Part XIII

LOGIC, MACHINE THEORY, AND COGNITIVE SCIENCE

41

The Logical and the Physical

ANDREW W. HODGES

This chapter on logic, machine theory and cognitive science will focus on the work of A. M. Turing (1912–54), who first related these topics, and whose ideas still dominate the interconnections after 50 years. In particular Turing pioneered the links between abstract logic and physical mechanisms. I shall here deal largely with a misconception which has recently found a wide circulation, but in the process I hope to shed light on Turing's foundations and also suggest a more positive point of new interest.

I will start with the scene in Bletchley Park, Buckinghamshire, England, in 1941. There Alan Turing is in charge of breaking the U-boat messages, as enciphered by the now-famous Enigma in its most fiendish variation. One of the many riddles wrapped inside the Enigma was that nine different code books were used to encipher what would now be called the session key. As revealed by his colleague I. J. Good (2000), Turing had devised a sophisticated statistical method for guessing which book had been used. The results of this work were then fed into the "Banburismus" method (Good, in Hinsley and Stripp 1993), based on giving numerical scores to the weight of evidence from various coincidences of letters, giving a scientific replacement to intuitive guessing. The actual counting of letter coincidences was done by WRNS servicewomen holding up punched paper tapes.

This scene, reminiscent of Searle's Chinese Room where people act out algorithms, is where Alan Turing was at a time when (as described in Hodges 1983: 214), he was first pondering questions of artificial intelligence through chess-playing algorithms. In 1941 he was also reading a book (Sayers 1941) by the detective novelist and religious writer Dorothy L. Sayers with reflections on determinism and creativity which, it seems, struck a peculiar chord in him. The question he apparently asked, judging by the comments on this book in Turing (1948), was whether these concepts were in fact exclusive as Dorothy L. Sayers supposed. Could not something purely *mechanical* nevertheless exhibit the features of *originality*? Perhaps there was a special meaning for Turing, whose mathematical methods were replacing the inspired guesswork of pre-scientific codebreaking. But to see the full point of this question, we must return to an earlier year. Alan Turing had made his name in 1936 with the concept of 'a machine', and we must see what he meant by it.

Turing's great paper "On Computable Numbers, with an application to the *Entscheidungsproblem*" (Turing 1936–7) was, as its awkward title said, intended to give a definite meaning to, and hence derive a definite conclusion to, Hilbert's *Entscheidungsproblem*. To do this, Turing had to characterize the most general process that a human mathematician might carry out as a definite method. This the Turing machine definition supplied. Church (1937a) characterized it thus: "a computing machine, occupying a finite space and with working parts of finite size." More technically, a Turing machine is limited to finitely many configurations (or states) and finitely many types of symbols, with only finitely many symbols written on a tape at any time. The extent of the tape used however, is unlimited. Thus the Turing machine embodies finite means, but unlimited time and space to work out their consequences.

This finiteness deserves attention, and an interesting aspect of Turing (1936–7) is his claim that we may assume only finitely many states of mind. Indeed it is a striking claim that unobservable 'states of mind' should be countable, let alone finite. The philosopher B. J. Copeland has contested the signficance of this finiteness restriction (Copeland 1992: 280), suggesting that Turing said a Turing machine could simulate a device with infinitely many states. But a 'machine' with infinitely many states could encode the answers to every mathematical question, thus rendering trivial the very problem that the 'machine' concept was intended to settle. Thus the finite-state restriction is *crucial*, as is the restriction to a finite alphabet of symbols. (The same goes for allowing infinitely many symbols to be printed on the tape, or roll of toilet paper as Copeland (1992) vividly describes it, at a finite time.)

Did Turing ever consider infinite-state machines? 'Calculable by finite means' is how Turing characterizes the mechanical, and this rules out an infinite-state machine, with infinite means. The *very significance* of algorithms is that they encode potentially *infinite* outputs by *finite* specifications. Extending them to allow drawing on an infinite store of data would miss the point of what 'calculation' involves.

In mathematics, Turing's non-trivial discovery was that defining a real number in a finite number of words is not the same as being able to calculate it effectively. Turing's work has great significance outside mathematics. In computer science, the vital thing is that Turing's universal machine and its mode of operation can be implemented in electronics. In cognitive science, Turing's interpretation of states of mind developed into a thesis of the computability of mental operations. But in 1936–9, Turing expressed his work mainly in terms of how it affected Church's thesis, which as a result is now often described as the Church–Turing thesis: that anything effectively calculable is computable by Turing machines.

