Notes

Introduction

1. Process thought refers to a wide and heterogeneous body of philosophical works that share similar sensibilities toward associations, sometimes also called relations (Barad 2007; Butler 2006; Dewey [1927] 2016; James [1912] 2003; Latour 1993b, 2013; Mol 2002; Pickering 1995; Serres 1983; Whitehead [1929] 1978). For process thinkers, as Introna put it (2016, 23), "relations do not connect (causally or otherwise) pre-existing entities (or actors), rather, relations enact entities in the flow of becoming." What things are is what they become in association to other entities, the association itself being part of the process. The emphasis is then put on the "how" rather than the "what": instead of asking what *is* something, process thinkers would rather ask how something *becomes*. This ontology is then about continuous performances instead of binary states. The present volume embraces this ontology of becoming.

2. At the end of the book, a glossary briefly defines technical terms used for this investigation (e.g., actant, collective world, constitution, course of action).

3. This unconventional conception of the social has been initially developed and popularized by Madeleine Akrich, Michel Callon, and Bruno Latour at the Centre de Sociologie de l'Innovation (Akrich, Callon, and Latour 2006; Callon 1986). It is important to note that even though this theoretical standpoint has somewhat made its way through academic research, it remains shared among a minority of scholars.

4. As pointed out by Latour (2005, 5–6), the Latin root *socius* that denotes a companion—an associate—fits well with the conception of the social as what emanates from the association among heterogeneous entities.

5. What connects the practitioners of the heterogeneous research community of *Science and Technology Studies* is the conviction that science is not just the expression of a logical empiricism; that knowledge of the world does not preexist; and that scientific and technological truths are dependent on collective arrangements, instrumentations, and dynamics (Dear and Jasanoff 2010; Jasanoff 2012). For a comprehensive introduction to STS, see Felt et al. (2016).

6. It is important to note that this lowering of capacity to act does not concern the sociology of attachments that precisely tries to document the appearance of delighted objects, as developed by Antoine Hennion (2015, 2017). At the end of chapter 5, I will discuss the important notion of attachment.

7. The notion of "composition"—at least as proposed by Latour (2010a)—is, in my view, an elegant alternative to the widely used notion of "governance." Both nonetheless share some characteristics. First, both notions suppose heterogeneous elements put together—collectives of humans, machines, objects, companies, and institutions trying to collaborate and persevere on the same boat. Second, they share the desire of a common world while accepting the irreducibility of its parts: for both notions, the irreducible entities that constitute the world would rather live in a quite informed community aware of different and competitive interests than in a distrustful and whimsical wasteland. Both composition and governance thus share the same basic topic of inquiry: how to step-by-step transform heterogeneous *collectives* into heterogeneous *common worlds*? Third, they both agree that traditional centralized decisional powers can no longer achieve the constitution of common worlds; to the verticality of orders and injunctions, composition and governance prefer the horizontality of compromises and negotiations. Yet they nonetheless differ on one crucial point: if governance still carries the hope of a smooth—yet heterogeneous—*cosmos*, composition promotes the need of a laborious and constantly readjusted *kakosmos* (Latour 2010a, 487). In other words, if control is still an option for governance, composition is committed to the always surprising "made to do" (Latour 1999b). It is this emphasis on the constant need for creative readjustments that makes me prefer the notion of "composition" over "governance."

8. The next two paragraphs derive from Jaton (2019, 319–320).

9. The single term "algorithm" became increasingly common in the Anglo-American critical literature from the 2000s onward. It would be interesting to learn more about the ways by which the term "algorithm" has come to take over other alternative terms (such as "software," "code," or "software-algorithm") that were also synonymously used in the past, especially in the 1990s.

10. In Jaton and Vinck (submitted), we closely consider the specific dynamic of the recent politicization of algorithms.

11. This controversy has been thoroughly analyzed in Baya-Laffite, Beaude, and Garrigues (2018).

12. As we will see in the empirical chapters of this book, it is not clear whether we should talk about computer scientists or engineers. But as the academic field of computer science is now well established, I choose to use the generic term "computer scientist" to refer to those who work every day to design surprising new algorithms.

13. For thorough discussions on this topic, see Denis (2018, 83–95).

14. Does it mean that "objective knowledge" is impossible? As we will see in chapters 4, 5, and 6, drawing such a conclusion is untenable: despite the irremediable limits of the inscriptions on which scientific practices heavily rely, these practices nonetheless manage to produce certified objective knowledge.

15. In their 2004 paper, Law and Urry build upon an argument initially developed by Haraway (1992, 1997).

16. This partly explains some hostile reactions of scientists regarding STS works on the "construction of scientific facts." On this topic, see Latour (2013, 151–178).

17. For recent examples, see Cardon (2015) and Mackenzie (2017).

18. In chapter 5, I will discuss at greater length the crucial importance of scientific literature for the formation of certified knowledge.

19. The term "infra-ordinary," as opposed to "extra-ordinary," was originally proposed by Pérec (1989). The term was later taken up in Francophone sociology, notably by Lefebvre (2013).

20. See, for example, Bishop (2007), Cormen et al. (2009), Sedgewick and Wayne (2011), Skiena (2008), and Wirth (1976). I will discuss some of these manuals in chapter 1.

21. However, it is crucial to remain alert to the performative aspects of manuals and classes. This topic is well studied in the sociology of finance; see, for example, MacKenzie, Muniesa, and Siu (2007) and Muniesa (2015).

22. This also often concerns social scientists interviewing renowned computer scientists (e.g., Seibel 2009; Biancuzzi and Warden 2009). As these investigations mainly focus on well-respected figures of computer science whose projects have largely succeeded, their results tend to be retrospective, summarized narratives occluding uncertainties and fragilities. On some limitations of biographic interviews, see Bourdieu (1986). On the problematic habit of reducing ethnography to interviews, see Ingold (2014).

23. For a presentation of some of the reasons why scholars started to inquire within scientific laboratory, see Doing (2008), Lynch (2014), and Pestre (2004).

24. On some of the problematic, yet fascinating, dynamics of this rapprochement between computer science and the humanities (literature, history, linguistics, etc.) that gave rise to digital humanities, see Gold (2012), Jaton and Vinck (2016), and Vinck (2016).

25. Among the rare attempts to document computer science work are Bechmann and Bowker (2019), Button and Sharrock (1995), Grosman and Reigeluth (2019), Henriksen and Bechmann (2020), and Mackenzie and Monk (2004). I will come back to some of these studies in the empirical chapters of the book.

26. After a thorough review of the contemporary critical studies of algorithms, Ziewitz (2016) warned that they could be about to reach a problematic impasse. Roughly put, the argument goes as follows: by mainly considering algorithms from a distance and in terms of their effects, critical studies are taking the risk of being stuck in a dramatic loop, constantly rehashing that algorithms are powerful because they are inscrutable, because they are powerful, because they inscrutable, and so on. The present volume can be considered an attempt at somewhat preventing such a drama from taking hold. In the conclusion, when I clarify the political aspect of this inquiry, I come back to this notion of algorithmic drama.

27. Theureau's work is unique in many ways. Building on the French ergonomics tradition (Ombredane and Faverge 1955) and critical readings of Newell and Simon's (1972) cognitive behaviorism as well as Varela's notion of "enactive cognition" (discussed in chapter 3), he has gradually proposed a simple yet effective definition of a course of action as an "observable activity of an agent in a defined state, actively engaged in a physically and socially defined environment and belonging to a defined culture" (Theureau 2003, 59). His analyses of courses of action involved in traffic management (Theureau and Filippi 2000), nuclear reactor control (Theureau et al. 2001), and musical composition (Donin and Theureau 2007) has led him to propose the notion of "courses-of-action centered design" for ergonomic studies.

