245322.

14. Hunter Heyck and David Kaiser, "Focus: New Perspectives on Science and the Cold War," *Isis* 101, no. 2 (2010): 362–66.

of boundaries between self and machine, and between capitalist and socialist Information Societies.

1. "PEOPLE WHO THINK LIKE WE DO": THE MAKING OF THE COLD WAR ALGORITHMIC COMMUNITY

In his pioneering study on the formation of the theoretical agenda of Computer Science, the historian Michael Mahoney analyzed computer experts' efforts at discipline building around the main question, "What should a theory of computing be about?" The question was far from trivial, as the computer was first constructed as a tool, not an object for scientific inquiry. "But computers had no nature," Mahoney could not avoid quipping, "until theorists created one for them, and different theorists had different ideas about the nature of computers."[15] Whereas the creation of the computer as a physical device was first and foremost associated with engineering efforts, computing theory developed by borrowing from a range of mathematical subdisciplines: mathematical logic, theory of automata, and formal semantics, among other fields. But the matter of theory was not an abstract question at all.

Janet Abbate recently showed that the experts' concerns over the intellectual agenda came with purse strings attached: government money flowed to computation centers in abundance, but primarily to fund their service activity for other sciences.[16] The leadership of the Association for Computing Machinery (ACM), the main professional society for academic computer experts, was greatly concerned with this dynamic. In September 1966, the president of the ACM, Anthony G. Oettinger, lamented that "while very large amounts of money go into computation these days this does not in itself furnish money for

15. Michael S. Mahoney, "Computer Science: The Search for a Mathematical Theory," in *Histories of Computing*, ed. Thomas Haigh (Cambridge, MA: Harvard University Press, 2011), 128–46, on 134. Also see Matti Tedre, *The Science of Computing: Shaping a Discipline* (CRC Press, 2014). For arguments that mathematical logic was foundational for both computing machinery and software, see Mark Priestley, *A Science of Operations: Machines, Logic and the Invention of Programming* (London: Springer, 2011).

16. Janet Abbate, "From Handmaiden to 'Proper Intellectual Discipline': Creating a Scientific Identity for Computer Science in 1960s America," in *Communities of Computing: Computer Science and Society in the ACM*, ed. Thomas Misa (New York: ACM Books, Morgan & Claypool, 2016), 25–46.

research and education in computer science."[17] Another ACM president and the founder of the Computer Science department at Stanford, George Forsythe, thought of affiliation with mathematics as a funding strategy: "I see no practical alternative today to our assuming the role of one of the mathematical sciences."[18] Intellectual and social alliances forged by computing community leaders made this strategy a relative success in the context of the exponential growth of the Cold War military-industrial-academic complex; the Space Race fostered the miniaturization of electronics, military-funded research in artificial intelligence, automatic translation, and most famously, computer networks, and intellectual hubs like Stanford flourished as regional intersections of expertise, qualified labor, and capital.[19]

During the 1960s, when numerous computer science departments opened at American universities, computer science emerged as a hybrid discipline in search of intellectual autonomy and professional identity.[20] However, in addition to theory, it was no less concerned with practical applications, thus blurring the conventional distinction between scientific and craft practice. As described by Nathan Ensmenger, the accomplishment in institutional consolidation of the discipline and its early success in attracting a large student

17. Anthony G. Oettinger, quoted in Abbate, "From Handmaiden" (ref. 16), 25. Also see William Aspray and Bernard O. Williams, "Arming American Scientists: NSF and the Provision of Scientific Computing Facilities for Universities, 1950–1973," *IEEE Annals of the History of Computing 16*, no. 4 (1994): 60–74.

18. George E. Forsythe, quoted in Abbate, "From Handmaiden" (ref. 16), 32. For full text, see George E. Forsythe, "President's Letter to the ACM Membership," *CACM* 9, no. 4 (1966): 244.

19. For more on Stanford and Silicon Valley, see: Stuart W. Leslie, *The Cold War and American Science: The Military-Industrial-Academic Complex at MIT and Stanford* (New York: Columbia University Press, 1993); Rebeca Lowen, *Creating the Cold War University: The Transformation of Stanford* (Stanford, CA: University of California Press: 1997); and Christophe Lecuyer, *Making Silicon Valley: Innovation and the Growth of High Tech, 1930–1970* (Cambridge, MA: MIT Press, 2006). The classical study of the influence of the security concerns over scientific research agenda is Paul Forman, "Behind Quantum Electronics: National Security as Basis for Physical Research in The United States, 1940–1960," *Historical Studies in the Physical and Biological Sciences* 18, no. 1 (1987): 149–229. On computer translation and Cold War intelligence concerns, see Michael D. Gordin, "The Dostoevsky Machine in Georgetown: Scientific Translation in the Cold War," *Annals of Science* 73, no. 2 (2016): 208–23.

20. Mahoney, "Computer Science" (ref. 15). For a study of institutional dynamics, see William Aspray, "Was Early Entry a Competitive Advantage? US Universities That Entered Computing in the 1940s," *IEEE Annals of the History of Computing* 22, no. 3 (2000): 42–87. For arguments about the Stanford computer science department as a model for other American Universities, see Joseph November, "George Forsythe and the Creation of Computer Science as We Know It," in Misa, *Communities of Computing* (ref. 16), 47–70.

population did not translate into easy solutions regarding the balance between academic standards and practically useful knowledge, a problem ever complicated by conflicted relationships with the computer industry.[21] Questions of identity and legitimacy interlocked.

The Cold War context in which the discipline was developing was not limited to that of the American military-industrial-academic complex. Although the CoCom (Coordinating Committee for Multilateral Export Controls) embargoes limited Western exports of computer technology to Eastern bloc countries throughout the period, they did not prevent interactions between the computer experts. On the contrary, as I have shown elsewhere, the new American discipline of computer science benefitted from international communication, intellectual coordination, and traffic of artifacts under the auspices of the International Federation for Information Processing (IFIP), the Algol (algorithmic language) group, and bilateral scientific agreements. The international scale of computer developments was a key argument used by the experts to enable interactions, and transnational encounters became important for grappling with ideas about professional identity and promoting the new field on both sides of the Iron Curtain.[22]

The 1974 speech given by Donald Knuth upon receiving the Turing Prize, the most prestigious award bestowed by the ACM, was a snapshot of the intellectual and practical concerns at the core of computer science. Today, Knuth still considers the development of the analysis of algorithms as an academic subject to be his proudest achievement. But he was also a generous teacher and prolific programmer: from 1968 into the 1990s, he trained several dozens of PhDs in computer science at Stanford; and between the 1970s and the 1990s, he was developing a pioneering open-source computerized typesetting system, TeX, which was widely adapted in commercial typesetting and by a thriving community of users and developers. Knuth's life-long insistence on providing mathematical and historical grounds for computer

21. For more on the dynamics between professionalization of programming and the establishment of computer science, see Nathan Ensmenger, *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise* (Cambridge, MA: MIT Press, 2010). For an overview of the curriculum debates, see Gopal K. Gupta, "Computer Science Curriculum Developments in the 1960s," *IEEE Annals of the History of Computing* 29, no. 2 (2007): 40–54.

22. Ksenia Tatarchenko, "The Cold War Origins of the International Federation for Information Processing," *IEEE Annals of the History of Computing* 32, no. 2 (2010): 46–57; and Tatarchenko, "'The Anatomy of an Encounter': Transnational Mediation and Discipline Building in Cold War Computer Science," in Misa, *Communities of Computing* (ref. 16), 199–227.

methods and his daily coding routines hold the key to understanding his Turing award speech.[23]

In 1974, Knuth spoke as the author of the highly influential, multivolume *The Art of Computer Programming*, addressing his peers to stress that programming was "both a science and an art."[24] After reminding his audience that the agenda of discipline building had been articulated on the pages of the ACM's journal as early as 1959, Knuth resumed the accomplishment with a touch of humor: "we have actually succeeded in making our discipline a science, and in a remarkably simple way: merely by deciding to call it 'computer science.'"[25] Behind Knuth's joke, however, was a serious and heated debate about both the name of the association and the name of the new discipline. During the 1970s, Knuth proclaimed his own preference on multiple occasions. According to him, computer science was the study of algorithms, and he would rather call it Algorithmics, a term that Knuth credited to another prominent member of computing community, Joseph F. Traub.[26]

The ACM itself almost changed its title. In 1965, George Forsythe failed to obtain enough support to change the organization's name to Association for Computing and Information Sciences. Although more than half of all members supported the change, it was less than the two-thirds required by the organization's bylaws.[27] In the end, although the ACM retained the name originating in the days of its creation in the late 1940s, its goals shifted as its membership increased. One important measure of change was in divergent discourses about computers and human reasoning.

