Turing's O-machines, Searle, Penrose and the Brain

B. Jack Copeland

ABSTRACT

In his PhD thesis (1938) Turing introduced what he described as 'a new kind of machine'. He called these 'O-machines'. The present paper employs Turing's concept against a number of currently fashionable positions in the philosophy of mind.

I

Johnson-Laird is surely right when he discerns three mutually exclusive positions in current thinking concerning the relationship between human mentality and computation (1987: 252). He suggests that the only alternative to these three positions is that consciousness is not 'scientifically explicable' (ibid.).

(1) The human brain (or, variously, mind or mindbrain) is a computer, equivalent to some Turing machine.

There are, of course, innumerable fine-grained renderings of this theory: the brain is variously thought to be a digital computer, an analogue computer, a stored-program machine, a program-controlled machine, a massively parallel distributed processor, etc.

(2) The activity of a human brain can be simulated perfectly by a Turing machine but the brain is not itself a computing machine.

Analogies offered in the literature include hurricanes, the motion of the planets around the sun, and the digestion of pizza. None of these phenomena is itself computational in nature yet, supposedly, all can be simulated perfectly by a Turing machine (in the sense that the machine can compute descriptions of the phenomena to any desired number of decimal places and on any temporal grid). (2) is Searle's position (1992, 1997).

(3) The brain's cognitive activity cannot in its entirety be simulated by a computing machine: a complete account of cognition will need 'to rely on non-computable procedures' (Johnson-Laird 1987: 252).

Penrose (1994) maintains a version of (3) which is so strong that it is not entailed by the conjunction of the negations of (1) and (2).

I shall be arguing that (1)-(3) do *not* exhaust the alternatives open for scientific consideration. At least two further hypotheses belong alongside (1)-(3):

(4) The brain is what Turing called an O-machine.

(5) The cognitive activity of the brain can be simulated perfectly by an O-machine, but the brain is not itself an O-machine; such simulation cannot be effected by a Turing machine.

O-machines (section II) are digital computing machines. They generate digital output from digital input by means of a step-by-step procedure consisting of repeated applications of a small, fixed number of primitive operations, the procedure unfolding under the control of a finite program of instructions which is stored internally in the form of data on the machine's tape. Thus even if (1) is false, the theory that the brain is a computing machine might nevertheless be true. Furthermore, the success of hypothesis (4) would vindicate functionalism. As with Turing machines, the description of an O-machine is silent about the underlying hardware.

I I

Turing introduced the concept of an O-machine in his PhD thesis (Princeton, 1938). The thesis was supervised by Church. It was subsequently published as Turing 1939. This paper is a classic of recursive function theory, yet it is seemingly little-known among philosophers of mind.

The primitive operations of an (ordinary) Turing machine are six in number: (i) move the tape left one square; (ii) move the tape right one square; (iii) read (i.e. identify) the symbol currently under the head; (iv)

write a symbol on the square of tape currently under the head (after first deleting the symbol already written there, if any); (v) change state; (vi) halt. These primitive operations are made available by unspecified subdevices of the machine - 'black boxes'. The question of what mechanisms might appropriately occupy these black boxes is not relevant to the machine's logical specification. Such matters belong to what Block has termed 'realisation science' (1990: 259). An O-machine is a Turing machine augmented with one or more primitive operations each of which returns the values of some function (on the natural numbers) that is not Turing-machine-computable. Each additional primitive operation is made available by a black box. Turing refers to these black boxes as 'oracles'. He remarks that an oracle works by 'unspecified means' and says that we need 'not go any further into the nature of [these] oracle[s]' (1939: 173). According to Turing's specification each oracle returns the values of a two-valued function. Let these values be written 0 and 1. Let *p* be one of the additional primitive operations. *p* is called into action by means of a special state χ, the call state. (Where an O-machine has several of these additional primitives a corresponding number of call states is required.) The machine inscribes the symbols that are to form the input to *p* on any convenient block of squares of its tape, using occurrences of a special symbol μ, the marker symbol, to indicate the beginning and the end of the input string. As soon as an instruction in the machine's program puts the machine into state χ, the input is delivered to the subdevice that effects *p*, which then returns the corresponding value of the function. On Turing's way of handling matters the value is not written on the tape. Rather a pair of states, the 1-state and the 0-state, is employed in order to record values of the function. A call to *p* ends with a subdevice placing the machine in one or other of these two

states according to whether the value of the function is 1 or 0. (When a function *g* is computable by an O-machine whose (only) oracle serves to return the values of a function *f*, then *g* is sometimes said to be computable *relative* to *f*.)