Turing gave this statement a definitive form in his Ph.D. thesis, submitted at Princeton in 1938, his supervisor being Church. This was later published as Turing (1939). Turing's formulation was:

> 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 λ-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. (166)

Thus, as Church and Gödel endorsed, the Turing machine definition, while equivalent in mathematical scope to lambda-calculus and recursive function theory, offers a convincing argument for why Church's thesis should be accepted. Turing's paper-tape definition also suggests, in a manner hard to define precisely, operations that can physically be *done*. The very word 'effect' means to *do* as opposed to postulate. Hence people sometimes distinguish Turing's thesis from Church's, though Turing himself never did this, and I shall continue to refer to this 1938 statement as the Church–Turing thesis.

But what did Turing mean by 'mechanical' or 'machine'? It is noteworthy that Turing does not make any qualifications; he does not say "carried out by a machine of a certain type"; he says "carried out by a machine." Nor did he ever devote any paper to the subject of "what is a machine." Newman (1955) said Turing came to his definition by analyzing the notion of a computing machine. Clearly Turing did indeed seize upon the concept of 'acting mechanically' implicit in the axiomatic program, turning it into the more definite 'machine,' and finding fascination in the mechanical thereafter – indeed developing it into our dominant technology. But as (Gandy 1988) put it, Newman was "subtly misleading"; Turing's 1936–7 analysis was of a *human being* computing in a mechanical way.

At this point I turn to the widely published and vividly expressed view of B. J. Copeland on just this question of what Turing meant by 'a machine.' It is my duty to point out a difficulty in Copeland's position. First, it is surprising that in his exposition (Copeland 1997) of the Thesis, Copeland omits the definitive 1938–9 statement, instead citing an informal version from Turing (1948). However that is not the most serious point, which is that Copeland (1997) holds that Turing's use of 'machine' was always made in a carefully restricted manner, and judiciously so because of the possibility of machines with *greater power* than Turing machines. Copeland explains that "For among a machine's repertoire of atomic operations there may be those that no human being unaided by machinery can perform." Copeland does not actually assert this was Turing's reasoning, in 1936 or subsequently, but his reader might well assume that this is why Turing made the definition he did. In fact Turing never suggested anything of the sort. Turing's thought stands within the natural and classical position, which is the other way round: it is to investigate whether and how a machine could possibly encompass the apparently greater faculties of human minds.

Copeland's suggestion of Turing having superhuman machines in mind clearly derives from his reading of Turing (1939), which is not mentioned in Copeland (1997), but is much discussed elsewhere (Copeland 1998, 1999; Copeland and Proudfoot 1999). In this paper Turing, after stating the Church–Turing thesis as quoted earlier, defined "a new type of machine." Copeland and Proudfoot (1999) quote this phrase as one of "Turing's Forgotten Ideas in Computer Science," and make it their mission to rescue it from the 'obscurity' of mathematical logic. Before succumbing to their excitement, however, we must analyze what Turing meant.

To understand Turing's "new type of machine" one must see what Turing had already defined. A vital point is that in (Turing 1936–7) the word 'machine' appeared in the definition of *choice-machine.* Such 'machines' are by definition *not purely mechanical*, being defined so as to require the decisions of 'an external operator.' Machines working without human choices, 'purely mechanically,' Turing called *automatic* or *a*-machines. Automatic machines are what (following Church's lead) we call Turing machines, and which Turing himself later (1948) called Logical Computing Machines. Such *a-machines* define the scope of computability and are the subject of the Church–Turing thesis, for that is concerned with whether a calculation can be carried out 'purely mechanically.' (Likewise, in Artificial Intelligence, attention is focused on *a*-machines. We should not be impressed by an AI program that relied upon human input!) Thus Turing used the word 'machine' for entities which are only *partially mechanical.* Perhaps this was courting confusion, but he was, of course, in accord with common usage.

An *oracle-machine* is likewise a machine only in the sense of being *partially mechanical.* Its specification, as a mathematical definition, is given as something merely postulated, *not* as an effective procedure. It differs from an *a*-machine in that it has one state in which it does *not* behave mechanically. Instead, the *o*-machine's next step then depends on an imaginary 'oracle' which has the power of giving the answer to any number-theoretic problem; as Turing shows this is equivalent to being able to answer the halting question for any Turing machine. Of the oracle Turing says, crucially, *"We shall not go any further into the nature of this oracle except that it cannot be a machine."* Turing describes the oracle as performing 'non-mechanical' steps. Thus an *o*-machine has a non-mechanical element, just as a choice-machine does.

Unfortunately Copeland, on the strength of the phrase "new type of machine," gives the vivid impression that the *o*-machine should be conceived as a *purely mechanical* device. Copeland and Proudfoot (1999) have written of how "the impact on the field of computer science could be enormous" if there were found "some practicable way of implementing an oracle." But the oracle is not a machine, so the question of 'implementing' it arises no more in Turing's exposition than does the question of 'implementing' the external operator's choices called for by a choice-machine.