28. At the beginning of chapter 4, I will briefly consider the problem of "representativeness."

Chapter 1

1. The general issue subtending my research has not fundamentally changed since the date at which I was awarded the research grant.

2. One of the particularities of the CSF was its international focus. During the official events I attended, deans regularly put forward the CSF's capacity to attract foreign students and researchers. This was especially true in the case of the Lab where I was the only "indigenous" scientific collaborator for nearly a year. The lingua franca was in line with this international environment; even though the Lab was located in a French-speaking region, most interactions, presentations, and documents were in English.

3. The history of the development of the charge-coupled device has been documented, though quite partially, in Seitz and Einspruch (1998, 212–228) and Gertner (2013, 250–265).

4. For an accessible introduction to CCDs and image sensors, see Allen and Triantaphillidou (2011, 155–173).

5. CMOS is a more recent variant of CCD where each pixel contains a photodetector *and* an amplifier. This feature currently allows significant size and power reduction

of image sensors. This is one of the reasons why CMOSs now equip most portable devices such as smartphones and compact cameras.

6. It is commonly assumed that the term pixel, as a contraction of "picture element," first appeared in a 1969 paper from Caltech's Jet Propulsion Lab (Leighton et al. 1969). The story is more intricate than that as the term was regularly used in emergent image-processing communities thoughout the 1960s. For a brief history of the term pixel, see Lyon (2006).

7. A digital signal is represented by n number of dimensions depending on the independent variables used to describe the signal. A sampled digital sound is, for example, typically described as a one-dimensional signal whose dependent variables—amplitudes—vary according to time (t); a digital image is typically described as a two-dimensional signal whose dependent variables—intensities—vary according to two axes (x, y), whereas audio-visual content will be described as a three-dimensional signal with independent variables (x, y, t). For an accessible introduction to digital signal processing, see Vetterli, Kovacevic, and Goyal (2014).

8. It was not the only research focus of the Lab. Several researchers also worked on CCD/CMOS architectures and sensors.

9. It is important to note that for digital image processing and recognition to become a major subfield of computer science, digital images first had to become stable entities capable of being processed by computer programs—a long-standing research and development endeavor. Along with the development, standardization, and industrial production of image sensors such as CCDs and, later, CMOSs, theoretical works on data compression—such as those of O'Neal Jr. (1966) on differential pulse code modulation; Ahmed, Natarajan, and Rao (1974) on cosine transform; or Gray (1984) on vector quantization—have first been necessary. The later enrollment of these works for the definition of the now-widespread International Organization for Standardization norm JPEG, approved in 1993, was another decisive step: from that moment, telecommunication providers, software developers, and hardware manufacturers could rely on and coordinate around one single photographic coding technique for digitally compressed representations of still images (Hudson et al. 2017). During the late 1990s, the growing distribution of microcomputers, their gradual increase in terms of processing power, and the development and maintenance of web technologies and standards have also greatly contributed to establishing digital image processing as a mainstream field of study. The current popularity of image processing for research, industry, and defense is thus to be linked with the progressive advent of multimedia communication devices and the blackboxing of their fundamental components operating now as standard technological infrastructure.

10. According to Japan-based industry association Camera & Imaging Products Association (to which, among others, Canon, Nikon, Sony, and Olympus belong), sales of digital cameras have dropped from 62.9 million in 2010 to fewer than

24.25 million in 2017 (Statista 2019). However, according to estimates generated by InfoTrends and Bitkom, the number of pictures taken worldwide increased from 660 billion to 1,200 billion over the same period (Richter 2017). This discrepancy is due, among other things, to the increasing sophistication of smartphone cameras as well as the popularity and sharing functionalities of social-media sites such as Instagram and Facebook (Cakebread 2017).

11. For example, Google, Amazon, Apple, Microsoft, and IBM all propose application programming interface products for image recognition (respectively, Cloud Vision, Amazon Rekognition, Apple Vision, Microsoft Computer Vision, and Watson Visual Recognition).

12. According to 2011 documents obtained by Edward Snowden, the National Security Agency intercepted millions of images per day throughout the year 2010 to develop computerized tracking methods for suspected terrorists (Risen and Poitras 2014). Chinese authorities also heavily invest in facial recognition for security and control purposes (Mozur 2018).

13. See, for example, *International Journal of Computer Vision*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, or *Pattern Recognition*.

14. See, for example, IEEE Conference on Computer Vision and Pattern Recognition, European Conference on Computer Vision, IEEE International Conference on Computer Vision, or IEEE International Conference on Image Processing.

15. Giving an example of the close relationships between academic and industrial worlds regarding image-processing algorithms, Jordan Fisher—chief executive officer of Standard Cognition, a start-up that specializes in image recognition for autonomous checkout—says in a recent *TechCrunch* article (Constine 2019): "It's the wild west—applying cutting-edge, state-of-the-art machine learning research that's hot off the press. We read papers then implement it weeks after it's published, putting the ideas out into the wild and making them production-worthy."

16. In 2016 and 2017, papers from Apple and Microsoft research teams won the best-paper award of the IEEE Conference on Computer Vision and Pattern Recognition, the most prestigious conference in image processing and recognition. Moreover, in 2018, Google launched *Distill Research Journal*, its own academic journal aiming at promoting machine learning in the field of image and video recognition.

17. This is for example the case in Knuth (1997a) where the author starts by recalling that "algorithm" is a late transformation of the term "algorism" that itself derives from the name of famous Persian mathematician Abū 'Abd Allāh Muhammad ibn Mūsa al-Khwārizmi—literally, "Father of Abdullah, Mohammed, son of Moses, native of Khwārizm," Khwārizm referring in this case to a region south of the Aral Sea (Zemanek 1981). Knuth then specifies that from its initial acceptation

as the process of doing arithmetic with Arabic numerals, the term algorism gradually became corrupted: "as explained by the *Oxford English Dictionary*, the word 'passed through many pseudo-etymological perversions, including a recent *algorithm*, in which it is learnedly confused' with the Greek root of the word *arithmetic*" (Knuth 1997a, 2).

18. See, for example, the (very) temporary definition of algorithms by Knuth (1997, 4): "The modern meaning for algorithm is quite similar to that of *recipe*, *process*, *method*, *technique*, *procedure*, *routine*, *rigmarole*."

19. See, for example, Sedgewick and Wade's (2011, 3) definition of algorithms as "methods for solving problems that are suited for computer implementation."

20. See also Cormen et al.'s (2009, 5) definition: "A well-defined computational procedure that takes some value, or set of values, as *input* and produces some value, or set of values, as *output* [being] thus a sequence of computational steps that transform the input into the output."

21. See also Dasgupta, Papadimitriou, and Vazirani's (2006, 12) phrasing: "Whenever we have an algorithm, there are three questions we always ask about it: 1. Is it correct? 2. How much time does it take, as a function of n? 3. And can we do better?" And also Skiena (2008, 4): "There are three desirable properties for a good algorithm. We seek algorithms that are correct and efficient, while being easy to implement."

Chapter 2

1. This chapter expands Jaton (2017). I thank Geoffrey Bowker, Roderic Crooks, and John Seberger for fruitful discussions about some of its topics.

2. Excerpts in quotes are literal transcriptions from audio recordings, slightly reworked for reading comfort. Excerpts not in quotes are retranscriptions from written notes taken on the fly.

3. In chapter 3, I critically discuss the computational metaphor of the mind on which many cognitive studies rely.

4. Studies on attention had already been engaged before the 1970s, notably through the seminal work of Neisser (1967) who suggested the existence of a pre-attentive stage in the human visual processing system.

5. Another important neurobiological model of selective attention method was proposed by Wolfe, Cave, and Franzel (1989). This model of selective attention method later inspired competing low-level feature computational models (e.g., Tsotsos 1989; Tsotsos et al. 1995).

