Can a machine think (anything new)? Automation beyond simulation

Article (Published Version)

AI & SOCIETY (2019) 34:813–824
https://doi.org/10.1007/s00146-018-0821-0

ORIGINAL ARTICLE

Can a machine think (anything new)? Automation beyond simulation

M. Beatrice Fazi[1]

Received: 18 January 2017 / Accepted: 24 January 2018 / Published online: 12 February 2018

Abstract
This article will rework the classical question 'Can a machine think?' into a more specific problem: 'Can a machine think anything new?' It will consider traditional computational tasks such as prediction and decision-making, so as to investigate whether the instrumentality of these operations can be understood in terms of the creation of novel thought. By addressing philosophical and technoscientific attempts to mechanise thought on the one hand (e.g. Leibniz's *mathesis universalis* and Turing's algorithmic method of computation), and the philosophical and cultural critique of these attempts on the other, I will argue that computation's epistemic productions should be assessed vis-à-vis the logico-mathematical specificity of formal axiomatic systems. Such an assessment requires us to conceive automated modes of thought in such a way as to supersede the hope that machines might replicate human cognitive faculties, and to thereby acknowledge a form of onto-epistemological autonomy in automated 'thinking' processes. This involves moving beyond the view that machines might merely simulate humans. Machine thought should be seen as dramatically alien to human thought, and to the dimension of lived experience upon which the latter is predicated. Having stepped outside the simulative paradigm, the question 'Can a machine think anything new?' can then be reformulated. One should ask whether novel behaviour in computing might come not from the breaking of mechanical rules, but from following them: from doing what computers do already, and not what we might think they should be doing if we wanted them to imitate us.

Keywords Automation · Simulation · Turing · Leibniz · Computation · Thought

1 An imitation game?

Can a machine think? Famously, Alan Turing believed this to be a bad question. He argued that it was ambiguous, and that it should be replaced with a test: an "imitation game" (Turing 1950, 433) that would bypass what he considered to be a dangerous haggling over definitions of the terms 'machines' and 'thinking', and which would instead link the issue of the intelligence of computing machinery to that of simulated behaviour.[1] That was in 1950, and questions about machine thought and issues about simulation have remained related ever since. In this article, I will rework Turing's classical problem 'Can a machine think?' into the more specific issue: 'Can a machine think anything new?' I will do so in order to entertain the possibility of breaking the common association between machine thought and simulation. In my view, addressing questions about the possibility of novelty in computation can afford a move away from what I will refer to as a simulative paradigm.[2] Asking whether a

✉ M. Beatrice Fazi
b.fazi@sussex.ac.uk

1 Sussex Humanities Lab, University of Sussex, Brighton, UK

[1] "I propose to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, 'Can machines think?' is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words" (Turing 1950, 433).

[2] The philosopher of science Thomas Kuhn (1962) used the term 'paradigm' to refer to the sets of normalising concepts and practices belonging to the science of a particular period of time. Of course, I acknowledge this usage of the term 'paradigm' and its popularity. However, I am not adopting the expression in this specific Kuhnian meaning, but rather, more broadly, I aim to argue that simulation constitutes a paradigm insofar as it is an established (although not unifying nor normalising) framework of reference (in fact, often, a benchmark of success) for the multifaceted field of AI. Equally, it is important to add here that by employing the term 'paradigm' I do not mean to imply that AI is a 'normal science' in Kuhn's sense (i.e. a settled, dogmatic and unquestioned ensemble of scientific practices).

machine can think anything new will then be a way for me to engage with issues pertaining to the possibility that computational processing might engender autonomous modes of automated epistemic production. In this sense, the question of novel thought in computation becomes a vehicle to tackle philosophical issues concerning what I will refer to here as the onto-epistemological autonomy of computational automation. I hope to show, therefore, that the question 'Can a machine think anything new?' can be a useful alternative to Turing's 'Can a machine think?': for where Turing's question led to an imitation game, and thereby to popular understandings of machine thought as simulative of human thought, the issue of novelty in computation helps to highlight precisely the opposite, i.e. the difference between machine thought and human thought. The novelty at stake here is, simply put, that of a new kind of thinking.

The history of computing is characterised by a certain degree of enthusiasm regarding the prospect of computing machines being able to produce forms of thought. This was true in the 1950s, when Turing asked whether machines can think, and it is true today, when computing is not just something that happens in university labs, but is instead a condition that few worldly activities can evade altogether. Yet, in addition to the persistent dream of a thinking machine, the suspicion that computational cognitive or intellectual capacities can, at best, merely imitate human ones has also endured from the mid-twentieth century up to the present time. Again, when it comes to the question of machine thought, simulation still seems to be our safest bet. If, from the 1950s to today, society and culture have often imagined the arrival of technologies that can think like humans do (and which, consequently, might also one day want what humans want), then the fact that these technical agents might not really be thinking as we are, but may only be fooling us into believing that they do, has often helped to ease persistent technocultural anxieties about the "rise of the robots" (Ford 2015), as well as to deflate technoscientific prospects of a "second machine age" (Brynjolfsson and McAfee 2014) in which machines substitute rather than complement humans.

The imitation of intelligent behaviour was the success threshold that Turing set when considering the cognitive capabilities of computing machinery. His imitation game heralded the emergence of artificial intelligence (AI) as a new scientific field of enquiry. It allowed machines to be credited as intelligent on the condition that a human was not able to distinguish that machine from another human on the basis of the answers given to questions put to both the machine and the human.[3] Whilst it is true that, by way of his imitation game, Turing discussed the circumstances under which one might determine that a digital computer can think, and whilst it is also true that he did not explicitly claim that all human thinking is computation (see Harnish 2002, 184), it is nonetheless the case that the imitation game opened up a comparative link between human cognition and mechanical calculation. In effect, Turing asked whether the algorithmic procedures of discrete-state machines are in principle capable of producing a sufficiently convincing display of cognitive behaviour.[4] The first AI initiatives that came after Turing's enquiry took on this implicit link and made it explicit, vis-à-vis the development of computationalist theories of the mind in what was then the newly emerging discipline of cognitive science.