The reader may suspect that further discussion is needed of what Turing meant by "cannot be a machine." Did he not merely mean, that it cannot be a *Turing machine.* Indeed he surely did: for in its context, the word 'machine' should mean the same as in the preceding statement of the Church–Turing Thesis, quoted above. There, it means what we call a 'Turing machine.' But this is no restriction on the force of Turing's remark, since he gave no indication that there could be any *other* types of purely mechanical machine.

If Turing had in mind the possibility of more general types of pure machine, he would have written that the oracle "cannot be an *a*-machine" or "cannot be a machine of any kind so far considered," or some such. Far from it: making himself even more categorical, he wrote that the *nature* of the oracle is that it cannot be a machine. Had Turing written that the *oracle* was a new kind of machine, Copeland would have his case. But by saying that the *o-machine* is a new kind of machine, Turing meant merely that it is a new type of 'not purely mechanical' machine.

As corroboration, note that if Turing had in mind the possibility of purely-mechanical machines other than *a*-machines, his 1938–9 statement would have required words like: '*understanding by a purely mechanical process one which could be carried out by an automatic machine of the type defined*'. In contrast, his actual statements, like Church's, identify the *a*-machine with the *most general* process that could be called 'purely mechanical.' Thus if he was saying the oracle was not an *a*-machine, he was saying it was not mechanical in *any* sense.

As already noted, Copeland (1997) chooses not to cite the Church–Turing thesis from Turing (1939), and so omits to analyze the use of 'machine' in this definitive version of it. Unfortunately an error in a later paper has denied Copeland the opportunity to show how he reconciles his concept of the 'new type of machine' with Turing's statement about the oracle. For Copeland (1999) quotes Turing as saying: "Of the nature of [an] oracle we shall say nothing." This truncation omits the essential substance. Further, Copeland uses the expression 'black box' to introduce the oracle, and says it could be conceptualized as having a tape with an infinite amount of information on it, giving the misleading impression that such physical images are Turing's.

Copeland also notes a difficulty in reconciling his standpoint with the endorsement in Church (1937b) of Turing's definition: "To define effectiveness as computability by an arbitrary machine, subject to restrictions of finiteness, would seem to be an adequate representation of the ordinary notion." Copeland (1997) states that 'arbitrary' refers to the arbitrary technical aspects of the way Turing machines or equivalent definitions are made. But no one could have read Church's sentence in so contrived a sense. Copeland asserts also that Church meant 'machine' to refer only to a machine mimicking the human calculator. In fact, Church (1937a) characterized the Turing machine in notably more general terms than this, and his words were that "in particular" a human calculator working to explicit instructions "can be regarded as a kind of Turing machine."

A more convincing claim would be that 'finiteness' restrictions *ipso facto* assert the logical possibility of machines with *infinitely* many states or symbols. But as indicated above, Turing never thought of such entities as 'machines,' and indeed the question of 'oracles' gives further evidence for this. For if Turing *had* wished to contemplate the oracle as a new kind of 'machine,' he could readily have done so by allowing a 'machine' to have infinitely many states. But Turing gave no such interpretation or definition, then or later.

Summarizing, Copeland misleads through his description of the *o*-machine as a new kind of machine. The oracle is not a machine. So *o*-machines are not purely mechanical. In fact they are *not machines*, if by 'machine' we mean something that works *independently.* Indeed the *whole point* of an oracle-machine is that it models non-mechanical steps. Unfortunately Copeland's exposition of the *o*-machine all rests upon this elementary confusion.

The reader may now wonder what possible reason Turing had to introduce a non-mechanical 'oracle.' One answer lies in some very interesting pure mathematics. I owe Feferman (1988) for an authoritative review. First, it is important to note that Turing (1939) is not focused on the *o*-machine definition. The paper is about a deep mathematical question, the fact that knowledge of one uncomputable number only provides *one* step in an infinite journey. The countable set of uncomputable numbers derived (computably) from this given one does not even scratch the surface of the (uncountable) totality of uncomputable numbers. This journey into the transfinite, subsequently formalized as 'relative computability,' is the subject of Turing's paper and its difficulty is what needed his genius; the oracle itself is trivial. Turing used the oracle concept to give a simple proof relating to the first step in this journey, but as Feferman points out, he could have proved it using even simpler cardinality arguments.

