Stanford Encyclopedia of Philosophy

Alan Turing

Alan Turing

First published Mon Jun 3, 2002; substantive revision Mon Sep 30, 2013

Alan Turing (1912–1954) never described himself as a philosopher, but his 1950 paper "Computing Machinery and Intelligence" is one of the most frequently cited in modern philosophical literature. It gave a fresh approach to the traditional mind-body problem, by relating it to the mathematical concept of computability he himself had introduced in his 1936–7 paper "On computable numbers, with an application to the Entscheidungsproblem." His work can be regarded as the foundation of computer science and of the artificial intelligence program.

1. Outline of Life........1
2. The Turing Machine and Computability........3
3. The Logical and the Physical........7
4. The Uncomputable........9
5. Building a Universal Machine........10
6. Building a Brain........11
7. Machine Intelligence........13
8. Unfinished Work........17
9. Alan Turing: the Unknown Mind........18
Bibliography........19
    Selected Works by Turing........19
        The Collected Works of A. M. Turing consists of 4 volumes:........20
    Secondary Literature........20

1. Outline of Life

Alan Turing's short and extraordinary life has attracted wide interest. It has inspired his mother's memoir (E. S. Turing 1959), a detailed biography (Hodges 1983), a play and television film (Whitemore 1986), and various other works of fiction and art.

There are many reasons for this interest, but one is that in every sphere of his life and work he made unexpected connections between apparently unrelated areas. His central contribution to science and philosophy came through his treating the subject of symbolic logic as a new branch of applied mathematics, giving it a physical and engineering content. Unwilling or unable to remain within any standard role or department of thought, Alan Turing continued a life full of incongruity. Though a shy, boyish, man, he had a pivotal role in world history through his role in Second World War cryptology. Though the founder of the dominant technology of the twentieth century, he variously impressed, charmed or disturbed people with his unworldly innocence and his dislike of moral or intellectual compromise.

Alan Mathison Turing was born in London, 23 June 1912, to upper-middle-class British parents. His schooling was of a traditional kind, dominated by the British imperial system, but from earliest life his fascination with the scientific impulse—expressed by him as finding the 'commonest in nature'—found him at odds with authority. His scepticism, and disrespect for worldly values, were never tamed and became ever more confidently eccentric. His moody humour swung between gloom and vivacity. His life was also notable as that of a gay man with strong emotions and a growing insistence on his identity.

His first true home was at King's College, Cambridge University, noted for its progressive intellectual life centred on J. M. Keynes. Turing studied mathematics with increasing distinction and was elected a Fellow of the college in 1935. This appointment was followed by a remarkable and sudden début in an area where he was an unknown figure: that of mathematical logic. The paper "On Computable Numbers…" (Turing 1936–7) was his first and perhaps greatest triumph. It gave a definition of computation and an absolute limitation on what computation could achieve, which makes it the founding work of modern computer science. It led him to Princeton for more advanced work in logic and other branches of mathematics. He had the opportunity to remain in the United States, but chose to return to Britain in 1938, and was immediately recruited for the British communications war.

From 1939 to 1945 Turing was almost totally engaged in the mastery of the German enciphering machine, Enigma, and other cryptological investigations at now-famous Bletchley Park, the British government's wartime communications headquarters. Turing made a unique logical contribution to the decryption of the Enigma and became the chief scientific figure, with a particular responsibility for reading the U-boat communications. As such he became a top-level figure in Anglo-American liaison, and also gained exposure to the most advanced electronic technology of the day.

Combining his ideas from mathematical logic, his experience in cryptology, and some practical electronic knowledge, his ambition, at the end of the war in Europe, was to create an electronic computer in the full modern sense. His plans, commissioned by the National Physical Laboratory, London, were overshadowed by the more powerfully supported American projects. Turing also laboured under the disadvantage that his wartime achievements remained totally secret. His ideas led the field in 1946, but this was little recognised. Frustrated in his work, he emerged as a powerful marathon runner, and almost qualified for the British team in the 1948 Olympic games.

Turing's motivations were scientific rather than industrial or commercial, and he soon returned to the theoretical limitations of computation, this time focussing on the comparison of the power of computation and the power of the human brain. His contention was that the computer, when properly programmed, could rival the brain. It founded the 'Artificial Intelligence' program of coming decades.

In 1948 he moved to Manchester University, where he partly fulfilled the expectations placed upon him to plan software for the pioneer computer development there, but still remained a free-ranging thinker. It was here that his famous 1950 paper, "Computing Machinery and Intelligence," (Turing 1950b) was written. In 1951 he was elected a Fellow of the Royal Society for his 1936 achievement, yet at the same time he was striking into entirely new territory with a mathematical theory of biological morphogenesis (Turing 1952).

This work was interrupted by Alan Turing's arrest in February 1952 for his sexual affair with a young Manchester man, and he was obliged, to escape imprisonment, to undergo the injection of oestrogen intended to negate his sexual drive. He was disqualified from continuing secret cryptological work. His general libertarian attitude was enhanced rather than suppressed by the criminal trial, and his intellectual individuality also remained as lively as ever. While remaining formally a Reader in the Theory of Computing, he not only embarked on more ambitious applications of his biological theory, but advanced new ideas for fundamental physics.

For this reason his death, on 7 June 1954, at his home in Wilmslow, Cheshire, came as a general surprise. In hindsight it is obvious that Turing's unique status in Anglo-American secret communication work meant that there were pressures on him of which his contemporaries were unaware; there was certainly another ‘security’ conflict with government in 1953 (Hodges 1983, p. 483). Some commentators, e.g. Dawson (1985), have argued that assassination should not be ruled out. But he had spoken of suicide, and his death, which was by cyanide poisoning, was most likely by his own hand, contrived so as to allow those who wished to do so to believe it a result of his penchant for chemistry experiments. The symbolism of its dramatic element—a partly eaten apple—has continued to haunt the intellectual Eden from which Alan Turing was expelled.

2. The Turing Machine and Computability

Alan Turing drew much between 1928 and 1933 from the work of the mathematical physicist and populariser A. S. Eddington, from J. von Neumann's account of the foundations of quantum mechanics, and then from Bertrand Russell's mathematical logic. Meanwhile, his lasting fascination with the problems of mind and matter was heightened by emotional elements in his own life (Hodges 1983, p. 63). In 1934 he graduated with an outstanding degree in mathematics from Cambridge University, followed by a successful dissertation in probability theory which won him a Fellowship of King's College, Cambridge, in 1935. This was the background to his learning, also in 1935, of the problem which was to make his name.

It was from the lectures of the topologist M. H. A. (Max) Newman in that year that he learnt of Gödel's 1931 proof of the formal incompleteness of logical systems rich enough to include arithmetic, and of the outstanding problem in the foundations of mathematics as posed by Hilbert: the “Entscheidungsproblem” (decision problem). Was there a method by which it could be decided, for any given mathematical proposition, whether or not it was provable?

The principal difficulty of this question lay in giving an unassailably correct and general definition of what was meant by such expressions as ‘definite method’ or ‘effective procedure.’ Turing worked on this alone for a year until April 1936; independence and isolation was to be both his strength, in formulating original ideas, and his weakness, when it came to promoting and implementing them.

