Minds & Machines – under review
(final version pre-review, Copyright Minds & Machines)

Invention, Intension and the Limits of Computation

Hajo Greif

30 June, 2020

Abstract This is a critical exploration of the relation between two common assumptions in anti-computationalist critiques of Artificial Intelligence: The first assumption is that at least some cognitive abilities are specifically human and non-computational in nature, whereas the second assumption is that there are principled limitations to what machine-based computation can accomplish with respect to simulating or replicating these abilities. Against the view that these putative differences between computation in humans and machines are closely related, this essay argues that the boundaries of the domains of human cognition and machine computation might be independently defined, distinct in extension and variable in relation. The argument rests on the conceptual distinction between intensional and extensional equivalence in the philosophy of computing and on an inquiry into the scope and nature of human invention in mathematics, and their respective bearing on theories of computation.

Keywords Artificial Intelligence · Intensionality · Extensionality · Invention in mathematics · Turing computability · Mechanistic theories of computation

This paper originated as a rather spontaneous and quickly drafted *Computability in Europe* conference submission. It grew into something more substantial with the help of the CiE reviewers and the author's colleagues in the Philosophy of Computing Group at Warsaw University of Technology, Paula Quinon and Paweł Stacewicz, mediated by an entry to the "Café Aleph" blog co-curated by Paweł.

Philosophy of Computing Group, International Center for Formal Ontology (ICFO), Warsaw University of Technology
E-mail: mail@hajo-greif.net
`https://hajo-greif.net`; `https://tinyurl.com/philosophy-of-computing`

1 Introduction

This is a critical exploration of the relation between two common assumptions concerning the differences between computation in humans and machines, and its bearing on cognitive inquiries: The first assumption is that at least some human-specific cognitive abilities are essentially non-computational, whereas the second assumption is that there are principled limitations to what machine-based computation can accomplish with respect to simulating or replicating these abilities. I take the conjunct of these two assumptions to be the most forceful anti-computationalist charge against the research programme of Artificial Intelligence (AI) in general and against the possibility of strong AI in particular.[1] Strong AI is understood here as a class of, paradigmatically digital, computing machines that display at least a significant subset of otherwise specifically human-like cognitive abilities, and that do so on a human or near-human level, where that set might include but will not be limited to meaningful and social language use, embodiment, intuition, inventiveness and consciousness.

The anti-computationalist views that build on a necessary connection between the two above assumptions typically proceed by defining machines in such a way that their operations necessarily, namely by definition, will be confined to the computational domain, while computation is defined so narrowly, namely as digital, Turing-Machine-computation, that it appears wholly or largely irrelevant to explaining or simulating human cognitive abilities.

I will argue in this essay that these definitions and the assumptions they support are problematic, and that the relation between them is tenuous. My argument will build on the observation that "it is far from obvious that the theoretical limits of human computation and the theoretical limits of machine computation need coincide" (Copeland 1997, 690). Contemporary accounts of computation and the computational mind, in particular "mechanistic" ones, will be discussed as alternative, broader yet more precise accounts of computation in physical systems and its bearing on cognition.Still, the following argument will remain partly sceptical of computationalism, admitting that the human mind might well be computational in some relevant respects while maintaining that it might not be not computational in *all* relevant respects. This argument should be viewed largely separate from the related-but-distinct question of computational models of human cognition in Artificial Intelligence.

With respect to the scope and style of this paper, I should remark that, despite its reference to physical computing, intensional identity and extensional equivalence, it does not carry an ambition to be a genuine contribution to the philosophy

[1] Such an anti-mechanist and basically essentialist stance is adopted by the most prominent classical critiques of classical AI, who argue that computer systems of any kind will always and by natural necessity lack some of the properties that are constitutive to human cognition. Dreyfus (1992) argues that computers rely on the ability of formalising knowledge, whereas human knowledge is in important respects and in key areas informal and based on embodiment and intuition. Searle (1980) argues that computers lack the ability to understand language, for want of being socialised into human social and linguistic contexts. To turn to a slightly more contemporary and significantly less essentialist critique of AI, Collins and Kusch (1998) argue that machines might be able to imitate human behaviours of high complexity in "mimeomorphic" fashion, but that they will not be able to deliberately take different behavioural routes to one and the same goal, as such "polimorphic" action would require an understanding of the purposes of seemingly variant behaviours.

of mathematics in general or the philosophy of computation from within those fields. It is far too general, non-technical and free-form for this purpose. Instead, it seeks to provide a slightly different perspective on a long-standing topic from a viewpoint more aligned with the history and philosophy of science.

The argument will be structured as follows: In Section 2, I will formulate a research question on computation in human beings and machines and suggest alternative, more informative versions of it along with a focus on physical computation. Section 3 offers a comparative look at computation in human beings and digital computing machines on the background of Alan Turing's theory of computability. Section 4 briefly outlines the concepts of intensional and extensional equivalence and its import on the present argument. Section 5 will discuss some of the possible non-computational aspects of human cognition, with exemplary focus on human inventiveness, and their implications on computational models of cognition. Section 6 will match the possible limitations of machine-based computation against the possible limitations of human inventiveness. Section 7 might not be able to provide fully-fledged conclusions, but it will offer some tentative lessons that might help to develop answers to the questions formulated in Section 2 at some other time, in some larger project.

2 Framing the problem

Given the assumed relation between computation and cognition outlined above, one might be able to formulate a research question that avoids the contentious and ill-defined notion of cognition in favour for the slightly better defined but also contentious notion of computation. After all, an understanding of what computation is will partly define what cognition can be. Minimally, computation is understood as execution of elementary arithmetical routines that produce definite solutions in finite time, where these routines only require a minimum of mathematical skill while in conjunction allowing for the solution of more complex logico-mathematical problems.

Even with such a minimal definition in place, there will be widely diverging interpretations of its relation to cognition. To the full-blown computationalist, computation (in a certain narrow sense) will plainly *be* cognition (Pylyshyn 1980). To the anti-computationalist, computation (in the same narrow sense) will be what cognition *is not*. To the pan-computationalist, everything, including cognition, is computation (in an extremely wide sense). Others redefine the concept of computation in such a way that it only includes the manipulation of meaningful or generally language-like structures, which makes it something other than the meta-mathematical tool as which it was intended. To the inhabitants of any position in between, an array of intermediate notions of computation will have to be matched against a variety of cognitive processes and their elements. Cognition will involve computation to some extent, at some level, and it allows for computational modelling to some extent, at some level. At least in most respects, the definition of computation is independent of the definition of cognition, while the latter typically at least partly depends on the former on any computational view. Hence, one possible way of formulating a research question is this:

Q_0 What is the difference between computation in human beings and machines?

