Yesterday's Algorithm: Penrose and the Gödel Argument

§1. The Gödel Argument.

Roger Penrose is justly famous for his work in physics and mathematics but he is *notorious* for his endorsement of the Gödel argument (see his 1989, 1994, 1997). This argument, first advanced by J. R. Lucas (in 1961), attempts to show that Gödel's (first) incompleteness theorem can be seen to reveal that the human mind transcends all algorithmic models of it[1]. Penrose's version of the argument has been seen to fall victim to the original objections raised against Lucas (see Boolos (1990) and for a particularly intemperate review, Putnam (1994)). Yet I believe that more can and should be said about the argument. Only a brief review is necessary here although I wish to present the argument in a somewhat peculiar form.

Let us suppose that the human cognitive abilities that underlie our propensities to acquire mathematical beliefs on the basis of what we call *proofs* can be given a classical cognitive psychological or computational explanation (henceforth I will often use 'belief' as short for 'belief on the basis of proof'). These propensities are of at least two types: the ability to appreciate chains of logical reasoning but also the ability to recognise 'elementary proofs' or, it would perhaps be better to say, elementary truths, such as if a = b and b = c then a = c, without any need or, often, any possibility of proof in the first sense. On this supposition, these propensities are the realization of a certain definite algorithm or program somehow implemented by the neural hardware of the brain. Following Penrose let us call this algorithm A. A is a specifiable algorithm which manifestly 'contains' elementary arithmetic, since we obviously can appreciate the elementary truths of arithmetic as well as proofs of arithmetical theorems including Gödel's incompleteness arguments themselves. Since A is algorithmic, it could be implemented by a Turing machine, T, which would then be a pure computational model of our mathematical propensities. There are various ways we can imagine such an implementation. A very simple one is to think of T as taking the numbers of formulas representing arithmetical propositions, under some suitable coding scheme, as inputs and returning 1 (standing for 'humanly believable on the basis of proof') or 0 (for 'not humanly believable'). Then, of course, we know there must be some mathematical statement, G(A), for which T will not produce an output in a finite time, so long as A (equally, T) is consistent[2]. If we know the structure of A (or T) then G(A) can be explicitly constructed and, by its construction, it must be *true* (given that A is consistent). That is, seeing as we do grasp the arguments that Gödel and Turing used, we can understand that G(A) must exist and must be true (again, given that A is consistent). However since, by hypothesis, A is what underlies our entire set of propensities for accepting mathematical truths it will be impossible for us to believe (on the basis of proof) that G(A) is true (because for

[1] Although it seems that Gödel himself endorsed some form of this argument, his views on the subject are complex and somewhat conflicting. For details see Wang (1993).

[2] I call A consistent just in case it will never lead to belief on the basis of proof in a pair of inconsistent mathematical statements (e.g. 0=1 and ~(0=1)). The sentence G(A) can be thought of as the mathematical proposition which suitably encodes: 'Turing machine T will not halt when given this statement as input'.

us to believe that G(A) is true just means that T yields output 1 in a finite time when G(A)'s number is T's input). But this is contrary to what we have just established. The argument concludes from this contradiction that no algorithm can completely account for our ability to acquire mathematical beliefs on the basis of proofs.

This argument is obviously subject to criticisms from a variety of standpoints. These can be clearly brought out by an examination of the key propositions in the argument. First, we must suppose that we can know A, that we can understand this algorithm:

(P1) We can thoroughly understand the entire structure of A.

We must also know that A is in fact the algorithm that underlies our mathematical 'belief generator' else our knowledge of A is of no more than academic interest. The fact that we can prove that G(A) is true would, in the absence of this knowledge, just show that A was *not* what underlies our mathematical beliefs in which case the argument would stall. So:

(P2) We know that A underlies our propensities to acquire mathematical beliefs on the basis of proofs.

Supposing that P1 and P2 are true, we must further believe that A is consistent. In fact for the argument to be entirely successful we must *know* that A is consistent, for if A is not consistent then Gödel's result simply does not apply to it:

(P3) We know that A is consistent.

P2 and P3 each mask a slight ambiguity for they could fail to hold in *two* ways: we might not know that A underlies our mathematical belief generator (or that A is consistent) – even when it does (is) – because of some internal cognitive failing (perhaps, for example, because A was too complex for us to understand and/or assess) or we might not know that A underlies our mathematical belief generator (is consistent) – whether or not we believed it – simply because A does *not* (is *not*).