The word ‘mechanical’ had often been used of the formalist approach lying behind Hilbert's problem, and Turing seized on the concept of the *machine.*Turing's solution lay in defining what was soon to be named the *Turing machine.* With this he defined the concept of ‘the mechanical’ in terms of simple atomic operations. The Turing machine formalism was modelled on the teleprinter, slightly enlarged in scope to allow a paper tape that could move in both directions and a ‘head’ that could read, erase and print new symbols, rather than only read and punch permanent holes.

The Turing machine is ‘theoretical,’ in the sense that it is not intended actually to be engineered (there being no point in doing so), although it is essential that its atomic components (the paper tape, movement to left and right, testing for the presence of a symbol) are such as *could* actually be implemented. The whole point of the formalism is to reduce the concept of ‘method’ to simple operations that can unquestionably be ‘effected.’

Nevertheless Turing's purpose was to embody the most general mechanical process as carried out by a *human being.* His analysis began not with any existing computing machines, but with the picture of a child's exercise book marked off in squares. From the beginning, the Turing machine concept aimed to capture what the *human mind* can do when carrying out a procedure.

In speaking of ‘the’ Turing machine it should be made clear that there are *infinitely many* Turing machines, each corresponding to a different method or procedure, by virtue of having a different ‘table of behaviour.’ Nowadays it is almost impossible to avoid imagery which did not exist in 1936: that of the computer. In modern terms, the ‘table of behaviour’ of a Turing machine is equivalent to a computer program.

If a Turing machine corresponds to a computer program, what is the analogy of the computer? It is what Turing described as a *universal* machine (Turing 1936, p. 241). Again, there are *infinitely many* universal Turing machines, forming a subset of Turing machines; they are those machines with ‘tables of behaviour’ complex enough to read the tables of other Turing machines, and then do what those machines would have done. If this seems strange, note the modern parallel that any computer can be simulated by software on another computer. The way that tables can read and simulate the effect of other tables is crucial to Turing's theory, going far beyond Babbage's ideas of a hundred years earlier. It also shows why Turing's ideas go to the heart of the modern computer, in which it is essential that programs are themselves a form of data which can be manipulated by other programs. But the reader must always remember that in 1936 there were no such computers; indeed the modern computer arose *out of* the formulation of ‘behaving mechanically’ that Turing found in this work.

Turing's machine formulation allowed the precise definition of the *computable:* namely, as what can be done by a Turing machine acting alone. More exactly, computable operations are those which can be effected by what Turing called *automatic* machines. The crucial point here is that the action of an automatic Turing machine is totally determined by its ‘table of behaviour’. (Turing also allowed for ‘choice machines’ which call for human inputs, rather than being totally determined.) Turing then proposed that this definition of ‘computable’ captured precisely what was intended by such words as ‘definite method, procedure, mechanical process’ in stating the*Entscheidungsproblem.*

In applying his machine concept to the *Entscheidungsproblem,* Turing took the step of defining*computable numbers*. These are those real numbers, considered as infinite decimals, say, which it is possible for a Turing machine, starting with an empty tape, to print out. For example, the Turing machine which simply prints the digit 1 and moves to the right, then repeats that action for ever, can thereby compute the number .1111111… A more complicated Turing machine can compute the infinite decimal expansion of π.

Turing machines, like computer programs, are countable; indeed they can be ordered in a complete list by a kind of alphabetical ordering of their ‘tables of behaviour’. Turing did this by encoding the tables into ‘description numbers’ which can then be ordered in magnitude. Amongst this list, a subset of them (those with ‘satisfactory’ description numbers) are the machines which have the effect of printing out infinite decimals. It is readily shown, using a ‘diagonal’ argument first used by Cantor and familiar from the discoveries of Russell and Gödel, that there can be no Turing machine with the property of deciding whether a description number is satisfactory or not. The argument can be presented as follows. Suppose that such a Turing machine exists. Then it is possible to construct a new Turing machine which works out in turn the Nth digit from the Nth machine possessing a satisfactory description number. This new machine then prints an Nth digit differing from that digit. As the machine proceeds, it prints out an infinite decimal, and therefore has a ‘satisfactory’ description number. Yet this number must by construction differ from the outputs of every Turing machine with a satisfactory description number. This is a contradiction, so the hypothesis must be false (Turing 1936, p. 246). From this, Turing was able to answer Hilbert's *Entscheidungsproblem* in the negative: there can be no such general method.

Turing's proof can be recast in many ways, but the core idea depends on the *self-reference*involved in a machine operating on symbols, which is itself described by symbols and so can operate on its own description. Indeed, the self-referential aspect of the theory can be highlighted by a different form of the proof, which Turing preferred (Turing 1936, p. 247). Suppose that such a machine for deciding satisfactoriness does exist; then apply it to its own description number. A contradiction can readily be obtained. However, the ‘diagonal’ method has the advantage of bringing out the following: that a real number may be *defined* unambiguously, yet be*uncomputable.* It is a non-trivial discovery that whereas some infinite decimals (e.g. π) may be encapsulated in a finite table, other infinite decimals (in fact, almost all) cannot. Likewise there are decision problems such as ‘is this number prime?’ in which infinitely many answers are wrapped up in a finite recipe, while there are others (again, almost all) which are not, and must be regarded as requiring infinitely many different methods. ‘Is this a provable proposition?’ belongs to the latter category.

This is what Turing established, and into the bargain the remarkable fact that anything that *is*computable can in fact be computed by *one* machine, a universal Turing machine.

It was vital to Turing's work that he justified the definition by showing that it encompassed the most general idea of ‘method’. For if it did not, the *Entscheidungproblem* remained open: there might be some more powerful type of method than was encompassed by Turing computability. One justification lay in showing that the definition included many processes a mathematician would consider to be natural in computation (Turing 1936, p. 254). Another argument involved a human calculator following written instruction notes. (Turing 1936, p. 253). But in a bolder argument, the one he placed first, he considered an ‘intuitive’ argument appealing to the *states of mind* of a human computer. (Turing 1936, p. 249). The entry of ‘mind’ into his argument was highly significant, but at this stage it was only a mind following a rule.

To summarise: Turing found, and justified on very general and far-reaching grounds, a precise mathematical formulation of the conception of a general process or method. His work, as presented to Newman in April 1936, argued that his formulation of ‘computability’ encompassed ‘the possible processes which can be carried out in computing a number.’ (Turing 1936, p. 232). This opened up new fields of discovery both in practical computation, and in the discussion of human mental processes. However, although Turing had worked as what Newman called ‘a confirmed solitary’ (Hodges 1983, p 113), he soon learned that he was not alone in what Gandy (1988) has called ‘the confluence of ideas in 1936.’