As it stands, this question is extremely unspecific though, as it would require the notions of computation and machine to be sufficiently defined beforehand. These definitional issues will be addressed in due course, but even with that clarification in place, the question remains by far too general and too little meaningful to be fully explicated, let alone answered in one short paper. However, there is a different, more semantic than terminological vagueness that is inherent to the above question. This vagueness lies in the possibility of unpacking Q_0 into two alternative questions:

Q_a What is the difference between a human when s/he computes and a machine when it computes, qua their activities?

Q_b What is the difference between a computing human being and a computing machine, qua their respective nature?

Of these two questions, only Q_b addresses ontological issues concerning the nature of computation, human beings and machines, but a prima facie plausible line of reasoning between them might be this: The activities (re Q_a) are different because the nature of human beings and machines (re Q_b) is different. They would have to accomplish the same computational tasks in the same way in order to possibly count as equivalent, and hence in the sense that is similar to what is called intensional identity (more about which in Section 4) in the philosophy of mathematics. It is this line of reasoning that I wish to put into question. A difference in activities does not pre-empt any judgement on a difference in nature. Conversely, we might find some form of analogy or equivalence in activities without having to assume analogy or equivalence in nature. Extensional equivalence in activities might be sufficient in many relevant cases.

If we want to grasp the extent of possible analogies and equivalences in nature instead, the initial question Q_0 might be rephrased in yet another way:

Q_c What are the limits of computation in human beings and machines respectively?

Again, the question is too fundamental and too general to be usefully addressed here, but the previously outlined argument might help to demarcate the domains of computation in human beings and machines respectively, and to compare them. In doing so, the discussion of the limits of, and differences in, computation will refer to the notion of physical computation, understood as the concrete ways in which computations are accomplished or "implemented" in concrete physical systems, of which there are various competing accounts.[2] The limits of physical computation are defined not by logico-mathematical principles alone, but in addition by concrete physical enablers and constraints. Physical computation does not enjoy the privilege of infinite time, infinite storage or other hypothetical conditions that can be mobilised in accounts of computability-in-principle in logic and mathematics. Instead, theories of physical computation task themselves with identifying the conditions under which any kind of processes, natural or other, will count as

[2] The concept of physical computation was first explicated by Putnam (1975). Its trivialising implication that every system implements every computation, given an arbitrarily chosen computational description (which was welcome to Putnam as a critic of computationalism and AI), led to a diversification of accounts that proposed causal, descriptive or counterfactual constraints on the sets of physical processes admitted to the domain of computation, most prominently by Chalmers (1995, 1996); Copeland (1996); Pitowsky (1990); Rescorla (2014).

a computation in light of a real-world-minded definition of computation and its elements.

If the computational processes under consideration here are physical processes that depend on the concrete conditions within and around the concrete systems in question, and if these concrete conditions work as enablers and constraints on what kind of computational systems human beings and machines can be, the domains of computation in human beings and machines might turn out to be quite independently determined and delimited. At the same instance, they might have a limited but instructive bearing on questions of cognition in either kind of system, to the extent that machine-based computation can or cannot realise certain cognitive processes, and to the extent that human computation serves as a model for machine computation. This is the case I will at least begin to make in what follows.

3 Delimiting computation and cognition

The theoretical machine first devised by Alan Turing (1936), introduced as the Logical Computing Machine (LCM) and later known as the Turing Machine, originally stood in the service of developing his theory of computability, which sought to apply the methods of arithmetic to solve meta-mathematical problems, and primarily needs to be understood in this light. Still, the design of this theoretical machine was then used to inform the design of real machines to a significant extent, which partly eclipsed its original purpose. On an abstract level, however, any system that adhered to the set of basic operational principles specified by Turing would be able to perform logical computations.

A basic and prima facie uncontroversial reconstruction of Turing's original (1936) definition of computation (which he famously never made fully explicit) might look like this:

c.1 *The domain condition:* The domain of computable functions is exhausted by the functions that are 'effectively calculable' in such a way that they can be solved, in principle, by an LCM.

c.2 *The specification condition:* An LCM comprises of a finite set of symbols, a finite set of possible states, a transition function and a potentially infinite memory.

Remarkably, the mechanical operational principles of the LCM specified in c.2 were modelled on the operations and "configurations" of mechanical typewriters – moving type heads over tape, writing, but also reading and erasing symbols (Hodges 1983, 96-98) – whereas the LCM's mathematical operational principles that delimit the domain of computation in c.1 were defined with respect to human computational abilities, and concretely modelled on the behaviour of human computers, who inadvertently bequeathed their class noun to digital computers (Copeland 1997, 2017): Their task was to accomplish complex calculations in a collective but centrally governed, namely algorithmic fashion, in which higher-order logico-mathematical operations are broken down into elementary arithmetical routines that could be accomplished with only a modicum of mathematical skills.

In human computing, thus conceived, only a small subset of cognitive abilities is involved. On this level, human computation is distinguished from higher-order human mathematical and other cognitive skills by not even potentially involving inventiveness, intuition, creativity or any other abilities of which we do not know whether or to what extent the are amenable to formalisation. Only this small subset of abilities is considered as a template or model for computation. Even more significantly, it is thus considered only in the form of the behaviours that it governs, not in its mode of realisation. Hence, mechanical computation in Turing's theoretical sense and machine computation in concrete devices are at least prima facie restricted to the tightly circumscribed analogy with the rule-governed behaviour of human computers. Neither the mode of realisation and therefore the inner nature of human computational skills nor the nature of other elements of human cognition will have a bearing on this analogy. Hence, a workable account of computation will not depend on a given state of human knowledge about the inner workings of any of these abilities.

This situation could be otherwise only under either of two strictly disjunct conditions: Machine computation would not be restricted to the behavioural analogy to human computation, as conceived of by Turing (1936),

e.1 if the concrete mode of realisation of computational abilities within human beings were a necessary precondition of the performance of *any* computation.
e.2 if machine computation in the theoretical and concrete sense were not restricted to LCM-type computation.