The organization was first established thanks to the community-building efforts of Edmund Berkeley, who was part of Howard H. Aiken's Harvard

23. For more on Knuth's biographical information, see his academic CV available at https://www-cs-faculty.stanford.edu/knuth/vita.pdf (accessed 15 Jun 2018). Also see Knuth's personal website featuring several autobiographical documents at https://www-cs-faculty.stanford.edu/~knuth/ (accessed 15 Jun 2018).

24. Donal Knuth, "Computer Programming as an Art," *CACM* 17, no. 12 (1974): 667–73, on 667.

25. Ibid. Also see Walter F. Bauer, Mario L. Juncosa, and Alan J. Perlis, "ACM Publication Policies and Plans," *JACM* 6, no. 2 (1959): 121–22.

26. At the time, Joseph F. Traub was the head of the computer science department at Carnegie Mellon University (1971–1979); later, he became the founding chairman of the Computer Science Department at Columbia University. His 1964 monograph introduced the notion of algorithmics: J. F. Traub, *Iterative Methods for the Solution of Equations* (Englewood Cliffs, NJ: Prentice-Hall, 1964).

27. George E. Forsythe, "President's Letter to the ACM Membership," *CACM* 8, no. 9 (1965): 541.

Mark II team and later drew on this experience for his famous book popularizing computer technology, *Giant Brains, or Machines That Think*.[28] Berkeley and his associates had an ambitious vision for the transformative power of computing and thus formulated the goals of the association: "advance the science, development, construction, and application of the machinery for computing, reasoning, and other handling of information."[29] The notion of "reasoning" among the foremost computer applications was not accidental; Berkeley staunchly believed in the possibility of human self-improvement, of thinking clearly and acting wisely. Since his time as a student of mathematics and logic at Harvard, where he was influenced by George Birkhoff, Berkeley was preoccupied with the potential of symbolic logic as a "modern method of thinking" for making optimal decisions about the ill-structured social problems of modern society.[30] By the 1960s, Berkeley's visionary social plans implicating professionalization of computer expertise to serve the public good did not remain at the forefront of the organization's goals. When, in 1972, ACM honored Berkeley as its singular founder at its 25th anniversary dinner, the evening soon turned sour. His acceptance speech denounced the ACM and computer industry leaders for neglecting the responsibility to care about how computers were being used in armed conflicts such as the Vietnam War.[31]

As the field professionalized, the notion of "thinking" in relation to computing acquired a set of different connotations.[32] For Knuth, the thinking

28. Berkeley's first major work was published in 1937 by the American Institute of Actuaries under the title of "Boolean algebra (the technique for manipulating 'and,' 'or,' 'not,' and conditions) and applications to insurance." It was reviewed in the *Journal of Symbolic Logic* by none other than Alonzo Church of Princeton. Church closed with a warm observation, noting the work's "novelty lies in the practical application of this technique in a hitherto unsuspected direction." Edmund C. Berkeley, "Boolean algebra (the technique for manipulating 'and', 'or', 'not', and conditions) and applications to insurance," *The Record of the American Institute of Actuaries* 26, part 2, no. 54 (1937): 373–414; reviewed by Alonzo Church in *Journal of Symbolic Logic* 3, no. 2 (1938): 90.

29. See Atsushi Akera, "Edmund Berkeley and the Origins of ACM," *CACM* 50, no. 5 (2007): 30–35; and Bernadette Longo, *Edmund Berkeley and the Social Responsibility of Computer Professionals* (New York: Association for Computing Machinery and Morgan & Claypool, 2015).

30. See George Birkhoff, "Science and Spiritual Perspective: A New Philosophy," *Century Magazine*, June 1929. More in Longo, *Edmund Berkeley* (ref. 29).

31. More in Longo, *Edmund Berkeley* (ref. 29).

32. The rise of the interdisciplinary field of cognitive science, which is one of the most famous developments of the time, is beyond the scope of this article. For a programmatic text arguing that human thinking is a kind of symbol manipulation, see the 1976 Turing Award speech by Allen Newell and Herbert Simon, "Computer Science as Empirical Inquiry: Symbols and Search," *CACM* 19, no. 3 (1976): 113–26. For an insightful analysis of the influence of Cold War

associated with computer science was thinking neither according to symbolic logic nor even according to mathematical principles. Thinking, as related to computers, was less Berkeley's nurtured thinking "in the right way" than an instinctive and joyous kind of thinking. Knuth's Turing award speech was about defining the cognitive labor that is required for the creation of the concrete artifact, namely, a beautiful and useful computer program.[33] "Good," the epithet used by Knuth, did not directly entail a moral judgment on the nature of the program's application. "Good" stood for "meaningful" as functional and understandable, as well as for error-free. In his speech, Knuth confessed: "I really enjoy writing computer programs; and I especially enjoy writing programs which do the greatest good, in some sense."[34] He believed this joy was shared by all programmers. According to Knuth, computer science retained its expressive aspect of an art form via the close interaction between the programmer and the machine.

Therefore, Knuth argued that computer systems should be designed to leave space for human insight and to balance automation with human-machine interaction. "One of the best ways to keep up the spirits of a system user," stated Knuth, "is to provide routines that he can interact with. We shouldn't make systems too automatic, so that the action always goes on behind the scenes; we ought to give the programmer-user a chance to direct his creativity into useful channels."[35] Knuth's arguments about the expressive power of programming as a source of a particular aesthetic and an experience of pleasure were not completely new to his audience. The American computer scientist contributed to a discourse that was already circulating among computer scientists internationally, and he explicitly drew on the language employed to describe the practice and challenges of the programming profession by the Soviet programmer and computer scientist, Andrei Ershov.[36]

---

context on the American cognitive sciences, see Jamie Cohen-Cole, *The Open Mind: Cold War Politics and the Sciences of Human Nature* (Chicago: Chicago University Press, 2014).

33. These two qualities allow Knuth to claim that programming represents a bridge between C. P. Snow's "two cultures." In "Computer Programming" (ref. 24), Knuth cites both the original publication and the 1964 follow-up of Snow's famous lecture: C. P. Snow, "The Two Cultures," *The New Statesman and Nation*, 6 Oct 1956, 413–14; and C. P. Snow, *The Two Cultures: and a Second Look. An Expanded Version of The Two Cultures and the Scientific Revolution* (Cambridge: Cambridge University Press, 1963).

34. Knuth, "Computer Programming" (ref. 24), 671.

35. Ibid., 672.

36. Ibid., 673, ref. 9.

In 1972, Ershov was invited to give a prestigious key speech at the American Joint Computer Conference in Atlantic City. The resulting "Aesthetics and the Human Factor in Programming" became his most well-known text, addressing questions that remained pertinent throughout the 1970s and to this day: How should software projects be organized? How should programmers be trained? What individual qualities should be sought and cultivated in programmers?[37] In the wake of the so-called software crisis, or the realization that the costs and failure risks associated with software were growing exponentially, Ershov called upon society to recognize and appreciate programming, and upon programmers to take upon themselves a sense of weighty social responsibility.[38] In a nutshell, the "human factor" was the programmer and the "aesthetics" was his aspiration to achieve a perfect combination of program and machine.

Unlike Knuth's emphasis on a contingent interpretation of the boundary between art and science, Ershov's speech conveyed a clear sense of historical mission for the new profession. One statement stressed that the programmer's way of thinking and his social role were to be nothing less than epoch-changing: "The programmer is the lynch-pin of a second industrial revolution [*sic*]; as such he should possess a revolutionary way of thinking."[39] The

37. Andrei Ershov, "Aesthetics and Human Factor in Programming," *CACM* 15, no. 7 (1972): 501–05; reprinted in: *Datamation* 18, no. 7 (1972): 62–67; *Computer Bulletin* 16, no. 7 (1972): 352–55; *The Honeywell Computer Journal* 6, no. 1 (1972): 23–26; and *Jurimetrics Journal* 13, no. 3 (1973): 142–49. Citations are according to the publication in the *CACM*.