6. The class of algorithms that calculates on low-level features quickly became interesting for the development of autonomous vehicles for which real-time image

processing was sought (Baluja and Pomerleau 1997; Grimson 1986; Mackworth and Freuder 1985).

7. Different high-level detection algorithms can nonetheless be assembled as modules in one same program that could, for example, detect faces and cars and dogs, and so on.

8. At that time, only two saliency-detection algorithms were published, in Itti, Koch, and Niebur (1998) and Ma and Zhang (2003). But the ground truths used for the design and evaluation of these algorithms were similar to those used in laboratory cognitive science. The images of these ground truths were, for example, sets of dots disrupted by a vertical dash. As a consequence, if these first two saliency-detection algorithms could, of course, process natural images, no evaluations of their performances on such images could be conducted.

9. Ground truths assembled by computer science laboratories are generally made available online in the name of reproducible research (Vandewalle, Kovacevic, and Vetterli 2009). The counterpart to this free access is the proper citation of the papers in which these ground truths were first presented.

10. An API, in its broadest sense, is a set of communication protocols that act as an interface among several computer programs. If APIs can take many different forms (e.g., hardware devices, web applications, operating systems), their main function is to stabilize and blackbox elements so that other elements can be built on top of them.

11. For a condensed history of contingent work, see Gray and Suri (2019, 48–63). On what crowdsourcing does to contemporary capitalism, see also Casilli (2019).

12. As Gray and Suri (2019, 55–56) put it: "Following a largely untested management theory, a wave of corporations in the 1980s cut anything that could be defined as 'non-essential business operations'—from cleaning offices to debugging software programs—in order to impress stockholders with their true value, defined in terms of 'return on investment' (in industry lingo, ROI) and 'core competencies.' … Stockholders rewarded those corporations that were willing to use outsourcing to slash costs and reduce full-time-employee ranks."

13. It is important to note, however, that on-demand work is not necessarily alienating. As Gray and Suri (2019, 117) noted: "[on-demand work] can be transformed into something more substantive and fulfilling, when the right mixture of workers' needs and market demands are properly aligned and matched. It can rapidly transmogrify into ghost work when left unchecked or hidden behind software rather than recognized as a rapidly growing world of global employment." Concrete ways to make crowdsourcing more sustainable have been proposed by the National Domestic Workers Alliance and their "Good Work Code" quality label. On this topic, see Scheiber (2016).

14. However, this shared unawareness toward the underlying processes of crowdsourcing may be valued and maintained for identity reasons, for as Irani (2015, 58) noted: "The transformation of workers into a computational service … serves not only employers' labor needs and financial interests but also their desire to maintain preferred identities; that is, rather than understanding themselves as managers of information factories, employers can continue to see themselves as much-celebrated programmers, entrepreneurs, and innovators."

15. Matlab is a privately held mathematical software for numerical computing built around its own interpreted high-level programming language. Because of its agility to design problems of linear algebra—all integers being considered scalars—Matlab is widely used for research and industrial purposes in computer science, electrical engineering, and economics. Yet, as Matlab works mainly with an interpreted programming language—just like the language Python that is now Matlab's main competitor for applied research purposes—its programs have to be translated into machine-readable binary code by an *interpreter* in order to make the hardware effectively compute data. This complex interpretative step makes it less efficient for processing heavy matrices than, for example, programs directly written in compiled languages such as C or C++. For a brief history of Matlab, see Haigh (2008).

16. In chapter 6, we will more thoroughly consider the relationship between groundtruthing and formulating activities.

17. The services of the crowdsourcing company costed the Lab around US$950.

18. The numerical features extracted from the training set were related, among others, to "2D Gaussian function," "spatial compactness," "contrast-based filtering," "high-dimensional Gaussian filters," and "element uniqueness." In chapter 6, using the case of the "2D Gaussian function," I will deal with these *formulating practices*.

19. This can be read as a mild critique of the recent, growing, and important literature on algorithm biases. Authors such as Obermeyer et al. (2019), Srivastava and Rossi (2018), and Yapo and Weiss (2018), among others, show that the results of many algorithms are indeed biased by the preconceptions of those who built them. Though this statement is, I believe, completely correct—algorithms derive from problematization practices influenced by habits of thought and action—it also runs the risk of confusing premises with consequences: biases are not the consequences of algorithms but, perhaps, are one of the things that make them come into existence. Certain biases expressed and materialized by ground truths can and, in my opinion, should be considered harmful, unjust, and wrong; racial and gender biases have, for example, to be challenged and disputed. However, the outcome of these disputes may well be other biases expressed in other potentially less harmful, unjust, and incorrect ground truths. As far as algorithms are concerned, one bias calls for another; hence the importance of asserting their existence and making them visible in order to, eventually, align them with values one wishes algorithms to promote.

20. Edwards (2013) uses the term "data image" instead of "ground truth." But I assume that both are somewhat equivalent and refer to digital repositories organized around data whose values vary according to independent variables (that yet need to be defined).

21. At the end of chapter 6, I will come back to the topic of machine learning and its contemporary labeling as "artificial intelligence."

22. This discussion has been reconstructed from notes in Logbook 3, May–October 2014.

23. However, it is interesting to note that BJ blames the reviewers of important conferences in image processing. According to him, the reviewers tend to privilege papers that make "classical improvement" over those that solve—and thus define—new problems. At any rate, there was obviously a problem in the framing of the Group's paper as the reviewers were not convinced by its line of argument. As a consequence, the algorithm could not circulate within academic and industrial communities and its existence remained, for a while, circumscribed to the Lab's servers.

II

1. In computer science and engineering, it is indeed well admitted that computer programming practices are difficult to conduct and their results very uncertain. On this well-documented topic, see Knuth (2002), Rosenberg (2008), and in a more literary way, Ullman (2012a, 2012b).

Chapter 3

1. My point of departure is arbitrary in the sense that I could have started somewhere else, at a different time. Indeed, as Lévy (1995) showed, the premises of what will be called "von Neumann architecture of electronic computers" can be found not only in Alan Turing's 1937 paper but also in the development of the office-machine industry during the 1920s, but also in the mechanic-mathematical works of Charles Babbage during the second half of the nineteenth century, but also in eighteenth century's looms programmed with punched cards, and so on, at least until Leibniz's work on binary arithmetic and Pascal's calculating machine. The history of the computer is fuzzy. As it only appears "after a cascade of diversions and reinterpretations of heterogeneous materials and devices" (Lévy 1995, 636), it is extremely difficult—in fact, almost impossible—to propose any unentangled filiation. Fortunately, this section does not aim to provide any history of the computer: It "just" tries to provide elements that, in my view, participated in the formation of one specific and influential document: von Neumann's report on the EDVAC.

2. For a more precise account of the design of firing tables in the United States during World War II, see Haigh, Priestley, and Rope (2016, 20–23) and Polachek (1997).

3. More than their effective computing capabilities—they required up to several days to be set up (Haigh, Priestley, and Rope 2016, 26) and their results were often less accurate than those provided by hand calculations (Polachek 1997, 25–27)—an important characteristic of differential analyzers was their capacities to attract computing experts around them. For example, by 1940, MIT, the University of Pennsylvania, and the University of Manchester, England—three important institutions for the future development of electronic computing—all possessed a differential analyzer (Campbell-Kelly et al. 2013, 45–50; Owens 1986). On the role of differential analyzers in early US-based computing research, see also Akera (2008, 38–45).

4. The assembling of the numerous factors affecting the projectiles started at the test range in Aberdeen where the velocities of the newly designed shells were measured (Haigh, Priestley, and Rope 2016, 20).