Condition e.1 would altogether undermine the original concept of computation, for being the exact opposite of the idea of computational operations to be multiply realisable in any suitably specified system that lies at the very heart of Turing's idea of the LCM. It would require type-identity where functional analogy and equivalence of operations were expressly intended to be sufficient. Condition e.1 is not only overly restrictive but also implausible, as any real-world machine that performs computations in the way specified for the LCM bears testimony to the possibility to performing computations in other than human-specific ways. Otherwise, the notion of human computation would have to be enriched to a point that contradicts the definition under consideration here.

In this latter spirit, condition e.2 would ask for computation to involve elements that Turing expressly excluded from his concept. Alternatively, it suggests this concept to be amended if deeper analogies are discovered or anticipated that make computation appear a less restricted phenomenon. Either way, the question would be whether the phenomenon in question, if retaining the label "computation" at all, can still be analysed in terms of Turing's theory of computation. This line of reasoning has been followed by various accounts, such as Miłkowski (2013, 2018); Piccinini (2015), that seek to unshackle the concept of computation from Turing's original definition and make it more compatible with contemporary psychology and neuroscience while others, such as Copeland (1996), seek to accomplish the latter by making Turing's notion more precise while otherwise staying true to it.

Once condition e.1 has been duly dismissed and condition e.2 duly qualified, any possible equivalence between human and machine computation can be more precisely described as follows: As far as internal processes that realise computations are concerned, human computation involves other processes than machine computation does, but a clearly delimited analogy in observable behaviours that accomplish a clearly delimited set of operations will suffice to establish a certain kind of equivalence between human and machine computation. The kind of equivalence in question will be extensional by definition. This general kind of equivalence became mainfest already in the development of the early theories of computation. It concerned the question whether the methods of determining the decidability of first-order logic in Turing's, Church's and other approaches at that time all produced the same result. Extensional equivalence refers to the observation shared between them that the various methods involved indeed produced the same result (Piccinini 2017).

4 Extensional equivalence and intensional identity

Extensional equivalence in this context, and some of its implications, will be best understood in light of the conceptual pair of extensionality and intensionality, which makes related appearances in logic, mathematics and the philosophy of language, beginning with the work of Gottlob Frege (1892). A sentence, phrase or other linguistic context is considered extensional, first, if co-referential terms within that context can be mutually substituted in truth-preserving fashion and, second, if the context allows for existential generalisation. ("Morning star" and "evening star" refer to the same celestial object, and if there are a morning star or evening star which are shining, there will exist at least one object that shines.)

In contrast, intensionality is a quality of linguistic contexts that depends on the sense, in Fregean terms, of its elements, in which identical referential relations (to the same celestial object) are differently expressed (as morning star and evening star respectively). Depending on such differences in their mode of presentation ("Art des Gegebenseins", Frege 1892, 26-7) to speakers, and depending on the speakers' respective knowledge, beliefs and predispositions, linguistic items may be differently used and perceived, with potentially differential effects on their truth or falsity even if they are coreferential. In particular, intensional contexts, which are sentences or phrases that report beliefs, neither guarantee a truth-preserving substitution of coreferential terms, nor do they guarantee the possibility of existential generalisation. (If "I believe that the morning star is shining" is true, it might still be false that "I believe that the evening star is shining", if I do not know that they are the same object, and I might be mistaken in my beliefs concerning the existence of shining objects to begin with.) In consequence, two sentences will be intensionally equivalent only if they not merely refer to the same subject matter but if they do so in the same, truth-preserving way, which is a condition that is difficult to meet in any other case than identity in all or in all relevant respects.

The extensional / intensional distinction has been mainly discussed in the context of philosophical semantics, where intensionality is often characterised as the mark of intentionality, understood as the quality of being meaningful, which in turn is considered the exclusive domain of thought and language. However, the origins of this distinction are at least equally close to questions of logic and mathematics as they are to language. Among other things, Frege's analysis provides the foundation of intensional logic, and it provides the critical backdrop for the intensional approach to mathematics (Feferman 1985).[3] In critique of Platonist externalism concerning mathematical forms, it highlights the importance of the mode of presentation of mathematical objects to human beings and the human

[3] For further discussions of intensionality in mathematics over the last decades, see Niebergall (2005); Peregrin (2018); Quinon (2019); Shapiro (1985); Sylvan (2003).

practices of endowing them with meaning. Paradigmatically, "two rules may have the same values at all arguments but they are not identified unless they are recognized to be the same as rules" (Feferman 1985, 44). However, the project explicitly refrains from answering questions of intensional identity while inquiring into the metaphysical foundations of mathematics. The metaphysical issues are even more evident in the mathematical concept of intensionality than in the linguistic one: what is the nature of a sign in relation both to a designated world affair and to its users?

The preceding tentative analysis of intensionality will help to answer the question whether analogies in behaviour and function are sufficient to support equivalence in computations. If identity in intensional properties between human and machine computation were required, it would likely never come to pass, first, because it demands that the mode of presentation of any function and any route towards a computational solutions would have to be identical for all systems under consideration. Second, however, the systems under consideration are sensitive to an identical mode of presentation in variant ways, depending on their respective constitution. To the extent they are designed to perform computations, the systems will perform computations that exemplify the same mathematical structure in different – paradigmatically neuronal, continuous versus electronic, discrete – ways in many cases precisely in order to be able to produce the same results. This is the first case against intensional identity.

Conversely, if the mode of presentation remains identical across various systems and therefore is insensitive to variations in their nature, the outcomes of the computational process, if there are any, are more likely to diverge than if the mode of presentation systematically maps their variations in nature. Intensional contexts are not truth-preserving precisely because contexts that are otherwise identical might produce diverging results for variantly disposed human beings. Even if one merely compares the operations of various digital computing devices, one and the same set of instructions and input variables might be straightforwardly computable for one system, require patches for another and be wholly unusable for a third, depending on such mundane technological factors as the processor architectures or operating systems used. This is the second case against intensional identity.

Moreover, even if one weakened the condition of intensional identity to "identical in every relevant respect", the question would remain what the relevant aspects are on the spectrum between producing an identical solution to an identical problem in extensionally equivalent ways and fundamentally being a type-identical system. In fact, the strategy adopted by anti-computationalists like Searle (1980) is to demand type-identity between the systems under consideration in order to grant them the status of computational equivalence: Only those physical systems which are human organisms compute in such a way that genuine intentional phenomena such as language understanding and consciousness will come to pass. The third case against intensional identity is not directed at the concept itself but at this specific line of argument, which not merely refuses to consider any possible form of equivalence apart from identity, but also ties identity to the presence of a an array of human-specific psychological phenomena and their qualities. These, however, are not *per se* part of the concept of intensionality. In a certain respect, the Searlean argument is fashioned as an answer to the Q_b version of the research question, where that answers amounts to a version of condition e.1 above: Any computation in the relevant sense would have to belong to the domain of human computation, which remains unattainable for machines by definition.