The very fact that the oracle is *not* necessary to his mathematics suggests that Turing may have had some extra-mathematical idea in mind when introducing it. There is indeed an interpretation which can be made from the context. The 1936–7 definition of computability arises from a human mind carrying out a rule; so one may well ask what Turing thought the mind was doing when *not* following a rule. The section in Turing (1939) headed "Interpretation of Ordinal Logics," tends to confirm this line of thought. It is about 'intuition' which as Turing explains, is how he considers the step of seeing the truth of a formally unprovable Gödel sentence; the whole point is that as Gödel showed, this step cannot be made 'mechanical.' Newman (1955) gives an interpretation of the oracle as the mathematician "having an idea" as distinct from "making mechanical use of a method."

This carries weight in view of Newman's unique cooperation with Turing in mathematical logic, and their wartime discussion of 'intuition,' but the interpretation needs care. The interpretative section of Turing's paper does not mention the 'oracle,' let alone *identify* the oracle with intuition, and this is probably for a good reason. The oracle does *far more* than any human being could: it knows the answer to all number-theoretic questions. And yet, as Turing must have known and Penrose (1994: 380) has emphasized, intuition, using a diagonal argument, can outdo an oracle-machine. Thus in another sense, when set against the mind, the oracle is *too weak.* We can only safely say that Turing's *general setting* for the introduction of uncomputable elements has to do with the role of the mind in apparently outdoing the mechanical.

Turing's one actual mention of an oracle in the substance of his paper refers to seeing "whether a given formula is a ordinal formula": the essential non-mechanical step that explains why repeated addition of new axioms cannot overcome the limitations of formal proof. I suspect that Turing's use of the lambda-calculus formalism, developed in his collaboration with and respect for Church, obscured what Turing worked out for himself more in the language of mechanical and non-mechanical steps, the latter corresponding to a mental act of 'seeing' intuitively.

Although Copeland (1999) correctly offers a mental context for the oracle, Copeland and Proudfoot (1999) suggest a very different technological picture to their popular readership. In a graphic illustration of a 'black box,' they suggest that "what Turing imagined" could be implemented as something like an electrical capacitor measured to infinite precision, and that one could solve a halting problem by reading off (say) the 8735439th binary place. Oracles, if discovered, might revolutionize computer science; and modern theorists of uncomputable physical effects are chided for not recognizing Turing's anticipation of their ideas. The words 'notional' and 'abstract,' are used to describe 'what Turing imagined,' but it is said that the oracle is abstract only in the same sense as is the Turing machine operation of scanning symbols, which clearly *can* be implemented.

Turing's oracle only amounts to a few lines of mathematical definition, so those expecting blueprints will be disappointed. It could be suggested, however, that physical black boxes with oracular properties might exist, even if Turing never aroused such a prospect himself. We must indeed distinguish between the historical question of what Turing entertained, and the scientific question of what actually is the case, and on the latter question research has certainly turned to more complex issues than Turing considered. The oracle, something 'fictional' in Turing's thought, as Penrose has described it, would be factual if in some way infinite data could be stored in a finite system. This question has stimulated investigations into the relationship between computability, continuous systems, and physical properties of universes real and imaginary. However, there is nothing in modern physics to suggest the crude 'black box' in Copeland and Proudfoot's illustration of 'what Turing imagined,' requiring measurements of unlimited precision! And nothing could be further from Turing's logical calculus, in which the whole point of the oracle is to investigate the structure of what can *not* be done mechanically. (One should note also that even if a 'black box' were found to emit an uncomputable sequence, in fact rather than in fiction, there is no reason to expect it to solve any identifiable logical problem.)

There is one physical context for discussing 'oracles,' which has both some historical pertinence and modern interest. This is not mentioned by Copeland and Proudfoot (1999) amidst their technological imagery, but it is alluded to by Copeland (1998, 1999): it is the physics of the *brain*. This is at least consistent with Turing's setting, that of the human mind appearing to outdo machines. This is also the context in which Penrose (1994: 379) introduces a discussion of oracles, in his discussion of how uncomputability might feature in an as yet unknown physical law. (See also Penrose (1996) for an introduction to Penrose's arguments about computability and Mind.)

Penrose might well ask how in 1938 Turing would have answered the question of how the brain performed these acts of intuition. This is not an ahistorical speculation because, although Turing never used the word 'brain' in his 1936–39 writing, in 1930–32 he had thought seriously of its physics, stimulated by A. S. Eddington's view that quantum mechanical physics removed the classic conflict of free-will with physical determinism (Hodges 1983: 63).

But lacking any trace of such a discussion in 1936–39, it appears that Turing then simply left open the question of how it can be that a physically embodied mind appears to do the non-mechanical 'intuition' involved in seeing the truth of Gödel statements. He would not have been alone; Gödel himself seems to have taken a view of mind as non-mechanical without trying to reconcile this with its embodiment in brains.