In my view, technocultural and technoscientific retreats into simulation should be read in parallel with these computationalist attempts to mechanise the mind, but also in parallel with the twentieth-century development of the ancient dream of finding and automating the rules of thinking. Philosophical criticisms of these attempts and of this dream should in turn be read vis-à-vis the belief that the automated cognitive activities of machines are just reductions or simplifications of the thinking that originates from lived experience. This belief, however, is still based on the simulative paradigm. In other words, it is grounded on the

[3] The field and name of 'artificial intelligence' were coined during the Dartmouth Summer Research Project, a 1956 conference held at Dartmouth College, in Hanover, New Hampshire. Although there was no agreement on a general theory or methodological programme (see Moor 2006), the belief that the Turing test (that is, a simplified version of Turing's 1950 imitation game) would be passed in less than a generation was widely shared amongst the AI pioneers who attended the event. So too was the conviction 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, 12). At present, however, and more than sixty years after the Dartmouth College conference, the Turing test has still not been passed. Moreover, it should also be added that passing the Turing test is not the most prominent effort of current research in AI. Focusing too strictly on the human performance of intelligent behaviour is considered to be distracting and restricting, and the imitation game is seen as an "elusive standard" (Moor 2003) for AI research. Whilst it is necessary to acknowledge this condition in order to give a correct depiction of AI research, the fact that the imitation game is not the primary focus of the field does not weaken this article's argument. This is because I understand and discuss the simulative paradigm as being predicated not uniquely on a striving to replicate the human in the machine, or the machine in the human, but in fact as an effort to make computation more akin to the onto-epistemological dimension of the lived and the living, upon which the human is arguably predicated.

Footnote 3 (continued)

[4] "Could a computer reply to an interrogator in a way indistinguishable from the way a human being might reply, whether adding numbers or scanning sonnets? This question (often expressed as whether a computer could pass the 'Turing-test') has three aspects. Might some future computer actually be able to answer in the ways imagined? Are effective procedures in principle capable of generating this performance, whether in humans or computers? And would such performance suffice for the attribution of intelligence to the computer? Turing's own answer in each case was 'Yes'" (Boden 2005, 4).

assumption that, at best, smart machines can only be said to simulate human thought, or to simulate the onto-epistemological conditions upon which human thought is predicated. My aim here is to show that, by taking some distance from the simulative paradigm, it is possible to challenge the all-enframing cognitivism inherent in the hope, and the effort, of automating thought. However, rather than discarding computational automation and its potential for thinking altogether, I also intend to demonstrate that it may be possible to open an autonomous onto-epistemological prospect for automated machine thought by freeing the latter from the expectation that automation is just an attempt at simulation, whether of human cognition, or of life and lived experience.

2 Calculative activities

The issue of novel thought in computation can be articulated in many ways. An obvious and important field of reference for this problem is the area of study that goes by the name of 'computational creativity'. Efforts in the field see the contributions made by artificial intelligence, cognitive science, psychology, philosophy and the arts towards exploring the possibility of artificially modelling and replicating creative behaviour using the mechanical means of computation. This heterogeneous field is relatively young, but it has already achieved some surprising results.[5] As often happens in AI research, this kind of experimentation has a mutual benefit: by trying to replicate human cognitive function in machines, it becomes possible to understand better what that function in the human is in the first place. So, by creating models of the multifaceted and protean qualities of human creativity, light is shed on the origin and the development of creative behaviour, regardless of whether that behaviour is natural or artificial.

It would be unfair to characterise computational creativity as being uniquely concerned with simulating human creativity in machines. As a matter of fact, the field considers the nature of creativity and of creativity involving computational activity at large.[6] At the same time, however, it should be noted that debates in this area of research tend to praise capacities such as insight, inspiration, intuition, ingenuity, interest, improvisation, interpretation or even intelligence itself, all of which are valued features of the creative human mind. In this respect, I would then say that, by addressing novel thought in computation through the parameters of computational creativity, or even creativity in general, there is a risk of retaining simulation as an implicit measure for success.[7] Seen in these terms, machines would be deemed to be capable of generating novelty insofar as they are recognised as doing something that one would characterise as creative if humans were doing it. It could be said that this mode of enquiry pursues computing machines that might be less computer-like: machines that might surprise us by doing what machines may not be likely to do successfully, such as crafting a moving speech, improvising jazz, or composing a beautiful dance. I believe, therefore, that what tends to be searched for in these practices is a type of thought process and of computational processing that is less automatic and less mechanical. In other words, one that resists or circumvents the stubborn automatism of logical machines.

I acknowledge the importance of this type of work, and I recognise that I am painting a picture of a varied area and community of research in broad brushstrokes. However, I must also clarify that my question about novel thought in computation is not based on the successes and failures of the field of computational creativity. I want to take a different route here to that which computational creativity might be seen to adopt, for I will not linger on the capacity of computers to surprise us, or to do what they are not supposed to do. On the contrary, I wish to consider machines that do exactly what they are supposed to do. I want to engage with the question: *can computing machines be generative of novelty in ways that are profoundly alien to humans, because they are so inherently computational?*

In order to develop this question, one must address traditional automated computational tasks, such as, for instance, prediction and decision-making. Both prediction and decision-making can be understood as typically *calculative*

[5] Colton, López de Mántaras and Stock comment that "computational creativity as a discipline has come of age. This maturity is evident in the amount of activity related to computational creativity in recent years; in the sophistication of the creative software we are building; in the cultural value of artifacts being produced by our software; and most importantly, in the consensus we are finding on general issues on computational creativity" (2009, 12). A survey of key research issues in the field is given in Cardoso et al. (2009).

[6] It should be noted that "[…] simulating thought processes in only one of [computational creativity's] aims" (Colton et al. 2009, 14). See also Colton and Wiggins, who characterise computational creativity as a "subfield of Artificial Intelligence (AI) research […] where we build and work with computational systems that can create artefacts and ideas" (2012, 21). This definition "does not rule out situations

Footnote 6 (continued)

where systems are deemed to be creative even though they behave in wholly different ways, and to different ends, from people" (ivi).