On the background both of these considerations and of the original observations concerning extensional equivalence between early theories of computation, there should be defensible routes towards equivalence between computation in human beings and machines. Such equivalence will paradigmatically be extensional, but intensional identity might have a circumscribed domain of application. Extensional equivalence might be established for all those levels and aspects of human cognition which are amenable to a computational account at all, and for all those levels and aspects of machine operations which are equally subsumable under such an account. The open questions to be discussed in the remainder of this essay are twofold: First, on the cognitive side, how much of human cognition is subsumable under a computational account, and what is the status of these levels and aspects? Second, on the computational side, will the concepts of computation involved remain restricted to LCM-style computation?

Before turning to the first question in the next section (5), I will briefly indicate a route towards an answer to the second question, which will be taken up again in Section 6. The currently most comprehensive and systematic account of computation that goes beyond Turing's LCM computation is the mechanistic account of computation (Piccinini 2015 and Piccinini and Craver 2011 in particular, but also Fresco 2014; Miłkowski 2013). It has the distinction of being both more general and more precise than Turing's account while maintaining his original idea of multiple realisability of computations. Abstracting from logico-mathematical problems and the rules of arithmetic while focusing on issues of physical implementation, the mechanistic account postulates the "medium-independence" and substrate-neutral "vehicles" of computation. Computation, thus understood, is the processing of vehicles of any sort – markings on tape, switches in electric circuits or activation states of neurones – in accordance with rules that are sensitive to relevant properties of those vehicles. The vehicles and their relevance are defined in light of the overall properties of the system of which they are part. A system that computes is a system that can be explained as having been designed or evolved in such a way that it contains an arrangement of organised features, or shortly a "mechanism", with the function of performing computations, thus defined.

The mechanistic account of computation aims to subsume a manifold of types of processes by means of defining them precisely enough to strike a balance between avoiding pan-computationalism and an undue restriction to LCM-style computation. Moreover, it allows to turn one's primary focus away from Q_b-type questions concerning the deeper nature of computing systems and towards an account of how they operate, and hence Q_a above.

5 The domain of human invention

If the previous discussion is to the point, the anti-computationalist's charge against equivalence between human and machine computation is either, in its stronger form, that it cannot be accomplished at all because it cannot be intensional, or, in its weaker form, that it may be extensional but then will remain restricted to those domains of human cognitive activities that can be described in terms of LCM-style computation. The former charge amounts to denying the possibility of computer-based analogues of human cognition, whereas the latter will amount to curtailing their power and scope. Against this charge, the mechanistic account of computation goes some way towards broadening the domain of computation by overcoming the restriction to LCM-style computation. However, unless it opts for pan-computationalism and its trivialising implications, it cannot rule out the possibility that there are key domains of human cognition that may still remain inaccessible to the broader mechanistic concept of computation.

There is a set of well-rehearsed arguments pro and contra cognition being an intrinsically and irreducibly embodied and situated phenomenon. It is usually taken to be the direct and symmetric counterpart of arguments pro and contra computationalism. As these debates cannot be covered here in any reasonable detail, and especially as one of the aims of this essay is to decouple some of the presuppositions that are usually made by either party in that dispute, I will take a slightly different, exemplary perspective on the relation between computation and cognition. I will consider the role of human invention in mathematics. This choice is not primarily motivated by the intention to make an argument by example, but by the prospect of connecting the following considerations to the notion of human and machine computation in light of possible limitations on human cognition in the very domain that gave rise to the computational paradigm.

One can provide at least three distinct views of the relevance of invention and intuition to mathematics, with various possible shades between them, that have a highly pertinent historical pedigree that manifested in foundational crisis of mathematics during the early 20^{th} century:[4]

i.1 *Logicism*: The role of human invention in mathematics is tightly circumscribed, as all possible mathematical concepts are founded in, and reducible to, a limited set of logical axioms, which themselves are self-evident and neither amenable nor requiring logical proof. These foundational axioms are discovered, not invented. All of arithmetic, even natural numbers can be derived from those foundational axioms. This is the logicist view that was initiated by Frege (1884) and became influential through the the *Principia Mathematica* (Whitehead and Russell 1910-1913), and informed analytic philosophy from Carnap to Wittgenstein. This view might be accompanied by, but is distinct from, the Platonic postulate that the existence and form of mathematical objects are independent of human thought, too. In any case, however, logicism is committed to the view that mathematical forms are meaningful first and foremost in an extensional sense, with all invention, convention and other human factors being subordinate to this condition.

i.2 *Formalism*: The role of human invention in mathematics is significant but circumscribed, as mathematics is conventional in the same sense in which natural language is conventional: it is based on signs whose shape is a matter of invention and agreement and may operate freely in this respect while, taken as a sign system, it is subject to definite rules that express necessary forms of reasoning. In particular, the mathematical language game is supposed to be uniform, complete and non-contradictory. This is the formalist view that

[4] The following reconstruction is largely based on the exchange between Carnap (1931); Heyting (1931); von Neumann (1931) and the accompanying discussion by various authors (1931) in the second volume of *Erkenntnis*, which occurred at a time when the foundational crisis had mostly subsided, and hence with the benefit of hindsight.

has been paradigmatically formulated by Hilbert (beginning in 1900). In a reversal of the logicist view (i.1), it sought to reduce logic to mathematics and expressly shunned ontological commitments concerning the reference of mathematical symbols. Not least by providing the critical backdrop for Turing (1936), it directly informed theories of computation. Its closest philosophical cousin is the hypothetico-deductive paradigm in philosophy of science (Popper 1959), according to which free-form, intuition-guided invention of hypotheses may become important to the context of discovery in many cases (see also Einstein's notion of an "Einfühlung in die Erfahrung", 1918, 31), whereas the context of justification is marked by hypotheis-testing and thereby fleshing out fundamental natural constraints.

i.3 *Intuitionism*: The role of human invention in mathematics is fundamental and only minimally constrained, as virtually all mathematical principles and objects are human-made, not only the conventions of mathematical notation. Mathematical objects have no independent existence, nor are they reducible to logic (i.1) or to the contingencies of linguistic construction (i.2). Mathematical objects are created by the human mind and exist in the human mind. The function of mathematical symbols and notation is to represent those mental entities. This is the intuitionist view that has been introduced by Brouwer (1907/1975) and that lost its influence on mathematics after the 1920ies, while having a parallel in later constructivist theories in philosophy. According to the intuitionist view, the only principled mathematical constraints on human inventiveness in mathematics lie in natural number concepts and elementary arithmetic operations – which arguably can be found to some extent in some animals, too (see Dehaene 2011; Fabry 2018). All remaining constraints are cognitive in nature, ultimately imposed by the structure and limits of human neuroanatomy and neurophysiology, and hence practical, not principled in kind.

