

Alan Turing and the Mathematical Objection

GUALTIERO PICCININI

Department of History and Philosophy of Science, University of Pittsburgh, 1017 Cathedral of Learning, Pittsburgh, PA 15260, USA; E-mail: gupst1@pitt.edu

Abstract. This paper concerns Alan Turing's ideas about machines, mathematical methods of proof, and intelligence. By the late 1930s, Kurt Gödel and other logicians, including Turing himself, had shown that no finite set of rules could be used to generate all true mathematical statements. Yet according to Turing, there was no upper bound to the number of mathematical truths provable by intelligent human beings, for they could invent new rules and methods of proof. So, the output of a human mathematician, for Turing, was not a computable sequence (i.e., one that could be generated by a Turing machine). Since computers only contained a finite number of instructions (or programs), one might argue, they could not reproduce human intelligence. Turing called this the "mathematical objection" to his view that machines can think. Logico-mathematical reasons, stemming from his own work, helped to convince Turing that it should be possible to reproduce human intelligence, and eventually compete with it, by developing the appropriate kind of digital computer. He felt it should be possible to program a computer so that it could learn or discover new rules, overcoming the limitations imposed by the incompleteness and undecidability results in the same way that human mathematicians presumably do.

Key words: artificial intelligence, Church-Turing thesis, computability, effective procedure, incompleteness, machine, mathematical objection, ordinal logics, Turing, undecidability

The 'skin of an onion' analogy is also helpful. In considering the functions of the mind or the brain we find certain operations which we can express in purely mechanical terms. This we say does not correspond to the real mind: it is a sort of skin which we must strip off if we are to find the real mind. But then in what remains, we find a further skin to be stripped off, and so on. Proceeding in this way, do we ever come to the 'real' mind, or do we eventually come to the skin which has nothing in it? In the latter case, the whole mind is mechanical (Turing, 1950, p. 454–455).

1. Introduction

This paper concerns British mathematician Alan Turing and his ideas on "mechanical intelligence," as he called it. In the late 1940s, Turing argued that digital computers could reproduce human thinking and, to measure their intelligence, he proposed the *Turing test*. Its *locus classicus* is a paper published in *Mind* in 1950, where the term "test" was used. For Turing, the Turing test was not an "operational definition of 'thinking' or 'intelligence' or 'consciousness'" (as sometimes maintained, e.g. by Hodges, 1983, p. 415) — the test only gave a sufficient condition for a machine to be considered intelligent, or thinking (Turing, 1950, p. 435). "Intelligence" and "thinking" were used interchangeably by Turing.



Minds and Machines 13: 23–48, 2003.

© 2003 Kluwer Academic Publishers. Printed in the Netherlands.

A decade earlier, his work in mathematical logic yielded the *Church-Turing thesis* (CT) and the concept of *universal computing machine*, which — as is generally recognized — were important influences on his machine intelligence research program. Turing's contributions, in turn, are fundamental to AI, psychology, and neuroscience. Nevertheless, a detailed, consistent history of Turing's research program, starting from his work in foundations of mathematics, has yet to be written.

Turing's views about machine intelligence are rooted in his thinking about mathematical methods of proof — this is the topic of this paper. The power and limitations of human mathematical faculties concerned him as early as the 1930s. By then, Kurt Gödel and others — including Turing himself — had shown that no finite set of rules, i.e. no uniform method of proof, could be used to generate all mathematical truths. And yet intelligent human beings, Turing maintained, could invent new methods of proof by which an unbounded number of mathematical truths could be proved. Instead, computers contained only finite instructions and, as a consequence, they could not reproduce human intelligence. Or could they? Turing called this the *mathematical objection* to his view that machines could think. In reply, he proposed designing computers that could learn or discover new instructions, overcoming the limitations imposed by Gödel's results in the same way that human mathematicians presumably do.¹

Most of the literature on Turing is written by logicians or philosophers who are often more interested in current philosophical questions than in Turing's ideas. More than historical tools, their research relies on philosophical analysis. The outcome, from a historiographical point of view, is a biased literature: Turing's words are interpreted in light of much later events, like the rise of AI or cognitive science, or he is attributed solutions to problems he didn't address, such as the philosophical mind-body problem. While trying to avoid such pitfalls, I'll occasionally point the reader to the existence of current debates. I hope such debates will benefit from a correct historical reconstruction of some of Turing's ideas.

In addition to the works published by Turing and his contemporaries, I have used unpublished material from the Alan Mathison Turing collection, King's College Library, Cambridge. This material is now available in published form (Copeland, 1999, forthcoming; The Turing Archive for the History of Computing <<http://www.AlanTuring.net>>). Most of the personal information originates from two biographies of Turing written by his mother, Ethel Sara Turing (1959), and by Andrew Hodges (1983). These two biographies provide useful details on Turing's life, but are not reliable when it comes to his intellectual development. Suffice it to say that Sara Turing, by her own admission, lacked the education necessary to understand her son's work, and that Hodges, when interpreting Turing's ideas, advocates the frustrating policy of omitting the evidence for most of his statements (Hodges, 1983, p. 541).

2. Computable Numbers

Turing's first formulation of CT, in his celebrated "On computable numbers, with an application to the *Entscheidungsproblem*," stated that the numbers computable by one of his machines "include all numbers which could naturally be regarded as computable" (Turing, 1936–1937, pp. 116, 135). That is, any calculation could be made by some Turing machine. This section concerns CT, Turing's use of "computable" and "machine" in his logic papers, and why his early work on computability should not be read as an attempt to establish or imply that the mind is a machine. In later sections, these explorations will help to illuminate Turing's remarks about mathematical faculties and, in turn, his reply to the mathematical objection.

Today, both the term "computable" and formulations of CT are utilized in many contexts, including discussions of the nature of mental, neural, or physical processes.² None of these uses existed at Turing's time, and their superposition onto Turing's words yields untenable results. For instance, according to a popular view, Turing's argument for CT was already addressing the problem of how to mechanize the human mind, while the strength of CT — perhaps after some years of experience with computing machines — eventually convinced Turing that thinking could be reproduced by a computer.³

This reading makes Turing appear incoherent. It conflicts with the fact that he, who reiterated CT every time he talked about machine intelligence, never said that the mechanizability of the mind was a consequence of CT. Quite the opposite: in defending his view that machines could think, he felt the need to respond to many objections, including the mathematical objection. Indeed, in his most famous paper on machine intelligence, Turing admitted: "I have no very convincing arguments of a positive nature to support my views. If I had I should not have taken such pains to point out the fallacies in contrary views" (Turing, 1950, p. 454). If one wants to understand the development of Turing's ideas on mechanical intelligence, his logical work on computability must be understood within its context. In the 1930s there were no working digital computers, nor was cognitive science on the horizon. A "computer" was a person reckoning with paper, pencil, eraser, and perhaps a mechanical calculator. Given the need for laborious calculations in industry and government, skilled individuals were hired as "computers." In this context, a "computation" was something done by a human computer.⁴

The origins of "Computable Numbers" can be traced to 1935, when Turing graduated in mathematics from King's College, Cambridge, and became a fellow of King's. In that year, he attended an advanced course on Foundations of Mathematics by topologist Max Newman. Newman, who became Turing's lifelong colleague, collaborator, and good friend, witnessed the development of Turing's work on computability, shared his interest in the foundations of mathematics, and read and commented on Turing's typescript before anyone else (Hodges, 1983, pp. 90–110).

In his biography of Turing as a Fellow of the Royal Society, Newman links “Computable Numbers” to the attempt to prove rigorously that the decision problem for first order logic, formulated by David Hilbert within his program of formalizing mathematical reasoning (Hilbert and Ackermann, 1928), is unsolvable in an absolute sense. “[T]he breaking down of the Hilbert programme,” said Newman (1955, p. 258), was “the application [Turing] had principally in mind.” In order to show that there is no effective procedure — or “decision process” — solving the decision problem, Turing needed:

...to give a definition of ‘decision process’ sufficiently exact to form the basis of a mathematical proof of impossibility. To the question ‘What is a “mechanical” process?’ Turing returned the characteristic answer ‘Something that can be done by a machine,’ and embarked in the highly congenial task of analyzing the general notion of a computing machine (ibid.).

Turing was trying to give a precise and adequate definition of the intuitive notion of effective procedure, as mathematicians understood it, in order to show that no effective procedure could decide first order logical provability. When he talked about computations, Turing meant sequences of operations on symbols (mathematical or logical), performed either by humans or by mechanical devices according to a finite number of rules — which required no intuition or invention or guesswork — and whose execution always produced the correct solution.⁵ For Turing, the term “computation” by no means referred to all that mathematicians, human minds, or machines could do.

However, the potential for anachronism exists even within the boundaries of foundations of mathematics. Members of the Hilbert school, until the 1930s, believed that *finitist* methods of proof, adopted in their proof theory, were identical to intuitionistically acceptable methods of proof.⁶ This assumption was questioned by Paul Bernays in the mid 30s, suggesting that “intuitionism, by its abstract arguments, goes essentially beyond elementary combinatorial methods” (Bernays, 1935a, p. 286; 1967, p. 502). Elaborating on Bernays, Gödel argued that “in the proofs of propositions about these mental objects insights are needed which are not derived from a reflection upon the combinatorial (space-time) properties of the symbols representing them, but rather from a reflection upon the meanings involved” (Gödel, 1958, p. 273).