38. For the original report that contributed to the diffusion of the notion of the "software crisis," see Peter Naur and Brian Randell, eds., *Software Engineering: Report on a Conference Sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968* (Brussels: Scientific Affairs Division, NATO, 1969). The NATO conference is one of the best known episodes of software historiography, and is too large to overview here. For an example, see Michael S. Mahoney, "Finding a History for Software Engineering," *IEEE Annals of the History of Computing* 25, no. 1 (2004): 8–19.

39. Ershov, "Aesthetics and Human Factor" (ref. 37), 503. Here the "second industrial revolution" is meant as the time when intellectual labor became industrialized as a result of the spread of computing technology. In the Russian language version of the paper, Ershov uses a Russian phrase that translates as "scientific-technical revolution." See A. P. Ershov, "O Chelovecheskom i esteticheskom faktorakh v programmirovanii," *Kibernetika*, no. 5 (1972): 95–99. On the Soviet usage of the theory of scientific-technical revolution, see Julian M. Cooper, "The Scientific and Technical Revolution in Soviet Theory" in *Technology and Communist Culture*, ed. Frederic Frelon (New York: Praeger, 1977), 146–75. For recent historical analyses, see Elena Aronova, "Big Science and 'Big Science Studies' in the United States and the Soviet Union during the Cold War," in Oreskes and Krige, *Science and Technology* (ref. 82), 393–429; Stefan Guth, "One Future Only: The Soviet Union in the Age of the Scientific-Technical Revolution," *Journal of Modern European History* 13, no. 3 (2015): 355–76.

"thinking" mentioned by Ershov corresponded to a description of the specific cognitive nature of programming, distinct from average intellectual activity. Ershov depicted programmers as "men whose work brings them to those limits of human knowledge which are marked by algorithmically unsolvable problems."[40] Like Knuth, Ershov emphasized the moment of human-machine interaction, the encounter bearing a transcendent potentiality.

Ershov did not shy away from borrowing religious symbolism to depict the "trinity" of roles that the programmer must take on to achieve the "familiar miracle" of a perfect machine-program combination: "He feels himself to be the father-creator of the program, the son-brother of the machine on which it runs, and the carrier of the spirit which infuses life into the program/machine combination."[41] However, contrary to Berkeley's anthropomorphic metaphors ascribing human qualities to machines, Ershov's metaphors implied a bidirectional relationship: the brotherhood residing in the act of programming.[42] Proclaiming the act of writing programs (the programs that fit particularly well for particular tasks running on particular machines) as a "triumph of human intelligence," the transcendent brotherhood between humans and machines implicitly entailed an international brotherhood of all programmers independently of where they were. The gendering of programming practice in these discourses was a constitutive element for enhancing the social status of the activity, very much as internationalism was contributing to its intellectual authority.[43] In the

40. Ershov, "Aesthetics and Human Factor" (ref. 37), 502.

41. Ibid., 504.

42. Compare with Louis Fein's ideas about human-machine interaction that circulated in the American press in the 1960s: "What the hell are we making these machines for, 'if not to free people?' Many scientists hope that in time the computer will allow man to return to the Hellenic concept of leisure, in which the Greeks had time to cultivate their minds and improve their environment while slaves did all the labor. The slaves, in modern Hellenism, would be the computers." Quoted in "The Cybernated Generation," *Time*, 2 Apr 1965, 84–91.

43. Gender imbalance in today's information technology has contributed to making the gender question one of the most developed subfields of the history of computing. For a collection of the U.S.-based, NSF-funded empirical studies of gender imbalance in computing, see J. McGrath Cohoon and William Aspray, eds., *Women and Information Technology: Research on Underrepresentation* (Cambridge, MA: MIT Press, 2006). The classic work on gender and early computing is Jennifer Light, "When Computers Were Women," *Technology and Culture* 40, no. 3 (1999): 455–83. Also see: Michael Mahoney, "Boys' Toys and Women's Work: Feminism Engages Software," in *Feminism in Twentieth-Century Science, Technology, and Medicine*, ed. Angela N. H. Creager, Elizabeth Lunbeck, and Londa L. Schiebinger (Chicago: University of Chicago Press, 2001), 169–85; Ensmenger, *The Computer Boys* (ref. 21); Ensmenger, "'Beards, Sandals, and Other Signs of Rugged Individualism': Masculine Culture within the Computing Professions," *Osiris* 30, ed. Erika Milam and Robert A. Nye (Chicago: Chicago University Press,

process, the social reality of the Soviet programming was erased, and portable professional ideal was articulated.[44]

Ironically, the power of location was mobilized by both Ershov and Knuth to create a mythologized past for the new discipline. Searching for the ancient roots of algorithm acquired an important significance for spelling out the algorithmic universalism that underlay the claims for programming as the case of reason's triumph. But origins are bound to places. "Your region," said Donald Knuth to the journalists from the Uzbek newspaper *Khorezmskaia Pravda*, in September 1979, "should become the mecca for all information technology specialists." This high praise was predicated on two assumptions. First was the peculiar vision of what computing was about: he also reminded readers that computers are made of sand and accomplish intelligent tasks thanks to programs—which was to say, thanks to algorithms. Second was the Persian identity of Al-Khwarizmi, emphasizing his alleged birthplace versus his scholarly workplace at the Abbasid Baghdad.[45]

So why would Knuth travel from Stanford to Uzbekistan? What was he doing there in the first place? As he himself explained, he had "wanted to make the pilgrimage to this place for many years, ever since learning that the word 'algorithm' was derived from the name of Al-Khwarizmi, the great ninth-century scientist whose name means 'from Khwarezm.'"[46] Whereas algorithmic thinking was a universal human property, locating algorithm's origins at Al-Khwarizmi's birthplace gave historical legitimacy to the contested agenda of computer science. Following an exchange of letters during the spring of 1977, Knuth and Ershov decided to co-organize an international conference in Urgench, which was the closest large city in the vicinity of the ancient city of

2015), 38–65; Thomas J. Misa, ed., *Gender Codes* (Hoboken, NJ: IEEE Computer Society Press, Wiley, 2010); Janet Abbate, *Recoding Gender: Women's Changing Participation in Computing* (Cambridge, MA: MIT Press, 2012); Marie Hicks, "De-Brogramming the History of Computing," *IEEE Annals of the History of Computing* 35, no. 1 (2013): 88; Hicks, *Programmed Inequality: How Britain Discarded Women Technologists and Lost Its Edge in Computing* (Cambridge, MA: MIT Press, 2017); and Irina Nikivincze, "Solving a Career Equation: The First Doctoral Women in Computer Science," in Misa, *Communities of Computing* (ref. 16), 71–90.

44. Elsewhere, I show how Ershov's integration into the masculine international milieu correlated with his depiction of a masculine professional ideal for programmers, despite the large proportion of women working in this occupation in the Soviet Union. See Ksenia Tatarchenko, "'The Computer Does Not Believe in Tears': Programming, Professionalization and Gendering of Authority," *Kritika* 18, no. 4 (2017): 709–39.

45. D. Knuth, "V Poiske," *Khorezmskaia Pravda*, 25 Sep 1979, f. 191/ l. 134.

46. Knuth, "Algorithms in Modern Mathematics and Computer Science," in Ershov and Knuth, *Algorithms* (ref. 9), 82.

Khiva, the heart of Khwarezmian oasis. The letter sent to potential participants of the meeting by the Soviet and American specialists stated:

> We believe the desert setting and the atmosphere of history that pervades the conference site will provide us with special opportunity to take our minds away from the everyday work that fills our work at home; it should help us to thoughts of a more penetrating, far-sighted, and philosophical nature.[47]

In addition to being an inspirational occasion and a change of scenery, the gathering was also supposed to bring about a more concrete result, namely to "be an important experience leading to future progress."[48] For the co-organizers, this "future progress" had a distinct connotation; although unable to change the name of "computer science," Knuth concentrated his efforts on promoting algorithms as the core of the new discipline.