One particular O-machine, the halting function machine, has as its 'classical' part the universal Turing machine specified by Turing in 1936 and as its 'nonclassical' part a primitive operation that returns the values of Turing's famous halting function $H(x,y)$ (Turing 1936). (The halting function is defined thus: $H(x,y)=1$ if and only if the x^{th} Turing machine eventually halts if set in motion with the integer *y* inscribed on its tape, say in binary code (think of *y* as being the machine's input); and $H(x,y)=0$ otherwise.) The halting function machine can compute many functions that are not Turing-machine-computable. This is because of the familiar phenomenon of the *reducibility* of one function to others (for example, multiplication is reducible to addition). All functions reducible to the halting function and/or the primitives of an ordinary Turing machine are computable by the halting function machine.

As previously remarked, Turing introduces O-machines without discussion of how the primitive operations might be implemented. Equally, in his earlier paper of 1936 there is no discussion of how the primitive operations of a Turing machine might be implemented. In both papers, the role played by his notional computing machines is to delineate classes of mathematical problems; the possibility of real existence is beside the point of this exercise. In 1936 it was, in fact, far from clear whether the construction of a universal Turing machine lay within the bounds of physical possibility. Once asked whether he thought that a universal Turing machine could actually be constructed, Turing dismissively replied that the machine

would need to be as big as the Albert Hall (this was related to me by Robin Gandy, who worked with Turing during the later part of the war). It was not until Turing became acquainted with electronic technology developed for a completely different purpose that he realised that the notional machines of his 1936 paper could actually be built. Perhaps his O-machines, too, will one day become an engineering reality. The existence of physical processes that cannot be simulated by Turing machine is certainly a logical possibility, there being no shortage of law-like regularities for such processes to exhibit: functions that are Turing-machine-computable form a relatively slender proper subset of the vast space of functions on the natural numbers (let alone of the vaster space of functions on the real numbers). Speculation as to whether there may actually be physical processes that cannot be simulated by Turing machine stretches back over at least four decades (for example Da Costa and Doria 1991; Doyle 1982; Geroch and Hartle 1986; Komar 1964; Kreisel 1967, 1974; Penrose 1989, 1994; Pour-El 1974; Pour-El and Richards 1979, 1981; Scarpellini 1963; Stannett 1990; Vergis et al 1986). If such processes do exist then perhaps future engineers will use them to implement the non-classical part of some O-machine.

Science fiction or not, this theorizing suffices to illustrate why it is an empirical matter whether or not the disjunction of hypotheses (1) and (2) is true.

III

Not so according to John Searle. Searle believes that it follows from Church's thesis, itself a broadly logical claim, that the activity of the brain

can be simulated by a Turing machine (whence (1)∨(2)). He writes as follows:

> Can the operations of the brain be simulated on a digital computer [read: Turing machine]? ... The answer ... seems to me ... demonstrably "Yes" ... That is, naturally interpreted, the question means: Is there some description of the brain such that under that description you could do a computational simulation of the operations of the brain. But given Church's thesis that anything that can be given a precise enough characterization as a set of steps can be simulated on a digital computer, it follows trivially that the question has an affirmative answer. The operations of the brain can be simulated on a digital computer in the same sense in which weather systems, the behavior of the New York stock market, or the pattern of airline flights over Latin America can. (1992: 200-201; see also 1997: 87.)

Searle's statement of Church's thesis is mistaken. Church's thesis (also known as 'Turing's thesis' and the 'Church-Turing thesis' (Church 1936, Turing 1936, Kleene 1967) is a proposition concerning the extent of what can be achieved by a human mathematician who is unaided by any machinery save paper and pencil, and who is working in accordance with 'mechanical' methods, which is to say, methods set out in the form of a finite number of exact instructions that call for no insight or ingenuity on the part of the person who is carrying them out. The Church-Turing thesis states that whatever can be calculated by a mathematician so working, even a mathematician idealised to the extent of being free of all constraints on time, patience, concentration, and so forth, can also be calculated by a Turing machine.