Exploiting this line of argument in a comment published in 1965, Gödel speculated about the possibility of effective but *non-mechanical* procedures to be distinguished from the effective *mechanical* procedures analyzed by Turing (Gödel, 1965, pp. 72–73). A non-mechanical effective procedure, in addition to mechanical manipulations, allowed for the symbols’ *meaning* to determine its outcome. Other logicians, interested in intuitionism, exploited similar considerations to raise doubts on CT.⁷ But many logicians preferred to reject both Gödel’s distinction and his view that any mathematical procedure could be regarded as non-mechanical yet effective at the same time. The issue cannot be pursued here.⁸

What concerns us is that Gödel's 1965 distinction, between mechanically and non-mechanically effective procedures, has been used in interpreting Turing's 1936–1937 words to suggest that his analysis applied *only* to the mechanical variety (Tamburrini, 1988, pp. 55–56, 94, 127, 146–154; Sieg, 1994, pp. 72, 96). The reason given is that Turing used “mechanical” as a synonym for “effectively calculable”: “a function is said to be ‘effectively calculable’ if its values can be found by a purely mechanical process” (Turing, 1939, p. 160). This remark is by no means exceptional in Turing's parlance. It makes clear, among other things, that the meaning of symbols couldn't affect calculations. So to say, instead, that Turing's definition was *restricted* to mechanical — as opposed to non-mechanical — effective procedures risks involving Turing in a debate that started after his time. In the 1930s and 1940s, neither Turing nor other proponents of formal definitions of “effectively calculable” drew Gödel's distinction.⁹ All we can say, from Turing's explications and terminological choices, is that for him, meanings were no part of the execution of effective procedures.¹⁰

He rigorously defined “effectively calculable” with his famous machines: a procedure was effective if and only if a Turing machine could carry it out. “Machine” requires a gloss. In the 1930s and 1940s, Turing's professionally closest colleagues read his paper as providing a general theory of computability, establishing what could and could not be computed — not only by humans, but also by mechanical devices.¹¹ Later, a number of authors took a more restrictive stance. Given the task of “Computable Numbers,” viz. establishing a limitation to what could be achieved in mathematics by effective methods of proof, it is clear that Turing machines represented (at the least) computational abilities of human beings. As a matter of fact, the steps these machines carried out were determined by a list of instructions, which must be understandable unambiguously by human beings.

But Turing's machines were not portrayed as understanding instructions — let alone intelligent. Even if they were anthropomorphically described as “scanning” the tape, “seeing symbols,” having “memory” or “mental states,” etc., Turing introduced all these terms in quotation marks, presumably to underline their metaphorical use (Turing, 1936–1937, pp. 117–118). If one thinks that carrying out a genuine, “meaningful” computation — as opposed to a “meaningless” physical process — presupposes understanding the instructions, one should conclude that only humans carry out genuine computations. Turing machines, in so far as they computed, were abstract and idealized representations of human beings. These considerations, among others, led some authors to a restrictive interpretation: Turing's theory bears on computability by humans not by machines, and Turing machines are “*humans* who calculate.”¹²

This interpretation is at odds with Turing's use of “computation” and “machine,” and with his depiction of his work. With all his insistence that his machines could mimic any human routine,¹³ he never said his machines should be regarded as abstract human beings — nor anything similar. We saw that, for him, a computation was a type of physical manipulation of symbols. His machines were introduced

to define rigorously this process of manipulation for mathematical purposes. As Turing used the term, machines were idealized mechanical devices; they could be studied mathematically because their behavior was precisely defined in terms of discrete, effective steps.

There is evidence that Turing, in 1935, talked about building a physical realization of his universal machine.¹⁴ Twelve years later, to an audience of mathematicians, he cited “Computable Numbers” as containing a universal digital computer’s design and the theory establishing the limitations of the new computing machines:

Some years ago I was researching on what might now be described as an investigation of the theoretical possibilities and limitations of digital computing machines. I considered a type of machine which had a central mechanism, and an infinite memory which was contained on an infinite tape. This type of machine appeared to be sufficiently general. One of my conclusions was that the idea of a ‘rule of thumb’ process and a ‘machine process’ were synonymous... Machines such as the ACE [Automatic Computing Engine] may be regarded as practical versions of this same type of machine (Turing, 1947, pp. 106–107).¹⁵

Therefore, a machine, when Turing talked about logic, was not (only) a mathematical idealization of a human being, but literally a hypothetical mechanical device, which had a potentially infinite tape and never broke down. Furthermore, he thought his machines delimited the *computing* power of any machine. This is not to say that, for Turing, every physical system was a computing machine or could be mimicked by computing machines. The outcome of a random process, for instance, could not be replicated by any Turing machine, but only by a machine containing a “random element” (1948, p. 9; 1950, p. 438).

Such was the scope of CT, the thesis that the numbers computable by a Turing machine “include all numbers which could naturally be regarded as computable” (Turing, 1936–1937, p. 116).¹⁶ To establish CT, Turing compared “a man in the process of computing ... to a machine” (*ibid.*, p. 117). He based his argument on cognitive limitations affecting human beings doing calculations. At the beginning of “Computable Numbers,” one reads that “the justification [for CT] lies in the fact that the human memory is necessarily limited” (*ibid.*, p. 117). In the argument, Turing used sensory limitations to justify his restriction to a finite number of primitive symbols, as well as memory limitations to justify his restriction to a finite number of “states of mind” (Turing, 1936–1937, pp. 135–136).

This argument for CT does not entail — nor did Turing ever claim that it did — that all operations of the human mind are computable by a Turing machine. His contention was, more modestly, that the operations of a Turing machine “include all those which are used in the computation of a number” by a human being (*ibid.*, p. 118). Since the notion of the human process of computing, like the notion of effectively calculable, is an intuitive one, Turing asserted that “all arguments which can be given [for CT] are bound to be, fundamentally, appeals to intuition, and for

this reason rather unsatisfactory mathematically” (ibid., p. 135). In other words, CT was not a mathematical theorem.¹⁷

From “Computable Numbers,” Turing extracted the moral that effective procedures, “rule of thumb processes,” or instructions explained “quite unambiguously in English,” could be carried out by his machines. This applied not only to procedures operating on mathematical symbols, but to any symbolic procedure so long as it was effective. It even applied to procedures that did not always generate correct answers to general questions, as long as these procedures were exhaustively defined by a finite set of instructions.¹⁸ A universal machine, if provided with the appropriate instruction tables, could carry out all such processes. This was a powerful thesis, but very different from the thesis that “thinking is an effective procedure.”¹⁹ In “Computable Numbers” Turing did not argue, nor did he have reasons to imply from CT, that human thinking could be mechanized.

He did prove, however, that no Turing machine — and by CT no uniform, effective method — could solve first order logic’s decision problem (Turing, 1936–1937, pp. 145–149). He added that, as far as he knew, a *non-uniform* process could generate a non-computable sequence — that is, a sequence no Turing machine could generate. Assume δ is a non-computable sequence:

It is (so far as we know at present) possible that any assigned number of figures of δ can be calculated, but not by a uniform process. When sufficiently many figures of δ have been calculated, an essentially new method is necessary in order to obtain more figures (ibid., p. 139).

Non-uniform processes — generating non-computable sequences — appeared again, in different guises, in Turing’s later work about foundations of mathematics and machine intelligence. These processes played a role in Turing’s next important logical work, where he commented on those mathematical faculties whose outputs could *not* be generated by Turing machines.

3. Ordinal Logics

Famously, Gödel (1931) proved his *incompleteness* theorems to the effect that, within formal systems like that of Alfred N. Whitehead and Bertrand Russell’s *Principia Mathematica* (1910–1913), not all arithmetical truths could be proved. A few years later, Alonzo Church (1936) and Turing (1936–1937) argued for CT, uniquely defining the notion of effective procedure, or uniform method of proof, independently of any particular formal system. Using their definitions, they showed that no effective procedure could prove all arithmetical truths: Gödel incompleteness applied to any (sufficiently powerful) formal system.²⁰ For most mathematicians, this ruled out the possibility of expressing all mathematics within one formal system. But many maintained that, in principle, human beings could still decide — perhaps by inventing new methods of proof — all mathematical statements.²¹

At the end of 1936, around the time he completed “Computable Numbers,” Turing went to Princeton, where he stayed until 1938. There, among other things, he worked on a Ph.D. dissertation under Church, which he later published as “Systems of Logic Based on Ordinals” (1939). In his doctoral work, Turing explored the possibility of achieving arithmetical completeness not by a logical system more powerful than that of *Principia Mathematica*, which he knew to be impossible, but by an infinite nonconstructive sequence of logical systems.

For each logical system L in the sequence, by Gödel incompleteness there was a true statement S_L unprovable by means of L . So if one started with a system L_1 , there would be an associated unprovable statement S_{L_1} . By adjoining S_{L_1} to L_1 , one could form a new system L_2 , which would be more complete than L_1 in the sense that more arithmetical theorems would be provable in L_2 than in L_1 . But by Gödel incompleteness, there would still be a true statement S_{L_2} unprovable within L_2 . So the process must be repeated with L_2 , generating a system L_3 more complete than L_2 , and so on. Turing showed that if this process was repeated infinitely many times, the resulting sequence of logical systems was complete, i.e. any true arithmetic statement could be derived within one or another member of the sequence. The sequence was nonconstructive in the sense that there was no uniform method (or Turing machine) that could be used to generate the whole sequence.²²

In a formal system, what counted as an axiom or a proof must be decidable by an effective process.²³ Since Turing’s method for generating ordinal logics violated this principle, he owed an explanation. In a section entitled “The purpose of ordinal logics,” he explained that he was proposing an alternative to the failed program of formalizing mathematical reasoning within one formal system (1939, pp. 208–210). Some authors have misinterpreted his interesting remarks about human mathematical faculties as implying that the human mind could not be reproduced by machines.²⁴ In this section, I’ll give a more accurate reading of Turing’s view.

In “The purpose of ordinal logics,” Turing asserted that he saw mathematical reasoning as the effect of two faculties, “intuition” and “ingenuity”²⁵:

The activity of the intuition consists in making spontaneous judgments which are not the result of conscious trains of reasoning. These judgments are often but by no means invariably correct (leaving aside the question what is meant by “correct”). Often it is possible to find some other way of verifying the correctness of an intuitive judgment. We may, for instance, judge that all positive integers are uniquely factorizable into primes; a detailed mathematical argument leads to the same result. This argument will also involve intuitive judgments, but they will be less open to criticism than the original judgment about factorization...

The exercise of ingenuity in mathematics consists in aiding the intuition through suitable arrangements of propositions, and perhaps geometrical figures or drawings. It is intended that when these are really well arranged the validity of the intuitive steps cannot seriously be doubted (Turing, 1939, pp. 208–209).