The Princeton logician Alonzo Church had slightly outpaced Turing in finding a satisfactory definition of what he called ‘effective calculability.’ Church's definition required the logical formalism of the *lambda-calculus.* This meant that from the outset Turing's achievement merged with and superseded the formulation of Church's Thesis, namely the assertion that the lambda-calculus formalism correctly embodied the concept of effective process or method. Very rapidly it was shown that the mathematical scope of Turing computability coincided with Church's definition (and also with the scope of the *general recursive functions* defined by Gödel). Turing wrote his own statement (Turing 1939, p. 166) of the conclusions that had been reached in 1938; it is in the Ph.D. thesis that he wrote under Church's supervision, and so this statement is the nearest we have to a joint statement of the ‘Church-Turing thesis’:

> A function is said to be ‘effectively calculable’ if its values can be found by some purely mechanical process. Although it is fairly easy to get an intuitive grasp of this idea, it is nevertheless desirable to have some more definite, mathematically expressible definition. Such a definition was first given by Gödel at Princeton in 1934… These functions were described as ‘general recursive’ by Gödel… Another definition of effective calculability has been given by Church… who identifies it with lambda-definability. The author [i.e. Turing] has recently suggested a definition corresponding more closely to the intuitive idea… It was stated above that ‘a function is effectively calculable if its values can be found by a purely mechanical process.’ We may take this statement literally, understanding by a purely mechanical process one which could be carried out by a machine. It is possible to give a mathematical description, in a certain normal form, of the structures of these machines. The development of these ideas leads to the author's definition of a computable function, and to an identification of computability with effective calculability. It is not difficult, though somewhat laborious, to prove that these three definitions are equivalent.

Church accepted that Turing's definition gave a compelling, intuitive reason for why Church's thesis was true. The recent exposition by Davis (2000) emphasises that Gödel also was convinced by Turing's argument that an absolute concept had been identified (Gödel 1946). The situation has not changed since 1937. (For further comment, see the article on the Church-Turing Thesis. The recent selection of Turing's papers edited by Copeland (2004), and the review of Hodges (2006), continue this discussion.)

Turing himself did little to evangelise his formulation in the world of mathematical logic and early computer science. The textbooks of Davis (1958) and Minsky (1967) did more. Nowadays Turing computability is often reformulated (e.g. in terms of ‘register machines’). However, computer simulations (e.g., Turing's World, from Stanford) have brought Turing's original imagery to life.

Turing's work also opened new areas for decidability questions within pure mathematics. From the 1970s, Turing machines also took on new life in the development of *complexity theory,* and as such underpin one of the most important research areas in computer science. This development exemplifies the lasting value of Turing's special quality of giving concrete illustration to abstract concepts.

3. The Logical and the Physical

As put by Gandy (1988), Turing's paper was ‘a paradigm of philosophical analysis,’ refining a vague notion into a precise definition. But it was more than being an analysis *within* the world of mathematical logic: in Turing's thought the question that constantly recurs both theoretically and practically is the relationship of the logical Turing machine to the physical world.

‘Effective’ means *doing,* not merely imagining or postulating. At this stage neither Turing nor any other logician made a serious investigation into the physics of such ‘doing.’ But Turing's image of a teleprinter-like machine does inescapably refer to something that could actually be physically ‘done.’ His concept is a distillation of the idea that one can only ‘do’ one simple action, or finite number of simple actions, at a time. How ‘physical’ a concept is it?

The tape never holds more than a finite number of marked squares at any point in a computation. Thus it can be thought of as being finite, but always capable of further extension as required. Obviously this unbounded extendibility is unphysical, but the definition is still of practical use: it means that anything done on a finite tape, however large, is computable. (Turing himself took such a finitistic approach when explaining the practical relevance of computability in his 1950 paper.) One aspect of Turing's formulation, however, involves absolute finiteness: the table of behaviour of a Turing machine must be finite, since Turing allows only a finite number of ‘configurations’ of a Turing machine, and only a finite repertoire of symbols which can be marked on the tape. This is essentially equivalent to allowing only computer programs with finite lengths of code.

‘Calculable by finite means’ was Turing's characterisation of computability, which he justified with the argument that ‘the human memory is necessarily limited.’ (Turing 1936, p. 231). The whole point of his definition lies in encoding infinite potential effects, (e.g. the printing of an infinite decimal) into finite ‘tables of behaviour’. There would be no point in allowing machines with infinite ‘tables of behaviour’. It is obvious, for instance, that any real number could be printed by such a ‘machine’, by letting the Nth configuration be ‘programmed’ to print the Nth digit, for example. Such a ‘machine’ could likewise store any countable number of statements about all possible mathematical expressions, and so make the *Entscheidungsproblem* trivial.

Church (1937), when reviewing Turing's paper while Turing was in Princeton under his supervision, actually gave a bolder characterisation of the Turing machine as an *arbitrary finite machine.*

> The author [i.e. Turing] proposes as a criterion that an infinite sequence of digits 0 and 1 be “computable” that it shall be possible to devise a computing machine, occupying a finite space and with working parts of finite size, which will write down the sequence to any desired number of terms if allowed to run for a sufficiently long time. As a matter of convenience, certain further restrictions are imposed on the character of the machine, but these are of such a nature as obviously to cause no loss of generality—in particular, a human calculator, provided with pencil and paper and explicit instructions, can be regarded as a kind of Turing machine.

Church (1940) repeated this characterisation. Turing neither endorsed it nor said anything to contradict it, leaving the general concept of ‘machine’ itself undefined. The work of Gandy (1980) did more to justify this characterisation, by refining the statement of what is meant by ‘a machine.’ His results support Church's statement; they also argue strongly for the view that natural attempts to extend the notion of computability lead to trivialisation: if Gandy's conditions on a ‘machine’ are significantly weakened then every real number becomes calculable (Gandy 1980, p. 130ff.). (For a different interpretation of Church's statement, see the article on theChurch-Turing Thesis.)

Turing did not explicitly discuss the question of the *speed* of his elementary actions. It is left implicit in his discussion, by his use of the word ‘never,’ that it is not possible for infinitely many steps to be performed in a finite time. Others have explored the effect of abandoning this restriction. Davies (2001), for instance, describes a ‘machine’ with an infinite number of parts, requiring components of arbitrarily small size, running at arbitrarily high speeds. Such a ‘machine’ could perform uncomputable tasks. Davies emphasises that such a machine cannot be built in our own physical world, but argues that it could be constructed in a universe with different physics. To the extent that it rules out such ‘machines’, the Church-Turing thesis must have at least some physical content.

True physics is quantum-mechanical, and this implies a different idea of matter and action from Turing's purely classical picture. It is perhaps odd that Turing did not point this out in this period, since he was well versed in quantum physics. Instead, the analysis and practical development of quantum computing was left to the 1980s. Quantum computation, using the evolution of wave-functions rather than classical machine states, is the most important way in which Turing machine model has been challenged. The standard formulation of quantum computing (Deutsch 1985, following Feynman 1982) does not predict anything beyond computable effects, although within the realm of the computable, quantum computations may be very much more efficient than classical computations. It is possible that a deeper understanding of quantum mechanical physics may further change the picture of what can be physically ‘done.’

4. The Uncomputable

Turing turned to the exploration of the *uncomputable* for his Princeton Ph.D. thesis (1938), which then appeared as *Systems of Logic based on Ordinals* (Turing 1939).

It is generally the view, as expressed by Feferman (1988), that this work was a diversion from the main thrust of his work. But from another angle, as expressed in (Hodges 1997), one can see Turing's development as turning naturally from considering the mind when following a rule, to the action of the mind when *not* following a rule. In particular this 1938 work considered the mind when seeing the truth of one of Gödel's true but formally unprovable propositions, and hence going beyond rules based on the axioms of the system. As Turing expressed it (Turing 1939, p. 198), there are 'formulae, seen intuitively to be correct, but which the Gödel theorem shows are unprovable in the original system.' Turing's theory of 'ordinal logics' was an attempt to 'avoid as far as possible the effects of Gödel's theorem' by studying the effect of adding Gödel sentences as new axioms to create stronger and stronger logics. It did not reach a definitive conclusion.