Of the views discussed here, the intuitionist (i.3) fared most poorly in terms of continued acceptance or even acknowledgement of merit, at least the latter of which can be safely ascribed to logicism (i.1). When it comes to continued acceptance, formalism (i.2) has been most successful. Despite or possibly *because* of the demonstrated failure of its completeness theorem, it nurtured computationalism.

If there are principled constraints on human inventiveness in mathematical matters, according to i.1, and if these are in some respects metaphysical, *pace* Carnap (1959), because there is a logical structure of the world with which we cannot argue, there still will be no implication that human inventiveness needs to be rejected or curtailed in its status in other areas than logic and mathematics. After all, religion, art an other areas of human cognitive activity might actually require a relative absence of the logic and a strong presence of inventiveness in order to thrive (as Carnap 1959 did not hesitate to admit). However, unless it can be demonstrated that these other areas of human thinking and activity are at least similarly central to the entire *conditio humana* as are logic and mathematics and may exert an influence on them, the domain of human cognition will remain relatively restricted to what logic and mathematics can accomplish. Human computation, in Turing's terms, will be one, arguably minor but genuinely representative, constituent of that relatively restricted domain. It will most likely be unable able to expand that domain.

If there is a significant domain for human invention in mathematical matters that is still subject a set of principled constraints, according to i.2, the domain of invention can be circumscribed by means of these constraints. If these constraints lie in the logical forms that human reasoning, qua reasoning, necessarily has to assume, they can be expressed accordingly, and hence at least potentially in computational terms. After all, i.2 is the approach most inclined towards granting the possibility of computational descriptions of the widest possible range of world affairs. Its conventionalist, quasi-linguistic premises are a central feature to resurface in the theories of computation that descended from it. On this background, the formalist may admit but is not compelled to acknowledge the existence of elements of human cognition that are essentially not computational in nature. If admitting for the existence of such elements, he or she will either be able to maintain that they might still be describable on the grounds of the conventions of a computational language or, if such a description is not forthcoming, that its domain can be defined *ex negativo*.

If, however, there are virtually no principled constraints on human inventiveness according to i.3, there will neither be a need for circumscribing the boundaries of its domain in logical or computational terms, nor will there be a possibility of doing so. Any and all formal principles that human thinkers do and possibly could devise will arise from a domain that is intrinsically not bound to or defined by such principles. In this case, there is no reason, on the one hand, to assume either that human cognition is computational in nature in any recognisable way or even that it may receive a comprehensive and authoritative description in the terms of one of the many possible descriptions that the human mind can devise. On the other hand, it is perfectly conceivable on this view that the power of human inventiveness is such that, given enough time and resources, more comprehensive and authoritative descriptions of human cognition may be devised than the expressly limited computational analogy devised by Turing or any of its descendants could furnish. The intuitionist view might even allow for the notion of machines to be invented that are cognitively more powerful than the unaided human mind. They might not be computing machines though.

6 Machine abilities and constraints

After the possible types of equivalence between human and machine computation have been clarified in Section 4, and after the possible extension of the domain of human cognition vis-a-vis human computation has been explored by proxy of the status of invention in mathematics in Section 5, what is still missing from the picture is an exploration of the domains of machine activities that might go beyond LCM-style computation. In order to obtain the other half of the picture, I will return to the bearing of variant concepts of computation (see Section 3 above) on the domain of machine computation.

According to definitions c.1 and c.2 on p. 5, everything that an LCM or 'Turing Machine' can solve is computable. It has been a matter of controversy whether Turing's conception of the LCM was meant to be strictly constrained by the behaviours of a human computer, defined as a person doing calculations with pencil and paper (Copeland 2017), or whether it was – even purposefully – indifferent to the human versus machine nature of the system doing the computations (Hodges

2008). Either way, Turing-computability as originally conceived did not concern principled abilities or limitations of machines *qua* machines. The LCM was introduced as a theoretical machine in the first place, while providing key elements of the design of digital computing machines first built a decade after Turing's 1936 publication.

Even though Turing introduced the concept of the LCM by proxy of the model of human activities of computation, the possible interpretations of how real or conceivable machines may simulate these activities will matter. After all, the specification condition that defines the elements of an LCM (c.2) has often been taken literally, as a blueprint of the functional structure of real machines. Considerations of machine computation will thereby be narrowed down to the original analogy with the behaviour of human computers, hence in disregard of the cognitive processes that realise this behaviour. All and only those machines which meet this specification, paradigmatically or exclusively digital computers, will rightfully count as computers. No machine that adheres to c.2 will be able to accomplish more than what is specified in c.1. Any machine that exceeds the abilities defined in c.1 will have to do so by means other than those specified in c.2.

Conversely, the domain condition that defines what process count as computable (c.1) has often been interpreted in a maximal sense, as implying not only that every function that *any* machine can solve is computable, but also that LCMs, as specified in c.2, will be in principle able simulate the operations of those other machines. The previous constraint that commits the notion of computation to the model of the behaviours of human computers would thereby be lifted, and a poorly defined prospect of LCM-style computational solutions to any function opened up – which is not what Turing appears to have had in mind.

This point can be elucidated by reference to Robin Gandy's "Thesis M" (1980, 124), which is derived but expressly distinct from Turing's thesis. It plainly states: "What can be calculated by a machine is computable". This thesis is intended to highlight the possibility that some machines may compute a function in ways that are not accessible to a human computer, paradigmatically in the case of machines that use parallel processing. Beyond this rather modest point, Thesis M, too, has been variously interpreted to mean that "whatever can be calculated by a machine can be calculated by a Turing machine" (Copeland 2017, 10) or that "anything that a machine can do is computable" (Hodges 2008, 86-7). These are quite distinct claims concerning the scope of possible accomplishments of machines: On the first view (according to Copeland's rephrasing), Turing computability will reign supreme, either within its own delimited domain because machines cannot accomplish more than what the LCM does, or, on a much stronger interpretation, the LCM may transcend its own boundaries by simulating the behaviour of machines that accomplish tasks that go beyond its original specification. On the second view (according to Hodges' rephrasing), the notion of computation is not restricted to LCM-style computation in the first place.

