



Computing Cultures: Historical and Philosophical Perspectives

Juan Luis Gastaldi¹

Published online: 13 December 2023

© The Author(s), under exclusive licence to Springer Nature B.V. 2023

1 From the Collective...

Computing has come a long way since Charles Babbage devised the idea that most, if not all, numerical calculations could be performed by machinery through one uniform process. In more than one sense, that trajectory is one of unparalleled technical and scientific progress. However, some facets of that process lend themselves less to such an evolutionary understanding. This is especially true of the successive images that our culture has projected onto computing machinery to address the latter's capabilities and shortcomings as well as its promises and dangers.

In the past two centuries, the image of computing in Western culture has undergone a major turn when a social and collective perspective was displaced by a mental and solipsistic one. Indeed, the birth of the modern idea of automatic computers in the work of Babbage was inseparable from the specific driving forces of the Industrial Revolution, at the top of which are the mechanization of human work and the division of labor. The details of this story have been the object of many historical and philosophical studies (e.g., Daston, 1994; Grattan-Guinness, 2003; Grier, 2005; Sack, 2019; Schaffer, 1994). This perspective reveals that, when Babbage considered replacing the work of human computers with calculating engines, the problem was not how to mechanically replicate the inner processes of an individual mind but how to reproduce the organized activity of a whole class of human beings. Computing was not a universal competence of individuals but the sociocultural role of those put to the particular task of performing computations within a brand-new way of organizing knowledge production through the division of labor. As of 1800, computers were a class of (typically unskilled) workers embedded in a sociocultural machine designed, as de Prony, the leader of the French *Bureau de Cadastre*, would say, to “manufacture logarithms as one manufactures pins” (quoted in Grattan-Guinness, 2003).

✉ Juan Luis Gastaldi
juan.luis.gastaldi@gess.ethz.com

¹ D-GESS, ETH Zürich, Clausiusstrasse 59, 8092 Zurich, ZH, Switzerland

As a consequence, Babbage's computing engines were projected after the mechanisms of inherently social processes in culture. Admittedly, to achieve his goal, Babbage could rely on a powerful and equally novel theory of operations, offered by the nascent English abstract or symbolical algebra (Koppelman, 1971). It was this formal "calculus", inspired by the recent advances of Continental analysis, that entertained Babbage's idea that all calculations could be performed through "one uniform process". But when it came to materializing this proto-theory of computation on a machine, the model was no other than the modern factory, aligning the collective work of humans with the efficiency afforded by the principles of the division of labor. Hence the two main components of Babbage's engine: the "mill" and the "store". In Babbage's engines, calculations are not carried out by processing the data saved in memory, but logarithms are manufactured by carrying numbers "from the mill to the store and from the store to the mill" in what was overtly pictured as "a real manufactory of figures" (Menabrea & Lovelace, 1843/2010).

This social model of computing was not limited to material aspects. It is well known that, to "regulate" the operations of the Analytical Engine (in modern words, to program it), Babbage borrowed the solution of using punched cards to the most prominent of all industries of his time: the textile industry, with its Jacquard-loom. Moreover, as Sack (2019) argues, work languages, the languages describing the actions of humans at work, had to be translated into machine language, a process taking place in the artisan's workshops, where one can identify the roots of software practices. It thus appears that, *at the moment of their inception, modern computing machines were devised, constructed, and addressed as a material scale model of cultural features and social relations*. As extravagant as it might sound from a contemporary perspective, the pioneer of computational psychology, Herbert Simon, did not exaggerate when he claimed that "the real inventor [of the digital computer] was Adam Smith" (quoted in Sack, 2019).

The collective image of computing will accompany the development of the Industrial Revolution throughout the 19th century, both mirroring and actively contributing to it. George Boole may very well push the limits of symbolical algebra to encompass logic and characterize his new logical calculus as the "laws of thought", the idea that a mechanical implementation of this calculus could represent "our thinking" will not be any more readily embraced. Toward the end of the century, we can still read none other than Gottlob Frege arguing that "Boolean formula-language only represents a part of our thinking; our thinking as a whole can never be coped with by a machine or replaced by purely mechanical activity" (Frege, 1880/1979). It is not surprising, then, that already in the 20th century, the young Shannon (1936) made no reference whatsoever to thinking machines in his celebrated logical analysis of digital circuits, pointing to automatic telephone exchanges and industrial motor-control equipment instead. So much so that Shannon's earliest work should be understood as offloading the psychological interpretation from Boole's logical calculus rather than investing digital circuits with thinking capabilities.