But now let us return to the Turing of 1941, who unlike Gödel has linked logic with the physical to astonishing effect. Turing's Enigma-breaking machines are demonstrating the power of algorithms, by following through logical implications. Machines have become practical, and aspects of guessing have become mechanical. And while defending human freedom with the machines his mind had made, Turing is reading *The Mind of the Maker* by Dorothy L. Sayers.

In Hodges (1997) I suggested that it was at this 1941 period that Turing concluded that the scope of the computable was *not* limited to processes where the mind follows a definite rule. Machines which modified their own rules of behavior would show features which had not been foreseen by anyone designing them. Such machines might be said to learn, and perhaps to act with the appearance of intelligence. From 1941 onwards Turing began to speak of such ideas to his Bletchley Park colleagues, and also to use the word *brain.* Again, my guess is that having confronted the problem of how the physical brain could support the appearances of non-mechanical 'intuition,' Turing concluded that the function of the brain *was* that of a machine, but one so complex that it could have the appearance of not following any rule. From this point Turing apparently became gripped by the potential of computability, and of his own discovery that all computable operations can be implemented on a single, universal, machine.

Thus, in this view it was during the war that Turing formulated both his central contribution to cognitive science and the practical universal machine. In his 1945 vision, algorithms are enough to account for all mental activity, including the kind previously thought of as non-mechanical intuition; the universal machine is enough for all algorithms, and electronics make practical a universal machine. In 1945 Turing embarked on his own independent design of what he called a "practical universal computing machine." (For a new account of the origin of the modern electronic stored-program computer, which credits Turing with giving von Neumann the central idea, see Davis 2000.)

Although Turing promoted the practical benefits of a computer, it was more the prospect of using it to simulate the brain that engaged him from the start. Thus even in Turing (1946), his technical report, a statement about the prospect of a machine showing intelligence in chess-playing appears. It makes a reference to *mistakes* in chess-playing, which, as expanded in later papers, betrays Turing's concern for answering the problem of how minds can see the truth of Gödel sentences. His postwar argument is that humans make mistakes, machines make mistakes, they are on a par, and that once infallibility is off the agenda, the Gödel argument does not apply. (This is the same argument that 50 years later Davis (2000: 197) upholds, as against Penrose's argument that a non-mechanical 'intuition' of truth is inescapable.) These remarks are to my mind evidence of how the postwar Turing had to respond to the implications of Gödel's work and his own 1938 discussion of 'intuition.'

It as not long before Turing (1948) described the problem of 'intelligent machinery' as that of how to create machines with 'initiative.' This is not the same word as the 'intuition' of 1938 but has the same role as describing that what the mind does when not apparently following a rule. But in Turing's postwar thought, initiative does not need uncomputable steps; it is as computable as the 'mechanical processes' even though this is against one's expectations of 'machines' (it is in this paper that he quotes from Dorothy L. Sayers). However it is necessary to depart from computations that follow the programmer's explicit plan. To this purpose Turing sketched nets of logical elements which, as Ince (1993) put it, can be said to predict the neural network approach. The paper of Copeland and Proudfoot (1996) on this subject is perhaps their greatest achievement since it has stimulated new scientific work (Teuscher 2001; Webster 1999). The modern climate is more favorable than were the 1970s and 1980s to Turing's viewpoint, in which advanced programming and evolutionary networks were not perceived as distinct alternatives, both being avenues for research in machine intelligence.

We have now reached the question of Turing's *postwar* writing on the concept of 'machine,' for Turing gave a classification of machines in that same paper (Turing

1948). Reflecting his greater acquaintance with physical machinery, Turing widened the scope of 'machine' and distinguished Logical Computing Machines from *continuous* and from *active* machines (e.g. 'a bulldozer'). Naturally, oracles do not feature in this discussion, since oracles are not machines. And Turing suggests nothing about the possibility of superhuman machine operations.

Copeland (1997) does not refer to this 1948 discussion, surprisingly as it is the closest Turing came to an essay on 'what is a machine.' But Copeland has usefully cited a later discussion by Turing's student Robin Gandy, who undertook the kind of formal analysis of computing machines that Newman (1955) attributed to Turing. Gandy (1980) introduced Thesis M: *"What can be calculated by a machine is computable."* Gandy showed that computability followed under very general assumptions about a mechanical system; and that if these conditions were weakened, *anything* could be calculable. 'Thesis M' allows Copeland to formulate what he claims of Turing: the essence of Copeland (1997) is that there is no evidence that Turing subscribed to Thesis M. It is true, as Copeland points out, that many writers have given versions of the Church–Turing thesis varying from what Church or Turing actually said, particularly in making sweeping statements about physical systems. Nor can we say quite how Turing would have responded to Gandy's formulation. But Copeland (1999) errs in claiming that Turing's definition of oracles precluded him from believing Thesis M. The reason is simple: Thesis M concerns machines, and Turing's oracle is *not* a machine.