In fact, both Thesis M and the most overbearing interpretations of Turing's programme have given rise to the maximal interpretation of the nature of computation, which includes the notion of conceivable machines that could, in principle, computationally solve more than the set of LCM-computable functions. This notion raises the question of whether we should still refer to whatever goes on in those more powerful machines as computation, and why we should do so. If we take Turing's definition at face value, computability is always relative to the design of his LCM, and it would take some justification to call Turing as witness in an argument for a notion of computation that is not covered by, let alone compatible with, his own definition.

However, to complement his own restrictive concept of computation, Turing introduced the notion of an "oracle", which could solve some or possibly all the functions that an LCM *cannot* solve by providing a distinct, "hypercomputational" mechanism.[5] Given that Turing did not specify what the nature of such a mechanism might be except "that it cannot be a machine" while, nonetheless, "with the help of the oracle we could form a new kind of machine" (Turing 1939, 173), there are two divergent interpretations of what an oracle could be and accomplish:

o.1 If any machine is necessarily restricted to computable functions in terms of Turing's LCM, there will be principled limitations on conceiving and building a machine that could solve any non-computable function. If there is an oracle, it will not be a machine (Hodges' interpretation).

o.2 If Turing's oracle could inform the operations of a real machine, a machine would become possible that solves non-Turing-computable functions. There will be no principled limitations on what a machine could potentially accomplish, but that machine would not be a LCM (Copeland's interpretation).

Matching this distinction against the distinction between different readings of the role of human inventiveness in mathematics in Section 5, we end up with two contrasting 'no principled limitations' claims concerning human inventiveness (i.3) and machine abilities (o.2) respectively, and with two contrasting 'principled limitations' claims (a stronger i.1 and a weaker i.2 versus o.1). These contrasting claims will have distinctive implications concerning the possibility and limitations of Artificial Intelligence (AI), especially with respect to the question whether machines are conceivable whose computational or hypercomputational abilities approximate or are equivalent to key elements of human cognition. I will refer to this latter possibility as "Strong AI". If we juxtapose the positions discussed under "i.n" and "o.n", the following landscape of hypotheses emerges:

h.1 If most or all mathematical principles are human inventions, with no principled boundaries (as in i.3), and if human beings could build a machine that solves non-Turing-computable functions that captures non-Turing-computable elements of the human mind (as in o.2), this machine would not be a Turing Machine but involve an oracle in a yet-to-be specified manner. Strong AI would be possible in principle but would have to match the near unlimited degrees of freedom of human inventiveness.

h.2 If most or all mathematical principles are human inventions, with no principled boundaries (as in i.3), and if human beings nonetheless remain unable to build machines that solve non-Turing-computable functions that capture non-Turing-computable elements of the human mind (as in o.1), it would still remain possible in principle to invent other, yet unspecified methods of solving those functions that do not involve machines. There would only be Turing Machines, hence no machine-based route to Strong AI.

[5] For the coining of the term "hypercomputation", see Copeland and Proudfoot (1999); for an overview of hypercomputational approaches, see Copeland (2002); for a recent example of a concrete approach, see Stacewicz (2019).

h.3 If all or at least some fundamental mathematical principles are not human inventions (as in i.1 and i.2 respectively), and if there are principled constraints on what a machine could provide in terms of computational solutions that would confine them to Turing-computability (as in o.1), the constraints in question will probably but not necessarily be stricter for machines than for humans. Any kind of AI would only be possible within the confines both of Turing computability and of the respective i.n condition.

h.4 If all or at least some fundamental mathematical principles are not human inventions (as in i.1 and i.2 respectively), and if there are no principled constraints on what machines could provide in terms of computational solutions that would confine them to Turing-computability (as in o.2), any remaining constraint could in principle be no stricter for machines than for humans. Strong AI would be possible but it would not be Turing-Machine-based AI.

This landscape of possibilities does not explicitly cover all in-principle imaginable scenarios, but it may include them as more or less plausible variants. Both h.1 and h.2 can be rendered in such a way as to appear straightforwardly anti-computationalist – if we take condition i.3 to mean that the human mind is non-computational in important respects, and if we assume that machines that embody some of its key properties are either impossible or would not be computing machines anymore. Conversely, a scenario in which humans and machines alike are confined to LCM-style computation would be an extreme version of h.3, in that it identifies the constraints on human inventiveness in particular and cognition in general with the boundaries of LCM-style computation. Likewise, h.4 can be pushed towards including a scenario in which human beings invent an artefact whose underlying computational principles they themselves could not possibly comprehend. This case would be qualitatively different from the epistemic opacity of advanced AI systems, where the issue lies in limitations on human cognitive tracking of their operations, so that we understand the principles but fail at the task of grasping their execution in scope, speed and complexity.

As instructive as these hypothetical scenarios may be, there is an aspect of h.3 and h.4 in particular that is more relevant to the present context. In postulating more restrictive i.n conditions and thus limiting the presumed degrees of freedom of human invention in the very domain of human cognition that also defines the boundaries of computation, they include the possibility that machine computation approximates, becomes equivalent to or even exceeds that more restricted set of human abilities, either on an LCM or a hypercomputational level. The question is what the nature of this equivalence would be in light of the intensional / extensional distinction outlined in Section 4, and how it may come to pass.

If the humans and machines involved are found to accomplish the same tasks and solve the same problems to the same degree but in distinct ways that each match their respective constitution and abilities, their operations will be extensionally equivalent. They will map the same solutions onto the same problems in distinct but rule-governed ways that ensure equivalence between them. If and when a strong version of the "formalist" i.2 condition applies, the accomplishments of machines will be in correspondence to key constituents of human cognitive abilities to the extent they trace necessary forms of human reasoning. The functions of the machines are specified in line with mathematical conventions that share these foundations but are otherwise malleable in light of human knowledge and purposes. According to h.3, the fact that LCM-style computation merely concerns an analogy between machine functions and the behaviour of human computers will impose a necessary and fairly strict limitation on machine abilities. In the most generic type of case, this analogy disallows, for example, parallel distributed processing in machines because this is not how a person calculates with pencil and paper. There will be no such principled limitation that would be imposed by h.4, as it allows for the invention of machines that transcend LCM-style computation by any means that come to pass but that might remain uninformative with respect to how human cognitive abilities might be computationally realised. LCM-style computation is unlikely to achieve much in this respect.