5. Although the differential equations defining the calculation of shells' trajectories are mathematically quite simple, solving them can be very complicated as one needs to model air resistance varying in a nonlinear manner. As Haigh, Priestley, and Rope (2016, 23) put it: "Unlike a calculus teacher, who selects only equations that respond to elegant methods, the mathematicians at the BRL couldn't ignore wind resistance or assign a different problem. Like most differential equations formulated by scientists and engineers, ballistic equations require messier techniques of numerical approximation."

6. Interesting to note that delay-line storage is originally linked to radar technology. More precisely, one problem of the radar technology in 1942 was that cathode-ray tube displays showed moving *and* stationary objects. Consequently, radar screens translated the positions of planes, buildings, or forests in one same messy picture extremely difficult to read. MIT's radiation laboratory subcontracted the development of a moving target indicator (MTI) to the Moore School in order to develop a system that could filter radar signals according to their changing positions. This was the beginning of delay-line storage technology at the Moore School, that at first had nothing to do with computing (Akera 2008, 84–86; Campbell-Kelly et al. 2013, 69–74). Radar technology also significantly helped the design of British highly confidential Colossus computer in 1943–1944 (Lévy 1995, 646).

7. By 1942, in order to speed up the resolution of ballistic differential equations, only a limited range of factors tended to be considered by human computers at the BRL. By simplifying the equations, more firing tables could be produced and distributed, but the drawback was that their precision tended to decrease (Polachek 1997). Of course, on the war front, once soldiers realized that the first volley was not adequately defined, they could still slightly modify the parameters of the long-distance weapon to increase its precision. Yet—and this is the crucial point—*in between* the first volley and the subsequent ones, the opposite side had enough time to take cover, hence making the overall long-distance shooting enterprise less effective. The

nerve of war was precisely the first long-distance volleys that, when accurate, could lead to many casualties. By extension, then, the nerve of war was also, to a certain extent, the ability to include more factors in the differential equations whose solutions were printed out in firing table booklets (Haigh, Priestley, and Rope 2016, 25).

8. Created in 1940, the National Defense Research Committee (NDRC) united the research laboratories of the US Navy and the Department of War with hundreds of US universities' laboratory. The NDRC initially had an important budget to fund applied research projects that could provide significant advantages on future battlefields. It also operated as an advisory organization as in the case of the ENIAC that was considered nearly infeasible due to the important amount of unreliable vacuum tubes it would require. On this topic, see Campbell-Kelly et al. (2013, 70–72).

9. The history of this contract could be the topic of a whole book. For a nice presentation of its most important moments, see Haigh, Priestley, and Rope (2016, 17–33).

10. Based on a proposal by Howard Aiken, the Harvard Mark 1 was developed by IBM for Harvard University between 1937 and late 1943. Though computationally slow, even for the standards of the time, it was an important computing system as it expressed an early convergence of scientific calculation and office-machine technologies. For a more in-depth history of the Harvard Mark 1, see Cohen (1999).

11. Though its shape varied significantly throughout its existence, the ENIAC was fundamentally a network of different units (accumulators, multipliers, and function tables). Each unit had built-in dials and switches. If adequately configured, these dials and switches could define one single operation; for example, "clear the values of the accumulator," "transmit a number to multiplier number 3," "receive a number," and so on. To start processing an operation, each configuration of dials and switches had to be triggered by a "program line" wired directly to the specific unit. All these "program lines" formed a network of wires connecting all the units for one specific series of operations. But as soon as another series of operations was required, the network of wires had to be rearranged in order to fit the new configurations of dials and switches. For more elements about the setup of ENIAC, see Haigh, Priestley, and Rope 2016 (35–57).

12. Von Neumann tried to hire Alan Turing as a postdoctoral assistant at Princeton. Turing refused as he wanted to return to England (MacRae 1999, 187–202).

13. The Manhattan Project was, of course, highly confidential and this prevented von Neumann from specifying his computational needs with the ENIAC team.

14. As suggested by Akera (2008, 119–120) and Swade (2011), and further demonstrated by Haigh, Priestley, and Rope (2014; 2016, 231–257), the notion of "stored program" is a historical artifact: "the 'stored program concept' was never proposed as a specific feature in the agreed source, the *First Draft*, and was only retroactively adopted to pick out certain features of the EDVAC design" (Haigh, Priestley, and Rope 2016, 256).

15. Shortly after the distribution of von Neumann's *First Draft*, Eckert and Mauchly distributed a much longer—and far less famous—counter-report entitled *Automatic High-Speed Computing: A Progress Report on the EDVAC* (Eckert and Mauchly 1945) in which they put the emphasis on the idealized aspect the *First Draft*. The stakes were indeed high for Eckert and Mauchly: if the idealized depiction of the EDVAC by von Neumann was considered a realistic description of the engineering project, no patent could ever be extracted from it. And this is exactly what happened. In 1947, the Ordnance Department's lawyers decided that the *First Draft* was the first publication on the project EDVAC, hence canceling the patents submitted by Eckert and Mauchly in early 1946 (Haigh, Priestley, and Rope 2016, 136–152).

16. This consideration of programming as an applicative and routine activity can also be found in the more comprehensive reports von Neumann coauthored in 1946 and 1947 with Arthur W. Burks and Herman H. Goldstine at Princeton Institute for Advanced Study (Burks, Goldstine, and von Neumann 1946; Goldstine and von Neumann 1947). In these reports, and especially in the 1947 report entitled *Planning and Coding of Problems for an Electronic Computing Instrument*, the implementation of instruction sequences for scientific electronic calculations is carefully considered. But while the logico-mathematical planning of problems to be solved is presented as complex and "dynamic," the further translation of this planning is mainly considered trivial and "static" (Goldstine and von Neumann 1947, 20). Programming is presented, in great detail, as a linear process that is problematic during its initial planning phase but casual during its implementation phase. What the report does not specify—but this was not its purpose—is that errors in the modeling and planning phases become manifest in the implementation phase (as it was often the case when the ENIAC was put in action), making empirical programming processes more whirlwind than linear.

17. In 1955, to alleviate the operating costs of the IBM 701 and the soon-to-be-released IBM 704, several of IBM's customers—among them Paul Armer of the RAND Corporation, Lee Amaya of Lockheed Aircraft, and Frank Wagner of North American Aviation—launched a cooperative association they named "Share." This customer association, and the many others that followed, greatly participated in the early circulation of basic suites of programs. On this topic, see Akera (2001; 2008, 249–274).

18. For a fine-grained historical account of this real-time computing project named "Whirlwind" that was initially designed as a universal aircraft simulator, see Akera (2008, 184–220).

19. For more thorough accounts of the SAGE project, see Redmond and Smith (1980, 2000), Jacobs (1986), Edwards (1996, 75–112), and Campbell-Kelly et al. (2013, 143–166).

20. According to Pugh (1995), this contract gave IBM a significant advantage on the early computer market.

21. In a nutshell, Thurstone Primary Mental Abilities (PMA) test was proposed in 1936 by Louis Leon Thurstone, by then the first president of the Psychometric Society. Originally intended for children, the test sought to measure intelligence differentials using seven factors: word fluency, verbal comprehension, spatial visualization, number facility, associative memory, reasoning, and perceptual speed. For a brief history of the PMA test and psychometrics, see Jones and Thissen (2007).