2 ...to the Individual...

The image described in the previous section will radically change in the years going from the mid-1930s to the mid-1950s. The case of Shannon is again symptomatic. While, in line with his previous work, his seminal theory of information was little, if at all, concerned with individual mental principles, the year 1955 will find him as one of the originators of the Dartmouth proposal for the development of Artificial Intelligence (AI), advancing the idea that “every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it” (McCarthy et al., 2006). Shannon’s personal project within that framework was notably entitled “Application of information theory concepts to computing machines and brain models”. It proposed to address the problem of “mechanized intelligence” by studying “the synthesis of brain models by the parallel development of a series of matched (theoretical) environments and corresponding brain models which adapt to them” (McCarthy et al., 2006).

Notice that the idea that machines can *be made to* simulate intelligence and theoretical principles can *be applied to* brain models implies that *they were not intended for that purpose in the first place*. It was precisely the originality of that somewhat unforeseen application that provided a decisive component of the so-called “cognitive revolution”. But the process leading to it involved a thorough transformation of how computers are perceived.

During those two transitional decades, Gigerenzer (2000) identifies two ways in which the new parallel between computers and humans was drawn. The first one, represented by von Neumann, finds its source in McCulloch and Pitts’s (1943) influential logical model of neural activity. Inspired by such a model, von Neumann (1945/1993) provided his famous description of a computer architecture in analogy with the human nervous system. Thus, the machine’s arithmetical and control units, together with the “memory”, corresponded, in his view, to the associative neurons, while the input and output “organs” were equivalent to the sensory and motor neurons. But a more decisive turning point in our story came from the second approach, represented by the work of Alan Turing. Because while McCulloch and Pitts’s model provided von Neumann with an appealing analogy of which he nonetheless acknowledged the limitations, Turing’s (1937) picture of “a man in the process of computing” actively guided the specification of his famous machine. Thus, after considering the “behavior” of a human computer as determined by his “state of mind” when observing symbols, Turing set out to “construct a machine to do the work of this computer”.

There is no need to stress the significance of Turing’s evocative image for the history of computing up to the present day. In particular, we know that his notion of a “mechanical procedure” associated with his picture of an individual human computer convinced Kurt Gödel of the generality of the definitions of computability previously advanced by him and Alonzo Church, as well as of their capacity to characterize what it means to compute *in general* (i.e., the Church-Turing thesis). We also know how the limits of the formal system underlying this general

characterization of computing (i.e., Gödel's undecidability results) encouraged Turing to further his approach in the direction of "intelligent machinery" which, unlike logical systems, should be allowed to make mistakes, as long as they could be progressively rectified through a training procedure. In all this quest, Turing's avowed guiding principle consisted in "following the human model as closely as we can", relying on a simplified version reduced to a brain without a body, and focusing on training it to "react in a disciplined manner to orders" (Turing, 1948/2004). Cultural aspects are not altogether absent from his new machine, nonetheless. They show up at the very end of his construction, in the form of a "search" for techniques in a space defined by the solutions found by other men, in a process that "must be regarded as carried out by the human community as a whole, rather than by individuals" (Turing, 1948/2004).

Significantly, Turing observed that by organizing an initially "unorganized machine" through training, a universal computer can still be achieved. In a movement that would have brought joy to Hegel, through Turing's image, computing became the place where the individual could be raised to the universal. At any rate, it is in this universal form that the individual image of computing has prevailed until today.

3 ...and Back?

As any contemporary historian knows, Babbage and Turing could be arguably viewed as mere names amidst complex historical dynamics. However, taken as pioneering expressions of their respective epochs, they serve here the purpose of illustrating our main point, namely, that *the modern development of computing has been accompanied by a major shift in what it means for a machine to compute in our culture*, going from a model inspired by *collective* mechanisms and social relations to one based on solipsistic mental principles of biological *individuals*.