In his investigation, Turing introduced the idea of an 'oracle' capable of performing, as if by magic, an uncomputable operation. Turing's oracle cannot be considered as some 'black box' component of a new class of machines, to be put on a par with the primitive operations of reading single symbols, as has been suggested by (Copeland 1998). An oracle is *infinitely more powerful*than anything a modern computer can do, and nothing like an elementary component of a computer. Turing defined 'oracle-machines' as Turing machines with an additional configuration in which they 'call the oracle' so as to take an uncomputable step. But these oracle-machines are*not purely mechanical.* They are only partially mechanical, like Turing's choice-machines. Indeed the *whole point* of the oracle-machine is to explore the realm of what *cannot* be done by purely mechanical processes. Turing emphasised (Turing 1939, p. 173):

> We shall not go any further into the nature of this oracle apart from saying that it cannot be a machine.

Turing's oracle can be seen simply as a mathematical tool, useful for exploring the mathematics of the uncomputable. The idea of an oracle allows the formulation of questions of *relative* rather than absolute computability. Thus Turing opened new fields of investigation in mathematical logic. However, there is also a possible interpretation in terms of human cognitive capacity. On this interpretation, the oracle is related to the 'intuition' involved in seeing the truth of a Gödel statement. M. H. A. Newman, who introduced Turing to mathematical logic and continued to collaborate with him, wrote in (Newman 1955) that the oracle resembles a mathematician 'having an idea', as opposed to using a mechanical method. However, Turing's oracle cannot actually be*identified* with a human mental faculty. It is too powerful: it immediately supplies the answer as to whether any given Turing machine is 'satisfactory,' something no human being could do. On the other hand, anyone hoping to see mental 'intuition' captured completely by an oracle, must face the difficulty that Turing showed how his argument for the incompleteness of Turing machines could be applied with equal force to oracle-machines (Turing 1939, p. 173). This point has been emphasised by Penrose (1994, p. 380). Newman's comment might better be taken to refer to the different oracle suggested later on (Turing 1939, p. 200), which has the property of recognising 'ordinal formulae.' One can only safely say that Turing's interest at this time in uncomputable operations appears in the *general setting* of studying the mental 'intuition' of truths which are not established by following mechanical processes (Turing 1939, p. 214ff.).

In Turing's presentation, intuition is in practice present in every part of a mathematician's thought, but when mathematical proof is formalised, intuition has an explicit manifestation in those steps where the mathematician sees the truth of a formally unprovable statement. Turing did not offer any suggestion as to what he considered the brain was physically doing in a moment of such ‘intuition’; indeed the word ‘brain’ did not appear in his writing in this era. This question is of interest because of the views of Penrose (1989, 1990, 1994, 1996) on just this issue: Penrose holds that the ability of the mind to see formally unprovable truths shows that there must be uncomputable physical operations in the brain. It should be noted that there is widespread disagreement about whether the human mind is really seeing the truth of a Gödel sentence; see for instance the discussion in (Penrose 1990) and the reviews following it. However Turing's writing at this period accepted without criticism the concept of intuitive recognition of the truth.

It was also at this period that Turing met Wittgenstein, and there is a full record of their 1939 discussions on the foundations of mathematics in (Diamond 1976). To the disappointment of many, there is no record of any discussions between them, verbal or written, on the problem of Mind.

In 1939 Turing's various energetic investigations were broken off for war work. This did, however, have the positive feature of leading Turing to turn his universal machine into the practical form of the modern digital computer.

5. Building a Universal Machine

When apprised in 1936 of Turing's idea for a universal machine, Turing's contemporary and friend, the economist David Champernowne, reacted by saying that such a thing was impractical; it would need ‘the Albert Hall.’ If built from relays as then employed in telephone exchanges, that might indeed have been so, and Turing made no attempt at it. However, in 1937 Turing did work with relays on a smaller machine with a special cryptological function (Hodges 1983, p. 138). World history then led Turing to his unique role in the Enigma problem, to his becoming the chief figure in the mechanisation of logical procedures, and to his being introduced to ever faster and more ambitious technology as the war continued.

After 1942, Turing learnt that electronic components offered the speed, storage capacity and logical functions required to be effective as ‘tapes’ and instruction tables. So from 1945, Turing tried to use electronics to turn his universal machine into practical reality. Turing rapidly composed a detailed plan for a modern stored-program computer: that is, a computer in which data and instructions are stored and manipulated alike. Turing's ideas led the field, although his report of 1946 postdated von Neumann's more famous EDVAC report (von Neumann 1945). It can however be argued, as does Davis (2000), that von Neumann gained his fundamental insight into the computer through his pre-war familiarity with Turing's logical work. At the time, however, these basic principles were not much discussed. The difficulty of engineering the electronic hardware dominated everything.

It therefore escaped observers that Turing was ahead of von Neumann and everyone else on the future of software, or as he called it, the ‘construction of instruction tables.’ Turing (1946) foresaw at once:

> Instruction tables will have to be made up by mathematicians with computing experiences and perhaps a certain puzzle-solving ability. There will probably be a great deal of work to be done, for every known process has got to be translated into instruction table form at some stage.
>
> The process of constructing instruction tables should be very fascinating. There need be no real danger of it ever becoming a drudge, for any processes that are quite mechanical may be turned over to the machine itself.

These remarks, reflecting the universality of the computer, and its ability to manipulate its own instructions, correctly described the future trajectory of the computer industry. However, Turing had in mind something greater: ‘building a brain.’

6. Building a Brain

The provocative words ‘building a brain’ from the outset announced the relationship of Turing's technical computer engineering to a philosophy of Mind. Even in 1936, Turing had given an interpretation of computability in terms of ‘states of mind’. His war work had shown the astounding power of the computable in mechanising expert human procedures and judgments. From 1941 onwards, Turing had also discussed the mechanisation of chess-playing and other ‘intelligent’ activities with his colleagues at Bletchley Park (Hodges 1983, p. 213). But more profoundly, it appears that Turing emerged in 1945 with a conviction that computable operations were sufficient to embrace *all* mental functions performed by the brain. As will become clear from the ensuing discussion, the uncomputable ‘intuition’ of 1938 disappeared from Turing's thought, and was replaced by new ideas all lying within the realm of the computable. This change shows even in the technical prospectus of (Turing 1946), where Turing referred to the possibility of making a machine calculate chess moves, and then continued:

> This … raises the question ‘Can a machine play chess?’ It could fairly easily be made to play a rather bad game. It would be bad because chess requires intelligence. We stated … that 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.

The puzzling reference to ‘mistakes’ is made clear by a talk Turing gave a year later (Turing 1947), in which the issue of mistakes is linked to the issue of the significance of seeing the truth of formally unprovable statements.

> …I would say that fair play must be given to the machine. Instead of it 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… 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.