[7] Interestingly, Turing tests for creative computing are popular and common in the field, although they are contested (see Colton and Wiggins 2012). The Neukom Institute for Computational Science at Dartmouth College (USA), for instance, is sponsoring the *Turing Tests in the Creative Arts*. This is a competition that, in its 2017 edition, encourages the creation of software able to: (1) write a short story; (2) compose sonnets; (3) accompany a human musician in a classical duet or jazz jam; (4) generate a digital dance partner for a human dancer. These four outcomes must be indistinguishable from those resulting from human effort. At the time of writing, the 2017 competition has just begun.

activities. As such, they are activities that can be effectively *mechanised via algorithmic means*. For instance, it is possible to argue that it was better predictions that allowed the IBM supercomputer Deep Blue to beat chess grandmaster Garry Kasparov in their historic match in 1997. Both players were trying to predict each other's moves. They were both calculating the consequences of changing the positions of the pieces on the board, and considering what their adversary would do next. Deep Blue won through the sheer brute force of its heuristics: it was able to evaluate two hundred million positions per second. It won because it predicted better, and it predicted better because it calculated faster and on a grander scale than a stressed-out Kasparov, whose thoughts, legend has it, were clouded by paranoia and suspicion.[8] Algorithmic decision-making also promises similarly successful outcomes, and just like prediction, decision-making is also calculative when it is carried out via computational means. Operational decisions, such as what trades to make, are calculatory staples of today's financial markets. It is not by following gut feelings that the quantitative finance models of automated algorithmic trading answer the question 'Should I buy this stock or not?' Instead, what one finds there is the deterministic operation of buying and selling securities on the market automatically, according to some pre-programmed strategy. This is decision-making on steroids, executing millions of orders and scanning multiple markets in fractions of seconds.[9]

It is easy to call both the supercomputer Deep Blue and the high-frequency trading algorithms 'smart'. Indeed, it is also tempting to call these machines "smarter than us", since "once computers achieve something at human level, they typically achieve it at a much higher level soon thereafter" (Armstrong 2014, 43). One might find it more difficult, however, to associate their smartness, and their ability to outsmart us, with the production of a form of epistemic novelty. Both the predictions of the supercomputer Deep Blue and the decisions of high-frequency trading (HFT) algorithms exemplify what computers do best: given certain premises, and by following certain instructions, they achieve certain goals. Or, to put it in more anthropomorphic terms, this is a case of selecting possible actions in order to steer the future towards a desired outcome. Both examples, then, are instances of an operational type of smartness that goes by the name of *instrumental rationality*, and which is, as Max Weber described in his study of industrial rationalisation, "determined by expectations as to the behaviour of objects in the environment and of other human beings" (1978, 24). These expectations "are used as 'conditions' or 'means' for the attainment of the actor's own rationally pursued and calculated ends" (ivi).

Of course, the affinity between calculating machines and instrumentally rational action, which "accepts itself simply as a tool" (Horkheimer 2012, vii), has been noted before. I am not going to reiterate the long history of the critical analogy between them here.[10] What I want to ask, rather, is whether this instrumentality can ever be understood in terms of the creation of novel thought. Arguably, instrumental reasoning is not inherently opposed to novelty. The instrumentality of reason is often exemplified, according to the Frankfurt School interpretation, by the cunning of Odysseus, who tied himself tightly to the mast of his ship in order to hear the beautiful song of the sirens whilst also avoiding the danger of steering the boat off its course (see Horkheimer and Adorno 1972). We know the myth, and what it means for us moderns who have been betrayed by (or, conversely, have betrayed) the Enlightenment, as Adorno and Horkheimer argued. However, there is nothing in this story that prevents us from believing that Odysseus' cost effective and efficient expedient is not a clever, inventive, although still perverse, form of novel problem-solving. The possibility of novelty within instrumentality would thus seem to be accepted when the instrumental rational agent is a human one.

Machine instrumentality is, however, different, and it is judged differently. The crux of this difference, I would claim, lies in the fundamental disparity between machines on the one hand, and life, the lived and the living on the other. In order to explain this central claim in my argument, it is first necessary to consider the question of novel thought in computation vis-à-vis an old hope and an old worry, both

8 These events, and Kasparov's feelings about them, are narrated in the 2003 documentary *Game Over: Kasparov and the Machine* by Vikram Jayanti. For the story of the match from the perspective of IBM and Deep Blue's creators, see Hsu (2002). The significance of Deep Blue's success for artificial intelligence is discussed in Morris (1997). In 2016, and nearly 20 years after the historic Deep Blue versus Kasparov chess match, AlphaGo, an AI developed by Google's AI company DeepMind, beat the champion human player Lee Sedol at a game of Go. Interestingly, DeepMind is said to describe AlphaGo as "more useful" than the Deep Blue computer that defeated Kasparov, insofar as this new AI does not consider all solutions to a problem, but instead "decide[s] between a few top scenarios" (Aron 2016, 6).

9 For an investigation of automated trading techniques from the sociological perspective of science and technology study, see MacKenzie (2016). The rise of AI trading is also recounted in the work of financial journalists Scott Patterson (2012) and Michael Lewis (2014).

10 Nonetheless, it is necessary to mention that the considerations of this affinity develop from a long tradition of critical enquiry about technology as a form of instrumental rationality. The Frankfurt School, most famously, addressed the role that technology plays in the modernising processes and organisation principles of society, as well as the manner in which technological rationalisation orients social formations and the meaning of culture (see Horkheimer 2012, 2013; Horkheimer and Adorno 1972). Beyond critical theory, the philosophy of AI itself recognises, albeit by a way of rather different perspective, that smart machines can be intelligent in a specifically instrumental sense (see, for instance, Bostrom 2014).

of which pertain to the possibility of mechanising thought, or in other words, to the prospect of making thinking procedural, inferential, and indeed both automatic and automated.

3 The automation of thought: Leibniz and Turing

A good starting point from which to address the expectations that have been invested (philosophically, scientifically and culturally) in the automation of thought can be offered by looking at the long-standing dream of *mathesis universalis*. Despite the erudition implied by the Greek-Latin origin of the expression, the actual meaning of *mathesis universalis* is not straightforward. For the sake of brevity, let us say that its most direct definition is that of a universal mathematical science. Aspirations towards such a universal mathematical science have been inherited from the ancient world, and became popular amongst the early modern philosophers of the sixteenth and seventeenth centuries.[11] Notably, they were also shared by Descartes, who described *mathesis universalis* as "a general science which explains all the points that can be raised concerning order and measure irrespective of the subject matter" (Descartes 1985, 19). It is, however, the Baroque polymath par excellence, Leibniz, whom I wish to address here, for Leibniz's philosophy will allow me to establish a conceptual link between *mathesis universalis*, the automation of thought and computing machines. Establishing this link is, in turn, important for the overall development of my argument about novelty qua the onto-epistemological autonomy of algorithmic automation. In this section, I will thus read Leibniz's philosophical attempt to mechanise thought in parallel with Turing's technical endeavour to show how thinking is already mechanical. I will do so in order to sketch a brief genealogy of the computationalist goal of mechanising the mind. As I have claimed above, this goal can be read in parallel with technoscientific concerns about simulation. By showing how they represent an enduring dream (and an equally enduring worry) in technoculture, I can slowly introduce my argument that questions about the automation of thought imply questions about the automation of being, and that they thus lead us to philosophically confront the onto-epistemological disparity between life and mechanism.