Turing did not see ingenuity and intuition as two independent faculties, perhaps working according to different principles (e.g., the first mechanical, the second non-mechanical). He said the use of one rather than the other varied case by case:

The parts played by these two faculties differ of course from occasion to occasion, and from mathematician to mathematician.²⁶ This arbitrariness can be removed by the introduction of a formal logic. The necessity for using the intuition is then greatly reduced by setting down formal rules for carrying out inferences which are always intuitively valid. When working with a formal logic, the idea of ingenuity takes a more definite shape. In general a formal logic, [sic] will be framed so as to admit a considerable variety of possible steps in any stage in a proof. Ingenuity will then determine which steps are the most profitable for the purpose of proving a particular proposition (ibid., p. 209).

The arbitrariness of the use of intuition and ingenuity, in different cases and by different mathematicians, motivated the introduction of formal logic — where all types of legitimate inferences were specified in advance, and proofs consisted of a finite number of those inferences.

Then, Turing discussed the relevance of Gödel incompleteness, implying that mathematics could never be fitted entirely within one formal system:

In pre-Gödel times it was thought by some that it would probably be possible to carry this program [of formalizing mathematical reasoning] to such a point that all the intuitive judgments of mathematics could be replaced by a finite number of these rules. The necessity for intuition would then be entirely eliminated (ibid., p. 209).

Turing was saying that, before Gödel's proof, some mathematicians tried to replace all intuitive mathematical judgments with a finite number of formal rules and axioms — which must be intuitively valid — eliminating the necessity of intuition in proofs. This having proved impossible, Turing proposed to do the opposite:

In our discussion, however, we have gone to the opposite extreme and eliminated not intuition but ingenuity, and this in spite of the fact that our aim has been in much the same direction. We have been trying to see how far it is possible to eliminate intuition, and leave only ingenuity (ibid., p. 209).

This passage seems self-contradictory, eliminating ingenuity at first, but claiming that, at the end of the day, only ingenuity will be left. Given Gödel incompleteness, Turing was explaining that he focused his research on what must be added to incomplete formal systems to make them *less* incomplete. At the same time, he “eliminated” ingenuity from his analysis in the following sense: he assumed that every time one needed to prove — possibly by a new method of proof — a new theorem discovered by intuition, one could find a proof. Given this assumption that all the needed proofs could be found, Turing concluded that, at the end of the hypothetical construction of an ordinal logic, all intuitive inferences would be replaced by proofs. Since proofs are the output of ingenuity, only ingenuity will

be left. So, Turing did concentrate on intuitive inferential steps, but only under the assumption that each step could eventually be replaced by a proof.

For this project, Turing's mathematical tools were special sequences of formal systems. In these sequences, no finite set of rules and axioms sufficed for all future derivations — new ones could be needed at any time:

In consequence of the impossibility of finding a formal logic which wholly eliminates the necessity of using intuition, we naturally turn to “non-constructive” systems of logic with which not all the steps in a proof are mechanical, some being intuitive (*ibid.*, p. 210).

Turing's contrast between intuitive and mechanical steps has been taken as evidence that, at least for a short period of his life, he endorsed an anti-mechanist view, maintaining that the mind couldn't be a machine (Hodges, 1988, p. 10; 1997, p. 22; Lucas, 1996, p. 111). But talking of anti-mechanism in this context is seriously misleading. It generates the pseudo-problem of why Turing never endorsed an anti-mechanist view, proposing instead a research program in machine intelligence. Turing defended his machine intelligence program by replying to the mathematical objection in a subtle way that will be discussed in the next section. We'll see that an anti-mechanist reading of Turing's remarks on ordinal logics makes his reply to the mathematical objection hardly intelligible.²⁷ We noted that Turing's explicitly advocated goal was the same as that of pre-Gödel proof-theorists, namely a metamathematical analysis of mathematical reasoning in which the use of intuition would be eliminated. Turing contrasted intuitive to mechanical (or formal) inferential steps *not* to invoke some “non-mechanical” power of the mind, but to distinguish between what could be justified within a given formal system — the mechanical application of rules — and what at some times needed to be added from the outside — like a new axiom, inferential rule, or method of proof.

The assumption of an unlimited supply of ingenuity, however, needed some justification. Turing provided it earlier in the paper. He wrote that the proposed non-constructive sequence of logics could still be accepted as “intellectually satisfying” if a certain condition was fulfilled:

We might hope to obtain some intellectually satisfying system of logical inference (for the proof of number-theoretic theorems) with some ordinal logic. Gödel's theorem shows that such a system cannot be wholly mechanical; but with a complete ordinal logic we should be able to confine the non-mechanical steps entirely to verifications that particular formulae are ordinal formulae (*ibid.*, p. 194).

Turing was pointing out that, in the non-constructive sequence of systems he was proposing, non-mechanical steps corresponded to verifications that particular formulae had a certain mathematical property — they were “ordinal formulae.” For some ordinal logics, the statement that a given formula was an ordinal formula was a number-theoretic statement (*ibid.*, pp. 210, 219). And, notwithstanding Gödel incompleteness, leading mathematicians working on foundations of mathematics, who shared a strong faith in the power of human reason, expected that all true

number-theoretic statements could be proved by one method or another.²⁸ This is why Turing assumed that each “intuitive” step could be, in principle, justified by a proof. The steps were called “intuitive” as opposed to “mechanical” because it was impossible to establish, once and for all, what method of proof must be used in any given case. In his metamathematical investigation, Turing sought to replace intuition with the equivalent to a non-uniform process for proving that appropriate formulae were ordinal formulae.

Nothing that Turing wrote in 1939 implied that the human mind was not a machine. Though Turing did not emphasize this point, nothing prevented each method of proof required by his ordinal logics from being implemented in a universal machine.

Turing’s crucial move was abandoning the doctrine that decidability must be achieved by a uniform method of proof (or a single effective procedure, or a single Turing machine), accepting that many methods (or machines) could be used. In “Computable Numbers,” Turing had already said that a non-uniform process, consisting of many methods, could generate a non-computable sequence. In a letter to Newman, in 1940, he explained his move: to an objection by Newman, he replied that it was too “radically Hilbertian” to ask that “there is ... some fixed machine on which proofs are to be checked.” If one took this “extreme Hilbertian” line, Turing admitted, “my ordinal logics would make no sense.” On the other hand:

If you think of various machines I don’t see your difficulty. One imagines different machines allowing different sorts of proofs, and by choosing a suitable machine one can approximate “truth” by “provability” better than with a less suitable one, and can in a sense approximate it as well as you please (Turing, 1940a).

That year, in another letter to Newman, Turing further explained the motivation for his ordinal logics. The “unsolvability or incompleteness results about systems of logic,” he said, amounted to the fact that “[o]ne cannot expect that a system will cover all possible methods of proof,” a statement he labeled “ β .”²⁹ The point was, again: “When one takes β) into account one has to admit that not one but many methods of checking up are needed. In writing about ordinal logics I had this kind of idea in mind.” Moreover, “[t]he proof of my completeness theorem ... is of course completely useless for the purpose of actually producing proofs ... The completeness theorem was written from a rather different point of view from most of the rest, and therefore tends to lead to confusion” (Turing, 1940b).

In explaining his motivation for studying ordinal logics, Turing was far from stating that the mind is not a machine. Quite the contrary: he wanted to show that by using many formal systems, whose proofs could be checked by as many machines, one could form stronger and stronger logical systems that would allow one to prove more and more arithmetical theorems, and the whole sequence of such logical systems would be complete, namely any arithmetical theorem would be provable by one or another member of the sequence.

Turing's 1939 paper and letters to Newman, read in their context, do not indicate concern with whether human ingenuity or intuition could be exhibited by machines. What they do show — not surprisingly — is that Turing clearly understood the consequences of Gödel incompleteness for the project of analyzing human mathematical reasoning with formal, mechanical systems. To generate a system with the completeness property, he proposed a strategy of adding new axioms to formal systems, where some additions could be seen as the invention of new methods of proof. This strategy sheds some light on Turing's "mathematical objection," his reply, and his related insistence on both inventing new methods of proof and machine learning.

4. Intelligent Machinery

Starting with his report on the ACE, and later with other reports, talks, and papers, Turing developed and promoted a research program whose main purpose was the construction of intelligent machines.³⁰ After a brief recapitulation of the main tenets of his view, I'll turn to the mathematical objection and how Turing's discussion of machine intelligence related to his work in logic.

Intelligence, for Turing, had to do with what people did, not with some essence hidden in their soul. Intelligent behavior was the effect of the brain, which Turing — usually — did not distinguish from the mind.³¹ Since the brain dealt with information,³² reproducing intelligence did not require building artificial neurons — perhaps surrounded by an artificial body — which would be impractical and expensive. What mattered was the logical structure of the machine (Turing, 1948, p. 13). This was one more reason to use universal digital computers, with their clear logical structure, as "artificial brains" (ibid.). How could one know when one had built an intelligent machine? Degrees of intelligence, for Turing, were no matter of scientific measurement. He once explained how humans attribute intelligence in a section titled "intelligence as an emotional concept": "the extent to which we regard something as behaving in an intelligent manner is determined as much by our own state of mind and training as by the properties of the object under consideration" (1948, p. 23). A similar rhetoric is deployed in the *Mind* paper to dispense with the question: "Can machines think?" It's a vague question, he said, "too meaningless to deserve discussion" (1950, p. 442); it depends on the terms "machine" and "think," whose meanings are subject to change (ibid., p. 433). Given this, according to Turing, one should attribute intelligence to a machine any time it displayed some interesting, unpredictable, human-like (symbolic) behavior (1947, p. 123; 1948, p. 23; 1950, p. 459).