At the conference, many papers addressed different aspects related to algorithms, from the origin of the term in Al-Khwarizmi's writings on arithmetic, to a general theory of algorithms, to the notion of algorithmic complexity, and to issues of design and implementation. Yet according to the paper by V. A. Uspenskii and A. L. Semenov, a precise definition of the algorithm itself remained elusive: "The concept of the algorithm like that of set and of natural number is such a fundamental concept that it cannot be explained through other concepts and should be regarded as undefinable one."[49] The mathematicians pointed out the recent wave of interest in algorithms and clarified the limits of current knowledge:

> The most important discovery in the science of algorithms was undoubtedly the discovery of the general notion of algorithm itself as a new and separate entity. We emphasize that this discovery should not be confused with the discovery of representative computational models (constructed by Turing, Post, Markov, Kolmogorov).[50]

In other words, neither formal constructions nor the usual, common-sense definitions of "algorithm" as a prescription were mathematical definitions, but rather should be understood as attempts to formally characterize algorithms

47. Ershov and Knuth, "Foreword," *Algorithms* (ref. 9), iii.

48. Ibid.

49. V. A. Uspenskii and A. L. Semenov, "What Are the Gains of the Theory of Algorithms: Basic Developments Connected with the Concept of Theory of Algorithm and with Its Application in Mathematics," in Ershov and Knuth, *Algorithms* (ref. 9), 100–234, on 106.

50. Ibid., 104.

and provide a meaningful explanation for them. For Uspenskii and Semenov, this difficulty of mathematical definition arose from the semantic nature of algorithms:

> [T]he algorithms themselves are objects of a special kind and have a property non-typical for mathematical objects—the semantic property "to have a meaning." . . . The meaning of an algorithm is imperative: an algorithm is to be performed. The theory of algorithms can be treated as a linguistics of imperative sentences. Mathematicians have not yet found out how to [deal] with the linguistic objects filled with meanings.[51]

In fact, most conference participants would agree that the intuitive understanding of "algorithm" would have existed since the earliest days of mathematical practice and thus was universal. Knuth's own interest in the history of algorithms reflected a conviction that an inborn human capacity for procedural thinking became professionalized, thanks to the need to program computers. At the conference, Knuth asked, "Is there an essential difference between an algorithmic viewpoint and the traditional mathematical worldview?"[52] Recollecting his own student days, Knuth pointed out how separate buildings were associated with different practices and thinking habits: "In one building I was a mathematician, in another I was a computer programmer, and it was as if I had a split personality." Next, Knuth conflated the notion of personality with that of professional identity and of disciplinary analytical tools to elaborate a list of professions associated with what he called "different thought processes." Drawing on the proportion of graduate students majoring in the field of computer science, Knuth speculated that "roughly 2% of all people think algorithmically, in the sense that they can rapidly reason about algorithmic processes." This natural inclination to "algorithmic thinking" sustained the global process of community and discipline building:

> I believe that the real reason underlying the fact that Computer Science has become a thriving discipline at essentially all of the world's universities, although it was totally unknown twenty years ago, is not that computers exist in quantity; the real reason is that the algorithmic thinkers among the scientists of the world never before had a home. We are brought together in Computer Science departments because we find people who think like we do.[53]

51. Ibid., 100–01.

52. Knuth, "Algorithms" (ref. 46), 82–99, 86.

53. Knuth, "Algorithms" (ref. 46), 87. To make explicit the algorithmic nature of some of the mathematical works that he cites later in the paper, Knuth rewrites the proofs in his own notation

This naturalized version of both the American institutionalization and the formation of the international community was strategic. It excluded many elements that both Ershov and Knuth had to deal with to make their international gathering possible. The communality of their own interests was not a simple coincidence fostered by natural ability to think alike, but a result of longstanding personal and institutional exchanges. Gathering an international conference around the notion of algorithm, the two scientists aptly constructed the event as an apolitical forum where eternal mathematical truths took precedence over ephemeral political issues. The limits to these apolitical boundaries and social and political structures enabling the gathering itself were too numerous to investigate in great detail in this essay: the local Uzbek authorities sponsored the conference to bolster a modern-looking image for the cotton-producing republic; the logistical obstacles of boundary crossing took a toll on some participants, who failed to make it on time; and even more importantly, a number of invitations were turned down by those computer scientists who choose another allegiance over their allegiance to apolitical algorithmic thinking, namely rising concerns regarding the violation of human rights in the Soviet Union.[54]

The logistics of the conference itself made plainly obvious that "algorithmic thinking" was neither natural nor a single force that could sustain community building. Moreover, there was an important nuance to the like-mindedness among the community members. The shared elements of Knuth's and Ershov's visions for the potential of algorithmic thinking itself were limited. The differences among their larger goals were no less prominent than the similarities. Notably, more like Edmund Berkeley, Ershov believed that

---

using Algol-like formalisms. The transformation of mathematical notation is a familiar device in the history of mathematics, highlighting the moment of interpretation built into the representation.

54. For an overview of the conference, including the perspective of the local community and stressing the limits of the interdisciplinary interactions, see Ksenia Tatarchenko, "Algorithm's Cradle: Commemorating Al Khwarizmi in the Soviet History of Mathematics, Cold War Computing, and Socialist Uzbekistan," in *Writing Histories of Ancient Mathematics, 18th–21st Centuries*, edited by Karine Chemla et al., manuscript. For a discussion of mathematical rationality, dissidence, and human rights in the Soviet Union, see Benjamin Nathans, "The Dictatorship of Reason: Aleksandr Vol'pin and the Idea of Rights under 'Developed Socialism'," *Slavic Review* 66, no. 4 (2007): 630–63. For an actor-produced memoir about the human rights movement within the American computing community, see Jack Minker, *Scientific Freedom and Human Rights: Scientists of Conscience During the Cold War* (Washington, DC: IEEE Computer Society Press, 2012).

computer specialists had a social mission to improve and uplift their home societies.[55] For Ershov, cultivating programming skills could strengthen any human intellect by improving one's ability to express and plan the actions. He was to turn this belief into a plan for the Soviet digital future.

2. FROM LOGOS TO ACTION: ONTOLOGY OF INFORMATION AND A NEW "NEW MAN"

Ershov saw little contradiction between his efforts to establish procedural thinking as a natural foundation for a new international discipline, and his goal to spread algorithmic thinking country-wide as a path to digital communism. A few facts about his personal and professional trajectory help explain his commitments. The extensive documentation left by Ershov—including vast correspondence, articles in popular science magazines, and dairies—testifies that he was hardly a double-faced cynic; on the contrary, his many activities resulted from a firm belief in a possibility of a better world.

From his experience of the German occupation in Ukraine during World War II to his integration into Moscow intelligentsia as a student of mathematics and programming at Moscow University in the mid-1950s, and later, throughout several decades of managing his own software group at the Computer Center of the Novosibirsk Akademgorodok, Ershov became deeply attached to the ideal of producing knowledge for social good and the good of his country.[56] His meetings with the American members of the research group developing the international algebraic computer language, Algol, in the late 1950s, was a life-defining event both for his intellectual orientation in computing and his espousal of scientific internationalism. In his poetry, such as this poem from January 1983, Ershov directly expressed the burden of the responsibilities he took upon himself in words combining biblical and Soviet cultural references: "I searched for knowledge in many

55. The similarity is not accidental. Berkeley was interested in socialism and visited the Soviet Union during his travels in Europe in 1934. For more on the interest in the Soviet experiment among the Western intellectual elite during the interwar era, see Katerina Clark, *Moscow, the Forth Rome: Stalinism, Cosmopolitanism, and the Evolution of Soviet Culture, 1931–1941* (Cambridge, MA: Harvard University Press, 2011); and Michael David-Fox, *Showcasing the Great Experiment: Cultural Diplomacy and Western Visitors to the Soviet Union, 1921–1941* (Oxford: Oxford University Press, 2011).

56. For a Russian-language biography, see I. A. Kraineva and N. A. Cheremnykh, *Put' programmista* (Novosibirsk: Nauchnoe izdatel'stvo SO RAN, 2011).

hidden places / so that the human does not live by bread alone."[57] This commitment to serving homeland was eventually materialized as a computer education reform that introduced the study of the generic Algorithmic Language. This language was defined as a form of notation for algorithms in textbooks and a descendant of Algol 60, but was also a tool for thinking and developing the skills of algorithmization among all Soviet youth.[58] This part outlines the evolution of Ershov's ideas from a distant professional ideal to an agenda for the national reform.