Turing's post-war view was that mathematicians make mistakes, and so do not in fact see the truth infallibly. Once the possibility of mistakes is admitted, Gödel's theorem become irrelevant. Mathematicians and computers alike apply computable processes to the problem of judging the correctness of assertions; both will therefore sometimes err, since seeing the truth is known not to be a computable operation, but there is no reason why the computer need do worse than the mathematician. This argument is still very much alive. For instance, Davis (2000) endorses Turing's view and attacks Penrose (1989, 1990, 1994, 1996) who argues against the significance of human error on the grounds of a Platonist account of mathematics.

Turing also pursued more constructively the question of how computers could be made to perform operations which did not appear to be 'mechanical' (to use common parlance). His guiding principle was that it should be possible to simulate the operation of human brains. In an unpublished report (Turing 1948), Turing explained that the question was that of how to simulate 'initiative' in addition to 'discipline'—comparable to the need for 'intuition' as well as mechanical ingenuity expressed in his pre-war work. He announced ideas for how to achieve this: he thought 'initiative' could arise from systems where the algorithm applied is not consciously designed, but is arrived at by some other means. Thus, he now seemed to think that the mind when*not* actually following any conscious rule or plan, was nevertheless carrying out some computable process.

He suggested a range of ideas for systems which could be said to modify their own programs. These ideas included nets of logical components ('unorganised machines') whose properties could be 'trained' into a desired function. Thus, as expressed by (Ince 1989), he predicted neural networks. However, Turing's nets did not have the 'layered' structure of the neural networks that were to be developed from the 1950s onwards. By the expression 'genetical or evolutionary search', he also anticipated the 'genetic algorithms' which since the late 1980s have been developed as a less closely structured approach to self-modifying programs. Turing's proposals were not well developed in 1948, and at a time when electronic computers were only barely in operation, could not have been. Fresh attention to them has been drawn by Copeland and Proudfoot (1996), and they have now have been tried out (Teuscher 2001).

It is important to note that Turing identified his prototype neural networks and genetic algorithms as *computable*. This has to be emphasised since the word 'nonalgorithmic' is often now confusingly employed for computer operations that are not explicitly planned. Indeed, his ambition was explicit: he himself wanted to implement them as programs on a computer. Using the term Universal Practical Computing Machine for what is now called a digital computer, he wrote in (Turing 1948):

> It should be easy to make a model of any particular machine that one wishes to work on within such a UPCM instead of having to work with a paper machine as at present. If one also decided on quite definite 'teaching policies' these could also be programmed into the machine. One would then allow the whole system to run for an appreciable

> period, and then break in as a kind of ‘inspector of schools’ and see what progress had been made. One might also be able to make some progress with unorganised machines…

The upshot of this line of thought is that all mental operations are *computable* and hence realisable on a universal machine: the computer. Turing advanced this view with increasing confidence in the late 1940s, perfectly aware that it represented what he enjoyed calling ‘heresy’ to the believers in minds or souls beyond material description.

Turing was not a mechanical thinker, or a stickler for convention; far from it. Of all people, he knew the nature of originality and individual independence. Even in tackling the U-boat Enigma problem, for instance, he declared that he did so because no-one else was looking at it and he could have it to himself. Far from being trained or organised into this problem, he took it on despite the prevailing wisdom in 1939 that it was too difficult to attempt. His arrival at a thesis of ‘machine intelligence’ was not the outcome of some dull or restricted mentality, or a lack of appreciation of individual human creativity.

7. Machine Intelligence

Turing relished the paradox of ‘Machine Intelligence’: an apparent contradiction in terms. It is likely that he was already savouring this theme in 1941, when he read a theological book by the author Dorothy Sayers (Sayers 1941). In (Turing 1948) he quoted from this work to illustrate his full awareness that in common parlance ‘mechanical’ was used to to mean ‘devoid of intelligence.’ Giving a date which no doubt had his highly sophisticated Enigma-breaking machines secretly in mind, he wrote that ‘up to 1940’ only very limited machinery had been used, and this ‘encouraged the belief that machinery was necessarily limited to extremely straightforward, possibly even to repetitious, jobs.’ His object was to dispel these connotations.

In 1950, Turing wrote on the first page of his Manual for users of the Manchester University computer (Turing 1950a):

Electronic computers are intended to carry out any definite rule of thumb process which could have been done by a human operator working in a disciplined but unintelligent manner.

This is, of course, just the 1936 universal Turing machine, now in electronic form. On the other hand, he also wrote in the more famous paper of that year (Turing 1950b, p. 460)

We may hope that machines will eventually compete with men in all purely intellectual fields.

How could the *intelligent* arise from operations which were themselves totally *routine and mindless*—‘entirely without intelligence’? This is the core of the problem Turing faced, and the same problem faces Artificial Intelligence research today. Turing's underlying argument was that the human brain must somehow be organised for intelligence, and that the organisation of the brain must be realisable as a finite discrete-state machine. The implications of this view were exposed to a wider circle in his famous paper, “Computing Machinery and Intelligence,” which appeared in *Mind* in October 1950.

The appearance of this paper, Turing's first foray into a journal of philosophy, was stimulated by his discussions at Manchester University with Michael Polanyi. It also reflects the general sympathy of Gilbert Ryle, editor of *Mind*, with Turing's point of view.

Turing's 1950 paper was intended for a wide readership, and its fresh and direct approach has made it one of the most frequently cited and republished papers in modern philosophical literature. Not surprisingly, the paper has attracted many critiques. Not all commentators note the careful explication of computability which opens the paper, with an emphasis on the concept of the universal machine. This explains why if mental function can be achieved by any finite discrete state machine, then the same effect can be achieved by programming a computer (Turing 1950b, p. 442). (Note, however, that Turing makes no claim that the nervous system should resemble a digital computer in its structure.) Turing's treatment has a severely finitistic flavour: his argument is that the relevant action of the brain is not only computable, but realisable as a totally finite machine, i.e. as a Turing machine that does not use any ‘tape’ at all. In his account, the full range of computable functions, defined in terms of Turing machines that use an infinite tape, only appears as being of ‘special theoretical interest.’ (Of uncomputable functions there is, *a fortiori*,no mention.) Turing uses the finiteness of the nervous system to give an estimate of about 10^9bits of storage required for a limited simulation of intelligence (Turing 1950b, p. 455).

The wit and drama of Turing's ‘imitation game’ has attracted more fame than his careful groundwork. Turing's argument was designed to bypass discussions of the nature of thought, mind, and consciousness, and to give a criterion in terms of external observation alone. His justification for this was that one only judges that other human beings are thinking by external observation, and he applied a principle of ‘fair play for machines’ to argue that the same should hold for machine intelligence. He dramatised this viewpoint by a thought-experiment (which nowadays can readily be tried out). A human being and a programmed computer compete to convince an impartial judge, using textual messages alone, as to which is the human being. If the computer wins, it must be credited with intelligence.

Turing introduced his ‘game’ confusingly with a poor analogy: a party game in which a man pretends to be a woman. His loose wording (Turing 1950b, p. 434) has led some writers wrongly to suppose that Turing proposed an ‘imitation game’ in which a machine has to imitate a man imitating a woman. Others, like Lassègue (1998), place much weight on this game of gender pretence and its real or imaginary connotations. In fact, the whole point of the ‘test’ setting, with its remote text-message link, was to *separate* intelligence from other human faculties and properties. But it may fairly be said that this confusion reflects Turing's richly ambitious concept of what is involved in human ‘intelligence’. It might also be said to illustrate his own human intelligence, in particular a delight in the Wildean reversal of roles, perhaps reflecting, as in Wilde, his homosexual identity. His friends knew an Alan Turing in whom intelligence, humour and sex were often intermingled.