As we saw, Turing was fond of saying that universal digital computers, like the ACE, were "practical versions of the universal machine" (1947, pp. 107, 112–113). Being universal, they could solve those problems "which can be solved by human clerical labour, working on fixed rules, and without understanding" (1945, pp. 38–39). Turing repeated similar generic formulations of CT every time he talked about

digital computers.³³ Nonetheless, CT would directly imply that universal machines could reproduce human thinking *only if* all human thinking could be put in the form of fixed rules. In “Intelligent Machinery,” Turing denied that the latter was the case:

If the untrained infant’s mind is to become an intelligent one, it must acquire both discipline and initiative. So far we have been considering only discipline. To convert a brain or machine into a universal machine is the extremest form of discipline... But discipline is certainly not enough in itself to produce intelligence. That which is required in addition we call initiative. This statement will have to serve as a definition. Our task is to discover the nature of this residue as it occurs in man, and try to copy it in machines (1948, p. 21).

For the remainder of the paper, Turing concentrated primarily on how a machine, by means other than finite instructions, could reproduce “initiative.”

In “Computable Numbers,” Turing proved that *no* Turing machine could prove all true mathematical statements, thereby, in principle, replacing all methods of proof. For any Turing machine, there existed mathematical questions the machine would not answer correctly. On the other hand, as we’ll see, Turing stressed that mathematicians invented new methods of proof — they could, in principle, answer all mathematical questions. So, if a machine were to have genuine intelligence, it would need to have more than discipline — it would need to be more than a Turing machine. In his work on ordinal logics, Turing showed that this limitation of Turing machines could be overcome by an infinite sequence of formal systems whose theorems could be generated (or checked) by a sequence of machines. Each machine, in turn, could be simulated by a universal machine. But no machine itself could infallibly choose all the machines in the sequence. Roughly, Turing thought that an intelligent machine, instead of trying to answer all possible questions correctly (which in some cases would lead to infinite loops), should sometimes stop computing, give the wrong answer, and try new instructions. If the new instructions answered the question correctly, the machine would have changed itself in a way that makes it less incomplete. In other words, the machine must “learn.” After all, human beings make many mistakes, but they learn to correct them. (Sometimes.) According to Turing, if a machine were able to learn (i.e., to change its instruction tables), Gödel incompleteness would be no objection to its intelligence. This explains Turing’s reply to the mathematical objection.³⁴ It is also relevant to Turing’s insistence on “child machines,” on various methods of “educating” or “teaching” machines, on “searches,” and on a random element being placed in machines (1948, pp. 14–23; 1950, pp. 454–460). These latter ideas are frequently mentioned, but their connection to Turing’s early work in logic has not been recognized.

In his reply, Turing made the *prima facie* puzzling claim that, “[i]f a machine is expected to be infallible, it cannot also be intelligent” (1947, p. 124). This claim is often cited but never clearly explained. Penrose suggests that, for Turing, if the algorithm followed by a machine were “unsound,” then the machine would overcome the limitations of Gödel incompleteness (Penrose, 1994, p. 129). A similar explication by Gandy is that, even though the formal system describing

a machine is consistent, the output of the machine could be inconsistent (Gandy, 1996, p. 134).³⁵ This reading has little relevance to Turing's words.

To reconstruct the mathematical objection, I will follow the chronological order of Turing's writings. The reasoning sketched above, about machine learning, was foreshadowed by an obscure remark in the report on the ACE. In listing some of the problems the ACE could solve, Turing mentioned the following chess problem: "Given a position in chess the machine could be made to list all the 'winning combinations' to a depth of about three moves on either side" (1945, p. 41). Then, Turing asked whether the machine could play chess, answering that it would play a "rather bad game":

It would be bad because chess requires intelligence. We stated at the beginning of this section that [in writing instruction tables] the machine should be treated as entirely without intelligence. There are indications however that it is possible to make the machine display intelligence at the risk of its making occasional serious mistakes. By following up this aspect the machine could probably be made to play very good chess (ibid., p. 41).

Turing would be more explicit about such "indications" at the end of his 1947 "Lecture," where, in connection with mechanical intelligence, he first talked in public about the importance of the machine's ability to learn.

It has been said that computing machines can only carry out the processes that they are instructed to do. This is certainly true in the sense that if they do something other than what they were instructed then they have just made some mistake. It is also true that the intention in constructing these machines in the first instance is to treat them as slaves, giving them only jobs which have been thought out in detail, jobs such that the user of the machine fully understands what in principle is going on all the time. Up till the present machines have only been used in this way. But is it necessary that they should always be used in such a manner? *Let us suppose we have set up a machine with certain initial instruction tables, so constructed that these tables might on occasion, if good reason arose, modify those tables.* One can imagine that after the machine has been operating for some time, the instructions would have altered out of all recognition, but nevertheless still be such that one would have to admit that the machine was still doing very worthwhile calculations. Possibly it might still be getting results of the type desired when the machine was first set up, but in a much more efficient manner. In such a case one would have to admit that the progress of the machine had not been foreseen when its original instructions were put in. It would be *like a pupil who had learnt much from his master, but had added much more by his own work.* When this happens I feel that *one is obliged to regard the machine as showing intelligence ... What we want is a machine that can learn from experience. The possibility of letting the machine alter its own instructions provides the mechanism for this,* but this of course does not get us very far (1947, pp. 122–123, emphasis added).

The analogy between learning from experience and altering one's instruction tables led directly to answering the mathematical objection, where human mathematicians are attributed the potential for solving all mathematical problems, and where training is compared to putting instruction tables in the machine.

It might be argued that there is a fundamental contradiction in the idea of a machine with intelligence. It is certainly true that "acting like a machine," has become synonymous with lack of adaptability. But the reason for this is obvious. Machines in the past have had very little storage, and there has been no question of the machine having any discretion. The argument might however be put into a more aggressive form. It has for instance been shown that with certain logical systems there can be no machine which will distinguish provable formulae of the system from unprovable, i.e. that there is no test that the machine can apply which will divide propositions with certainty into these two classes. Thus if a machine is made for this purpose it must in some cases fail to give an answer. On the other hand if a mathematician is confronted with such a problem he would search around and find new methods of proof, so that he ought eventually to be able to reach a decision about any given formula. This would be the argument. Against it I would say that fair play must be given to the machine. Instead of it sometimes giving no answer we could arrange that it gives occasional wrong answers. But the human mathematician would likewise make blunders when trying out new techniques. It is easy for us to regard these blunders as not counting and give him another chance, but the machine would probably be allowed no mercy. In other words then, if a machine is expected to be infallible, it cannot also be intelligent. There are several mathematical theorems which say almost exactly that. But these theorems say nothing about how much intelligence may be displayed if a machine makes no pretence at infallibility. To continue my plea for "fair play to the machines" when testing their I.Q. *A human mathematician has always undergone an extensive training. This training may be regarded as not unlike putting instruction tables into a machine.* One must therefore not expect a machine to do a very great deal of building up of instruction tables on its own. No man adds very much to the body of knowledge, why should we expect more of a machine? Putting the same point differently, the machine must be allowed to have contact with human beings in order that it may adapt itself to their standards. The game of chess may perhaps be rather suitable for this purpose, as the moves of the machine's opponent will automatically provide this contact (*ibid.*, p. 123–124, emphasis added).

The puzzling requirement that an intelligent machine not be infallible, then, has to do with the unsolvability result proved by Turing in "Computable Numbers." If a problem was unsolvable in that absolute sense, then no machine designed to answer correctly all the questions (constituting the problem) could answer them all. Sometimes it would keep computing forever — without ever printing out a result. If, instead, the machine were allowed to give the "wrong answer," viz. an output

that is not the correct answer to the original question, then there was no limit to what the machine could “learn” by changing its instruction tables. In principle, like a human mathematician with an ordinal logic, it could reach mathematical completeness.

The mathematical theorems “which say almost exactly that, if a machine is expected to be intelligent, it cannot also be infallible,” were: Gödel incompleteness theorems, Turing’s theorem about the decision problem, and the analogous result by Church.³⁶ In “Intelligent Machinery,” an objection to the possibility of machine intelligence was that these “results”:

...have shown that if one tries to use machines for such purposes as determining the truth or falsity of mathematical theorems and one is not willing to tolerate an occasional wrong result, then any given machine will in some cases be unable to give an answer at all. On the other hand the human intelligence seems to be able to find methods of ever-increasing power for dealing with such problems “transcending” the methods available to machines (1948, p. 4).

Turing’s reply to this “argument from Gödel’s and other theorems” was a condensed version of the 1947 response: the argument “rests essentially on the condition that the machine must not make mistakes. But this is not a requirement for intelligence” (ibid.).³⁷ The key was, once again, that the machine must be able to learn from trial and error, an issue to which much of the paper was devoted (see esp. pp. 11–12, 14–17, 21–23).

In the *Mind* paper, the “mathematical objection” was raised again — the short reply was based on a parallel between machine fallibility and human fallibility. Two elements were novel. First, formulating the objection in terms of Turing’s unsolvability result, rather than other similar mathematical results, was “most convenient to consider, since it refers directly to machines, whereas the others can only be used in a comparatively indirect argument.” Second, Turing pointed out that “questions which cannot be answered by one machine may be satisfactorily answered by another” (1950, pp. 444–445). The emphasis on learning was left to later sections of the paper, but the main point remained: “there might be men cleverer than any given machine, but then again there might be other machines cleverer again, and so on” (ibid., p. 445). Another version of the reply to the mathematical objection, analogous to the above ones, can be found in the typescript of a talk (1951a). Once again, if a machine were designed to make mistakes and learn from them, it might become intelligent. And this time, Turing subjected mathematicians to the same constraint: “this danger of the mathematician making mistakes is an unavoidable corollary of his power of sometimes hitting upon an entirely new method” (ibid., p. 2).

5. Conclusion

Making sense of Turing's reply to the mathematical objection — the objection that incompleteness results implied that machines could not be intelligent — requires an appropriate historical understanding of Turing's logical work. If he thought CT implied that the mind was mechanizable, then it's unclear why he replied to the mathematical objection at all, not to mention why he related machine learning to the reply. On the other hand, interpreting Turing's 1939 words as claiming that human intuition was a non-mechanical faculty, due to Gödel incompleteness, would fit with a belief in the possibility of non-mechanical effective procedures. But this sort of reading conflicts with both the fact that Turing never mentioned the possibility of non-mechanical effective procedures and the fact that he did not answer the objection — different from and stronger than his own mathematical objection — that a non-mechanical effective procedure would *not* be mechanizable. On the contrary, he repeated many times that all effective procedures could be carried out by a machine.