However, as we advanced at the beginning of these pages, the nature of this change is such that, unlike other transformations concerning the science and the practice of computing, *it cannot be assessed in terms of improvement, evolution, progress, or even decline and regression*. For there is no objective property of computing upon which the choice of one image or the other could be grounded. Notice that, by Turing's own admission, Babbage's Analytical Engine "was a universal digital computer" (Turing, 1950), making the defining property of a computer in our times independent of the individual or collective image we project on it.¹ The transition from one to the other is thus unmotivated by the objective principles of computing. In some cases, even the opposite can be argued: it is the choice of this or

¹ Turing's claim on this point has been the object of some discussion, pointing, among others, to the fact that, unlike Babbage's engine, in Turing's machine "the mechanisms used in reading a program are of the same kind as those used in executing it" (Gandy, 1988). Assuming this difference to be relevant, it would be difficult to pinpoint its source in the individual image that guided Turing's view.

that image of computing that can circumscribe a domain of experience and objects against which to assess the principles and properties of computing as objective.

Transformations of this kind have been thematized many times in the history and philosophy of science: paradigm shifts (Kuhn, 1970), changes of positivity (Foucault, 2002), differences in styles (Hacking, 1992), or maybe even conceptual changes (Posner et al., 1982). At each time, scholars have emphasized the incongruity and potential hazards associated with interpreting these transformations through an evolutionary lens. Rather than a constant and cumulative improvement, such changes resemble a *shift in perspective*: They can reveal hitherto unseen facets of an object, but cannot do so without concealing others that were previously fully accessible. To such an extent that one cannot expect to widen the perspective without performing a continual back and forth.

The emergence of an individual image of computing in our culture is unquestionably one of those shifts in perspective. As such, it certainly provided new insights and opened new horizons. Many significant effects in the theory and practice of computing can be attributed to them, such as advances in the research of programming languages, the construction of autonomous systems, or the idea of neural models of computing. But also fruitful developments in other fields, such as those brought about in linguistics and psychology through the cognitive revolution. However, the new horizons were gained at the cost of concealing or obscuring other equally essential facets of computing. An eloquent example is provided by Berry (2010), suggesting that Turing machines are conceived after the image of one man with only one pencil, with the troublesome consequence that parallelism remains excluded from his universal computer model by design. One could also think about problems related to network structures, software legacy, and pragmatic aspects of computing at large. Likewise, while the new perspective promoted novel developments in disciplines like psychology and cognitive linguistics, it hindered potential connections with other fields, such as anthropology, sociology, sociolinguistics, history, and in general, any non-cognitive approach to the humanities and the social sciences.

The risks of relying on a partial perspective on computing should be obvious. Although our culture may have embraced an individualistic image of computing, the collective principles and mechanisms that inform computer science and practices have not disappeared as a result. Neither have the effects of computing on our societies. Prior to implementing their computational model of the mind on a digital computer, Allen Newell and Herbert Simon simulated their program on “a computer constructed of human components”. Simon observed that “It was the task of each participant to execute his or her subroutine, or to provide the contents of his or her memory, whenever called by the routine at the next level above that was then in control” (quoted in Gigerenzer, 2000). As Gigerenzer (2000) points out, this situation looks no different from the one set up by de Prony’s at his *Bureau*. But we can indeed see a difference in the fact that *the concrete social relations between actual humans remain now obscured*—to them no less than to any observer—by the idea that the computer they collectively implement is a scale model of the brain instead of a scale model of society. Those relations do not appear now as corresponding to a social structure, but to a natural process disconnected from any other goal than the

rational and disinterested objective of accomplishing a task (here, the “purest” of all: proving a theorem). Newell and Simon’s experiment is not just a simple anecdote. We know how cybernetics raised the technologically-centered reorganization of social relations under the model of a task-oriented rational individual to the rank of a credo. More concretely, in his recent book, Gugerli (2022) offers a remarkable insight into how private and public institutions were transformed by the widespread adoption of general-purpose computers in the 1960s. The case of NASA’s Mission Control Center is a notable example, where the technical requirements of real-time computing for space travel motivated a complete institutional reorganization which “became a model, even an obsession, for people who liked to monitor, control, and command” (Gugerli, 2022).