In Leibniz, *mathesis universalis* still concerns a general framework of calculation. Nonetheless, the applications and implications of this general calculative framework extend much further than Cartesian issues of order and measure. With Leibniz, *mathesis universalis* is less a general mathematical science than a generalisation of the rules of thought itself. The project of *mathesis universalis*, in other words, becomes the project of a universal science of reasoning, which is in turn to be supported by the construction of what Leibniz called a *characteristica universalis*. This *characteristica universalis* is an abstract symbolism that, according to Leibniz's invention, would have unambiguously represented all that can be thought and expressed (see Leibniz 1989). Such a universal character, which in effect amounted to a universal conceptual language, was in turn to be supported by the calculus ratiocinator: a purely rational and ideal grammar that, for Leibniz, would have worked as a proper inference engine. Regrettably, Leibniz never described his project for a *mathesis universalis* in detail. His attempt can nonetheless be summarised in three steps: (1) to create a compendium of all human knowledge; (2) to select appropriate symbols to express this compendium; (3) to reduce rules of deduction to these symbols. By doing so, Leibniz aimed to demonstrate that it is possible to operate with concepts in a strictly formal manner. That is to say, that it was possible to find the general, symbolic rules for valid reasoning, and that once we had these rules, we would have the mechanism for never erring in our thinking. "*Calculemus*": let us calculate! In Leibniz's famous rationalist exhortation (see Leibniz 1951, 51), the valid reasoning which calculation expresses is the key to intellectual discovery, as well as the best tool we have for solving human disputes.

Leibniz himself might never have accomplished his plan for a *mathesis universalis*, yet a Leibnizian faith in calculation became decisive for twentieth-century interpretations of a universal formal science of reasoning, which have in turn influenced modern developments of mathematical logic. In this respect, it is also impossible to ignore the influence that Leibniz's dream of universal symbolic calculation has exerted upon the development of contemporary computing. It is useful here to refer directly to the establishment of the notion of computability by, once again, the mathematical genius of Alan Turing. More than a decade before he asked whether machines could think, Turing took on a pressing problem in mathematical logic: the *decision problem* (also known as *Entscheidungsproblem*). This problem was phrased by the most prominent mathematician of the time, David Hilbert, in terms of a question that asked whether a universal and definite method to decide whether a proposition is provable could be conclusively envisaged. In his 1936 paper 'On Computable Numbers, with an Application to the Entscheidungsproblem', Turing set out to answer Hilbert by addressing the decision problem from the perspective of computability, and by giving a precise definition of 'method' via that perspective. By answering in the negative to Hilbert's question (that is, by proving that certain functions cannot be computed), and thereby confirming Kurt Gödel's earlier results on undecidability in axiomatic systems (see

[11] Crapulli (1969) discusses the sixteenth-century development of the idea of *mathesis universalis*.

Gödel 2004), Turing then also proposed a specific method of computation that is said to have defined our current conception of mechanical calculation.

In Turing's 1936 formalisation of the notion of computability, to compute is to discretise a task into the *algorithmic form*. This means turning it into a rule-governed activity that can be addressed as a sequential succession of finite steps. In this sense, Turing's algorithmic method of computation expresses an automatic principle of deductive inference, which is considered universal insofar as it is capable of computing anything that can, in principle, be computed via the sequential succession of finite steps of an algorithm. The postulation of this universal method of computation is one of Turing's most influential achievements; an achievement that justifies the view that he 'invented' modern computers.[12] This characteristic of universality, I would argue, is also quite important from the point of view of the connection that I am making here between calculation and the automation of thought.

A universal calculative theory of reasoning, such as Leibniz's *mathesis universalis*, stands as an attempt to construct what I would call a *machine of thought*, and to do so by trying to give thinking an inferential, normative and procedural form. This calculative enterprise searches for an ordering technique to mechanically control and reproduce the structure of thinking procedures. Fast forward a couple of centuries, and one can note that the algorithmic method of calculation, which Turing established, is also predicated upon computation as a formalisation of 'valid reasoning', according to which a finite set of instructions accounts for the full operationality and predictability of a mechanical system. In this respect, Turing's algorithmic method of calculation becomes the technique proper to axiomatic reasoning: that is, reasoning that is fully automated insofar as it needs nothing but itself in order to prove its validity.

In proposing the algorithmic method of computation, Turing modelled the action of his then-hypothetical 'computing machine' on what he believed to be the discrete-state operations of a human mind engaged in crunching numbers.[13] *So, whilst Leibniz proposed a machine of thought, Turing treated thought as if its behaviour was already similar to that of a machine*. The character of universality is important in both cases. If calculation is valid reasoning (and both Leibniz and Turing believed it is), and if valid reasoning always aims to be universal (again, Leibniz and Turing believed this to be the case), then a valid calculative method is one that tries to be as general as possible. To be universal is thus to be general, but also, and most importantly, to be abstract. This is, in my view, a crucial point. Despite the differences between Leibniz's and Turing's respective projects (and there are plenty of differences between them), the Leibnizian faith in calculation and Turing's algorithmic method of computation share a confidence in abstraction, which is taken to be the safest ground for the operability of calculation. In this sense, aspirations to mechanise thought, or to show that thought could already be mechanical, are equally predicated upon the assumption that both proof and function can be placed outside of space and time, and outside of context and content.

The universal strength of mechanical calculation capitalises upon the detachment of the mechanical procedure from the particular; a detachment that is in turn instrumentalised into procedures to best resolve disputes, as Leibniz hoped, or to obtain an output from an input, as Turing expected. This *instrumentalisation of abstraction* is a key feature of the inferential and rule-based character of computational systems. For every calculating process, the computational system engages, and then re-engages again, with the general problem of determining consequences from a handful of symbolised premises. The automation of thought thrives on this procedural determinism of rules of inference: it is its complete determination that makes a machine (of thought, as of anything else) a machine.

4 The clash between life and mechanism

Let us linger, if only briefly, on this determinism, for doing so will allow us to consider the automation of thought in relation to my previous remarks about machine instrumentality, and about the fact that the latter is judged differently when it comes to the question of novel thought in computation.