Already in 1972, Ershov concluded his speech in Atlantic City with a declaration that the "highest" professional obligation was to spread the gospel of programming to all humanity:

> In the past ages, the ability to read and write was considered a rare, God-given gift, the destiny of a limited group of the specially chosen. In the present age of general literacy, we perceive reading to be a universally attainable accomplishment, but we are tempted to single out a new elite group, who become arbiters between the lay generality of mankind and the arcane informational model of the world hidden in the machine. Is it not however the highest aesthetic ideal of our profession to make the art of programming public property, and thereby to submerge our exclusiveness within a mature mankind?[59]

Despite the apparent grandiloquence of such words, Ershov also knew the power of small-scale actions on the ground. The reputation he earned as an

57. The line is a reference to one of the defining novels of the Thaw. See Denis Kozlov, "Naming the Social Evil: The Readers of *Novyi mir* and Vladimir Dudinstev's "*Not by Bread Alone*, 1956–59 and beyond," in *The Dilemmas of De-Stalinization: Negotiating Cultural and Social Change in the Khrushchev Era*, ed. Polly Jones (London and New York: Routledge, 2006), 80–98. Ershov's recurrent employment of religious expressions was most probably connected to his family traditions and his self-identity as a member of Russian intelligentsia. The end of the poem reads: "Carrying my own heavy cross, I have no way to know: will they crucify me or exalt to the heavens?" Ershov's usage of religious language could be situated within the larger context of the checkered relationship between the Soviet State and the Orthodox Church, including their temporary alliance during the WWII, when Ershov was a child. See Dimitry Pospielovsky, "The 'Best Years' of Stalin's Church Policy in the light of Archival Documents," *Religion, State and Society* 25, no. 2 (1997): 139–62. For the argument that the Soviet state's atheism amounted to taking on the church's roles and symbols, see Victoria Smolkin-Rothrock, "A Sacred Space Is Never Empty: The Soviet Atheism, 1954–1971" (PhD dissertation, University of California, Berkeley, 2010).

58. For an example, see A. P. Ershov, "Algoritmicheskii iazyk v shkol'nom kurse osnov informatiki i vychisletel'noi tekhniki," *Mikroprocessornye sredstva i sistemy*, no. 2 (1985): 48–51.

59. Ershov, "Aesthetics and Human Factor" (ref. 37), 505.

international authority on computing boosted his domestic legitimacy, helping him to sponsor unconventional initiatives, such as a non-academic interest in teaching children to program. Through the 1970s, a group at Ershov's department at the Akademgorodok Computer Center conducted experiments with programming education.[60] Ershov's department became home to a group of education and computing enthusiasts gathered from different Soviet computer hubs. In 1979, in a 30-page report entitled *School Informatics*, Ershov and two coauthors summarized the results of a decade of work at local after-school programs, via distant education, and regional summer schools for young programmers.[61] The successes of Siberian educational experiments confirmed the belief of Ershov and his team that children could indeed learn to program. Their report from 1979 concluded by proposing the introduction of informatics as a compulsory subject in Soviet secondary education and spelling out how to accomplish that kind of educational reform.[62]

In 1980, Ershov was invited to deliver a keynote address at the Third World Conference on Computer Education to be held in Lausanne, Switzerland, on July 27–31, 1981. Accepting this opportunity to speak to thousands of international experts in education, computing, and policy-making, Ershov produced a short popular account of his vision for computer education under the title of "Programming, the Second Literacy." The speech not only appeared in the official proceedings but was reprinted in many other venues and in several languages.[63] The phrase gained some international currency, thanks in part to

60. For a recent collective work focusing on the pedagogical experiments of the late Soviet period, see Il'ia Kuklin, Mariia Maiofis, and Petr Safranov, *Ostrova utopii: Pedagogicheskoe i sotsial'noe proektirovanie poslevoennoi shkoly (1940–1980e)* (Moskva: Novoe Literaturnoe Obozrenie, 2015).

61. A. P. Ershov, G. A. Zvenigorodskii, and Iu. A. Pervin, *Shkol'naia informatiki (kontseptsii, sostoianie, perspektivy)* (Novosibirsk, 1979). For an organizers' account of the Akademgorodok summer school for young programmers from the 1970s till the 2000s, see A. G. Marchuk, T. I. Tikhonova, and L. V. Gorodnyaya, "Novosibirsk Young Programmer's School: A Way to Success and Future Development," *Perspectives on Soviet and Russian Computing: First IFIP WG 9.7 Conference, SoRuCom 2006, Petrozavodsk, Russia, July 3–7, 2006, Revised Selected Papers*, ed. John Impagliazzo and Eduard Proydakov (Heidelberg, New York: Springer, 2011), 228–34.

62. Professionals were not the only Soviet group preoccupied with personal computers; on Soviet personal computer enthusiasts, see Zbigniew Stachntak, "Red Clones: The Soviet Computer Hobby Movement of the 1980s," *IEEE Annals of the History of Computing* 37, no. 1(2015): 12–23.

63. Andrei 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. In Russian: *Programmirovanie – vtoraia gramostnot'* (Novosibirsk: VTS SO AN SSSR 1981); reprinted in *Kvant*, no. 2 (1983): 2–7; in *Ekonomika i org. prom. pr-va*, no. 2 (1982): 143–46;

the growing number of publications on computers and education, before being forgotten when an instrumental understanding of computer literacy became the norm.[64] Before that happened, the phrase "programming, the second literacy" turned into the oft-recited slogan of the Soviet computer literacy campaign and a subject of heated contestation.

What did Ershov have in mind with these bold words, and what were their consequences? Ershov plainly spelled out some of the answers in the paper delivered in Lausanne. A close reading of the text reveals its underlying ontology of information and a conception of the human self that explains how a comparison articulated in 1972 could metamorphose into the 1985 reform slogan. From his 1972 professional ideal, to his 1980 computer education concept, Ershov was consistently preoccupied by the transformative potential of programming as human-machine communication. To promote that idea, he broadened his reach from close colleagues, to professional society, to education specialists and policy makers.

The opening of his speech, "Programming, the Second Literacy," was self-reflective. It was preoccupied with the perilous nature of the metaphor and the seeming incompatibility of its basic components, "literacy" and "programming":

> The title of this talk is, as I fully realize, a metaphor which may sound rather risky. On the one side of the copula is the name of an exotic, though already very popular, profession which requires specific abilities and an extensive training, while on the other side is the general virtue, the most fundamental property of modern man.[65]

The connections that Ershov established between literacy and programming were based on historicity, technological origins, and humans' "natural" cognitive capacity. They reflected the structural organization of the talk as a transition from the world of books, to the world of computers, to the world of programs, to the computer at school.

---

in *Chelovek i mashina* (Moskva: Znanie, 1985), 16–24. There were also Bulgarian, Czech, and Estonian versions.

64. For an overview of diverse meaning of computer literacy during the early 1980s, including references to Ershov's work, see Jack Culbertson, "Whither Computer Literacy," *Microcomputers and Education*, ed. Jack Culbertson and Luvern L. Cunningham (Chicago: University of Chicago Press, 1986), 109–30, on 112.

65. Andrei Ershov, "Programming, the Second Literacy." Citations are based on the online version of "Programming, the Second Literacy" at http://ershov.iis.nsk.su/archive/eadocbody.asp?btl=2&btid=265 (accessed 15 Jun 2018).

Ershov's Lausanne speech traced a historical progression from medieval to modern times, from literacy to programming. The timeline featured two major technological inventions, separated by five centuries—the printing press and the computer.[66] The printing press led to the spread of literacy. Likewise, the diffusion of computers enabled by the advent of microprocessors would require the universalism of programming skills. Both gave humans a way to enhance their cognitive capacity. But the relationship between the two skills was not seen as exactly linear. According to the speech, reading and writing stimulated the inherent human ability of abstraction, but also endangered the integrity of word and action by fostering a mindset of "bookishness," or knowledge separated from reality.

In this text, Ershov built upon the notion of procedural or algorithmic thinking to develop his ideas about goal-oriented, operational behavior empowered by programming practice. Whereas in the pre-print world, "the moment of truth was sensed not in the instant of mental inspiration, but after a goal had been achieved," programming, Ershov believed, would be able to restore the balance between mind and reality that print had destroyed when it separated knowledge from its applied context.[67] To program was to find a balance between logos and action, or as Ershov phrased it, quoting Shakespeare, "Suit the action to the word, the word to the action."[68] According to this framework, to program is to become one who both seeks truth and creates it. In Ershov's eyes, then, a programmer was nothing short of a poet, a poet with god-like powers of creation. This slippage from the initial analogy to the claims that programming skills amount to creation of a new reality was deliberate: "literacy and programming art . . . also supplement each other, forming a new harmony of human mind."[69]

It was this transition from analogy to direct claims that supports the text's circular structure and gives it a distinctive rhetorical power. Ershov concluded his talk with the same metaphor, "programming, the second literacy." This time, the expressive power of a metaphoric copula did not serve to reveal the

66. Although the chronological arc associating the printing press with electronic media became representative of media studies since the works of Marshall McLuhan, in the case of Ershov, such a timeline is best understood as a product of the Marxist approach to periodization and historical change. Marshall McLuhan, *The Gutenberg Galaxy: The Making of Typographic Man* (Toronto: University of Toronto Press, 1962).