The discussion by Copeland (1997) also neglects to engage with the fact that the general force of Turing's postwar arguments is that computable operations always suffice. How could he have made this thrust if, all the time, he had secretly in mind the potential of 'machines' to perform uncomputable operations? Copeland cites the careful definition of computability from Turing (1950) but ignores the central claim in that paper that Mind, including its appreciation of sonnets and the rest, can be imitated by a computer, that is computable. Copeland's position would be consistent with Turing's assertion of the computability of Mind only if Turing had believed that whereas minds were limited to the computable, there was a possibility of machines *not* so limited. But there is no such component detectable in Turing's thought. Despite this, Turing's later papers are searched for clues that he was leaving room for *o*-machines. For example, Copeland (1997) warns that in Turing (1947), a statement about 'machine process' should be read carefully to refer only to *a*-machines. It should indeed, but this is simply because Turing was here distinguishing Turing machines from the differential analyser, a *continuous* machine. (Morever he was in this passage advocating the the digital computer as *superior* in performance to the analog machine.)

In Copeland (1999) another claim is made relating 'oracles' to *randomness.* This is done through a paper of Church (1940), which gives a careful definition of an infinite random sequence, entailing that it must be an uncomputable sequence. Copeland goes on to describe a random sequence as an oracle-machine output, then to claim that Turing himself made this identification. But Turing did not allude to any connection between randomness and the infinite data stores implicit in an 'oracle.' On the contrary, Turing (1948, 1950) asserted that pseudo-random (i.e. computable) sequences would suffice for random effects, a statement he could never have made if he thought the uncomputable played a role. These were hardly idle comments; he as well as anyone on the planet in 1950 knew the significance of pseudo-random sequences, having in wartime extracted machine patterns from apparently random ciphertext; he had also addressed himself to the engineering of randomness in the Manchester computer. Besides, oracles were introduced by Turing to model not *randomness*, but a kind of infinite *knowledge.*

Turing gave scant analysis of 'randomness', but this very brevity and the rather cavalier treatment of underlying physics in Turing (1948, 1950) is telling. Had he ever had uncomputable effects in mind, he could not have been so terse. He treated the relationship of discrete to continuous systems in a similar way: in Turing (1948) merely asserting that the brain though continuous is effectively discrete, returning to this point again briefly in Turing (1950) to counter "the Argument from Continuity of the Nervous System."

After scouring Turing's later works in the hope of glimpsing an allusion to oracle-machines, Copeland (1999) concedes, "if Turing did think that *o*-machines other than partially random machines are physically possible, then perhaps he would have said as much, and he does not appear to have done so." Quite so. Copeland has not shown how he reconciles this conclusion with the statement in Copeland and Proudfoot (1998) that:

> Taking their cue from Turing's 1939 paper, a small but growing international group of researchers is interested in the possibility of constructing machines capable of computing more than the universal Turing machine . . . research in this direction could lead to the biggest change computing has seen since 1948. Hodges's Turing would regard their work as a search for the impossible. We suspect that the real Turing would think differently. (Copeland and Proudfoot 1998: 6)

In this account, Turing's imagined disposition towards 'constructing' oracle-machines is pronounced to be *so definite* that it is required to inform the general public that the standard view, as conveyed in Hodges (1997), is not "the real Turing."

Yet if Turing's use of 'machine' for the only partially mechanical has confused these writers, it surely must have perplexed others less eminent, and so perhaps it is as well that this confusion has emerged so publicly through Copeland's work. Copeland (1999) has also recently done the service of editing the radio broadcast made by Turing (1951), and in a preface, amidst a fruitless hunt for oracles, he correctly draws attention to a comment of Turing about the question of whether brains can be seen as machines. It is a remarkable comment which suggests that he had given more thought to physics since writing the 1948 and 1950 papers. Turing here describes the universal machine property, applying it to the brain, but says its applicability requires that the machine whose behavior 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."

I described in Hodges (1983: 441) how Turing in this passage harked back to his 1930–2 wonder about the physics of the brain. Now I would consider fuller analysis needed. It was Penrose's question about how Turing saw computability after 1936 that encouraged me to take Turing's 1938–9 reference to 'intuition' more seriously and suggest (Hodges 1997) that his thesis of computable mind probably came later, in about

1941. Now, again, prompted by Penrose's arguments, I take Turing's reference to quantum mechanics more seriously than in 1983, and see it as a link between his work on computability and the burst of work in physics just before his death.