Turing's ideas on mechanical intelligence were deeply related to his work in the foundations of mathematics. He formulated CT and inferred from it that any "rule of thumb," or "method of proof," or "stereotyped technique," or "routine," could be reproduced by a universal machine. This, for Turing, was insufficient ground for attributing intelligence to machines. He knew all too well the implications of his own unsolvability result, whose moral he took to be analogous to that of Gödel incompleteness. He was fond of saying that no Turing machine could correctly answer all mathematical questions, while human mathematicians might hope to do so. But he also thought that nothing prevented machines from changing, like the minds of mathematicians when we say that they learn. As long as this process of change was not governed by a uniform method, it could overcome incompleteness. Just as the power of a formal system could be augmented by adding new axioms, a universal digital computer could acquire new instruction tables, thereby "learning" to solve new problems. The process, like the process of generating more powerful formal systems, could in principle continue indefinitely. This was what ordinal logics were about, and how Turing's reply to the "mathematical objection" worked. If the machine could modify its instruction tables by itself, without following a uniform method, Turing saw no reason why it could not reach, and perhaps surpass, the intelligence of human mathematicians.

Acknowledgements

Parts of this paper were presented at *tcog*, University of Pittsburgh, fall 1999; JASHOPS, George Washington University, fall 1999; Hypercomputation, University College, London, spring 2000. I'm grateful to the audiences at those events for their helpful feedback. For different sorts of help while researching and writing this paper, I wish to thank Jacqueline Cox, Jerry Heverly, Lance Lugar, Peter

Machamer, Ken Manders, Rosalind Moad, Bob Olby, Elizabeth Paris, Merrilee Salmon, and especially Jack Copeland, Wilfried Sieg, and Becka Skloot.

Notes

¹The term “mathematical objection” was introduced by Turing (1950, p. 444); we’ll see how both the objection and Turing’s reply can be found in previous works by Turing, which remained unpublished at the time. Despite Turing being the first to publicly discuss the mathematical objection, his reply is rarely mentioned and poorly understood.

²For a survey of different uses see Odifreddi (1989, § I.8). (Odifreddi writes “recursive” instead of “computable.”)

³See e.g. Hodges (1983, esp. p. 108), also Hodges (1988, 1997), Leiber (1991, pp. 57, 100), Shanker (1995, pp. 64, 73) and Webb (1980, p. 220). Turing himself is alleged to have argued, in his 1947 “Lecture to the London Mathematical Society,” that “the Mechanist Thesis... is in fact *entailed* by his 1936 development of CT” (Shanker, 1987, pp. 615, 625). Since Shanker neither says what the Mechanist Thesis is, nor provides textual evidence from Turing’s lecture, it is difficult to evaluate his claim. If the Mechanist Thesis holds that the mind is a machine or can be reproduced by a machine, we’ll see that Shanker is mistaken. However, some authors — other than Turing — do believe CT to entail that the human mind is mechanizable (e.g., Dennett, 1978, p. 83; Webb 1980, p. 9).

⁴There did exist some quite sophisticated computing machines, later called analog computers. At least as early as 1937, Turing knew about the Manchester differential analyzer, an analog computer devoted to the prediction of tides, and planned to use a version of it to find values of the Riemann zeta function (Hodges, 1983, pp. 141, 155–158).

⁵See his argument for the adequacy of his definition of computation in Turing (1936–1937, pp. 135–138). The last qualification — about the computation being guaranteed to generate the correct solution — was dropped after “Computable Numbers.” In different writings, ranging from technical papers to popular expositions, Turing used many different terms to explicate the intuitive concept of effective procedure: “computable” as “calculable by finite means” (1936–1937), “effectively calculable” (1936–1937, pp. 117, 148; 1937, p. 153), “effectively calculable” as a function whose “values can be found by a purely mechanical process” (1939, p. 160), “problems which can be solved by human clerical labour, working to fixed rules, and without understanding” (1945, pp. 38–39), “machine processes and rule of thumb processes are synonymous” (1947, p. 112), “‘rule of thumb’ or ‘purely mechanical’” (1948, p. 7), “definite rule of thumb process which could have been done by a human operator working in a disciplined but unintelligent manner” (1951b, p. 1), “calculation” to be done according to instructions explained “quite unambiguously in English, with the aid of mathematical symbols if required” (1953, p. 289).

⁶E.g. Bernays (1935b, p. 89), Herbrand (1971, p. 274) and von Neumann (1931, pp. 50–51). Cf. Bernays (1967, p. 502) and van Heijenoort (1967, p. 618.)

⁷E.g., Kalmár (1959) and Kreisel (1972, 1987). For a treatment of this line of argument, starting with Bernays, see Tamburrini (1988, chapt. IV.)

⁸For defenses of CT in line with the tradition of Church and Turing, see Mendelson (1963) and Kleene (1987b). Most logicians still use “effective” and “mechanical” interchangeably; e.g. Boolos and Jeffrey (1974, p. 19) and Copeland (1996)

⁹E.g., see Church (1936, p. 90) and Post (1936, p. 291.)

¹⁰In his writings, Turing never mentioned anything resembling Gödel’s notion of non-mechanically effective procedure. From the point of view of Gödel’s distinction, as we saw in n. 5, some of the terms used by Turing for “effectively calculable” would be more restrictive than others (compare “purely mechanical process” to “rule of thumb process,” or “explainable quite unambiguously in English”). There is no reason to think that Turing’s explication of “effectively calculable” in terms of “purely mechanical” was intended to draw a distinction with respect to non-mechanical but effect-

ive procedures. Rather, it shows once again that Turing had no notion of non-mechanical effective procedure. In fact, Turing wrote that he used the term “effectively calculable” for “the intuitive idea without particular identification with any [formal] definition” (1939, p. 160; also 1937, p. 153). So he was explicating precisely the supposed intuitive meaning of the term “effectively calculable” in terms of mechanical processes. In the same paragraph, the term “mechanical process” was first used intuitively, and then identified with computability by Turing machines (*ibid.*, cf. also: “One of my conclusions [in 1936–1937] was that the idea of a ‘rule of thumb’ process and a ‘machine process’ were synonymous. The expression ‘machine process’ of course means one which could be carried out by the type of machine I was considering” (1947, p. 107)). In 1936, in contrast, Turing had introduced the “intuitive” notion of *computable numbers* not as those numbers that were computable by a mechanical process, but as those numbers “whose expression as a decimal are calculable by finite means” (1936–1937, p. 116). At that time, “finite,” “mechanical,” “effective,” “algorithmic,” etc., were used interchangeably to refer to the informal notion in question (cf. Gödel 1965, p. 72; Kleene 1987a, 55–56). Failure to recognize this leads to an unwarranted charge of ambiguity to Turing’s use of “finite” by Gandy (1988, p. 84), who was a student and friend of Turing in the latter part of Turing’s life.

¹¹E.g., see Church (1937; 1956, p. 52, n.119), Watson (1938, p. 448ff) and Newman (1955, p. 258). Kleene said: “Turing’s formulation comprises the functions computable by machines” (1938, p. 150). When von Neumann placed “Computable Numbers” at the foundations of the theory of finite automata, he introduced the problem addressed by Turing as that of giving “a general definition of what is meant by a computing automaton” (von Neumann 1951, p. 313). Most logic textbooks introduce Turing machines without qualifying “machine,” the way Turing did (see n. 2). More recently, doubts have been raised about the generality of Turing’s analysis of computability by machines (e.g., by Siegelmann, 1995).

¹²This widely cited phrase is in Wittgenstein (1980, sec. 1096). Wittgenstein knew Turing, who in 1939 attended Wittgenstein’s course on Foundations of Mathematics. Wittgenstein’s lectures, including his dialogues with Turing, are in Wittgenstein (1976). Discussions of their different points of views can be found in Shanker (1987) and Proudfoot and Copeland (1994). Gandy is more explicit than Wittgenstein: “Turing’s analysis makes no reference whatsoever to calculating machines. Turing machines appear as a result, as a codification, of his analysis of calculations by humans” (1988, p. 83–84). Sieg quotes and endorses Gandy’s statement (1994, p. 92; see also Sieg, 1997, p. 171). Along similar lines is Copeland (2000, pp. 10ff). According to Gandy and Sieg, “computability by a machine” is first explicitly analyzed in Gandy (1980).

¹³See n. 33.

¹⁴Newman (1954) and Turing (1959, p. 49). Moreover, in 1936 Turing wrote a précis of “Computable Numbers” for the French *Comptes Rendues*, containing a succinct description of his theory. The definition of “computable” is given directly in terms of machines, and the main result is appropriately stated in terms of machines:

On peut appeler ‘computable’ les nombres dont les décimales se laissent écrire par une machine . . . On peut démontrer qu’il n’y a aucun procédé général pour décider si une machine *m* n’écrit jamais le symbole 0 (Turing, 1936).

The quote translates as follows: We call “computable” the numbers whose decimals can be written by a machine — We demonstrate that there is no general procedure for deciding whether a machine *m* will never write the symbol 0. Human beings are not mentioned.

¹⁵See also *ibid.*, p. 93. Also, Turing machines “are chiefly of interest when we wish to consider what a machine could in principle be designed to do” (Turing, 1948, p. 6). In this latter paper, far from describing Turing machines as being humans who calculate, Turing described human beings as being universal digital computers:

It is possible to produce the effect of a computing machine by writing down a set of rules of procedure and asking a man to carry them out. Such a combination of a man with written instructions will be called a ‘Paper Machine.’ A man provided with paper, pencil, and rubber, and subject to strict discipline, is in effect a universal machine (Turing, 1948, p. 9).

Before the actual construction of the ACE, “paper machines” were the only universal machines available, and were used to test instruction tables designed for the ACE (Hodges, 1983, chapt. 6). Finally:

A digital computer is a universal machine in the sense that it can be made to replace ...any rival design of calculating machine, that is to say any machine into which one can feed data and which will later print out results (Turing, 1951c, p. 2).