For Leibniz, the determinism of calculation is total. If Leibniz believed that it is possible to mechanise the rules of thought into a procedure of automatic inference, this is because nature itself, for Leibniz, is fundamentally rational, or governed by a determinist logic. I am referring here to the Leibnizian formulation of a principle of *sufficient reason*, which postulates that everything has a cause or a reason for being what it is. It is exactly because the principle of sufficient reason has both logical and ontological validity that it is also possible to address Leibniz's dream of automating thought from the standpoint of his argument for the sufficient rationality of the real. It can in fact be argued that the Leibnizian invocation of calculation is grounded upon what he believed to be the 'pre-programming' of reality, a belief that is implicit in the principle of sufficient reason. That principle requires that whatever occurs is, as Leibniz himself put it,

[12] I have discussed at length Turing's 1936 paper 'On Computable Numbers, with an Application to the Entscheidungsproblem' in Fazi (2016).

[13] "We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions" (Turing 1936, 231).

"certain and determined beforehand" (1985, 151). In other words, it is possible to say that Leibniz's epistemological project is substantiated by a metaphysical principle.

Turing's position on ontological determinism is more difficult to reconstruct, partly because, despite having profoundly influenced philosophy, Turing was primarily a mathematician with explicitly logical, rather than ontological concerns. I acknowledge this, of course, yet I also wish to highlight that computation, via Turing's work, is formalised into an axiomatic process of showing how a conclusion necessarily follows from a set of premises. In my view, this deterministic activity of pinning down chains of reasoning into a fixed formal structure can be said to be an attempt to predicate deductive thought exclusively on its extra-empirical terms. On my reading of Turing's work, then, the epistemological leaks into the ontological, or their boundaries become somewhat permeable.

One can see the effect of this permeability in the computational theory of mind, which emerged in the second half of the twentieth century, partly as a consequence of the implicit links that Turing made between calculation and cognition. According to computationalist theories in cognitive science, "intelligent behavior is causally explained by computations performed by the agent's cognitive system (or brain)" (Piccinini 2009, 515). *Logos* (i.e. software) informs matter (i.e. hardware), then, or rather bypasses matter altogether to directly lead behaviour. In any case, the thin line between issues of knowledge and issues of reality blurs frequently.[14] I would further comment that this also seems to be the case amongst certain contemporary versions of information theory, such as that proposed by the mathematician Gregory Chaitin, who takes both Leibniz and Turing as guiding figures for his own work.[15] For Chaitin, and for the advocates of what is known as 'digital philosophy' or 'digital physics', "all is algorithm" (Chaitin 2007, 235): the determinism of algorithms is not just a metaphor used to explain the workings of nature and the mind; rather, this determinism amounts to how reality actually is and how it actually operates.

It is crucial to stress that past and present projects for the automation of thought do not concern only epistemology, but involve an ontological prospect too; that questions about the automation of thought, in other words, also concern questions about the automation of being. Acknowledging this point brings to the fore an as yet unresolved issue in philosophy: namely, the relation between thought and being, and the ways in which that relation might be established, or simply recounted, via universalising tools such as abstraction. Moreover, acknowledging this issue also highlights something important about the question of novel thought in computation. I have discussed how machine thought adheres to the deterministic and deductive rules of computational formalisation. In this respect, if computation seems incapable of producing novelty, this is because algorithmic formalisation does not construct any knowledge, but instead only breaks it down into a finite and discrete manipulation of symbols and instructions. However, recognising that questions about the automation of thought involve questions about the automation of being also helps us to consider here that views concerning the absence of novelty in computation are not only, or are not uniquely, based on the fact that, in machines, everything is pre-programmed. As a matter of fact, there exist (and there have existed for some time) machine programs that possess a degree of complexity sufficient enough to allow them to 'learn' to modify their behaviour and perform operations without explicit instructions.[16] What I wish to argue, then, is that the popular belief about the impossibility of novelty in computation finds its justification in the observation that to compute is to formally abstract, and thus to generalise and reduce into a logical and deterministic relation the dynamism of life and of thought that comes from lived experience. To put this otherwise: my contention here is that the belief about the impossibility of novelty in computation could be said to be a consequence, or at least to reflect, the fundamental ontological disparity between machines on the one hand, and the non-mechanical on the other.

This is an ontological disparity that is also mirrored in the (ontological, as well as epistemological) difference between automated and lived thought. The mechanical thought associated with computation involves, inevitably, abstracting from life, from sensation, from experience itself; in other words, from those grounds upon which, it is said, everything that could account for genuine epistemic as well as ontological production might arise. Contra formalism and the formalisation of thought, thinking processes that are immanent to lived experience are instead always particular because they are instantiated in spatiotemporal and affective dynamics. As such, they can only be reduced to their logical representation by way of the approximation and generalisation of those dynamics. These approximations and generalisations are precisely what twentieth-century criticisms of

[14] For Putnam (1967), for instance, mental states are functional states, and thus detachable from the 'hardware' of the brain. On the other hand, in the materialist version of computationalism advanced by Paul and Patricia Churchland, there is only hardware, and minds are brains, which are computing. See Churchland and Churchland (1990), Churchland and Sejnowski (1992).

[15] See Chaitin (2005).

[16] The present enthusiasm about the AI technique known as 'machine learning' concerns precisely this capability. This enthusiasm is manifest in Domingos (2015), who argues that "the quest for the ultimate learning machine will remake our world". I will return to the case of machine learning in the last section of this article.

instrumental reason, as well as wider philosophical arguments that emphasise the singularity and incommensurability of the lived, warned against. They warned that a thinking process that detaches itself from the particular is totalising, and thus dangerous, for “the concept of rationality that underlies our contemporary industrial culture […] contain[s] defects that vitiate it essentially” (Horkheimer 2013, vii), and “the rule of freedom” that is instantiated by this concept, “once brought to pass, necessarily turns into its opposite: the automatizing of society and human behaviour” (Horkheimer 2012, ix–x).

Moreover, on this view, a thinking process that detaches itself from the particular is also somewhat ontologically ‘inferior’. This inferiority is due to the fact that the thinking associated with computing involves, once again, a structuring of formal relations, in which there is no room for the generative potential of life, the living and the lived. Post-structuralist, existentialist and phenomenological positions in philosophy have, in this respect, equally led the charge against the prospect of reducing the infinity and variability of lived thought to the finitude of an automated procedure. The eccentricities of thought that comes from Being or beings are thus vindicated against functionalism, operationalisation, universalism, instrumentalisation, and anything that wants to obstruct the ontological dynamic of thinking or entrap it within the programmable, abstracted and predefined structure.