This thesis carries no implication concerning the extent of what can be calculated by a *machine* (say one that operates in accordance with a finite program of instructions), for among the machine's repertoire of primitive operations there may be those that no human being unaided by machinery can perform. Nor does the thesis imply that each process admitting of a precise characterisation 'as a set of steps' can be simulated by a Turing machine, for the steps need not be ones that a human mathematician working in accordance with some mechanical method can carry out. Trivially, the processing of an O-machine is always characterisable as a set of steps, namely, the set of steps specified by the machine's program. Employing the thesis espoused by Searle yields the absurdity that an O-machine can be simulated by a Turing machine. Searle's attempt to recruit Church's thesis in support of (2) is entirely fallacious.

An O-machine's program may call for primitive operations that a human clerk working by rote and unaided by machinery is incapable of carrying out (for otherwise, by the real Church-Turing thesis, whatever can be calculated by an O-machine can be calculated by a Turing machine - a contradiction). It follows that there is no possibility of Searle's Chinese room argument being successfully deployed against the new functionalism offered by hypothesis (4) (which Searle will presumably find as 'anti-biological' as other functionalisms). This argument (Searle 1980, 1989) depends upon the human clerk who occupies the Chinese room being able to carry out by hand, using paper and pencil, each operation that the program in question calls for (or in one version of the argument, to carry them out in his or her head). Turing originally introduced the Turing machine as a *model* of a human clerk engaged in mathematical calculation (1936: 231), and so, of course, each primitive operation of a Turing machine is indeed one that a

human clerk can carry out. The same is true of the electronic machines fashioned after the universal Turing machine. As Turing himself put it:

> 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. (Turing 1950: 1.)

O-machines, on the other hand, conspicuously fail to satisfy this ground condition of the Chinese room argument. Searle says:

> Because programs are defined purely formally or syntactically, and because minds have an intrinsic mental content, it follows immediately that the program by itself cannot constitute the mind. The formal syntax of the program does not by itself guarantee the presence of mental contents. I showed this a decade ago in the Chinese room argument. ... A computer, me for example, could run the steps in the program for some mental capacity, such as understanding Chinese, without understanding a word of Chinese. (1992: 200.)

But the program of an O-machine, too, no less than in the case of a Turing machine, is 'defined purely formally or syntactically'. If there is any implication from 'is defined purely formally' to 'is neither constitutive of nor sufficient for the mind', it is not one that can be established by the Chinese room argument.

Searle's mistake concerning the extent of what is asserted by the Church-Turing thesis is, in fact, a common one, which can be frequently encountered in recent writing on the philosophy of mind. For example, the Churchlands, like Searle, purport to deduce from Church's Thesis that the mind-brain can in principle be simulated by a Turing machine (1983: 6). They further assert that Turing's

> results entail something remarkable, namely that a standard digital computer, given only the right program, a large enough memory and sufficient time, can compute *any* rule-governed input-output function. That is, it can display any systematic pattern of responses to the environment whatsoever. (1990: 26.)

Turing had no results which entail this. Rather, he had a result that entails the opposite. The various functions that, in 1936, he proved to be not Turing-machine-computable, for example, the halting function, are mathematical characterisations of systematic patterns of responses to the environment that cannot be displayed by a standard digital computer (even one unfettered by resource constraints).

Perhaps the introduction of a term for the fallacy embraced here by Searle and the Churchlands will assist in its extinction. Since the fallacy is invariably committed in the name of either Church or Turing, I suggest the *Church-Turing fallacy*. One commits the Church-Turing fallacy by believing that the Church-Turing thesis, or some formal result proved by Turing or Church, secures the truth of the proposition that the brain can be simulated by a Turing machine. Here is a third example of the fallacy at work:

> If you assume that [consciousness] is scientifically explicable ... [and] [g]ranted that the [Church-Turing] thesis is correct, then ... [i]f you believe [functionalism] to be false ... then ... you [should] hold that consciousness could be modelled in a computer program in the same way that, say, the weather can be modelled ... [and if] you accept functionalism ... you should believe that consciousness is a computational process. (Johnson-Laird 1987: 252.)