Here, Turing formulated CT with respect to all calculating machines, without distinguishing between analog and digital computers. This fits well with other remarks by Turing, which assert that any function computable by analog machines could also be computed by digital machines (Turing, 1950, 451–452). And it strongly suggests that, for him, any device mathematically defined as giving the values of a non-computable function, that is, a function no Turing machine could compute — like the “oracle” in Turing (1939, pp. 166–167) — could not be physically constructed.

¹⁶The generality of Turing’s terminology, using the term “naturally,” asking “What are the possible processes [and not *mechanical* processes] which can be carried out in computing a number?” (Turing, 1936–1937, p. 135), and especially the conclusion that he wanted to reach, i.e. the *absolute* unsolvability of the decision problem, are more evidence that neither Gödel’s notion of “effective but non-mechanical procedure” nor the possibility of a computing machine more powerful than Turing machines had a place here. If there were effective but non-mechanical procedures, or machines computing non-computable functions, perhaps one of them could solve the decision problem. But Turing concluded that the decision problem “can have no solution” (ibid., p. 117). The term “absolutely unsolvable” was used in this connection by Post (1936, p. 289; Post attributes the term to Church in Post, 1965, p. 340; see also Post’s remarks in his 1936, p. 291; 1944, p. 310).

¹⁷CT is usually regarded as an unprovable thesis for which there is compelling evidence. For the opposing view that Turing proved CT as a theorem, see Gandy (1980, p. 124; 1988, p. 82); Mendelson (1990) challenges the view that CT cannot be proved. For a careful criticism of this revisionism about the provability of CT, see Folina (1998).

¹⁸After the 1930s, the term “effective” or “algorithmic” was often applied in the generalized sense, which included procedures calculating the values of partial rather than total functions. A Turing machine computing a partial function was called “circular” (Turing, 1936–1937, p. 119). For some inputs, circular machines did not generate outputs of the desired type.

¹⁹According to Shanker, this was Turing’s “basic idea” (1995, p. 55). But Turing never made such a claim. Sieg is therefore correct in writing that Turing “does not show that *mental* processes cannot go beyond mechanical ones” (Sieg and Byrnes, 1996, p. 117). However, Sieg is probably misreading Gödel in writing that “Gödel got it wrong, when he claimed that Turing’s argument in the 1936 paper was intended to show that ‘mental processes cannot go beyond mechanical procedures’” (Mundici and Sieg, 1995, p. 14; Sieg and Byrnes, 1996, p. 103). In that claim, Gödel did not write “processes,” as Sieg and his collaborators have him do, but “procedures” (Gödel, 1972). If by “mental procedures” Gödel meant effective procedures that a human being could carry out, as is plausible, then Turing was, indeed, arguing that “mental procedures cannot go beyond mechanical procedures.” However, we have seen that Gödel understood effectiveness differently from Turing. While on several occasions Gödel discussed “understanding” and “reflecting on meaning” in the context of what could be effectively calculated, Turing, in line with many other mathematicians, did not discuss such mentalistic notions — perhaps he considered them too obscure to be relevant (cf. n. 10 and 16).

²⁰Cf. Gödel (1965, p. 71) and Kleene (1988, pp. 38–44).

²¹E.g., Gödel (1951), Post (1965, p. 417) and Wang (1974, pp. 315–316); Turing expressed this opinion in Turing (1947, p. 123).

²²Turing’s completeness result applies to arithmetical statements that can be represented as Π_1^0 sentences, i.e. sentences of the form $\forall x(Rx)$. For a more detailed and complete account of Turing’s work on ordinal logics, including their importance for recursive function theory, see Feferman (1988).

²³For a classic formulation by Turing’s doctoral supervisor, see Church (1956, esp. section 07).

²⁴Hodges (1988, 1997) and Lucas (1996).

²⁵A third, which allowed mathematicians to distinguish “topics of interest from others,” was left out of his analysis because “the function of the mathematician,” in the context of foundations of mathematics, was “simply to determine the truth or falsity of propositions” (Turing, 1939, p. 208).

²⁶Sara Turing (1959) and Hodges (1983) report many episodes where Turing answered mathematical questions almost immediately, surprising his teachers or colleagues. Other people would take hours to solve the same problems (e.g. Turing, 1959, pp. 13, 27–28; also Newman, 1955, p. 255).

²⁷For attempts at solving this pseudo-problem, or expressions of puzzlement over Turing’s alleged change of mind, see Feferman (1988, p. 132), Hodges (1988, p. 10; 1997, pp. 28–29, 51) and Lucas (1996, p. 111).

²⁸See n. 21.

²⁹By “unsolvability and incompleteness results” Turing meant primarily, and respectively, the unsolvability of Hilbert’s decision problem, established by Church and himself, and Gödel’s two incompleteness theorems. Despite their differences, Turing said that the relevant common consequence of these results is that no formal system could exhaust all methods of proof. He later used this formulation in framing the mathematical objection.

³⁰In understanding Turing’s views on machine intelligence, I was helped by Colvin (1997). There is much evidence that studying the possibility of mechanical intelligence was Turing’s main goal from the beginning of his work on digital computers. Both Turing (1959) and Hodges (1983) cite several conversations in which Turing expressed his interest. For example, “Sometimes round 1944 he [Turing] talked to me about his plans for the construction of a universal computer and of the service such a machine might render to psychology in the study of the human brain” (Turing, 1959, p. 94; see also p. 88). In a letter to Ashby, Turing wrote that “In working on the ACE I am more interested in the possibility of producing models of the action of the brain than in the practical applications to computing” (Turing, undated). The report on the ACE already contained some obscure remarks on a machine possibly “displaying” intelligence (Turing, 1945). Turing addressed questions about machine intelligence at some length in a lecture to the London Mathematical Society (1947). The first paper fully devoted to the problem of machine intelligence was a report to the National Physics Laboratory, where Turing was working at the time (1948). The latter included all the important ideas that were then published in the famous *Mind* paper (1950), and more (e.g., see Copeland and Proudfoot, 1996). Turing also expressed his ideas in less formal occasions, where he didn’t add much of significance for our purposes (1951a, c, 1952, 1954).

³¹A good example is the quote at the beginning of the present paper, where Turing talked of “the functions of the mind or the brain.” But some passages in “Computing Machinery and Intelligence” indicate that attributing to Turing a strict materialism would be simplistic: “I do not wish to give the impression that there is no mystery about consciousness... But I do not think that these mysteries necessarily need to be solved before we can answer the question with which we are concerned in this paper” (Turing, 1950, p. 447); “the statistical evidence, at least for telepathy, is overwhelming” (ibid., p. 453). Turing seems to suggest that neither consciousness nor telepathy are a necessary condition for an individual to be intelligent.

³²Turing used the term “information” without definition, though he knew Shannon and some of his work (Hodges, 1983, pp. 250–252, 274, 410–411). In the ACE report, the brain was already said to be a machine (1945, pp. 103, 104). In the 1947 “Lecture,” it was a “digital computing machine” (1947, pp. 111, 123), which was distinguished from analog computers. In “Intelligent Machinery,” Turing changed his position and said the brain was a “continuous controlling” machine (1948, p. 5), i.e., an analog computer devoted to processing information. The latter view was repeated in Turing (1950, p. 451). But Turing also claimed that “brains very nearly fall into this class [of discrete controlling machinery, i.e., digital computers], and there seems every reason to believe that they could have been made to fall genuinely into it without any change in their essential properties” (1948, p. 6). For Turing, at any rate, the functions computed by analog computers could also be computed by digital computers so that, under the circumstances of the Turing test, the two kinds of machines could not be distinguished (1950, pp. 451–452).

³³Other examples include: digital computers “can be made to do any rule of thumb process” (1947, pp. 106–107, 112), “as soon as any technique becomes at all stereotyped it becomes possible to devise a system of instruction tables which will enable the electronic computer to do it [the act made possible by the technique] for itself” (ibid., p. 121). It sufficed to program the universal machine to do the job. Things that could be programmed included games like chess and mathematical derivations in a formal system (ibid., p. 122). CT applied to digital computers was repeated in Turing (1948, pp. 6–7; 1950, p. 436; 1951b, p. 1).

³⁴Usually, the mathematical objection is attributed to Lucas (1961), who formulated it using Gödel’s first and second incompleteness theorems. [Lucas acknowledged the discussion contained in Turing (1950), as well as others successive to it, on p. 112 of Lucas, 1961; also, a bibliography about the early debate can be found in Lucas (1970, pp. 174–176).] Since Lucas proposed his version of it, the mathematical objection has been hotly debated. Recent formulations can be found in Lucas (1996), Penrose (1994) and Wright (1995). Recent rebuttals can be found in Chalmers (1995) and Detlefsen (1995). Turing discussed the mathematical objection not in terms of Gödel incompleteness, but in terms of his machines. Lucas’s well-known formulation may have made Turing’s discussion less transparent, explaining why his reply is little understood. Moreover, Turing’s best-known paper on machine intelligence, the *Mind* paper, contains a shorter and less perspicuous reply than other papers. An embryonic form of the mathematical objection was formulated by Post as early as 1921 (Post, 1965, p. 417), based on his remarkable anticipation of Gödel, Church, and Turing’s results on incompleteness and undecidability, but Post’s work remained unpublished until much later (Davis, 1965, p. 338). Turing appears to be the first person to raise the objection and give a reply in a published form. An interesting question is whether anyone proposed the mathematical objection to him, or whether he formulated it himself. He did not cite anyone in this respect, but did use the ambiguous phrase “[t]hose who hold to the mathematical argument” (Turing, 1950, p. 445), which suggests that the mathematical objection was at least informally discussed at the time.

³⁵Penrose’s interpretation is reiterated by Grush and Churchland (1995, p. 325). Neither Penrose, nor Grush and Churchland, nor Gandy give reasons favoring their interpretation, which appears to be influenced by the shape the discussion of the mathematical objection took after the 1960s.

³⁶In “Intelligent Machinery,” as references for the mathematical theorems, Turing gave Gödel (1931), Church (1936) and Turing (1936–1937). In Turing 1950, he added the names of Kleene and Rosser, and a reference to Kleene (1935).