22. One important insight of the EDSAC project was to use the new concept of program to initialize the system and make it translate further programs from non-binary instructions into binary strings of zeros and ones. David Wheeler, one of Maurice Wilkes' PhD students, wrote in 1949 such very first program he called "Initial Orders" (Richards 2005). This type of program whose function was to transform other programs into binary (the only code cathode-ray tubes, magnetic core, or microprocessors can interact with) were soon called "assemblers" and cast to linguistic terms such as "translation" and "language" (Nofre, Priestley, and Alberts 2014). During the 1950s, as multiple manufacturers invested in the electronic computer market, many different assemblers were designed, thereby creating important problems of compatibility: as (almost) every new computer organized the accumulator and multiplier registers slightly differently, a new assembler was generally required. The problem lay in the one-to-one relationship between an assembler and its hardware. Since an assembler had one instruction for one hardware operation, every modification in the operational organization of the hardware required a new assembler. Yet—and this was the crucial insight of Grace Hopper and then John Backus from IBM (Campbell-Kelly et al. 2014, 167–188)—if, instead of a program with a one-to-one relationship with the hardware, one could provide a more complex program that would transform lines of code into another program with somehow equivalent machine-instructions, one may be able to stabilize computer programming languages since any substantial modification of the hardware could be integrated within the "transformer" program that lay in between the programmer's code and the hardware. This is the fundamental idea of *compilers*, programs that take as input a program written in so-called high-level computer language and outputs *another* program—often called "executable"—whose content can interact with specific hardware. In the late 1950s, besides their greater readability, a tremendous advantage of the first high-level computer programming languages such as FORTRAN or COBOL over assembly language lay in their compilers whose constant maintenance could compensate and "absorb" the frequent modifications of the hardware. For example, if two different computers both had a FORTRAN compiler—a crucial and costly condition—the same FORTRAN program could be run on both computers despite their different internal organizations.

23. Between 1964 and 1967, IBM invested heavily in the development of an operating system for its computer System 360. The impressive backlogs, bugs, and overheads of this colossal software project made Frederick Brooks—its former manager—call it "a multi-million-dollar mistake" (Brooks 1975).

24. In 1968, an article by cofounder of Informatics General Corporation Werner Frank popularized the idea that the cost of software production will outpace the cost of computer hardware in the near future (Frank 1968). Though speculative in many respects, this claim was fairly reused and embellished by commentators until the 1980s. Though Frank himself later acknowledged that he unintentionally generated a myth (Frank 1983), this story "reinforced a popular perception that programmer productivity was lagging, especially compared to the phenomenal advances in computer hardware" (Abbate 2012, 93).

25. The topic of "logical statement performances" is recurrent in behavioral studies of computer programming, especially during the 1970s. This has to do with a controversy initiated by Edsger Dijkstra over the GOTO statement as allowed by high-level computer programming languages such as BASIC or early versions of FORTRAN (Dijkstra 1968). According to Dijkstra, these branch statements that create "jumps" inside a program make the localization of errors extremely tedious and should thus be avoided. He then proposed "structured programming," a methodology that consists in subdividing programs in shorter "modules" for more efficient maintenance (Dijkstra 1972). Behavioral studies of computer programming in the 1970s typically tried to evaluate the asserted benefits of this methodology.

26. To prove his second incompleteness theorem, Gödel first had to show that any syntaxic proposition could be expressed as a number. Turing's 1937 demonstration highly relied on this seminal insight. On the links between Gödel's incompleteness theorem and Turing's propositions regarding the *Entscheidungsproblem*, see Dupuy (1994, 22–30).

27. Neural networks, particularly those defined as "deep" and "convolutional," have recently been the focus of much attention. However, it is important to note that the notion of neural networks as initially proposed by McCulloch and Pitts (who preferred to use the notion of "networks of neurons") in their 1943 paper, and later taken up by von Neumann in his 1945 report, is very different from its current acceptance. As Cardon, Cointet, and Mazières (2018) have shown, McCulloch and Pitts's neural networks that were initially logical activation functions were worked on by Donald O. Hebb (1949) who associated them with the idea of learning, which was itself reworked by, among others, Frank Rosenblatt (1958, 1962) and his notion of Perceptron. The progressive probabilization of the inference rules suggested by Marvin Minsky (Minsky and Papert 1970), the works on the back-propagation algorithm (Werbos 1974; LeCun 1985; Rumelhart, Hinton, and Williams 1986) and on Boltzmann machines (Hinton, Sejnowski, and Ackley 1984) then actively participated in the association of the notions of "convolution" (LeCun et al. 1989) and, more recently, "depth" (Krizhevsky, Sutskever, and Hinton 2012). The term "neural network" may have survived this translation process but it now refers to very different world-enacting procedures. At the end of chapter 6, I will consider this topic related to machine learning and artificial intelligence.

28. The division between "extended things" and "thinking things" derives, to a large extent, from Cartesian dualism. For thorough discussions of Descartes's aporia, see the work of Damasio (2005).

29. As we saw in chapter 2, saliency detection in image processing is directly confronted with this issue. Hence the need to carefully frame and constrict the saliency problem with appropriate ground truths.

30. One may trace these critics back to the Greek Sophists (Cassin 2014). James (1909) and Merleau-Ponty (2013) are also important opposition figures. In developmental psychology, the "social development theory" proposed by Vygotsky (1978) is also a fierce critic of cognitivism.

Chapter 4

1. To conduct this project, I had to become competent in Python, PHP, JavaScript, and Matlab programming languages.

2. It is important to note that this line-by-line translation is what is experienced by the programmer. In the trajectory of INT and most other interpreters, the numbered list of written symbols is translated into an abstract syntax tree that does not always conserve the line-by-line representation of the Editor.

3. It is difficult to know exactly how INT managed to deal with these three values at T1. It may by default consider that only the first two values of image-size—width and height—generally matter.

4. In the Matlab programming language, every statement that is not conditional and that does not end with an semicolon is, by default, printed by the interpreter in the Command Window. This is different from many other high-level programming languages for which printing operations should be specified by an instruction (typically, the instruction "`print`").

5. In chapter 5, where I will consider the formation of mathematical knowledge, I will more thoroughly examine the shaping of scientific facts as proposed by STS.

6. This may be a limitation of *Software Studies*, as for example presented in Fuller (2008) and in the journal *Computational Culture*. By considering completed code, these studies tend to overlook the practical operations that led to the completion of the code. Of course, this glance remains important as it allows us to consider the performative effects of software-related cultural products, something my action-oriented method is not quite able to do.

7. The successive operations required to assemble chains of reference in the case of program-testing are well documented, though in a literary way, by Ullman (2012b).

8. It is interesting to note that DF's alignment practices would have been greatly facilitated by the next version of Matlab. Indeed, the 2017 version of Matlab's

interpreter automatically recognizes this type of dimension error during matrix incrementation processes and directly indicates the related breakpoint, the line at which the problem occurred (in our case, at line 9).

9. Donald Knuth, one of the most prominent programming theorists, stressed the importance of program intelligibility by proposing the notion of *literate programming*: a computer programming method that primarily focuses on the task of explaining programs to fellow programmers rather than "just" instructing computers.

10. To my knowledge, there are only three exceptions: Vinck (1991), Latour (2006), and Latour (2010b).

11. This discussion has been reconstructed from notes in Logbook 8, November 2015–March 2016.

12. Some STS authors use the term "script" to define these particular narratives that engage those who enunciate them (Akrich 1989; Latour 2013). If I use the term "scenario," it is mainly for sake of clarity as "script" is often used by computer scientists and programmers—and myself in this book—to describe small programs such as PROG.

Chapter 5

1. Here, my style of presentation and use of scenes are greatly inspired by Latour (1987).

2. I am following here Rosental's (2003) book.

3. I am following here the work of MacKenzie (1999).

4. This is taken from Logbook 1, October 2013–February 2014.

5. With their distinction between *apodeixis* (rigorous demonstration) and *epideixis* (rhetorical maneuvering), Platonists philosophers may have initiated such grand narratives (Cassin 2014; Latour 1999). According to Leo Corry (1997), this way of presenting mathematics culminated with Bourbaki's structuralist conception of mathematical truth. On this topic, see also Lefebvre (2001, 56–68). For a philosophical exploration of grand narratives, see the classic book by Lyotard (1984).