No less common in the literature are statements which, while not explicitly involving the Church-Turing fallacy, make exaggerated claims on behalf of

Turing machines. For example, the entry on Turing in the recent 'A Companion to the Philosophy of Mind' contains the following assertions: 'we can depend on there being a Turing machine that captures the functional relations of the brain', for so long as 'these relations between input and output are functionally well-behaved enough to be describable by ... mathematical relationships ... we know that some specific version of a Turing machine will be able to mimic them' (Guttenplan 1994: 595). Also typical are the following:

> If a mental process can be functionally defined as an operation on symbols, there is a Turing machine capable of carrying out the computation. (Fodor 1981: 130; see also 1983: 38-39.)
>
> *any* process which can be formalised so that it can be represented as a series of instructions for the manipulation of discrete elements can, at least in principle, be reproduced by [a universal Turing machine]. (Dreyfus 1992: 72.)

The logical availability of hypothesis (4) gives the lie to all of these claims. The relations between the inputs and outputs of an O-machine, say the halting-function machine, are certainly 'functionally well-behaved enough to be describable by ... mathematical relationships'. There is nothing *unmathematical* or *ill-defined* about the halting function! Contrary to the claims by Fodor and Dreyfus, hypothesis (4) postulates mental processes consisting of operations on discrete symbols that cannot be carried out by a universal Turing machine. Turing machines and O-machines alike execute 'series of instructions for the manipulation of discrete elements'; by definition, each additional primitive operation of an O-machine is 'an operation on symbols' - in essence, an operation that replaces a binary string with 1 or 0. The O-machines point up the fact that the notion of a

symbol-processing machine is more general than the notion of a Turing machine.

IV

As is well known, Penrose argues on the basis of formal results due to Gödel and Turing that propositions (1) and (2) are false (1989, 1990, 1994). In a section of the latter rather inconspicuously positioned in the midst of a chapter on quantum theory and the brain (1994, chapter 7, section 9), Penrose explains that his argument can be extended to apply to oracle machines:

> the arguments of Part I of this book can be applied equally well against an oracle-machine model of mathematical understanding as they were against the Turing-machine model, almost without change. (1994: 380; see also 1996, sects 3.10, 13.2.)

This is why, in section I, I described Penrose's version of the anti-computationalist hypothesis (3) as being sufficiently strong as to be not entailed by the conjunction of the negations of (1) and (2).

There is, as Penrose is clearly aware, a whiff of *reductio ad absurdum* about this. Let the *first-order* O-machines be those whose (only) oracle returns the values of Turing's halting function $H(x,y)$ (as in the case of the halting function machine described earlier). Similarly, the second-order O-machines are those that possess an oracle which can say whether or not any given first-order O-machine eventually halts if set in motion with such-and-such input; and so on for third-order, and in general α-order. Penrose's argument, originally marketed as demonstrating that human mathematicians do not use a knowably sound Turing-machine algorithm in

order to ascertain mathematical truth (e.g. in 1994 ch. 2), appears to be so powerful that it can equally well be employed to show, given any number α at all for which there is a notation, that human mathematicians do not use, in ascertaining mathematical truth, any knowably sound procedure capable of being executed by an α-order oracle machine. Penrose's argument moves relentlessly up through the orders, stopping nowhere. This discovery evidently discomforts Penrose:

> The final conclusion of all this is rather alarming. For it suggests that we must seek a non-computable physical theory that reaches beyond every [recursive] level of oracle machines (and perhaps beyond). No doubt there are readers who believe that the last vestige of credibility of my argument has disappeared at this stage! I certainly should not blame any reader for feeling this way. (1994: 381.)

Penrose does hint at a way out:

> [I]t need not be the case that human mathematical understanding is in principle as powerful as *any* oracle machine at all. ... [T]he conclusion 𝔾 [human mathematicians are not using a knowably sound algorithm in order to ascertain mathematical truth] does *not* necessarily imply that human insight is powerful enough, in principle, to solve each instance of the halting problem. Thus, we need not necessarily conclude that the physical laws that we seek reach, in principle, beyond every computable level of oracle machine (or even reach the first order). We need only seek something that is not equivalent to *any* specific oracle machine (including also the zero*th*-order machines, which are Turing

> machines). Physical laws could perhaps lead to something that is just *different*. (1994: 381.)