³⁷The claim shifted slightly. In his “Lecture” Turing had said that, though the mathematical theorems excluded the possibility of a machine which is both infallible and intelligent, they did not exclude the possibility of an intelligent but fallible machine. In that occasion, he said nothing about to what extent intelligence is compatible with fallibility. Now, more blatantly, intelligence itself does not require infallibility.

References

- Benacerraf, P. and Putnam, H. (1964), *Philosophy of Mathematics*, Englewood Cliffs, NJ: Prentice-Hall.
- Bernays, P. (1935a), ‘Sur le platonisme dans les mathématiques’, *L’enseignement mathématique* 34, pp. 52–69, reprinted in P. Benacerraf and H. Putnam (1964), *Philosophy of Mathematics*, Englewood Cliffs, NJ: Prentice-Hall, pp. 274–286.
- Bernays, P. (1935b), ‘Quelques points essentiels de la métamathématique’, *L’enseignement mathématique* 34, pp. 70–95.
- Bernays, P. (1967), ‘Hilbert, David’, in P. Edwards, ed., *Encyclopedia of Philosophy*, Vol. 3, New York: Macmillan and Free Press, pp. 496–504.
- Boolos, G. and Jeffrey, R. (1974), *Computability and Logic*, New York, NY: Cambridge University Press.

- Chalmers, D., ed. (1995), 'Symposium on Roger Penrose's *Shadows of the Mind*', *Psyche* 2, <http://psyche.cs.monash.edu.au>.
- Church, A. (1936), 'An Unsolvable Problem in Elementary Number Theory', *The American Journal of Mathematics* 58, pp. 345–363, reprinted in M. Davis (1965), *The Undecidable*, Ewlett, NY: Raven Press, pp. 88–107.
- Church, A. (1937), 'Review of Turing (1936–1937)', *Journal of Symbolic Logic* 2, pp. 42–43.
- Colvin, S.I. (1997), *Intelligent Machinery; Turing's Ideas and Their Relation to the Work of Newell and Simon*, unpublished M.S. Dissertation, Carnegie Mellon University.
- Copeland, B.J. (1996), 'The Church-Turing Thesis', in E.N. Zalta, ed., *Stanford Encyclopedia of Philosophy*, <<http://plato.stanford.edu>>, as consulted in January 1999.
- Copeland, B.J. (1999), 'A Lecture and Two Radio Broadcasts on Machine Intelligence by Alan Turing', in K. Furukawa, D. Michie, and S. Muggleton, eds, *Machine Intelligence 15*, Oxford: Oxford University Press.
- Copeland, B.J. (2000), 'Narrow Versus Wide Mechanism: Including a Re-Examination of Turing's Views on the Mind-Machine Issue', *The Journal of Philosophy* XCVI, pp. 5–32.
- Copeland, B.J. (forthcoming), *The Essential Turing*, Oxford: Oxford University Press.
- Copeland, B.J. and Proudfoot, D. (1996), 'On Alan Turing's Anticipation of Connectionism', *Synthese* 108, pp. 361–377.
- Davis, M. (1958), *Computability and Unsolvability*, New York: McGraw-Hill.
- Davis, M. (1965), *The Undecidable*, Ewlett, NY: Raven Press.
- Davis, M. (1987), 'Mathematical Logic and the Origin of Modern Computers', in E.R. Phillips, ed., *Studies in the History of Mathematics* (The Mathematical Association of America), pp. 137–165, reprinted in R. Herken (1988), ed., *The Universal Machine: A Half-Century Survey*, New York: Oxford University Press, pp. 149–174.
- Dennett, D.C. (1978), *Brainstorms: Philosophical Essays on Mind and Psychology*, Cambridge, MA: MIT Press.
- Detlefsen, M. (1995), 'Wright on the Non-Mechanizability of Intuitionist Reasoning', *Philosophia Mathematica* 3, pp. 103–119.
- Eddington A.E. (1928), *The Nature of the Physical World*, Cambridge: Cambridge University Press.
- Feferman S. (1988), 'Turing in the Land of $O(z)$ ', in R. Herken, ed., *The Universal Machine: A Half-Century Survey*, New York: Oxford University Press, pp. 113–147.
- Folina, J. (1998), 'Church's Thesis: Prelude to a Proof', *Philosophia Mathematica* 6, pp. 302–323.
- Gandy, R. (1980), 'Church's Thesis and principles for mechanisms', in J. Barwise, H.J. Keisler, and K. Kunen, eds., *The Kleene Symposium*, Amsterdam: North-Holland, pp. 123–148.
- Gandy, R. (1996), 'Human versus Mechanical Intelligence', in P.J.R. Millican and A. Clark, eds., *Machines and Thought: The Legacy of Alan Turing*, Oxford: Clarendon, pp. 125–136.
- Gödel, K. (1931), 'On Formally Undecidable Propositions of Principia Mathematica and Related Systems I', *Monatshefte für Mathematik und Physik* 38, pp. 173–198, reprinted in M. Davis (1965), *The Undecidable*, Ewlett, NY: Raven Press, pp. 5–38.
- Gödel, K. (1934), 'On Undecidable Propositions of Formal Mathematical Systems', in M. Davis (1965), *The Undecidable*, Ewlett, NY: Raven Press, pp. 41–71.
- Gödel, K. (1951), 'Some Basic Theorems on the Foundations of Mathematics and Their Implications', edited text of Josiah Gibbs Lecture delivered at Brown University, in S. Feferman et al., eds., (1995), *Collected Works, Vol III*, Oxford: Oxford University Press, pp. 304–323.
- Gödel, K. (1958), 'On a Hitherto Unutilized Extension of the Finitary Standpoint', *Dialectica* 12, pp. 280–287, reprinted in S. Feferman et al., eds., (1990), *Collected Works, Vol II*, Oxford: Oxford University Press, pp. 241–251.
- Gödel, K. (1965), 'Postscriptum to 'On Undecidable Propositions of Formal Mathematical Systems'', in M. Davis (1965), *The Undecidable*, Ewlett, NY: Raven Press, pp. 71–73.
- Gödel, K. (1972), 'Some Remarks on the Undecidability Results', in S. Feferman et al., eds., *Collected Works, Vol. II*, Oxford: Oxford University Press, 1990, pp. 305–306.

- Grush, R. and Churchland, P.S. (1995), 'Gaps in Penrose's Toilings', *Journal of Consciousness Studies* 2, pp. 10–29; reprinted in P.M. Churchland and P.S. Churchland (1998), *On the Contrary: Critical Essays, 1987–1997*, Cambridge, MA: MIT Press, pp. 205–229, 324–327.
- Herbrand, J. (1971), *Logical Writings*, W.D. Goldfarb, ed., Dordrecht: Reidel.
- Herken, R., ed. (1988), *The Universal Machine: A Half-Century Survey*, New York: Oxford University Press.
- Hilbert, D. and Ackermann, W. (1928), *Grundzüge der theoretischen Logik*, Berlin: Springer.
- Hodges, A. (1983), *Alan Turing: The Enigma*, New York: Simon and Schuster.
- Hodges, A. (1988), 'Alan Turing and the Turing Machine', in R. Herken, ed., *The Universal Turing Machine: A Half-Century Survey*, New York: Oxford University Press, pp. 3–15.
- Hodges, A. (1997), *Turing; a Natural Philosopher*, London: Phoenix.
- Ince, D.C., ed. (1992), *Collected Works of A.M. Turing: Mechanical Intelligence*, Amsterdam: North Holland.
- Kálmár, L. (1959), 'An Argument Against the Plausibility of Church's Thesis', in A. Heyting, ed., *Constructivity in Mathematics*, Amsterdam: North-Holland, pp. 72–80.
- Kleene, S.C. (1935), 'General Recursive Functions of Natural Numbers', *American Journal of Mathematics* 57, pp. 153–173, 219–244.
- Kleene, S.C. (1938), 'On Notation for Ordinal Numbers', *Journal of Symbolic Logic* 3, pp. 150–155.
- Kleene, S.C. (1943), 'Recursive Predicates and Quantifiers', *Transactions of the American Mathematical Society* 53, pp. 41–73, reprinted in M. Davis (1965), *The Undecidable*, Ewlett, NY: Raven Press, pp. 254–287.
- Kleene, S.C. (1952), *Introduction to Metamathematics*, Amsterdam: North-Holland.
- Kleene, S.C. (1987a), 'Gödel's Impression on Students of Logic in the 1930s', in P. Weingartner and L. Schmetterer, eds., *Gödel Remembered*, Napoli: Bibliopolis, pp. 49–64.
- Kleene, S.C. (1987b), 'Reflections on Church's Thesis', *Notre Dame Journal of Formal Logic* 28, pp. 490–498.
- Kleene, S.C. (1988), 'Turing's Analysis of Computability', in R. Herken, ed., *The Universal Machine: A Half-Century Survey*, New York: Oxford University Press, pp. 17–54.
- Kreisel, G. (1972), 'Which Number-Theoretic Problems Can Be Solved in Recursive Progressions on Π_1^1 -paths Through O', *Journal of Symbolic Logic* 37, pp. 311–334.
- Kreisel, G. (1987), 'Gödel's Excursions Into Intuitionistic Logic', in P. Weingartner and L. Schmetterer, eds., *Gödel Remembered*, Napoli: Bibliopolis, pp. 65–179.
- Leiber, J. (1991), *An Invitation to Cognitive Science*, Cambridge, MA: Basil Blackwell.
- Lucas, J.R. (1961), 'Minds, Machines and Gödel', *Philosophy* 36, pp. 112–127.
- Lucas, J.R. (1970), *The Freedom of the Will*, Oxford: Clarendon Press.
- Lucas, J.R. (1996), 'Minds, Machines, and Gödel: A Retrospect', in P.J.R. Millican and A. Clark, eds., *Machines and Thought; The Legacy of Alan Turing*, Oxford: Clarendon Press, pp. 103–124.
- Mendelson, E. (1963), 'On Some Recent Criticism of Church's Thesis', *Notre Dame Journal of Formal Logic* 4, pp. 201–205.
- Mendelson, E. (1990), 'Second Thoughts about Church's Thesis and Mathematical Proofs', *Journal of Philosophy* 88, pp. 225–233.
- Michie, D. (1982), *Machine Intelligence and Related Topics; an Information Scientist's Weekend Book*, New York: Gordon and Breach Science.
- Mundici, D. and Sieg, W. (1995), 'Paper Machines', *Philosophia Mathematica* 3, pp. 5–30.
- Newman, M.H.A. (1954), Obituary Notice for Alan Turing, *The Times*, 16 June.
- Newman, M.H.A. (1955), 'Alan Mathison Turing,' in *Biographical Memoirs of Fellows of the Royal Society*, London: The Royal Society, pp. 253–263.
- Odifreddi, P. (1989), *Classical Recursion Theory*, Amsterdam: North-Holland.
- Penrose, R. (1994), *Shadows of the Mind; A Search for the Missing Science of Consciousness*, Oxford: Oxford University Press.
- Post, E. (1936), 'Finite Combinatorial Processes. Formulation I', *Journal of Symbolic Logic* 1, pp. 103–105, reprinted in M. Davis (1965), *The Undecidable*, Ewlett, NY: Raven Press, pp. 289–291.