6. Yet "likes" and "retweets" that support claims published on Facebook or Twitter may, sometimes, work as significant external allies. On this topic, see Ringelhan, Wollersheim, and Welpe (2015).

7. Before the 1878 foundation of the *American Journal of Mathematics* (AJM), there was no stable academic facility for the publication of mathematical research in the United States (Kent 2008). The situation in England was a bit different: built on the ashes of the *Cambridge and Dublin Mathematical Journal*, the *Quarterly Journal of Pure and Applied Mathematics* (QJPAM) published its first issue in 1855 (Crilly 2004). Yet for both Kempe's and Heawood's papers, the editorial boards of their journals—as

indicated on their front matters—were rather small compared with today's standards: five members for *AJM* in 1879 (J. J. Sylvester, W. E. Story, S. Newcomb, H. A. Newton, H. A. Rowland) and four members for *QJPAM* in 1890 (N. M. Ferrers, A. Cayley, J. W. L. Glaisher, A. R. Forsyth).

8. According to the document in American Association for Artificial Intelligence (1993).

9. See, for example, the *Journal of Informetrics*.

10. In a nutshell, Kempe circumscribed the problem to maps drawn on a plane that contain at least one region called "country" with fewer than six neighbors. He could then limit himself to five cases, countries from one to up to five neighbors. Proving that "four colorability" is preserved for countries with three neighbors was, obviously, not a problem. Yet in order to prove it for countries with four neighbors, Kempe used an argument known as the "Kempe chains" (MacKenzie 1999, 19–20). This argument stipulates that for a country X with four neighbor countries A, B, C, D, two opposite neighbor countries, say A and C, are either joined by a continuous chain of, say, red and green countries, or they are not. If they are joined by such a red-green chain, A can be colored red and C can be colored green. But as we are dealing with a map drawn on a plane, the two other opposite neighbor countries of X—B and C—*cannot* be joined by a continuous chain of blue and yellow countries (one way or another, this chain is indeed interrupted by a green or red country). As a consequence, these two opposite neighbor countries can be colored blue and X can be colored yellow. Four colorability is thus preserved for countries with four neighbors. Kempe thought that this method also worked for countries with five neighbors. But Heawood's figure shows a case of failure of this method where E's red-green region (vertically cross-hatched in figure 5.1) intersects B's yellow-red region (horizontally cross-hatched), thus forcing both countries to be colored red. Consequently, X has to be colored differently than red, blue, yellow, and green. In such a case, four colorability is *not* preserved.

11. On this topic, see the work of Lefebvre (2001).

12. For rhetorical habits in the life sciences, see Latour and Woolgar (1986, 119–148) and Knorr-Cetina (1981, 94–130). For a thorough comparison among scientific disciplines—excluding mathematics—see Penrose and Katz (2010).

13. Despite the efforts made by Serres (1995, 2002).

14. There was, of course, no scientific institution at that time; experimental protocols, peer witnessing, and, later, academic papers are products of the seventeenth century (Shapin and Shaffer 1989). Yet, as Netz (2003, 271–312) showed, theorems written on wax tablets and parchments did circulate among a restricted audience of (very!) skeptical readers.

15. This is at least Netz's (2003, 271–304) hypothesis, supported by the work of Lloyd (1990, 2005). As Latour summarized it: "It is precisely *because* the public life in

Greece was so invasive, so polemical, so inconclusive, that the invention, by 'highly specialized networks of autodidacts', of *another way* to bring an endless discussion to a close took such a tantalizing aspect" (Latour 2008, 449).

16. So surprising that this careful and highly specialized method of conviction mastered by a peripheral community of autodidacts who took great care to stick to forms was soon "borrowed" by Plato and extended to content in order to, among other things, silence the Sophists. This is at least the argument made by Cassin (2014), Latour (1999b, 216–235), and Netz (2004, 275–282).

17. Aristotle seems to be one of the first to compile geometrical texts and systematize their logical arguments (Bobzien 2002). During late antiquity, commentators such as Eutocius annotated many geometrical works and compiled their main results to facilitate their systematic comparisons (Netz 1998). According to Netz (2004), these collections of standardized geometrical compilations further helped Islamic mathematicians such as al-Kwarizmi and Khayyam to constitute the algebraic language.

18. During the late nineteenth century's so-called crisis of foundations in mathematics, the formalist school—headed by David Hibert—tried to establish the foundations of mathematics on logical principles (Corry 1997). This led to famous failures such as Russell and Whitehead's three volumes of *Principia Mathematica* (Whitehead and Russell 1910, 1911, 1913). Thanks to the philological work of Netz, we now better understand why such an endeavor has failed: it was the very practice of mathematics—lettered diagrams carefully indexed to small Greek sentences—that led to the formulation of the rules of logic and not the other way round.

19. Except, to a certain extent, Lefebvre (2001) and Mialet (2012). It seems then that Latour's remark remains true: few scholars have had the courage to do a careful anthropological study of mathematics (Latour 1987, 246).

20. This is taken from Latour (1987, chapter 2) and Wade (1981, chapter 13).

21. This is taken from Pickering and Stephanides (1992) and Hankins (1980, 280–312).

22. Very schematically, peptides are chemical elements made of chains of amino acids. They are known for interacting intimately with hormones. As there are many different amino acids (twenty for the case of humans), there exists—potentially—billions of different peptides made of combinations of two to fifty amino-acids. It is important to note that in 1972, at the time of Guillemin's experiment, peptides could already be assembled—and probed—within well-equipped laboratories.

23. At the time of Hamilton, the standard algebraic notation for a complex number—so-called absurd quantities such as square roots of negative numbers—was $x+iy$, where $i^2=-1$ and x and y are real numbers. These advances in early complex algebra were problematic to geometers: if positive real numbers could be considered measurable quantities, negative real numbers and their square roots were difficult to represent as shapes on a plane. A way to overcome this impasse was to consider x and y as

coordinates of the end point of a segment terminating at the origin. Therefore, "the *x*-axis of the plane measured the real component of a given complex number represented as such a line segment, and the *y* axis the imaginary part, the part multiplied by *i* in the algebraic expression" (Pickering and Stephanides 1992, 145). With this visualization of complex numbers, algebraic geometers such as Hamilton could relate complex geometrical operations on segments and complex algebraic operations on equations. A bridge between geometry and complex algebra was thus built. Yet geometry is not confined to planes: if a two-dimensional segment $[0, x+iy]$ can represent a complex number, there is a priori no reason why a three-dimensional segment $[0, x+iy+jz]$ could not represent another complex number. Characterizing the behavior of such a segment was the stated goal of Hamilton's experiment.

24. Hamilton's inquiry into the relationships between complex number theory and geometry was not a purely exploratory endeavor. As Pickering and Stephanides noted, "the hope was to construct an algebraic replica of transformations of line segments in three-dimensional space and this to develop a new and possibly useful algebraic system appropriate to calculations in three-dimensional geometry" (Pickering and Stephanides 1992, 146).

25. Contrary to Hamilton, ancient Greek geometers could only refer to their lettered diagrams with short but still cumbersome Greek sentences (Netz 2003, 127–167). Along with Greek geometers' emphasis on differentiation, the absence of a condensed language such as algebra—that precisely required compiled collections of geometrical works in order to be constituted (Netz 1998)—may have participated in limiting the scope of ancient Greek geometrical propositions (Netz 2004, 11–54).

26. Regarding these instruments, it is worth mentioning that here we retrieve what we were discussing about in the last section: all of them—except, perhaps, noncommutative algebra—are blackboxed polished facts that were, initially, written claims. Rat pituitary cell cultures, algebraic notations, radioimmunoassays, coordinate spaces and even Pythagoras's theorem all had to overcome trials in order to gain conviction strength and become established, certified facts.