If we consider the mechanistic concept of computation instead of the LCM, the distinction between h.3 and h.4 will assume a different shape: it will concern possible deeper analogies between computational operations inherent in human cognition and machine computation. Here, a pertinent example would be biologically realistic neural network modelling (which at the level of basic imolementation remains committed to LCM-computational principles though). The extension of the domains of human and machine computation could be made equivalent on the basis of any such analogy that might still be discovered. The question will become whether there are relevant and genuine non-computational elements of human cognition, and whether and to what extent analogues of these could be implemented in hypercomputing machines.

If, in contrast, the humans and machines involved are presented with the same tasks in the same way, and if they also solve them in the same way in all relevant respects, their operations will be intensionally identical. As a minimum condition, both the mode of presentation and the solution will remain insensitive to any variation in internal realisation of the solution process. A pertinent example of what is required here would be an enriched version of the Logical Neurone model (McCulloch and Pitts 1943): If there is a structure and a set of operational principles that is realised both by a logical calculus and by the human nervous system, and if it can be demonstrated that the same processes describable by the same logical calculus will occur in a human nervous system and a computer implementation of that calculus, provided that both are presented with the same input variables in the same way, a case for intensional identity might be made. The requirements involved would be very steep though, as the requisite mode of presentation might be that of visual perception and also include elements of, necessarily embodied, self-perception. If and when a strong version of the "logicist" i.1 condition is applied to cases of this kind, the degrees of freedom of human invention in mathematics that are involved will be tightly constrained by a logical structure of the world to which all well-founded mathematical principles correspond. To the extent that computing machines are organised and operate along these principles, they will make reference to that same logical structure of the world.

Given that disallowing any variation in internal realisation of the relevant functions would forestall any comparison between human and machine computation, and given that such a requirement is not part of the concept of intensionality, the degrees of variation that are still reconcilable with the condition of an identical mode of presentation will have to be determined through empirical investigation. The logicist approach i.1 is quite well-disposed towards identifying these conditions though, as it postulates a logical structure of the world that may serve as a common external reference point. To the extent that human cognition partakes in that logical structure of the world, some degree of equivalence between its functions and operations and the functions and operations of machines will be conceivable. To the extent that human cognition does not partake in that logical structure, it will remain beyond the reach of any machine that embodies the computational principles that build on that structure. Either way, the basis of comparison cannot be LCM-style computation, as it only concerns behaviour-based analogies with human computers, which will be too narrow an analogy. Even on the weaker concept of intensionality proposed here, the comparison will concern aspects of internal realisation or implementation, and hence will be better based on a mechanistic concept of computation that is able to formulate identity conditions with sufficient precision and generality.

7 Conclusion

The synthesis of the discussion in the preceding sections is this: In pursuit of tentative and partial answer to question Q_a, Section 3 demonstrated that the concept of computation was originally modelled on human computers but designed in such a way as to be implementable in a certain kind of machines, namely Turing's LCM. It has later been refined in order to cover a broader range of phenomena by means of a more abstract notion of "mechanical" computation that can be implemented in a greater variety of systems, both natural and artificial. Section 4 explored the possible types of equivalence between human and machine computation, concluding that extensional equivalence between certain effects of certain operations in human beings and machines can be established, whereas intensional identity needs to be considered an undue requirement for either, especially in light of the subsidiary requirement that they would have to be identical in nature in every relevant respect. Section 5 developed an overview of the scope of those human cognitive abilities which might go beyond the domain of extensional equivalence between computational operations, with particular focus on the relation between human invention in mathematics and computation. Finally, section 6 juxtaposed this image with an image of the domain of possible machine activities that go beyond LCM-style computation.

In the course of this argument, I have tried to provide a partial answer to the Q_c version of the initial research question by proxy of providing a partial answer to its Q_a version. However, that answer is *not* one that tells us what computation in human beings and machines actually is and what the differences between them really are in a metaphysical sense or by logico-mathematical proof. It is much more open-ended in at least two ways. First, my partial and pragmatically minded answer focuses on the activities of human beings and machines, not on their nature, in that it highlights the importance of extensional equivalence of operations with respect to solving a certain set of problems. Intensional identity in the strong sense mobilised by anti-computationalists is not required and, although partly defensible in a weaker sense, remains difficult to unequivocally establish. Second, I have made case for the relation between human and machine abilities to be treated an open question, and as multi-faceted rather than as a strict dichotomy. Any possible decision for one position or another will have rich and normatively relevant implications. On most of the more tenable accounts outlined above, the domains of human cognition and machine computation will be distinct in kind and extension, but this will be a matter of empirical investigation and concrete human invention.

Ultimately, I have made a case for reconsidering the concept of machine that is involved here. Where anti-computationalists defined machines as precisely those entities which never might and never will match human accomplishments, so that any hypothetical machine that might transgress this boundary would cease to be perceived and referred to as a machine (as Collins and Kusch 1998 appear to argue), the arguments brought forward here remain agnostic with respect to this criterion. Any principled limitations on the possibility and the abilities of conceivable artefacts do not fall into one with the factual limits of one's imagination. After all, Turing (1950, 442) made a point of using his LCM for pushing the envelope of the meaning of the word "machine" at a time when the paradigm of machines still were power machines and when computing machines where only emerging. On all accounts, he has been at least moderately successful in pushing that envelope.