Turing was in fact sensitive to the difficulty of separating ‘intelligence’ from other aspects of human senses and actions; he described ideas for robots with sensory attachments and raised questions as to whether they might enjoy strawberries and cream or feel racial kinship. In contrast, he paid scant attention to the questions of authenticity and deception implicit in his test, essentially because he wished to by-pass questions about the reality of consciousness. A subtle aspect of one of his imagined ‘intelligent’ conversations (Turing 1950b, p. 434) is where the computer imitates human intelligence by giving the *wrong answer* to a simple arithmetic problem. But in Turing's setting we are not supposed to ask whether the computer ‘consciously’ deceives by giving the impression of innumerate humanity, nor why it should wish to do so. There is a certain lack of seriousness in this approach. Turing took on a second-rank target in countering the published views of the brain surgeon G. Jefferson, as regards the objectivity of consciousness. Wittgenstein's views on Mind would have made a more serious point of departure.

Turing's imitation principle perhaps also assumes (like ‘intelligence tests’ of that epoch) too much of a shared language and culture for his imagined interrogations. Neither does it address the possibility that there may be kinds of thought, by animals or extra-terrestrial intelligences, which are not amenable to communication.

A more positive feature of the paper lies in its constructive program for research, culminating in Turing's ideas for ‘learning machines’ and educating ‘child’ machines (Turing 1950b, p. 454). It is generally thought (e.g. in Dreyfus and Dreyfus 1990) that there was always an antagonism between programming and the ‘connectionist’ approach of neural networks. But Turing never expressed such a dichotomy, writing that both approaches should be tried. Donald Michie, the British AI research pioneer profoundly influenced by early discussions with Turing, has called this suggestion ‘Alan Turing's Buried Treasure’, in an allusion to a bizarre wartime episode in which Michie was himself involved (Hodges 1983, p. 345). The question is still highly pertinent.

It is also a commonly expressed view that Artificial Intelligence ideas only occurred to pioneers in the 1950s *after* the success of computers in large arithmetical calculations. It is hard to see why Turing's work, which was rooted from the outset in the question of mechanising Mind, has been so much overlooked. But through his failure to publish and promote work such as that in (Turing 1948) he largely lost recognition and influence.

It is also curious that Turing's best-known paper should appear in a journal of philosophy, for it may well be said that Turing, always committed to materialist explanation, was not really a philosopher at all. Turing was a mathematician, and what he had to offer philosophy lay in illuminating its field with what had been discovered in mathematics and physics. In the 1950 paper this was surprisingly cursory, apart from his groundwork on the concept of computability. His emphasis on the sufficiency of the computable to explain the action of the mind was stated more as a hypothesis, even a manifesto, than argued in detail. Of his hypothesis he wrote (Turing 1950b, p. 442):

> …I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted. I believe further that no useful purpose is served by concealing these beliefs. The popular view that scientists proceed inexorably from established fact to established fact, never being influenced by any unproved conjecture, is quite mistaken. Provided it is made clear which are proved facts and which are conjecture, no harm can result. Conjectures are of great importance since they suggest useful lines of research.

Penrose (1994, p.21), probing into Turing's conjecture, has presented it as ‘Turing's thesis’ thus:

> It seems likely that he viewed physical action in general—which would include the action of a human brain—to be always reducible to some kind of Turing-machine action.

The statement that all physical action is in effect computable goes beyond Turing's explicit words, but is a fair characterisation of the implicit assumptions behind the 1950 paper. Turing's consideration of ‘The Argument from Continuity in the Nervous System,’ in particular, simply asserts that the physical system of the brain can be approximated as closely as is desired by a computer program (Turing 1950b, p. 451). Certainly there is nothing in Turing's work in the 1945–50 period to contradict Penrose's interpretation. The more technical precursor papers (Turing 1947, 1948) include wide-ranging comments on physical processes, but make no reference to the possibility of physical effects being uncomputable.

In particular, a section of (Turing 1948) is devoted to a general classification of ‘machines.’ The period between 1937 and 1948 had given Turing much more experience of actual machinery than he had in 1936, and his post-war remarks reflected this in a down-to-earth manner. Turing distinguished ‘controlling’ from ‘active’ machinery, the latter being illustrated by ‘a bulldozer’. Naturally it is the former—in modern terms ‘information-based machinery’—with which Turing's analysis is concerned. It is noteworthy that in 1948 as in 1936, despite his knowledge of physics, Turing made no mention of how quantum mechanics might affect the concept of ‘controlling’. His concept of ‘controlling’ remained entirely within the classical framework of the Turing machine (which he called a Logical Computing Machine in this paper.)

The same section of (Turing 1948) also drew the distinction between *discrete* and *continuous*machinery, illustrating the latter with ‘the telephone’ as a continuous, controlling machine. He made light of the difficulty of reducing continuous physics to the discrete model of the Turing machine, and though citing ‘the brain’ as a continuous machine, stated that it could probably be treated as if discrete. He gave no indication that physical continuity threatened the paramount role of computability. In fact, his thrust in (Turing 1947) was to promote the digital computer as *more powerful* than analog machines such as the differential analyser. When he discussed this comparison, he gave the following informal version of the Church-Turing thesis:

> One of my conclusions 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 [i.e. Turing machines]

Turing gave no hint that the discreteness of the Turing machine constituted a real limitation, or that the non-discrete processes of analog machines might be of any deep significance.

Turing also introduced the idea of ‘random elements’ but his examples (using the digits of π) showed that he considered *pseudo-random* sequences (i.e. computable sequences with suitable ‘random’ properties) quite adequate for his discussion. He made no suggestion that randomness implied something uncomputable, and indeed gave no definition of the term ‘random’. This is perhaps surprising in view of the fact that his work in pure mathematics, logic and cryptography all gave him considerable motivation to approach this question at a serious level.

8. Unfinished Work

From 1950 Turing worked on a new mathematical theory of morphogenesis, based on showing the consequences of non-linear equations for chemical reaction and diffusion (Turing 1952). He was a pioneer in using a computer for such work. Some writers have referred to this theory as founding Artificial Life (A-life), but this is a misleading description, apt only to the extent that the theory was intended, as Turing saw it, to counter the Argument from Design. A-life since the 1980s has concerned itself with using computers to explore the logical consequences of evolutionary theory without worrying about specific physiological forms. Morphogenesis is complementary, being concerned to show which physiological pathways are feasible for evolution to exploit. Turing's work was developed by others in the 1970s and is now regarded as central to this field.

It may well be that Turing's interest in morphogenesis went back to a primordial childhood wonder at the appearance of plants and flowers. But in another late development, Turing went back to other stimuli of his youth. For in 1951 Turing did consider the problem, hitherto avoided, of setting computability in the context of quantum-mechanical physics. In a BBC radio talk of that year (Turing 1951) he discussed the basic groundwork of his 1950 paper, but this time dealing rather less certainly with the argument from Gödel's theorem, and this time also referring to the quantum-mechanical physics underlying the brain. Turing described the universal machine property, applying it to the brain, but said that its applicability required that the machine whose behaviour is to be imitated

> …should be of the sort whose behaviour is in principle predictable by calculation. We certainly do not know how any such calculation should be done, and it was even argued by Sir Arthur Eddington that on account of the indeterminacy principle in quantum mechanics no such prediction is even theoretically possible.