Since the early days of the individual turn, the gap between sociocultural aspects of computing and the individual lens through which we understand them has continued to widen. Nowhere is this gap more visible than in the recent developments of AI. Take, for instance, the emblematic case of Large Language Models (LLMs). Even the most elementary analysis should be able to bring to light that these computational models explore and reproduce the regularities in the expression of *collective* linguistic practices as recorded in a digital textual corpus. If only because of its sheer size, necessitated by the inherent characteristics of the computational model, no individual could have possibly produced such a corpus, nor processed it as part of a training. Let alone the social conditions of the human labor involved at the different stages of those model’s production. Whatever the computational principles underlying the production of an LLM, the collective nature of their source, results, and consequences is inescapable, and its cultural inscription would be obvious even for Turing under his 1948 description.

And yet, even if “the social theory of meaning we need to understand this technology took shape long before the technology itself”, as Underwood (2023) correctly points out, even if many insightful attempts have been made to renew those theories in the face of the development of digital technologies (e.g., Maniglier, 2011; Manovich, 2020), even if an increasing number of recognized leaders in the field of AI have started casting doubts on a “mental imagery” mostly fueled by science fiction (e.g., Bottou & Schölkopf, 2023), such models keep being untiringly presented, addressed, and assessed as individual intelligent agents, endowed with individual minds, supposedly grounded on individual brain-like processes. Given the obscurity in which our current image of computing leaves the sociocultural conditions of our computational practices, it should come as no surprise that a convincing critical approach is struggling to take form. Indeed, the philosophical discussion surrounding LLMs in recent years has ended up being organized around the “AI Consciousness” (e.g., Chalmers, 2023) vs. “Stochastic Parrots” (Bender et al., 2021) axis. If the arguments were to be summarized in two words, one could say that the former argues that since these computational models can capture meaning, they are like humans, while the latter maintains that since these models are not like humans, they cannot capture meaning. It thus becomes evident how both positions remain trapped in a perspective where computational models of language are, and cannot but be, computational models of individual linguistic agents. For that reason, they both fail to see what a collective image of computing associated with a social

theory of meaning would readily see: that computational models of language, by their very (non-individual) nature, might help to reveal cultural principles of meaning; and that, far from being a weakness, that is precisely where the strength of these computational models lies.

The case of LLMs is exemplary of a prevalent issue in the current state of the AI field, where the whole spectrum of collective issues is urged to be addressed from the individualistic standpoint of an “ethics”, thus hampering the development of an epistemology, a sociology, an anthropology, or even a politics of AI. Although AI is far from exhausting the diverse range of computing practices of our time, it reveals the potential effects of an overly narrow idea of what computers are and what place they occupy among us. During the same twenty years in which we have seen the remarkable developments of a new AI wave, we have also witnessed a no less spectacular rise of global challenges such as climate change, the increase of inequalities, the resurgence of authoritarianism, the outbreak of a pandemic, and the proliferation of war. Primarily focused on developing individual services like personalized medicine, personal assistants, and self-driving cars, AI models’ positive contribution to addressing the global challenges of our era remains to be seen. And there are reasons to suspect that we will not see it without *bringing the culture back to the machine*. For those challenges are collective in nature. No individual can reverse climate change, no matter how much garbage they recycle; no individual can redistribute global wealth, overturn an authoritarian regime, prevent the spread of a deadly disease, or stop a war. And on this point, for once, we can be sure that what is true of individual human agents is also true of computational ones.

4 For the History and Philosophy of Computing

Under the title “Computing Cultures: Historical and Philosophical Perspectives”, the present Special Issue aims to enrich our image of computing by bringing the cultural dimension back at the center of computing practices, devices, and knowledge. At a moment when it has become impossible to experience our collective lives without computers, a solipsistic understanding of computing falls short of the necessary intelligibility that our situation requires. The hope is that by revealing the multiple cultural aspects of computing, we can help to reduce the gap between what computers do among us and what we think they do, ultimately contributing to addressing the entanglement of computing practices with the main cultural challenges our epoch is facing.