On that enigmatic note Penrose leaves it. Just when we seem to be approaching a crucial part of the exposition, he suddenly falls silent. What, indeed, does Penrose *mean* here?

It is natural to think of the functions, or problems, that are solvable by a first-order oracle machine as being *harder* that those solvable by Turing machine, and those solvable by a second-order oracle machine as being harder still, and so forth. (To say that a Turing machine or O-machine 'solves a problem' is to say that when the machine is given the problem, suitably encoded, on its tape it halts with the answer, 1 (Yes) or 0 (No), under its head.) It is customary in recursion theory to say that a class of problems of equal hardness are of the same *degree*. Problems that are solvable by Turing machine are said to be of degree 0. Let me write 1 for the degree of problems that are solvable by a first-order oracle machine (but not by Turing machine). It is known that there are degrees *between* 0 and 1 (Friedberg 1957, Sacks 1964; Simpson 1977 is a survey of the area). That is to say, there are classes of problems that are too hard to be solved by Turing machine and yet are less hard than some of the problems that a first-order oracle machine can solve. Here 'less hard' has the precise sense that while a first-order machine can solve any of the problems in such a class, an O-machine that is equipped only with an oracle for delivering the solutions ('Yes' or 'No') to problems in that class is unable to solve every problem that the first-order machine can solve. This notion of degrees lying between 0 and 1 seems to make sense of much of what Penrose says about what it is that he seeks (although certainly not of the specific claim that 'it need not be the case that human mathematical understanding is in principle

as powerful as *any* oracle machine at all'). For some degree between 0 and 1, the 'physics of mind' is exactly that hard. This is certainly a coherent position; and for all that anyone presently knows, such may in fact be the case.

However, it is now possible to toughen up the threat of *reductio*. Let *i* (for 'intermediate') be an O-machine of the sort just described (any arbitrarily selected one), and let *I* be the set of all machines with the same oracle as *i*. (That is to say, members of *I*, like first-order O-machines, differ in their programs, not in the primitive operations that each can perform.) Do mathematicians use, in ascertaining mathematical truth, a knowably sound procedure that can be executed by a machine in *I*? Apparently not, if Penrose's Gödelian argument is sound. For just as the argument can be pressed to yield the conclusion that mathematicians do not use a knowably sound procedure that can be executed by an α-order O-machine, it can equally well be pressed to show the same regarding machines in *I*. The two cases appear to be parallel in all relevant respects; in particular, it can be shown that there is no machine in *I* that can calculate all values of the halting function for machines in *I* (i.e. the function whose definition is just like that of $H(x,y)$ given above, except that the phrase 'the x^{th} Turing machine' is replaced by 'the x^{th} machine in *I*'). So what knowably sound procedure *can* mathematicians be supposed to use (recall that *i* was arbitrarily selected)? It is by no means clear how an upholder of the Gödelian argument might respond to this difficulty.

University of Canterbury
Christchurch
New Zealand
j.copeland@phil.canterbury.ac.nz

REFERENCES

Block, N. 1990. The computer model of the mind. In *An Invitation to Cognitive Science*, ed. D.N. Osherson, H. Lasnik. Vol.3: *Thinking*. Cambridge, Mass.: MIT Press.

Church, A. 1936. An unsolvable problem of elementary number theory. *American Journal of Mathematics* 58: 345-363.

Churchland, P.M., Churchland, P.S. 1983. Stalking the wild epistemic engine. *Noûs* 17: 5-18.

Churchland, P.M., Churchland, P.S. 1990. Could a machine think? *Scientific American* 262 (Jan.): 26-31.

da Costa, N.C.A., Doria, F.A. 1991. Classical physics and Penrose's thesis. *Foundations of Physics Letters* 4: 363-73.

Doyle, J. 1982. What is Church's thesis? Laboratory for Computer Science, MIT.

Dreyfus, H.L. 1992. *What Computers Still Can't Do: A Critique of Artificial Reason*. Cambridge, Mass.: MIT Press.