There is nothing about oracles in Turing's 1951 sentences. (Note also that even though now expanding the concept of 'machine,' Turing still addresses the question of whether a machine can do as much as the mind, not Copeland's question of superhuman machines.) The point is that Turing here characterized the nature of quantum physics as possibly *unpredictable in principle*. Now, much is (rightly) made of Turing's contact with von Neumann in connection with the origin of the digital computer, but it is less well-known that Turing's first contact with von Neumann's work came in 1933 from studying the mathematical foundations of quantum mechanics (Hodges 1983: 79). So, as Turing knew so early, it is the *reduction or measurement process* for which there is no prediction even in principle; the evolution of the wave-function by Schrödinger's equation is predictable.

It is unlikely that Turing here was suggesting Penrose's view of quantum mechanics. More probably, he was seeking to reformulate quantum mechanics as a predictable theory when in 1953–4 he pursued this interest in physics. He wrote to Gandy, partly perhaps in jest, "I'm trying to invent a new Quantum Mechanics but it won't really work. How about coming here next week and making it work for me?" (Turing 1953–4). He was apparently focusing on the problem of the 'watched pot paradox' of wave function reduction (Gandy 1954; Hodges 1983: 495). Nevertheless Turing did acknowledge that here lay a fundamental problem for anyone assuming the computability of brain function.

Turing's interest in state-reduction, and the lack of a rule for it, should not be confused with prospect of *quantum computation* as developed since the 1980s. Quantum computation does not cross the boundary of computability, and moreover depends on the *predictability* of unitary evolution. Yet the elementary applications of quantum computation, as applied in quantum cryptography, have already led to procedures depending on non-local effects which cannot usefully be formulated as classical algorithms. This is enough to show that logic and physics can no longer be kept apart. The interpretation of the Church–Turing Thesis must necessarily be influenced by this development. In 1983 I used *the logical and the physical* as an organizing principle for the life and work of Alan Turing. But perhaps I did not follow this principle far enough. If we look to Turing for a prophecy of developments beyond the Turing machine, our best bet lies in his hint that the full discussion of computability requires the as yet incompletely known laws of quantum mechanics.

It is notable that in his 1951 talk Turing also raised the question of interpreting Gödel's theorem, and with less assertiveness than in 1950 that the problem went away through 'mistakes.' Thus although the philosophical detective, B. J. Copeland, has handcuffed the wrong suspect, through mistaking the identity of the oracle, I agree with him that the case as regards Turing's thought in his last period is not entirely closed. Turing did not draw a connection between Gödel's theorem and quantum mechanics, as Penrose does, but he did point to just these areas as leaving open and awkward questions.

Turing probably took the name 'oracle' from Shaw's *Back to Methuselah* which he enjoyed seeing performed at Princeton. Shaw doubted that our current human span is long enough to gain sufficient wisdom. Turing said, perhaps echoing this pessimism, that no individual can do very much in a life. But the collective mind allows us greater optimism. In 1900, Planck's quantum, and Hilbert's problem of the consistency of arithmetic, soon joined by Russell's paradox, were unrelated. A century of investigations into the logical and physical have led to quantum computing, a connection unimaginable in 1900. The next century may see more unexpected developments, though perhaps no individual (or machine) more surprising than Alan Turing.

References

Church, A. (1937a) Review of Turing, 1936–7. *Journal of Symbolic Logic*, 2, 42.

Church, A. (1937b) Review of Post 1936. *Journal of Symbolic Logic*, 2, 43.

Church, A. (1940) On the concept of a random sequence. *Bull. Amer. Math. Soc.*, 46, 130–5.

Copeland, B. J. (1992) *Artificial Intelligence: A Philosophical Introduction*. Oxford: Blackwell.

Copeland, B. J. (1997) The Church–Turing thesis, in E. N. Zalta (ed.), *Stanford Encyclopaedia of Philosophy*, http://plato.stanford.edu

Copeland, B. J. (1998) Turing's o-machines, Searle, Penrose and the brain. *Analysis*, 58.2, 128–38.

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. and Proudfoot, D. (1996) On Alan Turing's anticipation of connectionism. *Synthese*, 108, 361–77.

Copeland, B. J. and Proudfoot, D. (1998) Enigma variations. *Times Literary Supplement* (London), July 3, 1998.

Copeland, B. J. and Proudfoot, D. (1999) Alan Turing's forgotten ideas in computer science. *Scientific American*, April.

Davis, M. (2000) *The Universal Computer*. New York: W. W. Norton.

Feferman, S. (1988) Turing in the Land of O(Z). In Herken (1988).