The global and collective nature of such challenges requires a comprehensive perspective on computing. Not only have cultural phenomena increasingly become the object of computational analysis, but computational practices have also proven inseparable from the cultural environment in which they evolve. For this reason, bringing the culture back into computing cannot mean replacing our current image of computing with the one that once accompanied its emergence. It means, more profoundly, multiplying the perspectives in such a way that we simultaneously see that culture can not only be the object of computing but also its condition. Those multiple perspectives should reveal that, for any aspect of

computing, there are as many cultural effects as there are cultural circumstances that contribute to making it possible. The deliberately ambiguous title “Computing Cultures” intends to convey this double sense where culture is both the object and the subject of computing: cultures that are computed, cultures that compute.

In this pursuit, the constitution of a transdisciplinary environment is crucial. A space where art is as necessary as engineering, anthropological insights as important as psychological models, and the critical perspectives of history and philosophy as decisive as the axioms and theorems of theoretical computer science. For more than a decade, the DHST/DLMPST Commission for the “History and Philosophy of Computing” (HaPoC) has contributed to constructing such an interdisciplinary community and environment. HaPoC strives to bring together historians, philosophers, computer scientists, social scientists, designers, manufacturers, practitioners, artists, logicians, mathematicians, each with their own experience and expertise, to take part in the collective construction of a comprehensive and diverse image of computing.

This Special Issue follows the 6th edition of the HaPoC biennial Conference, which was hosted at the ETH Turing Centre in Zurich, Switzerland, in October 2021. It features papers stemming from presentations selected in that event, as well as contributions from authors outside the Conference. The papers can be seen as a representative sample of the kind of approach developed by the HaPoC community, exhibiting, at once, high diversity, solid expertise, and critical perspective. However, their diversity does not contradict their coherent contribution to the whole, connecting, as with a dotted line, the entire space that we have just traversed. Thus, Friedman (2023) focuses on the traditional connection between computing and weaving practices, showing that it can be traced as far back as to Leibniz’s work. Daylight (2023) proposes an original reading of the nature of Turing’s claims concerning his computational models. Astarte (2023) and Jones (2023) reveal the many paths explored in the search for a theory of concurrency that the lack of parallelism in Turing machines left as an open problem. Waszek (2023) performs a critical assessment of Simon and Larkin’s computational approach to cognitive psychology. Carissimo and Korecki (2023) discuss the limitations of applying optimization processes valid for formal and computational systems to complex and open social systems. Finally, Buda and Primiero (2023) provide an enhanced model of computational artifacts incorporating pragmatic dimensions inherent to any computational device.

Every editing project has its own set of regrets, and this one is no exception. Despite our best efforts to encourage submissions from the most diverse areas in the humanities, the final result does not include contributions from disciplines such as anthropology, sociology, economy, literature, or art, whose perspectives would have been precious to this issue. The few submissions within this scope did not make it through the rigorous reviewing process. This situation is indicative of the prevailing state of research in academia and industry, where the consistent crossings of disciplinary boundaries, and the institutions that encourage them, are the exception and not the rule. Yet another proof that the task of HaPoC is more relevant today than ever before.

Funding This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under Grant Agreement No 839730