67. Ershov, "Programming, the Second Literacy" (ref. 65).

68. Ibid. Ershov quotes from Shakespeare's *Hamlet*, Act 3, Scene 2.

69. Ibid.

qualities of its components; instead it was there to mobilize for an imperative agenda: to "shorten a child's way to intellectual maturity, increase his activity, improve his preparedness for occupational performance."[70] The metaphor became a call for action—to teach children to program and to enable them to become creative participants in the era of this "second" industrial revolution: the age of computerization and automation.

At the core of this move between knowing and making was a particular understanding of the human self and information. According to Ershov, both possessed the capacity to belong simultaneously to two typically separate existential realms, the natural and the artificial.[71] The book and the computer shared a distinct characteristic that separated them from other things and commodities: they belonged to an infinite dimension. However, "Books and computers are different [exceptional]. Both books and computers embody an information model of the world in all its multivariousness and changeability."[72] According to this transgressive ontological status, the computer was not a mere tool but a "partner" in human evolution. The very necessity for such acts of transcendence implied moral valences.

In his speech Ershov outlined the threats of modern society, "escapism and passivity," and spelled out a solution: "We . . . call for a more active and responsible attitude to life. But what is it like? Plainly speaking, it is an ability to formulate a program of action and to follow it." If passivity was the result of a recognizably Western, bourgeois self—"when educated and cultured people lack an active attitude toward life"—then learning to program was part of

70. Ibid.

71. For a discussion of the disciplinary struggles over the "ownership" of information, its meanings and ontology between Ershov and the All-Union Institute for Scientific and Technical Information (VINITI), see: Iu. Iu. Chernyi, "Kak ponimal informatiku akademik Andrei Petrovich Ershov," in *Trudy SORUCOM-2011* (Novgorod, 2011), 341–49. There was another important disciplinary boundary that Ershov had to negotiate, between informatics (*informatika*), cybernetics, and mathematics. The historical study of this question is beyond the purview of this article. Ershov himself addressed the issue on multiple occasions, for an example see A. P. Ershov, "Stanoviashchaiasia nauka," *Nauka v Sibiri*, 11 Sep 1986. The issue of boundary-making had a direct effect on Ershov's choice to sideline earlier Soviet works on programmable education, such as those by Lev Landa; see Lev Landa, *Algorithmization in Learning and Instruction* (Englewood Cliffs, NJ: Educational Technoogy Publications, 1974). On the intellectual and institutional context of Soviet cybernetics and American psychology shaping these Soviet pedagogical research projects, see Ekaterina Babintseva, "The Management of the Mind: American Influence on the Soviet Psychology of Learning," paper presented at "Midwest Junto for History of Science," Bloomington, Indiana, 24–26 Mar 2017.

72. Ershov, "Programming, the Second Literacy" (ref. 65).

forming a new, virtuous person. Such a self-actualized human would have the power to overcome dichotomies. "The second literacy," wrote Ershov, "is not only the ability to write computer instructions, but also the way to bring up a man who is resolute and prudent at the same time."[73]

With these words Ershov turned programming from something low and technical into a high and morally significant occupation. The authority of such a transformation was corroborated on several levels: tables and numbers, encyclopedic definitions, and references to the work of Seymour Papert, the American pioneer of artificial intelligence and computer education for children. However, Ershov framed his text not as a contemporary debate among specialists, but as an ongoing conversation about human virtues in Western civilization. His citations come from the Bible, Shakespeare, Comenius, Montaigne, Goethe, Molière, Kipling, Gorky, and even Arthur Clarke.[74]

Describing his own text as an "invasive sermon" and aiming to impress and entertain his audience, Ershov also requested the collaboration of an illustrator. Mikhail Zlatkovsky, who then served as the artistic editor at the popular science journal *Khimia i Zhizn'* (Chemistry and Life) and would go on to become a celebrated political cartoonist in the 1990s, accepted the work and created 32 graphics.[75] A number of Zlatkovsky's images not only illustrate, but complicate and subvert Ershov's text. In one image, a perplexed man opens his bodysuit and beholds unruly tapes of biological code sprawling out of his chest. The illustration makes Ershov's reference to the human organism as "stuffed with programs" both more meaningful and more disturbing.[76] The zipper opening calls to mind the act of dissection and

73. Ibid. For an analysis of virtues and technics for cultivating the Soviet "self" in a broader historical context, see Oleg Kharkhordin, *The Collective and the Individual in Russia: A Study of Practices* (Berkeley, CA: University of California Press, 1999).

74. All the authors mentioned by Ershov were part of the Soviet classics. In 1980, Arthur Clarke was not yet the controversial author he would become with his 1982 *2010: Odyssey Two*. Clarke's works were popular in the Soviet Union; his authority was based on his role as a popularizer of science and the scientifically literate descriptions of space travel in his fiction. However, the Russian translation of his 1968 novel *2001: Space Odyssey* appeared with significant cuts in the ending, considered too mystical for the Soviet reader. See I. A. Efremov, "O romane Artura Klarka 'Kosmicheskaia Odisseia 2001 goda'," *Molodaia Gvardia*, no. 10 (1970), 260–62.

75. For more on Zlatkovsky, see his website at http://www.zlatkovsky.ru (accessed 15 Jun 2018).

76. Lily Kay studied the incorporation of cybernetic discourse into biology; see Lily E. Kay, *Who Wrote the Book of Life?: A History of the Genetic Code* (Stanford University Press, 2000).

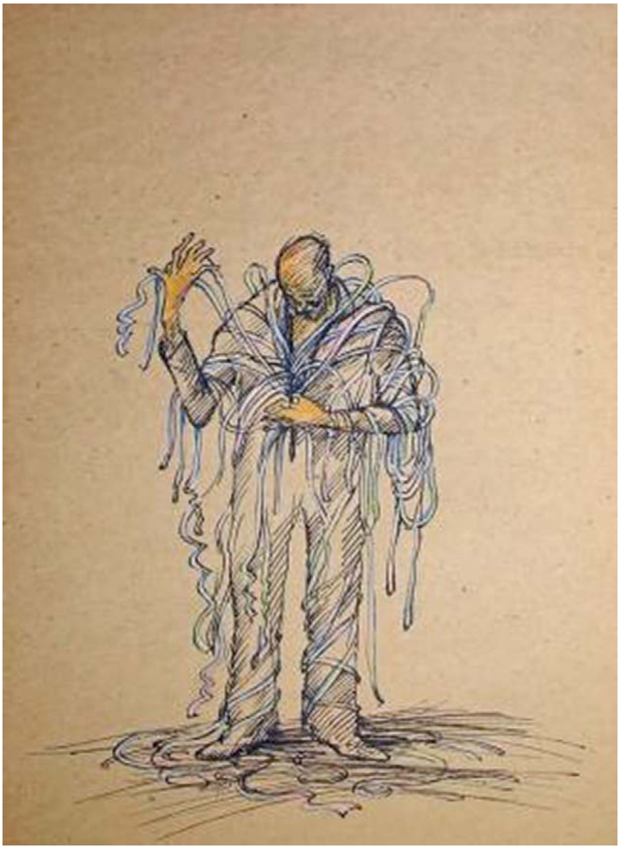

FIG. 1.

points to potential violence involved in acts of (self-)inspection of the human body and mind (Fig. 1).[77]

Similarly, Ershov's stress on self-control exercised via self-awareness, achieved thanks to programming, could be denigrated into a closed loop of self-censure. Another image Ershov used in his presentation depicts a human puppet pulling his own string (Fig. 2).