From this perspective, the automation of thought, or its full mechanisation, is not the dream that Leibniz dreamt, but in fact a nightmare. This nightmare finds its latest expression in some of the ways in which automation is culturally perceived today as posing a sort of existential threat to society. Many cultural media outlets have attested to this. From the tabloid to the scientific journal via the glossy coffee table spread, the message that is broadcasted is clear: the robots are out to get us. In my view, and following from what I have discussed so far, the threat that is culturally perceived to be posed by automation must be understood not only in terms of a collective fear for people’s existence or survival. By saying this, of course, I am not underestimating real concerns about unemployment caused by technology, or the faster and faster pace at which artificial intelligence is developing.[17] I believe, however, that whenever one asks questions such as, for instance, whether machines will threaten our jobs, one is not just asking about the economic implications of automation. What is also implicitly already assessed is our relation to mechanical procedures, and thus the extent to which these mechanical procedures exist in opposition to what is not, and might never be, mechanised. The problem of automation is existential, then, insofar as it involves recognising an ontological clash between life and mechanism. It concerns the ways in which machines, thought and mechanical thought exist, endure or simply are.

5 Alien thought

It is only at this point in the discussion that I can fully advance the view that philosophical, scientific and cultural withdrawals into a simulative paradigm mirror a generalised technical, scientific and cultural suspicion that an intelligent automatism might not in fact be so special, insofar as it is a reduction, an approximation, or at best just a simulation of what life (to be understood both as the living and the lived) will never be, i.e. automatic. I can finally fully advance this claim because I have now demonstrated that this suspicion taps into long-standing ontological and epistemological issues about the nature of thought and its relation to lived experience. It is also at this point in the discussion, however, that it is necessary to mention that a preoccupation with an onto-epistemological disparity between machines and life does not only characterise philosophy and critical theory, but belongs to computer science too. In recent decades, the classical algorithmic model of computation (i.e. Turing’s model) has been challenged by *interactive, embodied* and *situated* conceptualisations of the computing machine, which aim to complement the algorithmic rule with environmental inputs, so as to bring the physicality and sociality of the ‘real world’ (which is always leaning towards the behavioural richness of living and lived experience) into its computational representation.

In this respect, it could be said that attempts to mechanise the mind have given rise to attempts to humanise the machine. This reflects the necessities of contemporary computing, which has to interact with and respond to the world at an unprecedented scale and speed. For me, however, there is more involved at a conceptual level. Twenty-first century society is not only witnessing attempts to make machines more human or more social, but also attempts to make them akin to the generative dimension of lived existence upon which human and social conditions are predicated. So, for instance, from an aesthetic and sociocultural perspective, computational systems are made akin to matter and life by associating them with empirical variability, or with the affective immanence of sensation. Moreover, within computer science itself, algorithms have been programmed to adapt, mutate and evolve like living organisms, and ‘reflexivity’ and ‘emergence’ are not only contemporary key words in the natural sciences, but in computing too.

This is another side of understanding computation as simulation. Although this other side can also be seen as

[17] Some of these concerns are discussed in Brynjolfsson and McAfee (2011), Frey and Osborne (2013), Carr (2015), Ford (2015) and Mason (2015).

a move away from the computationalism of much cognitive science (insofar as what it is pursued is not, strictly speaking, a computational explanation of a phenomenon), I believe that the contemporary debate concerning this topic remains, nonetheless, stuck in a computational metaphor of some kind. Just as the mechanisation of the mind and the humanisation of the machine go together, the counterpart to the naturalisation of machines can be said to be the mechanisation of nature. What one finds here, of course, is not the clockwork mechanistic universe of Newton, but rather the dynamic structuralism of ants, bird flocks and neurons, which are all said to compute, or to do something similar to computation, insofar as they transmit and process information whilst organising themselves from simple procedures all the way up to complex behaviour. In this scenario, it is difficult to determine what simulates what, or who simulates whom. The walls of Searle's Chinese room have come down to encompass nature itself.[18]

When they are looked at from this perspective, conflicting positions such as philosophical criticisms of the mechanisation of thought on the one hand, and technoscientific attempts to enlarge the domain of computability to include the human and the natural on the other, appear to have a common problem and a common target: formalism itself. The simulative paradigm reinforces this attack by aiming to intervene within formalisation: humanising it, I would argue, but also naturalising it. These operations, I would also claim, can be addressed as counterparts of similar simulative endeavours, which point, however paradoxically, not away from formalisation but towards it: for machines that can be humanised, there are minds that can be mechanised, and if a computer can do what nature does it is because nature, some say, is already computing.[19]

In this scenario, whatever direction we choose (away or towards formalisation) we are still stuck in an imitation game. And yet, once it is recognised that this entrapment is predicated upon a striving towards either making formalism more empirical, or towards making the empirical more formal, it becomes possible to envisage a way out of this jam. The exit that I propose concerns accepting the disparity between machines and life, and in fact involves radicalising this disparity *by accepting computational formalisation in its distinct specificity*. I am arguing that computation's epistemic productions should be assessed vis-à-vis the logico-quantitative character of those formalisms that underpin computing. This involves engaging with the epistemic contributions of computational formalisation by allowing automated modes of thought to operate independently of any aspirations towards replicating human cognitive faculties, and of the dimension of lived experience upon which these human cognitive faculties are ontologically predicated. Similarly, however, to engage with the epistemic contributions of computational formalisation also involves recognising that the automated modes of thought that are engendered by computation, precisely because they are formalised and formalising, propose *a different order of intelligibility*. This order of intelligibility corresponds to a machine ontology that cannot be simply dismissed as reductive, precisely because moving away from the simulative paradigm means that the assumption that the machine only represents humans has been invalidated.

My point is that if we, as a society, are to engage with computation's possibility of producing forms or modes of thought, then we need to start by acknowledging this other order of intelligibility, together with the way in which the computational procedure stands alongside the living organism, the physical object, and also the mathematical idea, *as a distinct entity*. By saying this I am not, of course, advocating a Platonism of sorts, because what I wish to look at are computational processes in their actuality, as opposed to viewing them in purely ideal terms. Similarly, I am not advocating the regression into a mind and body dualism. Instead, what I am supporting is the philosophical challenge of understanding the legitimacy of the processual and mechanical nature of this distinct entity, i.e. the computational procedure. This is a challenge that cannot be subsumed under the affirmation of the ontological superiority of life or machines over one another.