References

- Astarte, T. K. (2023). From Monitors to Monitors: A Primitive History. *Minds and Machines*. <https://doi.org/10.1007/s11023-023-09632-2>
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency* (pp. 610–623). Association for Computing Machinery.
- Berry, G. (2010). Penser, modéliser et maîtriser le calcul informatique. Course at the Collège de France. Retrieved 15 October, 2023, from <https://www.college-defrance.fr/fr/agenda/cours/penser-modeliser-et-maitriser-le-calcul-informatique>
- Bottou, L., & Schölkopf, B. (2023). Borges and AI. arXiv:2310.01425
- Buda, A. G., & Primiero, G. (2023). A Pragmatic Theory of Computational Artefacts. *Minds and Machines*. <https://doi.org/10.1007/s11023-023-09650-0>
- Carissimo, C., & Korecki, M. (2023). Limits of Optimization. *Minds and Machines*. <https://doi.org/10.1007/s11023-023-09633-1>
- Chalmers, D. J. (2023). Could a Large Language Model be Conscious? arXiv:2303.07103
- Daston, L. (1994). Enlightenment Calculations. *Critical Inquiry*, 21(1), 182–202.
- Daylight, E. (2023). True Turing: A Bird's-Eye View. *Minds and Machines*. <https://doi.org/10.1007/s11023-023-09634-0>
- Foucault, M. (2002). *Archaeology of Knowledge*. Routledge.
- Frege, G. (1880/1979). Boole's Logical Calculus and the Concept-Script. In *Posthumous writings* (pp. 9–52). Blackwell.
- Friedman, M. (2023). Leibniz and the Stocking Frame: Computation, Weaving and Knitting in the 17th Century. *Minds and Machines*. <https://doi.org/10.1007/s11023-023-09623-3>
- Gandy, R. (1988). The Confluence of Ideas in 1936. In *A Half-century Survey on the Universal Turing Machine* (pp. 55–111). Oxford University Press, Inc.
- Gigerenzer, G. (2000). *Adaptive Thinking: Rationality in the Real World*. Oxford University Press.
- Grattan-Guinness, I. (2003). The Computation Factory: De Prony's Project for Making Tables in the 1790s. In *The History of Mathematical Tables: From Sumer to Spreadsheets*. Oxford University Press.
- Grier, D. A. (2005). *When Computers Were Human*. Princeton University Press.
- Gugerli, D. (2022). *How the World Got into the Computer. The Emergence of Digital Reality*. Chronos Verlag.
- Hacking, I. (1992). 'Style' for Historians and Philosophers. *Studies in History and Philosophy of Science Part A*, 23(1), 1–20. [https://doi.org/10.1016/0039-3681\(92\)90024-Z](https://doi.org/10.1016/0039-3681(92)90024-Z)
- Jones, C. B. (2023). Three Early Formal Approaches to the Verification of Concurrent Programs. *Minds and Machines*. <https://doi.org/10.1007/s11023-023-09621-5>
- Koppelman, E. (1971). The Calculus of Operations and the Rise of Abstract Algebra. *Archive for History of Exact Sciences*, 8(3), 155–242.
- Kuhn, T. S. (1970). *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press.
- Maniglier, P. (2011). Processing Culture: 'Structuralism' in the History of Artificial Intelligence. In S. Franchi & F. Bianchini (Eds.), *The Search for a Theory of Cognition: Early Mechanisms and New Ideas* (pp. 149–175). Amsterdam: Rodopi.
- Manovich, L. (2020). *Cultural Analytics*. The MIT Press.
- McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955. *AI Magazine*, 27(4), 12.
- McCulloch, W. S., & Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*. <https://doi.org/10.1007/BF02478259>
- Menabrea, L. F., & Lovelace, A. A. (1843/2010). Sketch of the Analytical Engine, with Notes by the Translator, Ada Augusta, Countess of Lovelace. In H. P. Babbage (Ed.), *Babbage's Calculating Engines: Being a Collection of Papers Relating to Them; Their History and Construction* (pp. 6–50). Cambridge University Press.

- Posner, G. J., Strike, K. A., Hewson, P. W., & Gertzog, W. A. (1982). Accommodation of a Scientific Conception: Toward a Theory of Conceptual Change. *Science Education*, 66(2), 211–227.
- Sack, W. (2019). *The Software Arts*. The MIT Press.
- Schaffer, S. (1994). Babbage's Intelligence: Calculating Engines and the Factory System. *Critical Inquiry*, 21(1), 203–227.
- Shannon, C. E. (1936). *A Symbolic Analysis of Relay and Switching Circuits (Master of Science)*. MIT: Department of Electrical Engineering.
- Turing, A. (1937). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2–42(1), 230–265. <https://doi.org/10.1112/plms/s2-42.1.230>
- Turing, A. (1948/2004). *Intelligent Machinery (1948)*. The Essential Turing. Oxford University Press.
- Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, LIX(236), 433–460.
- Underwood, T. (2023). The Empirical Triumph of Theory. Retrieved 15 October, 2023, from <https://critiq.wordpress.com/2023/06/29/the-empirical-triumph-of-theory/>
- von Neumann, J. (1945/1993). First Draft of a Report on the EDVAC. *IEEE Annals of the History of Computing*, 15(4), 27–75. <https://doi.org/10.1109/85.238389>
- Waszek, D. (2023). Informational Equivalence but Computational Differences? Herbert Simon on Representations in Scientific Practice. <https://doi.org/10.1007/s11023-023-09630-4>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.