Finally, the key visual citation of the speech is the victorious runner (Fig. 3). A man cutting through a perforated tape at the finish line represents the

77. Figures 1, 2, and 3 can be found in *Academician A. Ershov's Archive*, http://ershov.iis.nsk.su/ru/second_literacy/article. The potential violence associated with self-inspection is explicit in the etymology of the "analysis." The word "analysis" is a noun of action, first meaning "a breaking up, a loosening, releasing," from the Greek verb "lyein," or "to unfasten."

FIG. 2.

generation of computer literate people while calling to mind the early Soviet representations of young athletes.

This image of the virtuous programmer of the future, coming straight down from the Soviet iconography of a "new man," highlights the political message behind Ershov's text. "Programming, the Second Literacy" was a manifesto for the revolution of the mind. Famously, print was "the arm of the proletariat," and Lenin had said that the illiterate person "stands outside politics."[78] In the 1920s, Soviet literacy campaigns were meant to bring about a new kind of humanity.[79]

78. Quoted in Peter Kenez, *The Birth of the Propaganda State: Soviet Methods of Mass Mobilization, 1917–1929* (Cambridge: Cambridge University Press, 1985), 72.

79. A monograph-length study is Charles E. Clark, *Uprooting Otherness: The Literacy Campaign in NEP-era Russia* (Selinsgrove, PA: Susquehanna University Press, 2000). For an overview encompassing a larger chronological period, see Ben Eklof, "Russian Literacy Campaigns

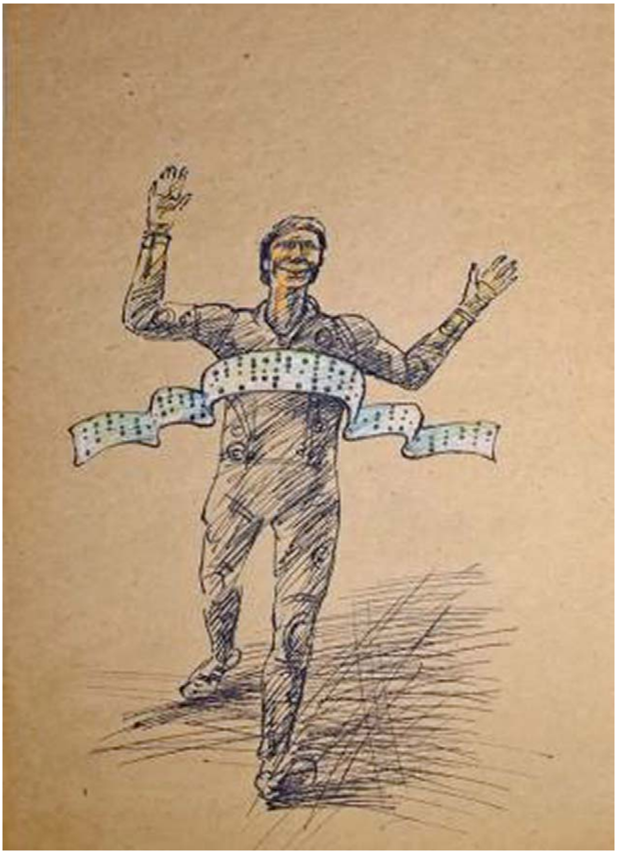

FIG. 3.

In the 1980s, Ershov borrowed from this Soviet cultural baggage to recreate that revolutionary enthusiasm for his own skill of choice, programming. The rhetorical coupling of programming with literacy allowed Ershov to align and accommodate three revolutionary discourses associated with the Soviet Revolution of 1917, the Scientific-Technical Revolution of the 1960s, and the Personal Computer Revolution of the 1980s. Spelled out as a metaphor, "Programming, the Second Literacy" was already a slogan, a public move from concept to realization, from the world of ideas to the world of politics.

1861–1939," in *National Literacy Campaigns and Movements: Historical and Comparative Perspectives*, ed. Robert F. Arnove and Harvey J. Graff (New Brunswick, NJ: Transaction Publishers, 2008), 123–45. For an early discussion of the social implications of the movement, see Sheila Fitzpatrick, *Education and Social Mobility in the Soviet Union 1921–1934* (Cambridge: Cambridge University Press, 1979), 168–77.

On September 1st, 1985, ninth-graders began to study a new subject, "The Basics of Informatics and Computing Technology." Textbooks were to be printed, teachers trained, and computers delivered to some 60,000 schools across the fifteen Soviet republics, all tasks suffering numerous setbacks. Despite the infamous Soviet reputation for centralization, the scale of the endeavor led to a great variety of local experiences: one indication of the reform's range and scope is that great efforts had to be made to translate and publish the textbook in the national languages of Soviet republics. At first, schools received only a few computers and relied on provisional solutions such as using paper tools, programmable calculators, and shared computing facilities. In many locations, students, teachers, and parents alike had a poor understanding of the goal of the curriculum developed by Ershov and his team, which went beyond concrete skills and aimed instead to develop "algorithmic thinking." Even before the official advent of *glasnost*, the new government's policy of open discussion, enthusiasts and critics of the reform became invested in the debate over the meaning of computer literacy, a debate that took place across various types of media and even spilled beyond Soviet borders. Covering a wide range of issues, from the choice of hardware to the social implications of human-machine interaction, the debate—even more than the reform itself—revealed a society preoccupied with information technology, modes of Socialist lifestyle, and its own future.[80] The Soviet Union's collapse only a few years after the implementation of computer education reform does not lessen its relevance.[81] Although the debates about "algorithmic thinking" ceded their

80. For more on Soviet press as a space of important debates during the Thaw, see Denis Kozlov, *The Readers of* Novyi Mir*: Coming to Terms with the Stalinist Past* (Cambridge, MA: Harvard University Press, 2013). The 1980s debates over computer literacy pose some methodological puzzles (as it took place across a variety of media including different paper-based outlets and film) and requires a separate analysis. For a contemporaneous overview, see Stephen T. Kerr, "Educational Reform and Technological Change: Computing Literacy in the Soviet Union," *Comparative Education Review* 35, no. 2 (1991): 222–54. For a comparative perspective, see Margo Boenig-Liptsin, "Making Citizens of the Information Age: A Comparative Study of the First Computer Literacy Programs for Children in the United States, France, and the Soviet Union, 1970–1990" (PhD diss., Harvard University, 2015).

81. The importance of the reform for the actual computational practice during the last years of the Soviet Union is significant. I have traced the reform's influence on the formation of communities among the users of programmable calculators and their coordination via popular science magazines; see Ksenia Tatarchenko, "'The Man with a Micro-calculator': Digital Modernity and Late Soviet Computing Practices," in *Exploring Early Digital: Communities and Practices*, ed. Thomas Haigh, manuscript; and "Right to be Wrong: Gaming, Science Fiction, and Cybernetic Imaginary in *Kon-tiki: A Path to the Earth* (1985–1986)," manuscript.

place to the debates about "computational thinking," the reform's main question is still with us: Who is the true citizen of the Information Age?

Today, the question provokes discussions around two main axes. One familiar notion is that of the "digital divide" and of the poverty separating the "haves" from the "have-nots," those with little or no access to industrial technology. Another is that of Big Data generated by computers-turned-ubiquitous-media-machines, and its influence on our pubic, private, and political identities. However, as was discussed above, the issues of access and of digital selfhood did not surface with the dot-com bubble. The two parts of this essay have demonstrated that both Cold War professional and socialist identities were influenced by questions about the nature of human-machine interaction. As opposed to the capitalist Information Society predicated on the personal possession of computers as black-boxed commodities, its socialist double revolved around a national diffusion of programming skills in pursuit of a new kind of virtuous citizenry excelling at self-expression and self-control.

CONCLUSION: ALGORITHMIC PASSAGES

Algorithmic thinking was not an ideal form of Cold War rationality but an actor category mobilized for a variety of scenarios involving human and machine. Asking who and how used "algorithmic" to characterize the activity of the mind contributes to a better understanding of the historical values and agency associated with the notion of the algorithm. The described commonalities, interdependencies, and differences defining the goals of the historical actors reveal how this notion was polarized within a set of oppositions familiar to modern societies in general: between nature and nurture, between expert and lay knowledge, and between access to information technologies versus control over them. At the same time, the modalities of "algorithmic" and "thinking" conjuncture also lay bare the embeddedness of these rhetorical usages, their attachment to the particular figures and their passages at particular locations, from Stanford to Urgench, from Novosibirsk to Atlantic City, from Lausanne to Moscow.