Gandy, R. O. (1954) Letter to M. H. A. Newman, in King's College archive, to appear in the remaining volume of *The Collected Works of A. M. Turing*, ed. R. O. Gandy and C. E. M. Yates.

Gandy, R. O. (1980) Principles of mechanisms. In J. Barwise, H. J. Keisler and K. Kunen (eds.), *The Kleene Symposium*. Amsterdam: North-Holland.

Gandy, R. O. (1988) The confluence of ideas in 1936. In Herken (1988).

Good, I. J. (2000) Turing's anticipation of empirical Bayes in connection with the cryptanalysis of the Naval Enigma. *J. Stats. Comp. & Simul.*, 66: 101–11; and in the remaining volume of *The Collected Works of A. M. Turing*, ed. R. O. Gandy and C. E. M. Yates.

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

Hinsley, F. and Stripp, A. (eds.) (1993) *Codebreakers*. Oxford: Oxford University Press.

Hodges, A. (1983) *Alan Turing: The Enigma*. London: Burnett; New York: Simon & Schuster; new editions London: Vintage 1992; New York: Walker, 2000.

Hodges, A. (1997) *Turing: A Natural Philosopher*. London: Phoenix; New York: Routledge, 1999. Included in *The Great Philosophers* (eds. R. Monk and F. Raphael) London: Weidenfeld and Nicolson, 2000.

Ince, D. C. (1993) Preface to Turing (1948) in D. C. Ince (ed.), *Mechanical Intelligence*, vol. III of *The Collected Works of A. M. Turing*. Amsterdam: North-Holland, 1993.

Newman, M. H. A. (1955) *Alan M. Turing*, Biographical memoirs of the Royal Society.

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

Penrose, R. (1994) *Shadows of the Mind*. Oxford, New York: Oxford University Press.

Penrose, R. (1996) Beyond the doubting of a shadow. *Psyche Electronic Journal*, http://psyche.csse.monash.edu.au/v2/psyche-2–23-penrose.html

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

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

Turing, A. M. (1936–7) On computable numbers with an application to the *Entscheidungsproblem. Proc. London Maths. Soc.*, ser. 2, 42, 230–65; also in M. Davis, (ed.), *The Undecidable*. New York: Raven, 1965.

Turing, A. M. (1939) Systems of logic defined by ordinals. In M. Davis (ed.), *The Undecidable.* New York: Raven, 1965. (This was also Turing's 1938 Ph.D. thesis, Princeton University.)

Turing, A. M. (1946) Proposed Electronic Calculator, report for National Physical Laboratory, Teddington; published in B. E. Carpenter and R. W. Doran (eds.), *A. M. Turing's ACE Report of 1946 and Other Papers*, Cambridge, MA: MIT Press, 1986; and in D. C. Ince (ed.), *Mechanical Intelligence*, vol. III of *The Collected Works of A. M. Turing*. Amsterdam: North-Holland, 1993.

Turing, A. M. (1947) Lecture to the London Mathematical Society, typescript in King's College, Cambridge, available in Turing Digital Archive, http://www.turingarchive.org. Published in B. E. Carpenter and R. W. Doran (eds.), *A. M. Turing's ACE Report of 1946 and Other Papers*, Cambridge, MA: MIT Press, 1986; and in D. C. Ince (ed.), *Mechanical Intelligence*, vol. III of *The Collected Works of A. M. Turing*. Amsterdam: North-Holland, 1993.

Turing, A. M. (1948) Intelligent machinery, unpublished NPL report, typescript in Turing Archive, King's College, Cambridge, available in Turing Digital Archive, http://www.turingarchive.org. Published (ed. D Michie) in *Machine Intelligence* 7 (1969), and in D. C. Ince (ed.), *Mechanical Intelligence*, vol. III of *The Collected Works of A. M. Turing.* Amsterdam: North-Holland, 1993.

Turing, A. M. (1950) Computing machinery and intelligence. *Mind*, 49, 433–60.

Turing, A. M. (1951) BBC Radio talk, typescript in King's College, Cambridge, available in Turing Digital Archive, http://www.turingarchive.org. Published (ed. B. J. Copeland) in K. Furukawa, D. Michie and S. Muggleton (eds.), *Machine Intelligence* 15. Oxford: Oxford University Press, 1999.

Turing, A. M. (1953/4) Letter to R. O. Gandy, undated, marked by Gandy "1953 or 54." In Turing Archive, King's College, Cambridge, section D/4, available in Turing Digital Archive, http://www.turingarchive.org.

Webster, C. S. (1999) Unorganized machines and the brain. Available from http://home.clear.net.nz/pages/cw/unorganized.htm.