27. This topological characteristic of mathematical laboratories may be a reason why they have rarely been sites for ethnographic inquiries (Latour 2008, 444).

28. Of course, as we saw in chapter 4, such inscriptions are meaningless without the whole series of inscriptions previously required to produce them. It is only by aligning the "final" inscriptions to former ones, thus creating a chain of reference, that Guillemin can produce information about his peptide (Latour 2013, chapter 3).

29. Here we retrieve something we already encountered in chapters 3 and 4: the "cognitive" practice of aligning inscriptions. Just as DF in front of his computer terminal, Brazeau, Guillemin, and Hamilton never stop grasping inscriptions they acquire from experiments. These inscriptions can, in turn, be considered takes suggesting further actions.

30. Again, this is taken from Latour (1987, chapter 2) and Wade (1981, chapter 13).

31. Again, this is taken from Pickering and Stephanides (1992) and Hankins (1980, 280–312).

32. Brazeau and Guillemin published their results in *Science* (Brazeau et al. 1973). After having presented his results at the Royal Irish Academy in November 1843, Hamilton published a paper on quaternions in *The London, Edinburg and Dublin Philosophical Magazine and Journal of Science* (Hamilton 1844). An important thing to note about quaternions is that after Hamilton named them that way, he still had to define the complex quantities k^2, ik, kj, and i^2 in order to complete his system. According to a letter Hamilton wrote in 1865, the solution to this problem—the well-known $i^2=j^2=k^2=ijk=-1$—appeared to him as he was walking along the Royal Canal in Dublin. If this moment was indubitably important, it would be erroneous to call it "the discovery of quaternions" (Buchman 2009). As shown by Pickering and Stephanides (1992), quaternions were already defined as objects before the attribution of values to the imaginary quantities' products. In fact, when compared with the experimental work required to define the problem of these products' values, what happened on Dublin's Royal Canal appears relatively minor.

33. This is the recurrent problem of biographies of important mathematicians; as they tend to use nature to explain great achievements, they often ignore the many instruments and inscriptions that were needed to shape the "discovered" objects. Biographies of great mathematicians are thus often—yet not always (see the amazing comic strip *Logicomix* [Doxiàdis et al. 2010])—unrealistic stories of solitary geniuses chosen by nature.

34. Accepting the dual aspect of nature—the *consequence* of settled controversies as well as the *retrospective cause* of noncontroversial facts—provides a fresh new look at the classical opposition between Platonism and Intuitionism in the philosophy of mathematics. It seems indeed that the oddity of both Platonism—for which mathematical objects come from the outer world of ideas—and Intuitionism—for which mathematical objects come from the inner world of human consciousness—comes from their shared starting point: they both consider certified noncontroversial mathematical facts. Yet as soon as one accounts for controversies in mathematics—that is, mathematics *in the making*—nature from above (the outer-world of ideas) or nature from below (the inner-world of human consciousness) cannot be considered resources anymore as both are precisely what is at stake during the controversies. It is interesting to note, however, that both antagonist unempirical conceptions of the origin of mathematics led to important performative disagreements about the practice of mathematics, notably through the acceptance, or refusal, of the law of excluded middle. On this fascinating topic, see Rotman (2006) and Corry (1997).

35. According to Netz (2004, 181–186), the constant search for differentiation and originality in ancient mathematical texts had the effect of multiplying individual

proofs of similar problems stated differently. In short, Greek geometers were not interested in systems; they were interested in authentic proofs with a specific "aura" (Netz 2004, 58–63).

36. Netz suggests that the polemical dynamics of ancient mathematical texts prevented Greek mathematicians from normalizing their works, demonstrations, and problems. As he noted: "The strategy we have seen so far—of the Greek mathematician trying to isolate his work from its context—is seen now as both prudent and effective. It is prudent because it is a way of protecting the work, in advance, from being dragged into inter-textual polemics over which you do not have control. And it is effective because it makes your work shine, as if beyond polemic. When Greek mathematicians set out the ground for their text, by an explicit introduction or, implicitly, by the mathematical statement of the problem, what they aim to do is to wipe the slate clean: to make the new proposition appear, as far as possible, as a *sui generis* event—the first *genuine* solution of the problem at hand" (Netz 2004, 62–63).

37. To a certain extent, as we will shall see in chapter 6, mathematical software such as Wolfram Mathematica and Matlab can be considered repositories of polished, compiled, and standardized mathematical certified knowledge.

38. Very schematically, a neuron cell is made of three parts. There is first the "dendrite": the structure that allows a neuron to receive an electro-chemical signal. There is then the "cell body": the spherical part of the neuron that contains the nucleus of the cell and reacts to the signal. There is finally the "axon": the extended cell membrane that sends information to other dendrites.

39. It is important to note that the inevitable losses that go along with reduction processes can be used to criticize the products of these reductions. This is exactly what I did in chapter 3 when I was dealing with the computational metaphor of the mind. I used what some reductions did not take into account in order to criticize the product of these reductions.

Chapter 6

1. BJ's face-detection algorithm computes the size of a face as the ratio of the area of the face-detection rectangle to the size of the image; hence the very small size-values of faces in figure 6.3.

2. Remember that this comparison exercise was the main reason why the Group's paper on the algorithm was initially rejected by the committee of the image-processing conference (see chapter 2).

3. It is important to note that this spreadsheet form required not so trivial Matlab parsing scripts written by the Group. The construction of a ground-truth database thus also sometimes requires computer programming practices as described in chapter 4.

4. Napier initiated the theory of logarithms mainly to facilitate manual numerical calculations, notably in astronomy. On this topic, see the old but enjoyable work by Cajori (1913).

5. This discussion was reconstructed from notes in Logbook 2, February 2014–May 2014.

6. With lower-level programming languages such as C or C++, it might be trickier to transform this scenario into a completed program.

7. If it is not time consuming to approximate square roots of positive real numbers, it is more complicated to get precise results. Nowadays, computers start by expressing the positive real number in floating point notation $m * 2^e$ where m is a number between 1 and 2 and e is its exponent (MacKenzie 1993). Thanks to this initial translation, computer languages can then use the Newton-Raphson iteration method to calculate the *reciprocal* of square root before finally multiplying this result with the initial real number to get the final answer. Calculating k-means of five clusters is also not that trivial. It can be summarized by a list of six operations: (1) place five arbitrary random centroids within the given dataset; (2) compute the distances of every point of the dataset from all centroids; (3) assign every point of the dataset to its nearest centroid; (4) compute the center of gravity of every centroid-assigned group of points; (5) assign each centroid to the position of the center of gravity of its group; and (6) reiterate the operation until no centroid changes its assignment anymore.

8. Remember that INT stands for the Matlab interpreter that translates instructions written in the Editor into machine code, the only language that can make processors trigger electric pulses.

9. Information retrieved from Matlab Central Community Forum (MATLAB Answers 2017)

10. This discussion has been reconstructed from notes in Logbook 3, February–May 2014.

11. This discussion has been reconstructed from notes in Logbook 3, February–May 2014.

12. Fei-Fei Li is now a professor at Stanford University. Between 2017 and 2018, she was chief scientist at Google Cloud.

13. Image classification in digital image processing consists of categorizing the content of images into predefined labels. For an accessible introduction to image classification, see Kamavisdar, Saluja, and Agrawal (2013).