Fodor, J.A. 1981. The mind-body problem. *Scientific American* 244 (Jan.): 124-32.

Fodor, J.A. 1983. *The Modularity of Mind*. Cambridge, Mass.: MIT Press.

Friedberg, R.M. 1957. Two recursively enumerable sets of incomparable degrees of unsolvability (solution of Post's problem, 1944). *Proceedings of the National Academy of Sciences* (USA) 43: 236-238.

Geroch, R., Hartle, J.B. 1986. Computability and Physical Theories. *Foundations of Physics* 16: 533-550.

Guttenplan, S. 1994. *A Companion to the Philosophy of Mind*. Oxford: Blackwell.

Johnson-Laird, P. 1987. How could consciousness arise from the computations of the brain?. In *Mindwaves*, ed. C. Blakemore, S. Greenfield. Oxford: Basil Blackwell.

Kleene, S.C. 1967. *Mathematical Logic*. New York: Wiley.

Komar, A. 1964. Undecidability of macroscopically distinguishable states in quantum field theory. *Physical Review*, second series, 133B: 542-544.

Kreisel, G. 1967. Mathematical logic: what has it done for the philosophy of mathematics? In *Bertrand Russell: Philosopher of the Century*, ed. R. Schoenman. London: George Allen and Unwin.

Kreisel, G. 1974. A notion of mechanistic theory. *Synthese* 29: 11-26.

Penrose, R. 1989. *The Emperor's New Mind Concerning Computers, Minds, and the Laws of Physics*. Oxford: Oxford University Press.

Penrose, R. 1990. Précis of *The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics*. *Behavioural and Brain Sciences* 13: 643-655 and 692-705.

Penrose, R. 1994. *Shadows of the Mind: A Search for the Missing Science of Consciousness*. Oxford: Oxford University Press.

Penrose, R. 1996. Beyond the doubting of a shadow: a reply to commentaries on *Shadows of the Mind*. *Psyche: An Interdisciplinary Journal of Research on Consciousness* 2(23) [http://psyche.cs.monash.edu.au].

Pour-El, M.B. 1974. Abstract computability and its relation to the general purpose analog computer. *Transactions of the American Mathematical Society* 199: 1-28.

Pour-El, M.B., Richards, I. 1979. A computable ordinary differential equation which possesses no computable solution. *Annals of Mathematical Logic* 17: 61-90.

Pour-El, M.B., Richards, I. 1981. The wave equation with computable initial data such that its unique solution is not computable. *Advances in Mathematics* 39: 215-239.

Sacks, G.E. 1964. The recursively enumerable degrees are dense. *Annals of Mathematics* second series, 80: 300-312.

Scarpellini, B. 1963. Zwei unentscheitbare Probleme der Analysis. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 9: 265-289.

Searle, J. 1980. Minds, brains, and programs. *Behavioural and Brain Sciences* 3: 417-424.

Searle, J. 1989. *Minds, Brains and Science: the 1984 Reith Lectures.* London: Penguin.

Searle, J. 1992. *The Rediscovery of the Mind.* Cambridge, Mass.: MIT Press.

Searle, J. 1997. *The Mystery of Consciousness.* New York: New York Review.

Simpson, S.G. 1977. Degrees of unsolvability: a survey of results. In *Handbook of Mathematical Logic,* ed. J. Barwise. Amsterdam: North-Holland.

Stannett, M. 1990. X-machines and the halting problem: building a super-Turing machine. *Formal Aspects of Computing* 2: 331-341.

Turing, A.M. 1936 .On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* series 2, 42 (1936-37): 230-265.

Turing, A.M. 1938. Systems of logic based on ordinals. A dissertation presented to the faculty of Princeton University in candidacy for the degree of Doctor of Philosophy.

Turing, A.M. 1939. Systems of logic based on ordinals. *Proceedings of the London Mathematical Society* series 2, 45: 161-228.

Turing, A.M. 1950. *Programmers' Handbook for the Manchester Electronic Computer*. University of Manchester Computing Laboratory.

Vergis, A., Steiglitz, K., Dickinson, B. 1986. The complexity of analog computation. *Mathematics and Computers in Simulation* 28: 91-113.