In summary, the algorithm's multiple roles—a subject of mathematical definition, an element of computer science's theoretical corpus, or a school textbook flowchart visualization, to name just a few—depended on its movement across different communities. But the algorithm did not travel across disciplinary, national, or generational divides on its own; it was carried back and forth by an international scientific community that crossed the Cold War divide. Therefore, when used as an adjective describing modes of cognition, the algorithm acts as a historical tracer: it testifies to the extend of the bottom-up agency of individuals and communities at particular historical conjunctures. Recognizing such traces is the key step to reflect on the significance of "Cold War" in Cold War computing and to account for the central role that the algorithm acquired during the quarter century following the end of the bipolar world.

Recent works in Science and Technology Studies map the entangled geographies of expertise transcending the international politics of the nation-state and demonstrate the powerful legacies of Cold War technopolitics in today's globalized world.[82] But the Socialist histories of computing have little role in these conversations.[83] This essay is not an argument against the scholarship that highlighted the divisive role of the proverbial Iron Curtain—such as politicized discursive milieu and military build-up, as well as incompatible economic regimes and export controls. Rather, it demonstrated that such descriptions are incomplete: computing expertise and visions of Information Societies were predicated on an entanglement between the competing technological modernities. Beyond mathematical "war tools" operating within and generating closed spaces of military installations, or "efficiency tools," the data processing commodities diffused from the American center to global periphery, Cold War computing was also a pursuit to redefine the scenarios of

82. See Gabrielle Hecht, ed., *Entangled Geographies: Empire and Technopolitics in the Global Cold War* (Cambridge, MA: MIT Press, 2011); Naomi Oreskes and John Krige, eds., *Science and Technology in the Global Cold War* (Cambridge, MA: MIT Press, 2014); and Ksenia Tatarchenko and Christopher J. Phillips, "Mathematical Superpowers: The Politics of Universality in a Divided World," *HSNS* 46, no. 5 (2016): 549–55. The most influential account of Cold War as a global phenomenon is Odd Arne Westad, *The Global Cold War* (Cambridge: Cambridge University Press, 2007).

83. The overviews in computing history typically omit the Soviet experience altogether; for an example, see Martin Campbell-Kelly, William Aspray, Nathan Ensmenger, and Jeffery R. Yost, *Computer: A History of the Information Machine* (Philadelphia: Westview Press, 2014). Moreover, post–Cold War, English-language historical scholarship on the subject is scarce: Slava Gerovitch, *From Newspeak to Cyberspeak: A History of Soviet Cybernetics* (Cambridge, MA: MIT Press, 2002); and Benjamin Peters, *How Not to Network a Nation: The Uneasy History of the Soviet Internet* (Cambridge, MA: MIT Press, 2016). For a non-Soviet version of a socialist cybernetic utopianism, see Eden Medina's history of the project Cybersyn, *Cybernetic Revolutionaries: Technology and Politics in Allende's Chile* (Cambridge, MA: MIT Press, 2011). She has recently revisited this subject as a historical case of algorithmic regulation; "Rethinking Algorithmic Regulation," *Kybernetes* 44, no. 6/7 (2015): 1005–19.

human-machine interaction.[84] The notion of algorithmic thinking was one of the key sites for such redefinitions.

If computers were the new "machines of the Cold War," they also partook in generating Cold War subjectivities. Acknowledging the notion of "algorithmic thinking" as an actor category reopens the question raised by the coauthors of *How Reason Almost Lost Its Mind: The Strange Career of Cold War Rationality*. As the two superpowers competed, human-machine systems could embody instrumental rationality, where "algorithmic" stood for evacuating human judgment in multiple expert communities operating in the world under the shadow of nuclear apocalypse.[85] This is not how the computer scientists, who defined their new discipline as the pursuit of knowledge about such systems, represented algorithmic rationality, however. They drew on both the enlightenment tradition of self-control and the romantics' quest for self-expression to imagine a universal utopia of human and machine brotherhood. The key point here is not simply to attest to a coexistence of various Cold War rationalities brought about by the computer, but to call attention to the loss of the explanatory power when imposing artificial analytical and geographical limits on the relations connecting the algorithm to rationality.[86]

Unlike the familiar notions of "closed world" and "mechanical rationality," the trajectory of "algorithmic thinking" across the East-West political divide

84. See Paul Edwards, *The Closed World* (ref. 2); Atsushi Akera, *Calculating a Natural World: Scientists, Engineers, and Computers During the Rise of U.S. Cold War Research* (Cambridge, MA: MIT Press, 2008). For a more recent work on computers as military tools, see Rebecca Slayton, *Arguments that Count: Physics, Computing, and Missile Defense, 1949–2012* (Cambridge, MA: MIT Press, 2013). James Cortada has published extensively on computers as data processing tools; for a global history of computer diffusion, see James W. Cortada, *The Digital Flood: The Diffusion of Information Technology across the U.S., Europe, and Asia* (New York: Oxford University Press, 2012). For a transnational study emphasizing the local dynamics of appropriation, see Corinna Schlombs, "Productivity Machines: Transatlantic Transfers of Computing Technology and Culture in the Cold War" (PhD Dissertation, University of Pennsylvania, 2010). On human-machine coupling, see David Mindel, *Between Human and Machine: Feedback, Control, and Computing before Cybernetics* (Baltimore, MD: Johns Hopkins University Press, 2002); and Ron Kline, *Or Why We Call Our Age the Information Age* (Baltimore, MD: Johns Hopkins University Press, 2015).

85. Paul Erickson, Judy L. Klein, Lorraine Daston, Rebecca Lemov, Thomas Sturm, and Michael D. Gordin, *How Reason Almost Lost Its Mind: The Strange Career of Cold War Rationality* (Chicago: Chicago University Press, 2013). For a case study of expert systems for international relations, see Joy Rohde, "Pax Technologica: Computers, International Affairs, and Human Reason in the Cold War," *Isis* 108, no. 4 (2017): 792–813.

86. For a discussion of larger stakes in the history of mathematics associated with algorithm and mathematical reasoning, see Karine Chemla, "What is at Stake in Mathematical Proofs from Third-Century China?" *Science in Context* 10, no. 2 (1997): 227–51.

and across professional and public spheres outlines the embroiled contours of Cold War computing and the interdependence of capitalist and socialist subjectivities. The variety of meanings associated with the phrase "algorithmic thinking" developed as a result of personal encounters, intertextual references, and shared ideas. However, the very same actors, who made many efforts to cross the divide, contributed to maintaining the separation as they worked to realize alternative visions of the Information Society. The improbable paths taken by the idea of the algorithm, the "algorithmic passages" mapped out in this essay, situate the past experiences of relationships between humans and between humans and machines on their own terms. Recent studies of post-Soviet computing communities reveal that these past experiences directly shape the current expertise in digital technology.[87]

Thus, the Cold War travels of the algorithm shed light on our present-day concerns; they disturb the circular narrative that computerization and globalization are intrinsically coupled. Viewing computerization and Big Data from the perspective of these algorithmic passages is a way out of the teleological march of information technologies and a way to grasp more fully the nature of digital power. The task of the transnational and global history of computing is to localize the claims of universality and to demonstrate how digital technologies operate at the intersection of mathematical abstraction with material and social dimensions: algorithms are to be performed; minds come with bodies; and technological changes beget social conflicts.[88]

87. Mario Biagioli and Vincent Lepinay, *From Russia with Code: Russian Computer Scientists Abroad* (Durham, NC: Duke University Press, 2019, forthcoming).

88. Transnational history remains both a promising and a challenging endeavor. See the reflections relating to different areas of scholarship: for Soviet history, Michael David Fox, "The Implications of Transnationalism," *Kritika* 12, no. 4 (2001): 885–904; for history of technology, E. B. A. van der Vleuten, "Toward a Transnational History of Technology: Meanings, Promises, Pitfalls," *Technology and Culture* 49, no. 4 (2008): 974–94; for history of science, Simone Turchetti, Nestor Herran, and Soraya Boudia, eds., "Transnational History of Science," special issue, *BJHS 45*, no. 3 (2012); for history of computing, Corinna Schlombs, "Toward International Computing History," *IEEE Annals of the History of Computing* 28, no. 1 (2006): 108, and Fabian Prieto-Ñañez, "Postcolonial Histories of Computing," *IEEE Annals of the History of Computing* 38, no. 2 (2016): 2–4.