Copeland (1999) has rightly drawn attention to this sentence in his preface to his edition of the 1951 talk. However, Copeland's critical context suggests some connection with Turing's ‘oracle.’ There is is in fact no mention of oracles here (nor anywhere in Turing's post-war discussion of mind and machine.) Turing here is discussing the possibility that, when seen as as a *quantum-mechanical machine* rather than a classical machine, the Turing machine model is inadequate. The correct connection to draw is not with Turing's 1938 work on ordinal logics, but with his knowledge of quantum mechanics from Eddington and von Neumann in his youth. Indeed, in an early speculation, influenced by Eddington, Turing had suggested that quantum mechanical physics could yield the basis of free-will (Hodges 1983, p. 63). Von Neumann's axioms of quantum mechanics involve two processes: unitary evolution of the wave function, which is predictable, and the measurement or reduction operation, which introduces unpredictability. Turing's reference to unpredictability must therefore refer to the reduction process. The essential difficulty is that still to this day there is no agreed or compelling theory of when or how reduction actually occurs. (It should be noted that ‘quantum computing,’ in the standard modern sense, is based on the predictability of the unitary evolution, and does not, as yet, go into the question of how reduction occurs.) It seems that this single sentence indicates the beginning of a new field of investigation for Turing, this time into the foundations of quantum mechanics. In 1953 Turing wrote to his friend and student Robin Gandy that he was ‘trying to invent a new Quantum Mechanics but it won't really work.’

At Turing's death in June 1954, Gandy reported in a letter to Newman on what he knew of Turing's current work (Gandy 1954). He wrote of Turing having discussed a problem in understanding the reduction process, in the form of

> …‘the Turing Paradox’; it is easy to show using standard theory that if a system start in an eigenstate of some observable, and measurements are made of that observable N times a second, then, even if the state is not a stationary one, the probability that the system will be in the same state after, say, 1 second, tends to one as N tends to infinity; i.e. that continual observation will prevent motion. Alan and I tackled one or two theoretical physicists with this, and they rather pooh-poohed it by saying that continual observation is not possible. But there is nothing in the standard books (e.g., Dirac's) to this effect, so that at least the paradox shows up an inadequacy of Quantum Theory as usually presented.

Turing's investigations take on added significance in view of the assertion of Penrose (1989, 1990, 1994, 1996) that the reduction process must involve something uncomputable. Probably Turing was aiming at the opposite idea, of finding a theory of the reduction process that would be predictive and computable, and so plug the gap in his hypothesis that the action of the brain is computable. However Turing and Penrose are alike in seeing this as an important question affecting the assumption that all mental action is computable; in this they both differ from the mainstream view in which the question is accorded little significance.

Alan Turing's last postcards to Robin Gandy, in March 1954, headed ‘Messages from the Unseen World’ in allusion to Eddington, hinted at new ideas in the fundamental physics of relativity and particle physics (Hodges 1983, p. 512). They illustrate the wealth of ideas with which he was concerned at that last point in his life, but which apart from these hints are entirely lost. A review of such lost ideas is given in (Hodges 2004), as part of a larger volume on Turing's legacy (Teuscher 2004).

9. Alan Turing: the Unknown Mind

It is a pity that Turing did not write more about his ethical philosophy and world outlook. As a student he was an admirer of Bernard Shaw's plays of ideas, and to friends would openly voice both the hilarities and frustrations of his many difficult situations. Yet the nearest he came to serious personal writing, apart from occasional comments in private letters, was in penning a short story about his 1952 crisis (Hodges 1983, p. 448). His last two years were particularly full of Shavian drama and Wildean irony. In one letter (to his friend Norman Routledge; the letter is now in the Turing Archive at King's College, Cambridge) he wrote:

Turing believes machines think

Turing lies with men

Therefore machines do not think

The syllogistic allusion to Socrates is unmistakeable, and his demise, with cyanide rather than hemlock, may have signalled something similar. A parallel figure in World War II, Robert Oppenheimer, suffered the loss of his reputation during the same week that Turing died. Both combined the purest scientific work and the most effective application of science in war. Alan Turing was even more directly on the receiving end of science, when his sexual mind was treated as a machine, against his protesting consciousness and will. But amidst all this human drama, he left little to say about what he really thought of himself and his relationship to the world of human events.

Alan Turing did not fit easily with any of the intellectual movements of his time, aesthetic, technocratic or marxist. In the 1950s, commentators struggled to find discreet words to categorise him: as ‘a scientific Shelley,’ as possessing great ‘moral integrity’. Until the 1970s, the reality of his life was unmentionable. He is still hard to place within twentieth-century thought. He exalted the science that according to existentialists had robbed life of meaning. The most original figure, the most insistent on personal freedom, he held originality and will to be susceptible to mechanisation. The mind of Alan Turing continues to be an enigma.

But it is an enigma to which the twenty-first century seems increasingly drawn. The year of his centenary, 2012, witnessed numerous conferences, publications, and cultural events in his honor. Some reasons for this explosion of interest are obvious. One is that the question of the power and limitations of computation now arises in virtually every sphere of human activity. Another is that issues of sexual orientation have taken on a new importance in modern democracies. More subtly, the interdisciplinary breadth of Turing's work is now better appreciated. A landmark of the centenary period was the publication of *Alan Turing, his work and impact* (eds. Cooper and van Leeuwen, 2013), which made available almost all aspects of Turing's scientific oeuvre, with a wealth of modern commentary. In this new climate, fresh attention has been paid to Turing's lesser-known work, and new light shed upon his achievements. He has emerged from obscurity to become one of the most intensely studied figures in modern science.

Bibliography

Selected Works by Turing

- 1936, ‘On computable numbers, with an application to the Entscheidungsproblem’, *Proc. London Maths. Soc.* (Series 2), 42: 230–265; also in Davis 1965 and Gandy and Yates 2001; [Available online].

- 1939, ‘Systems of logic defined by ordinals’, *Proc. Lond. Math. Soc.*, Ser. 2, 45: 161–228; This was Turing's Ph.D. thesis, Princeton University (1938), published as *Alan Turing's systems of logic: The Princeton thesis*, A. W. Appel (ed.), Princeton: Princeton University Press, 2012; also in

Davis 1965 and in Gandy and Yates 2001.