- Post, E. (1944), 'Recursively Enumerable Sets of Positive Integers and their Decision Problems', *Bulletin of the American Mathematical Society* 50, pp. 284–316, reprinted in M. Davis (1965), *The Undecidable*, Ewlett, NY: Raven Press, pp. 305–337.
- Post, E. (1965), 'Absolutely Unsolvable Problems and Relatively Undecidable Propositions; Account of an Anticipation', in M. Davis (1965), *The Undecidable*, Ewlett, NY: Raven Press, pp. 340–433.
- Pratt, V. (1987), *Thinking Machines; The Evolution of Artificial Intelligence*, Oxford: Basil Blackwell.
- Proudfoot, D., and Copeland, B. J. (1994), 'Turing, Wittgenstein and the Science of the Mind', *Australasian Journal of Philosophy* 72, pp. 497–519.
- Rogers, H. (1967), *Theory of Recursive Functions and Effective Computability*, New York: McGraw-Hill.
- Shanker, S.G. (1987), 'Wittgenstein versus Turing on the Nature of Church's Thesis', *Notre Dame Journal of Formal Logic* 28, pp. 615–649.
- Shanker, S.G. (1995), 'Turing and the Origins of AI', *Philosophia Mathematica* 3, pp. 52–85.
- Sieg, W. (1994), 'Mechanical Procedures and Mathematical Experience', in A. George, ed., *Mathematics and Mind*, New York: Oxford University Press, pp. 71–117.
- Sieg, W. (1997), 'Step by Recursive Step: Church's Analysis of Effective Calculability', *Bulletin of Symbolic Logic* 3, pp. 154–180.
- Sieg, W. and Byrnes, J. (1996), 'K-graph Machines: Generalizing Turing's Machines and Arguments', in P. Hájek, ed., *Gödel '96*, Berlin: Springer, pp. 98–119.
- Siegelmann, H. (1995), 'Computation Beyond the Turing Limit', *Science* 268, pp. 545–548.
- Tamburrini, G. (1988), *Reflections on Mechanism*, unpublished Ph.D. dissertation, Columbia University.
- Turing, A.M. (1936), 2pp. Typescript of précis of Turing (1936–1937), made for *Comptes Rendues*. AMT K4, Contemporary Scientific Archives Centre (CSAS), King's College Library, Cambridge.
- Turing, A.M. (1936–1937), 'On Computable Numbers, with an Application to the Entscheidungsproblem', reprinted in M. Davis (1965), *The Undecidable*, Ewlett, NY: Raven Press, pp. 116–154.
- Turing, A.M. (1937) 'Computability and λ -definability' *Journal of Symbolic Logic* 2, pp. 153–163.
- Turing, A.M. (1939), 'Systems of Logic Based on Ordinals', *Proceedings of the London Mathematical Society, Ser. 2* 45, pp. 161–228, reprinted in M. Davis (1965), *The Undecidable*, Ewlett, NY: Raven Press, pp. 155–222.
- Turing, A.M. (1940a), Letter to Newman, dated "early 1940?" by R. O. Gandy, AMT D/2, Contemporary Scientific Archives Centre, King's College Library, Cambridge. Printed in Copeland, forthcoming.
- Turing, A.M. (1940b), Letter to Newman, dated 21 April, "1940" added by R.O. Gandy, AMT D/2, Contemporary Scientific Archives Centre, King's College Library, Cambridge. Printed in Copeland, forthcoming.
- Turing, A.M. (1945), 'Proposal for Development in the Mathematical Division of an Automatic Computing Engine (ACE)', reprinted in D.C. Ince (1992), ed., *Collected Works of A.M. Turing: Mechanical Intelligence*, Amsterdam: North Holland. pp. 20–105.
- Turing, A.M. (1947), 'Lecture to the London Mathematical Society on 20 February 1947', reprinted in D.C. Ince (1992), ed., *Collected Works of A.M. Turing: Mechanical Intelligence*, Amsterdam: North Holland. pp. 87–105.
- Turing, A.M. (1948), 'Intelligent Machinery', reprinted in D.C. Ince (1992), ed., *Collected Works of A.M. Turing: Mechanical Intelligence*, Amsterdam: North Holland. pp. 87–106.
- Turing, A.M. (1950), 'Computing Machinery and Intelligence', in D.C. Ince (1992), ed., *Collected Works of A.M. Turing: Mechanical Intelligence*, Amsterdam: North Holland.
- Turing, A.M. (1951a), 'Intelligent Machinery, a Heretical Theory,' Lecture given to '51 Society' at Manchester (c) 1951, AMT B.4, Contemporary Scientific Archives Centre, King's College Library, Cambridge. Printed in E.S. Turing (1959), *Alan M. Turing*, Cambridge: Heffer & Sons, pp. 128–134, and in B.J. Copeland (1999), 'A Lecture and Two Radio Broadcasts on Machine

- Intelligence by Alan Turing', in K. Furukawa, D. Michie, and S. Muggleton, eds., *Machine Intelligence 15*, Oxford: Oxford University Press.
- Turing, A.M. (1951b), *Programmers' Handbook for the Manchester Electronic Computer*. University of Manchester Computing Laboratory. Also available at <http://www.AlanTuring.net/programmers_handbook>.
- Turing, A.M. (1951c), 'Can digital computers think?' Typescript of talk broadcast in BBC Third Programme 15 May 1951, AMT B.5, Contemporary Scientific Archives Centre, King's College Library, Cambridge. Printed in B.J. Copeland (1999), 'A Lecture and Two Radio Broadcasts on Machine Intelligence by Alan Turing', in K. Furukawa, D. Michie, and S. Muggleton, eds., *Machine Intelligence 15*, Oxford: Oxford University Press.
- Turing, A.M. (1952), 'Can automatic calculating machines be said to think?' Typescript of broadcast discussion in BBC Third Programme, 14 and 23 January 1952, between M.H.A. Newman, A.M. Turing, Sir Geoffrey Jefferson, R.B. Braithwaite, AMT B.6, Contemporary Scientific Archives Centre, King's College Library, Cambridge. Printed in B.J. Copeland (1999), 'A Lecture and Two Radio Broadcasts on Machine Intelligence by Alan Turing', in K. Furukawa, D. Michie, and S. Muggleton, eds., *Machine Intelligence 15*, Oxford: Oxford University Press.
- Turing, A.M. (1953), 'Digital Computers Applied to Games', in B.V. Bowden, ed., *Faster Than Thought*, London: Pittman, pp. 286–310, reprinted in D.C. Ince (1992), ed., *Collected Works of A.M. Turing: Mechanical Intelligence*, Amsterdam: North Holland, pp. 161–185.
- Turing, A.M. (1954), 'Solvable and Unsolvable Problems', reprinted in D.C. Ince (1992), ed., *Collected Works of A.M. Turing: Mechanical Intelligence*, Amsterdam: North Holland, pp. 187–203.
- Turing, A.M. (undated), Letter to Ashby, NPL M11, Archives of the Science Museum Library, London. Also available at <http://www.AlanTuring.net/turing_ashby>.
- Turing, E.S. (1959), *Alan M. Turing*, Cambridge: Heffer & Sons.
- van Heijenoort, J. ed., (1967) *From Frege to Gödel*, Cambridge, MA: Harvard University Press.
- von Neumann, J. (1931), 'The Formalist Foundations of Mathematics', *Erkenntnis* 2, pp. 91–121, reprinted in P. Benacerraf and H. Putnam (1964), *Philosophy of Mathematics*, Englewoods Cliffs, NJ: Prentice-Hall, pp. 50–54.
- von Neumann, J. (1951), 'The General and Logical Theory of Automata', in A. H. Taub, ed., (1963), *Collected Works; Vol V*, London: Pergamon Press, pp. 288–328.
- Wang, H. (1974), *From Mathematics to Philosophy*, New York: Humanities Press.
- Watson, A.G.D. (1938), 'Mathematics and Its Foundations', *Mind* 47, pp. 440–451.
- Webb, J.C. (1980), *Mechanism, Mentalism, and Metamathematics*, Dordrecht: Reidel.
- Whitehead, A.N. and Russell, B. (1910–1913), *Principia Mathematica*, Cambridge: Cambridge University Press.
- Wilkes, M.V. (1985), *Memoirs of a Computer Pioneer*, Cambridge, MA: MIT Press.
- Wittgenstein, L. (1976), Wittgenstein's Lectures on the Foundations of Mathematics Cambridge, 1939, in C. Diamond, ed., Ithaca, NY: Cornell University Press.
- Wittgenstein, L. (1980), *Remarks on the Philosophy of Psychology*, Vol. 1, G.E.M. Anscombe, and G.H. von Wright, eds., Chicago: University of Chicago Press.
- Wright, C. (1995), 'Intuitionists Are Not (Turing) Machines', *Philosophia Mathematica* 3, pp. 86–102.