Importantly, moving beyond the simulative paradigm would allow us to extend the technocultural debate about computational automation from popular tales of intelligent machines that will one day outsmart us to questions about thinking per se. In this respect, moving beyond the simulative paradigm would also allow us to recast *the metaphysical question of the nature of thought*, precisely because the computational turn, and the present expansion of computational automation, with its epistemological consequences, have brought that question to the fore again. In my view, recasting the question of the nature of thinking vis-à-vis computational automation means opening up new avenues of speculation, so as to take seriously the possibility that

[18] In mentioning the 'Chinese room', I am referring to John Searle's famous argument against the computationalist presumptions of what Searle calls 'strong AI'. Searle (1980) argues that to compute is not the same thing as to understand or to behold any kind of consciousness. A program might make a computer behave in a way that demonstrates a degree of human-level intelligence, for instance, by making it speak Chinese. However, the machine does not have any understanding or consciousness of the faculty of speaking Chinese: it is only processing symbols. Syntax, in other words, does not account for semantics. For Searle, then, computers only simulate (human) cognition.

[19] I am referring here to pancomputationalist positions in 'digital physics' and 'digital philosophy' (see Fredkin 2003), but also to heterogeneous standpoints in what is known as 'natural computing' (see De Castro 2006; Shasha and Lazere 2010).

computational processes of decision-making and prediction might constitute a novel modality of thought: not despite the deterministic, deductive, axiomatic, procedural and operational character of these processes, but rather because of it. IBM super computer Deep Blue did not beat Kasparov because it thought like him. Quite the opposite: it won because it thought in a manner that was completely different to the way in which he thought, and indeed from the ways in which we all think. Equally, the decision-making of high-frequency trading performs on a different epistemic scale. So, whilst I agree that formalisation, and also computation itself, might obscure what is exciting and creative about reasoning in humans, I would equally say that the simulative paradigm obscures what is exciting and potentially creative in a mode of reasoning that is purely computational, and which is, insofar as it is purely computational, dramatically alien to us.

The case of machine learning can help me to explain this claim. I mentioned this AI technique earlier in this essay.[20] Here I can add that machine learning is a type of artificial intelligence that endows computers with the capacity to learn from large data sets. This technique is much talked about today, as it lies at the basis of many promising developments. Along with the excitement, however, comes a *caveat*. It is often said that machine learning is a 'black box'. In other words, it is stressed that, although these algorithms have a great impact upon society, they cannot be held accountable for the consequences of this impact because they are proprietary and thus closed to public scrutiny.[21] Of course, I agree that deciphering the black boxes of machine learning is becoming more and more urgent, because their influence upon our world is becoming greater and greater each day. The point that I would like to consider here, however, is that machine learning is also a black box in another, more technical, sense. Because they are modelled to act as deep neural nets, these techniques are often very opaque, or indeed illegible, because they are so complex that they cannot even be understood by the programmers that created them. Machine learning works, but we do not fully understand how.

The claim that 'we do not know how they work' is, for me, profoundly interesting insofar as it might allow us to say that, de facto, machine learning is making a crack in the simulative paradigm, which has been looming over AI research since Turing's imitation game in 1950. Whilst, to an extent, we can say these algorithms 'learn' because they improve themselves based on their exposure to stimuli, in reality we cannot judge this improvement, or this supposed learning, according to parameters of simulated behaviour, precisely because we cannot really tell what is going on.

It is at this point, exactly when facing the prospect of this *alien thought*, that I find that asking 'Can a machine think?' might be an ambiguous, and yet by no means meaningless operation.[22] As discussed in the opening section of this article, Turing recognised the poignancy of this question, but he also believed that the question risked giving rise to unproductive and frustrating attempts to define 'machines' and 'thought'. However, if the question of whether a machine can think is *not* addressed from a simulative perspective, it becomes possible to realise that it is in fact extremely important to respond to the need to define, and to keep defining, the terms 'machine' and 'thought'. What I propose to address here is indeed not just any type of thought, but a very special and peculiar one. This is a mode or form of thought that is procedural, instrumental and operational, and one that is also always quantifying, rationalising and discretising. The machine that is the subject of this investigation is also not just any machine, and Turing himself recognised this when he stated that the "machines concerned in the [imitation] game" (1950, 435) are "digital computers" (ibid., 436), which are in fact "discrete state machines" (ibid., 439). First and foremost, Turing's digital computers are machines that are arithmetic, highly formalised and formalising, and which pertain more to the laws of abstraction than to the laws of matter and life. As such, they are also machines that can be universal, for all the good and bad that this universalism implies.

Inevitably, then, *conceptions of novelty must also be reframed as specific to these discrete machines*. The aim of this article was to show that the possibility of novel thought in computation becomes specific to what algorithmic automation is and does. Just as the 'machine thought' that I have been concerned with is unique to the discrete and formal operations of computation, so the 'novelty' in computation that I am pointing towards is epitomised by the onto-epistemological specificity of these operations. For example, if we look for ingenuity and intuition as conditions for novelty, we might be disappointed and never find them in the discrete algorithmic operationality of machine thought, for the same reason that a computer can easily win a chess game or even bring about a 'flash crash' of the market, and yet still cannot pass the Turing test. However, having stepped outside the simulative paradigm, the question 'Can a machine think anything new?' can eventually be reformulated so as to

[20] See note 16.

[21] Frank Pasquale has eloquently described contemporary society's current "gaps in knowledge" about algorithmic technologies, and argued for a "more intelligible social order" (2016, 2–3).

[22] The AI pioneer Joseph Weizenbaum also talked of AI as 'alien', although in a different context to my own argument. He wrote: "No other organism, and certainly no computer, can be made to confront genuine human problems in human terms. And, since the domain of intelligence is, except for a small set of formal problems, determined by man's humanity, every other intelligence, however great, *must necessarily be alien to the human domain*" (1976, 223, my emphasis).

ask whether a novel behaviour can come not from breaking mechanical rules, but from following them: from doing, in other words, what computers do already, and not what we might think they should be doing if we wanted them to imitate us. Novelty, then, can be found in the distinctiveness of the mode of thought that machines express, and which arises from self-determinate and self-determining rule following. In this sense, and to conclude: this reformulated question about novel thought in computation involves re-defining and enlarging the prospect of what rule-based thought might be, along with reconsidering what rules themselves are. The issue that comes to the fore, then, is whether novelty in computation might be described as predicated on a form of ontological and epistemological autonomy in the normatively instrumental dimension of computational automation.