References

Brouwer LEJ (1907/1975) On the Foundations of Mathematics, vol 1, North-Holland, Amsterdam, pp 11–101. Translation of *Over de Grondslagen der Wiskunde*, 1907

Carnap R (1931) Die logizistische Grundlegung der Mathematik. Erkenntnis 2:91–105

Carnap R (1959) The elimination of metaphysics through logical analysis of language. In: Ayer AJ (ed) Logical Positivism, Free Press, New York, pp 60–81

Chalmers D (1995) On implementing a computation. Minds and Machines 4:391–402

Chalmers D (1996) Does a rock implement every finite-state automaton? Synthese 108:309–333

Collins HM, Kusch M (1998) The Shape of Actions: What Humans and Machines Can Do. MIT Press, Cambridge

Copeland BJ (1996) What is computation? Synthese 108(3):335–359, DOI 10.1007/BF00413693

Copeland BJ (1997) The broad conception of computation. American Behavioral Scientist 40(6):690–716, DOI 10.1177/0002764297040006003

Copeland BJ (2002) Hypercomputation. Minds and Machines 12:461–502

Copeland BJ (2017) The Church-Turing thesis. In: Zalta EN (ed) The Stanford Encyclopedia of Philosophy, spring 2020 edn, The Metaphysics Research Lab, Stanford, p html, URL `https://plato.stanford.edu/archives/sum2020/entries/church-turing/`

Copeland BJ, Proudfoot D (1999) Alan Turing's forgotten ideas in computer science. Scientific American 280(4):76–81

Dehaene S (2011) The Number Sense: How the Mind Creates Mathematics, 2nd edn. Oxford University Press, Oxford/New York

Dreyfus HL (1992) What Computers Still Can't Do. A Critique of Artificial Reason. MIT Press, Cambridge/London

Einstein A (1918) Motive des Forschens. In: Warburg E, Planck M (eds) Zu Max Plancks sechzigstem Geburtstag. Ansprachen, gehalten am 26. April 1918 in der Deutschen Physikalischen Gesellschaft, Berlin, C.F. Müllersche Hofbuchhandlung, Karlsruhe, pp 29–32

Fabry RE (2018) Turing redux: Enculturation and computation. Cognitive Systems Research 52:397–808

Feferman S (1985) Intensionality in mathematics. Journal of Philosophical Logic 14:41–55

Frege G (1884) Die Grundlagen der Arithmetik. Eine logisch mathematische Untersuchung über den Begriff der Zahl. Wilhelm Koebner, Breslau

Frege G (1892) Über Sinn und Bedeutung. Zeitschrift für Philosophie und Philosophische Kritik 100(1):25–50

Fresco N (2014) Physical Computation and Cognitive Science. Springer, New York

Gandy R (1980) Church's thesis and principles for mechanisms. In: Barwise J, Keisler HJ, Kunen K (eds) The Kleene Symposium, Studies in Logic and the Foundations of Mathematics, vol 101, Elsevier, pp 123–148, DOI 10.1016/S0049-237X(08)71257-6

Heyting A (1931) Die intuitionistische Grundlegung der Mathematik. Erkenntnis 2:106–115

Hilbert D (1900) Mathematische Probleme – Vortrag, gehalten auf dem Internationalen Mathematiker-Kongreß zu Paris 1900. Nachrichten von der Königlichen Gesellschaft der Wissenschaften zu Göttingen Mathematisch-Physikalische Klasse 1900(3):253–297
Hodges A (1983) Alan Turing: The Enigma. Simon & Schuster, New York
Hodges A (2008) What did Alan Turing mean by "machine"? In: Husbands P, Holland O, Wheeler M (eds) The Mechanical Mind in History, MIT Press, Cambridge/London, pp 75–90
McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biology 5(4):115–133, DOI 10.1007/BF02478259
Miłkowski M (2013) Explaining the Computational Mind. MIT Press, Cambridge
Miłkowski M (2018) From computer metaphor to computational modeling: The evolution of computationalism. Minds and Machines 28:515–541, DOI 10.1007/s11023-018-9468-3
von Neumann J (1931) Die formalistische Grundlegung der Mathematik. Erkenntnis 2:116–121
Niebergall KG (2005) Intensionality on philosophy and metamathematics. In: Kahle R (ed) Intensionality, Lecture Notes in Logic, vol 22, Cambridge University Press, Cambridge, pp 123–159
Peregrin J (2018) Intensionality in mathematics. In: Piazza M, Pulcini G (eds) Truth, Existence and Explanation: FilMat 2016 Studies in the Philosophy of Mathematics, Springer, Cham, pp 57–70, DOI 10.1007/978-3-319-93342-9_4
Piccinini G (2015) Physical Computation: A Mechanistic Account. Oxford University Press, Oxford
Piccinini G (2017) Computation in physical systems. In: Zalta EN (ed) The Stanford Encyclopedia of Philosophy, The Metaphysics Research Lab, Stanford, p html, URL `https://plato.stanford.edu/archives/sum2017/entries/computation-physicalsystems`
Piccinini G, Craver C (2011) Integrating psychology and neuroscience: Functional analyses as mechanism sketches. Synthese 183(3):283–311, DOI 10.1007/s11229-011-9898-4
Pitowsky I (1990) The Physical Church Thesis and physical computational complexity. Iyyun: The Jerusalem Philosophical Quarterly 39:81–99
Popper KR (1959) The Logic of Scientific Discovery. Hutchinson, London
Putnam H (1975) Mind, Language and Reality. Philosophical Papers, Vol. 2. Cambridge University Press, Cambridge
Pylyshyn Z (1980) Computation and cognition: Issues in the foundations of cognitive science. The Behavioral and Brain Sciences 3:111–169
Quinon P (2019) Can church's thesis be viewed as a carnapian explication? Synthese DOI 10.1007/s11229-019-02286-7, special Issue "Intensionality in Mathematics"
Rescorla M (2014) A theory of computational implementation. Synthese 191(6):1277–1307, DOI 10.1007/s11229-013-0324-y
Searle JR (1980) Minds, brains, and programs. The Behavioral and Brain Sciences 3:417–457
Shapiro S (ed) (1985) Intensional Mathematics, Studies in Logic and the Foundations of Mathematics, vol 113. Elsevier, Amsterdam
Stacewicz P (2019) Uncomputable numbers and the limits of coding in computer science. Studia Semiotyczne – English Supplement 30:107–127, DOI 10.26333/stsen.xxx.06
Sylvan R (2003) The importance of nonexistent objects and of intensionality in mathematics. Philosophia Mathematica 11(1):20–52, DOI 10.1093/philmat/11.1.20
Turing AM (1936) On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society s2-42:230–265
Turing AM (1939) Systems of logic based on ordinals. Proceedings of the London Mathematical Society 2:161–228
Turing AM (1950) Computing machinery and intelligence. Mind 59:433–460
Various authors (1931) Diskussion zur Grundlegung der Mathematik. Erkenntnis 2:135–151
Whitehead AN, Russell B (1910-1913) Principia Mathematica, vol 1-3. Cambridge University Pres, Cambridge