14. The beginnings of the ImageNet ground truth project were difficult. As Gershgorn noted it: "Li's first idea was to hire undergraduate students for $10 an hour to manually find images and add them to the dataset. But back-of-the-napkin math

quickly made Li realize that at the undergrads' rate of collecting images it would take 90 years to complete. After the undergrad task force was disbanded, Li and the team went back to the drawing board. What if computer-vision algorithms could pick the photos from the internet, and humans would then just curate the images? But after a few months of tinkering with algorithms, the team came to the conclusion that this technique wasn't sustainable either—future algorithms would be constricted to only judging what algorithms were capable of recognizing at the time the dataset was compiled. Undergrads were time-consuming, algorithms were flawed, and the team didn't have money—Li said the project failed to win any of the federal grants she applied for, receiving comments on proposals that it was shameful Princeton would research this topic, and that the only strength of proposal was that Li was a woman" (Gershgorn 2017).

15. To minimize crowdworkers' labeling errors, Fei-Fei Li and her team asked different workers to label the same image—one label being considered a vote, the majority of votes "winning" the labeling task. However, depending on the complexity of the labeling task—categories such as "Burmese cat" being difficult to accurately identify—Fei-Fei Li and her team have varied the levels of consensus required. To determine these content-related required levels of consensus, they have developed an algorithm whose functioning is, however, not detailed in the paper (Deng et al. 2009, 252).

16. Once assembled, the ImageNet dataset and ground truth did not generate immediate interest among the image recognition community. Far from it: the first publication of the project in the 2009 Computer Vision and Pattern Recognition (Deng et al. 2009) was taken from a poster stuck in a corner of the Fontainebleau Resort at Miami Beach (Gershgorn 2017).

17. In a nutshell, ILSVRC challenges, in the wake of PASCAL VOC challenges, consist of two related components: (1) a publicly available ground truth and (2) an annual competition whose results are discussed during dedicated workshops. As Russakovsky et al. summarized it: "The publically released dataset contains a set of manually annotated training images. A set of test images is also released, with the manual annotations withheld. Participants train their algorithms using the training images and then automatically annotate the test images. These predicted annotations are submitted to the evaluation server. Results of the evaluation are revealed at the end of the competition period and authors are invited to share insights at the workshop held at the International Conference on Computer Vision (ICCV) or European Conference on Computer Vision (ECCV) in alternate years" (Russakovsky et al. 2015, 211).

18. AlexNet, as the algorithm presented in Krizhevsky, Sutskever, and Hinton (2012) ended up being called, has brought back to the forefront of image processing the convolutional neural network learning techniques developed by Joshua Bengio, Geoffrey Hinton, and Yann LeCun since the 1980s. Today, convolutional neural networks for text, image, and video processing are ubiquitous, empowering products

distributed by large tech companies such as Google, Facebook, or Microsoft. Moreover, Bengio, Hinton, and LeCun received the Turing Prize Award in 2018, generally considered the highest distinction in computer science.

19. These criticisms were summarized by Marvin Minsky, the head of the MIT Artificial Intelligence Research Group, and Seymour Papert in their book *Perceptrons: An Introduction to Computational Geometry* (1969).

20. Boltzmann machines are expansions of spin glass-inspired neural networks. By including a stochastic decision rule, Ackley, Hinton, and Sejnokwski (1985) could make a neural network reach an appreciable learning equilibrium. As Domingos explained, "the probability of finding the network in a particular state was given by the well-known Boltzmann distribution from thermodynamics, so they called their network a Boltzmann machine" (Domingos 2015, 103).

21. As noted in Cardon, Cointet, and Mazières (2018), there is a debate regarding the anteriority of backprop algorithm: "This method has been formulated and used many times before the publication of [Rumelhart Hinton, and Williams 1986]'s article, notably by Linnainmaa in 1970, Werbos in 1974 and LeCun in 1985" (Cardon, Cointet, and Mazières 2018, 198; my translation).

22. This second marginalization of connectionists during the 1990s can be related to the spread of Support Vector Machines (SVMs), audacious learning techniques that are very effective on small ground truths. Moreover, while SVMs manage to find, during the learning of the loss function, the global error minimum, convolutional neural networks can only find local minimums (a limit that will prove to be less problematic with the advent of large ground truths, such as ImageNet, and the increase in the computing power of computers). On this specialized topic, see Domingos (2015, 107–111) and Cardon, Cointet, and Mazières (2018, 200–202).

Conclusion

1. Though, like Negri, this book is drawn to the idea of contributing to founding a philosophy capable of going beyond modernity understood as "the definition and development of a totalizing thought that assumes human and collective creativity in order to insert them into the instrumental rationality of the capitalist mode of production" (Negri 1999, 323).

2. Curiously, even though Negri explicitly positions himself as an opponent of the Anglo-American liberal tradition, his conclusions regarding the dual aspect of insurrectional acts are quite aligned with propositions made by American pragmatist writers such as Walter Lippmann and John Dewey. Indeed, whereas for these two authors, the political can only be expressed by means of issues that redefine our whole living together (Dewey [1927] 2016; Lippmann [1925] 1993; Marres 2005), for Negri, the political, as Michael Hardt notes, "is defined by the forces that challenge

the stability of the constituted order … and the constituent processes that invent alternative forms of social organization. … The political exists only where innovation and constituent processes are at play" (Hardt 1999, ix).

3. This, I believe, is a potential way of somewhat reconciling Negri—at least, his writings—with the great German legal tradition that he is also explicitly opposed to. If Negri is certainly right to refuse the exteriority of constituent power vis-à-vis constituted power, thus emptying legal constitutions of any power of political innovation, he is probably wrong to dismiss Georg Jellinek's and Hans Kelsen's propositions as to the scriptural, and therefore ontological, weight of constituent texts. On this tension between *Sollen* (what ought to be) and *Sein* (what is) within constitutive processes, see Negri (1999, 5–35) as well as Jellinek ([1914] 2016) and Kelsen (1991).

4. This is the topic of Anne Henriksen's and Cornelius Heimstädt's PhD theses (currently being conducted at Aarhus University and Mines ParisTech, respectively), as well as Nick Seaver's forthcoming book (Seaver forthcoming).

5. The moral economy of blockchain technology is the topic of Clément Gasull's PhD thesis, currently being conducted at Mines ParisTech.

6. This is part of Vassileios Gallanos's PhD thesis, currently being conducted at the University of Edinburgh.

This is a section of doi:10.7551/mitpress/12517.001.0001

The Constitution of Algorithms

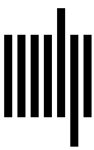
Ground-Truthing, Programming, Formulating

By: Florian Jaton

Citation:

The Constitution of Algorithms: Ground-Truthing, Programming, Formulating
By: Florian Jaton
DOI: 10.7551/mitpress/12517.001.0001
ISBN (electronic): 9780262363235
Publisher: The MIT Press
Published: 2021

The open access edition of this book was made possible by generous funding and support from Arcadia – a charitable fund of Lisbet Rausing and Peter Baldwin



The MIT Press

The open access edition of this book was made possible by generous funding from Arcadia—a charitable fund of Lisbet Rausing and Peter Baldwin.

This book was set in Stone Serif and Stone Sans by Westchester Publishing Services.

Library of Congress Cataloging-in-Publication Data

Names: Jaton, Florian, author. | Bowker, Geoffrey C., writer of foreword.
Title: The constitution of algorithms : ground-truthing, programming, formulating / Florian Jaton ; foreword by Geoffrey C. Bowker.
Description: Cambridge, Massachusetts : The MIT Press, [2020] | Series: Inside technology | Includes bibliographical references and index.
Identifiers: LCCN 2020028166 | ISBN 9780262542142 (paperback)
Subjects: LCSH: Algorithms--Case studies. | Computer programming--Case studies. | Algorithms--Social aspects. | Mathematics--Philosophy.
Classification: LCC QA9.58 .J38 2020 | DDC 518/.1--dc23
LC record available at https://lccn.loc.gov/2020028166