References

Armstrong S (2014) Smarter than us: the rise of machine intelligence. MIRI, Berkley

Aron J (2016) AI is going for go. New Sci 229(3058):6

Boden MA (2005) Introduction. In: Boden MA (ed) The philosophy of artificial intelligence. Oxford University Press, Oxford, pp 1–21

Bostrom N (2014) Superintelligence: paths, dangers, strategies. Oxford University Press, Oxford

Brynjolfsson E, McAfee A (2011) Race against the machine: how the digital revolution is accelerating innovation, driving productivity, and irreversibly transforming employment and the economy. Digital Frontier Press, Lexington

Brynjolfsson E, McAfee A (2014) The second machine age: work, progress, and prosperity in a time of brilliant technologies. W. W. Norton & Company, New York

Cardoso A, Veale T, Wiggins GA (2009) Converging on the divergent: the history (and future) of the international joint workshops in computational creativity. AI Magazine 30(3):15–22

Carr N (2015) The glass cage: where automation is taking us. The Bodley Head, London

Chaitin G (2005) Meta maths: the quest for omega. Atlantic Books, London

Chaitin G (2007) Leibniz, information, math and physics. In: Thinking about Gödel and Turing: essays on complexity, 1970–2007. World Scientific, Singapore, pp 227–240

Churchland PS, Churchland P (1990) Could a machine think? Sci Am 262(1):26–31

Churchland PS, Sejnowski TJ (1992) The computational brain. The MIT Press, Cambridge MA

Colton S, Wiggins GA (2012) Computational creativity: the final frontier? In: ECAI 2012. 20th European Conference on Artificial Intelligence. IOS Press, Amsterdam, pp 21–26

Colton S, López de Mántaras R, Stock O (2009) Computational creativity: coming of age. AI Magazine 30(3):11–14

Crapulli G (1969) Mathesis universalis: genesi di un' idea nel XVI secolo. Edizioni dell' Ateneo, Roma

De Castro LN (2006) Fundamentals of natural computing: basic concepts, algorithms, and applications. Chapman & Hall/CRC, Boca Raton FL

Descartes R (1985) The philosophical writings of Descartes. Volume I. Translated by J Cottingham, R Stoothoff, D Murdoch. Cambridge University Press, Cambridge

Domingos P (2015) The master algorithm: how the quest for the ultimate learning machine will remake our world. Allen Lane, London

Fazi MB (2016) Incomputable aesthetics: open axioms of contingency. Computational Culture 5. http://computationalculture.net/article/incomputable-aesthetics-open-axioms-of-contingency. Accessed 13 Jan 2017

Ford M (2015) The rise of the robots: technology and the threat of mass unemployment. Oneworld, Richmond

Fredkin E (2003) An introduction to digital philosophy. Int J Theor Phys 42(2):189–247. https://doi.org/10.1023/A:1024443232206

Frey CB, Osborne MA (2013) The future of employment: how susceptible are jobs to computerisation? Oxford Martin School, Oxford. http://www.oxfordmartin.ox.ac.uk/downloads/academic/The_Future_of_Employment.pdf. Accessed 12 Jan 2017

Gödel K (2004) On formally undecidable propositions of the Principia Mathematica and related systems I. In: Davis M (ed) The undecidable: basic papers on undecidable propositions, unsolvable problems and computable functions. Translated by E Mendelson. Dover Publications, Mineola, pp 4–38

Harnish RM (2002) Minds, brains, computers: an historical introduction to the foundations of cognitive science. Blackwell, Oxford

Horkheimer M (2012) The critique of instrumental reason. Translated by MJ O'Connell and others. Verso, London

Horkheimer M (2013) Eclipse of reason. Bloomsbury Academic, London

Horkheimer M, Adorno TW (1972) Dialectic of enlightenment. Translated by J Cumming. Herder and Herder, New York

Hsu FH (2002) Behind Deep Blue: building the computer that defeated the world chess champion. Princeton University Press, Princeton

Khun TS (1962) The structure of scientific revolutions. University of Chicago Press, Chicago

Leibniz GW (1951) The Art of Discovery. In: Wiener PP (ed) Leibniz selections. Translated by PP Wiener. Charles Scribner's Sons, New York, pp 50–58

Leibniz GW (1985) Theodicy. Translated by EM Huggard. Open Court, La Salle IL

Leibniz GW (1989) Preface to a Universal Characteristic (1678-79). In: Philosophical Essays. Translated by R Ariew and D Garber. Hackett Publishing Company, Indianapolis, pp 5–10

Lewis M (2014) Flash boys: cracking the money code. Allen Lane, London

MacKenzie D (2016) A material political economy: automated trading desk and price prediction in high frequency trading. Soc Stud Sci. https://doi.org/10.1177/0306312716676900

Mason P (2015) Postcapitalism: a guide to our future. Allen Lane, London

McCarthy J, Minsky ML, Rochester N, Shannon CE (2006) A proposal for the Dartmouth summer research project on artificial intelligence. AI Magazine 27(3):12–14

Moor JH (ed) (2003) The Turing test: the elusive standard of artificial intelligence. Kluwer Academic, Dordrecht

Moor J (2006) The Dartmouth college artificial intelligence conference: the next fifty years. AI Magazine 27(4):87–91

Morris R (ed) (1997) Kasparov vs. Big Blue: the significance for artificial intelligence. Papers for the 1997 AAAI Workshop. Technical Report WS-97-04. AAAI Press, Menlo Park CA

Pasquale F (2016) The black box society: the secret algorithms that control money and information. Harvard University Press, Cambridge

Patterson S (2012) Dark pools: the rise of A.I. trading machines and the looming threat to Wall Street. Random House Business, London

Piccinini G (2009) Computationalism in the philosophy of mind. Philosophy Compass 4(3):515–532. https://doi.org/10.1111/j.1747-9991.2009.00215.x

Putnam H (1967) Psychological predicates. In: Capitan WH, Merrill DD (eds) Art, mind, and religion: proceedings of the 1965 Oberlin Colloquium in Philosophy. University of Pittsburgh Press, pp 37–48

Searle JR (1980) Minds, brains, and programs. Behav Brain Sci 3(3):417–424. https://doi.org/10.1017/S0140525X00005756

Shasha D, Lazere C (2010) Natural computing: DNA, quantum bits, and the future of smart machines. W.W. Norton & Company, New York

Turing AM (1936) On computable numbers, with an application to the Entscheidungsproblem. Proc Lond Math Soc 42:230–265. https://doi.org/10.1112/plms/s2-42.1.230

Turing AM (1950) Computing machinery and intelligence. Mind 59(236):433–460. https://doi.org/10.1093/mind/LIX.236.433

Weber M (1978) Economy and society: an outline of interpretive sociology. Volume I. Translated by E Fischoff and others. University of California Press, Berkley

Weizenbaum J (1976) Computer power and human reason: from judgment to calculation. W. W. Freeman and Company, San Francisco