- 1946, *Proposed Electronic Calculator*, report for National Physical Laboratory, Teddington; published in *A. M. Turing's ACE report of 1946 and other papers*, B. E. Carpenter and R. W. Doran (eds.), Cambridge, Mass.: MIT Press, 1986; also in *Collected Works* (Volume 1).
- 1947, ‘Lecture to the London Mathematical Society on 20 February 1947’, in *A. M. Turing's ACE report of 1946 and other papers*, B. E. Carpenter and R. W. Doran (eds.), Cambridge, Mass.: MIT Press, 1986; also in *Collected Works* (Volume 1).
- 1948, ‘Intelligent Machinery’, report for National Physical Laboratory, in *Machine Intelligence 7*, B. Meltzer and D. Michie (eds.) 1969; also in *Collected Works* (Volume 1).
- 1950a, *Programmers' Handbook for the Manchester Electronic Computer*, Manchester University Computing Laboratory. [Available online in PDF].
- 1950b, ‘Computing machinery and intelligence’, *Mind*, 50: 433–460; also in Boden 1990,*Collected Works* (Volume 1), and [Available online].
- 1951, BBC radio talk, in *The Essential Turing*, B. J. Copeland (ed.), Oxford: Clarendon Press, 2004.
- 1952, ‘The chemical basis of morphogenesis’, *Phil. Trans. R. Soc. London B* 237: 37–72; also in*The Collected Works of A. M. Turing: Morphogenesis*, P. T. Saunders (ed.), Amsterdam: North-Holland, 1992.

The Collected Works of A. M. Turing consists of 4 volumes:

- Volume 1: *Mechanical Intelligence*, D.C. Ince (ed.), Amsterdam: North-Holland, 1992.
- Volume 2: *Morphogenesis*, P. T. Saunders (ed.), Amsterdam: North-Holland, 1992.
- Volume 3: *Pure Mathematics*, J. L. Britton (ed.), Amsterdam: North-Holland, 1992.
- Volume 4: *Mathematical Logic*, R. O. Gandy and C. E. M. Yates, Amsterdam: North-Holland, 2001.

The following, single-volume work contains much of the *Collected Works* and adds extensive modern commentary:

- *Alan Turing, his work and impact,* S. B. Cooper and J. van Leeuwen (eds.), Amsterdam: Elsevier, 2013.

Secondary Literature

- Boden, M. (ed.), 1990, *The Philosophy of Artificial Intelligence*, Oxford: Oxford University Press.
- Church, A., 1937, Review of Turing 1936–7, *Journal of Symbolic Logic*, 2: 42.

- ––––, 1940, ‘On the concept of a random sequence’, *Bull. Amer. Math. Soc.*, 46: 130–135.
- Copeland, B. J., 1998, ‘Turing's o-machines, Searle, Penrose and the brain’, *Analysis*, 58(2): 128–138.
- ––––, 1999, ‘A lecture and two radio broadcasts on machine intelligence by Alan Turing’, in*Machine Intelligence 15*, K. Furukawa, D. Michie, and S. Muggleton (eds.), Oxford: Oxford University Press.
- –––– (ed.), 2004, *The Essential Turing*, Oxford: Clarendon Press.
- Copeland, B. J. and D. Proudfoot, 1996, ‘On Alan Turing's anticipation of connectionism’,*Synthese*, 108: 361–377.
- Davies, E. B., 2001, ‘Building infinite machines’, *British Journal for the Philosophy of Science*, 52 (4): 671–682.
- Davis, M., 2000, *The Universal Computer*, New York: Norton.
- –––– (ed.), 1958, *Computability and Unsolvability*, New York: McGraw-Hill; New York: Dover (1982).
- –––– (ed.), 1965, *The Undecidable*, New York: Raven.
- Dawson, J. W., 1985, Review of Hodges (1983), *Journal of Symbolic Logic*, 50: 1065–1067.
- Deutsch, D., 1985, ‘Quantum theory, the Church-Turing principle and the universal quantum computer’, *Proc. Roy. Soc.* A, 400: 97–115.
- Diamond, C. (ed.), 1976, *Wittgenstein's Lectures on the Foundations of Mathematics, Cambridge, 1939*, Hassocks: Harvester Press.
- Dreyfus, H. L. and S. E. Dreyfus, 1990, ‘Making a mind versus modelling the brain: artificial intelligence back at a branch-point’, in Boden 1990.
- Feferman, S., 1988, ‘Turing in the Land of O(Z)’, in (Herken 1988); also in Gandy and Yates (eds.) 2001.
- Feynman, R. P., 1982, ‘Simulating physics with computers’, *Int. Journal of Theoretical Physics*, 21: 467–488.
- Gandy, R. O., 1954, Letter to M. H. A. Newman, in Gandy and Yates, 2001.
- ––––, 1980, ‘Principles of Mechanisms’, in *The Kleene Symposium*, J. Barwise, H. J. Keisler and K. Kunen (eds.), Amsterdam: North-Holland.
- ––––, 1988, ‘The confluence of ideas in 1936’, in Herken 1988.
- Gandy, R. O. and C. E. M. Yates (eds.), 2001, *The Collected Works of A M. Turing: Mathematical Logic*, Amsterdam: North-Holland.

- Gödel, K., 1946, ‘Remarks before the Princeton Bicentennial Conference on problems in mathematics’, in Davis 1965.

- Herken R., (ed.), 1988, *The Universal Turing Machine: A Half-Century Survey*, Berlin: Kammerer und Unverzagt; Oxford: Oxford University Press.

- Hodges, A., 1983, *Alan Turing: the Enigma*, London: Burnett; New York: Simon & Schuster; London: Vintage, 1992, 2012; Princeton University Press, 2012.

- –––, 1997, *Turing, a natural philosopher*, London: Phoenix; New York: Routledge (1999); included in *The great philosophers*, R. Monk and F. Raphael (eds.), London: Weidenfeld and Nicolson (2000).

- –––, 2004, What would Alan Turing have done after 1954, in *Alan Turing: life and legacy of a great thinker*, C. Teuscher (ed.), Berlin: Springer Verlag.

- –––, 2006, Review of Copeland 2004, *Notices of the American Mathematical Society*, 53: 1190–1199.

- Ince, D. C., 1989, Preface to Turing 1948, in Ince (ed.) 1992.

- Lassègue, J., 1998, *Turing*, Paris: les Belles Lettres.

- Minsky, M. L., 1967, *Computation: Finite and Infinite Machines*, Englewood Cliffs, N.J.: Prentice-Hall.

- Newman, M. H. A., 1955, ‘Alan Mathison Turing’, *Biographical memoirs of the Royal Society (1955)*, 253–263.

- Penrose, R., 1989, *The Emperor's New Mind*, Oxford: Oxford University Press.

- –––, 1990, Précis of *The Emperor's New Mind, Behavioral and Brain Sciences*, 13: 643–655.

- –––, 1994, *Shadows of the Mind*, Oxford: Oxford University Press.

- ––– 1996, ‘Beyond the doubting of a shadow: A Reply to Commentaries on *Shadows of the Mind*’, in *Psyche: An Interdisciplinary Journal of Research on Consciousness*, Volume 2.

- Sayers, D., 1941, *The Mind of the Maker*, London: Methuen.

- Teuscher, C., 2001, *Turing's Connectionism*, London: Springer-Verlag UK.

- ––– (ed.), 2004, *Alan Turing: Life and Legacy of a Great Thinker*, Berlin: Springer-Verlag.

- Turing, E. S., 1959, *Alan M. Turing*, Cambridge: Heffers; republished by Cambridge University Press, 2012.

- von Neumann, J., 1945, ‘First draft of a report on the EDVAC’, University of Pennsylvania; first printed in N. Stern, *From Eniac to Univac: an appraisal of the Eckert-Mauchly machines*, Bedford MA: Digital Press, 1981.

- Whitemore, H., 1986, *Breaking the Code*, London: S. French.