However, given each of P1 through P3 the argument is surely successful insofar as they cannot be all true. If we assume P1 and P2 then we cannot know that A is consistent, for if we did know this then G(A) would be a statement we would have to admit was true but which we could not admit was true (by P2). If P2 and P3 are assumed then P1 must be false, for if we *did* understand A we could then construct G(A) and by this construction procedure, which we already know that we understand, arrive at a statement we must but cannot endorse. P1 and P3 entail that we could not know that A underlies our mathematical belief generator, since otherwise we could generate G(A) via P1 and know that it was true via P3 which would contradict P2. Penrose's own route is this last one. Furthermore, he interprets the falsity of P2 in the stronger of the two ways outlined above; P2 is false simply because A does *not* underlie our mathematical belief generator.

§2. A Miraculous Selection of Errors.

Given this inconsistent triad, it is easy to list the basic ways that Penrose could be wrong: P1 might be false, P3 might be false, or P2's falsity might be due not to A failing to underlie our mathematical belief generator but rather to some sort of cognitive deficiency on our part. A very common response is to deny P3 (and Penrose goes to great lengths in his attempt to show that our mathematical belief generator is consistent and knowably so). George Boolos's remarks are particularly clear here (although these remarks are directed to Penrose (1989) I doubt that Boolos's opinion has changed):

> Penrose has said nothing that shows that we can recognize the truth of the Gödel sentence for ZF or for any other reasonable approximation to the whole of mathematics that we

> ourselves use. What we can see the truth of is this conditional proposition: The Gödel sentence for ZF is ZF-unprovable (and therefore true) *if* ZF is consistent. We cannot see that the Gödel sentence is true precisely because we cannot see that ZF is consistent. We may hope or believe that it is, but we do not know it, that therefore cannot see it ... Can we really 'see' that '0=1' is not sitting at the bottom of some lengthy, intricate, and ingenious proof perhaps involving concepts and arguments of a kind of which today we are completely unaware? (1990, pp. 655, 656, original emphasis)

However, it seems to me that such a response to Penrose is extremely puzzling. Our judgement as to the consistency of some system is *not* dependent upon that system's being able to prove its own consistency (i.e. generate a formula that states, e.g. '0=1' is not provable). For if that was the sole basis, how could we trust it? If the system was inconsistent, it could generate this formula as well (see Smullyan (1992, p. 109)). Furthermore, Boolos allows that we do know that certain systems, such as Peano Arithmetic, are consistent even though they cannot prove their own consistency. Presumably, we know this because we can see that a certain model satisfies the axioms of the system at issue[3], hence that they are true in that model and so must be consistent. Boolos speculates that it is possible that someday someone will prove that 0=1 but if this is a real possibility, as opposed to a mere statement of the possible fortunes of ZF when subjected to *our* mathematical belief generator, then we could use it to undercut our belief in the consistency of any mathematical system to which the application of either consistency proofs or a model require us to use our own mathematical belief generator in their assessment (which is to say, *all* such systems). I think this would be to go too far, certainly too far for Professor Boolos who has said about a system of elementary Peano arithmetic, here labelled Z:

> Corollary 2 shows that -Prov(0=1), which 'expresses the consistency of Z', is not provable in Z. ('If Z is consistent', one might be tempted to add. But Z is consistent) Boolos and Jeffrey (1980, p. 188)

So what prevents us from similarly knowing that A is consistent? Just Gödel's theorem? Now this is a kind of miracle: the consistency of an algorithm designed by evolution and culture, implemented in our brains, ready for study by all the cognitive sciences *must* be unknowable for reasons of pure logic! The argument is akin to some used in the physical sciences which depend upon the anthropic principle and it suffers the same problem: the argument provides no *explanation* of what it is about this particular algorithm that makes it such that it cannot be known to be consistent. The algorithm is obviously a complex one, but there is no proof that its complexity precludes us from understanding it or from being able to apply a model to it which satisfies its basic postulates. I can see no reason, for example, why A could not be made of a large number of modules each of which was relatively simple and whose interactions were one-by-one graspable. There is certainly no need to imagine that our algorithm is a straightforward emulation of ZF – its 'postulates' are likely to be numerous and more directly 'mathematical' as well as more obviously true than the axioms of set theory (hence Boolos's remark that 'none of the axioms of set theory forces itself on us the way "x + 0 = x" does' (1990, p. 656) is unlikely to be directly relevant to our own algorithm). The alternative view of Boolos *et. al.* is disturbingly reminiscent of what they like least about appeals to Gödel's theorem: that

[3] Obviously, relative consistency proofs (such as Gentzen's proof that arithmetic is consistent) will not help us here unless we already have faith in the consistency of the higher systems in which these proofs are couched.

there are significant limits set to the world and its workings simply by this theorem which we can know about without any empirical investigation.

A similar worry besets the alternative ways of evading Penrose's conclusion. It might be urged that we can simply deny that A is consistent. There is, after all, lots of evidence that human thought processes are somewhat less than a model of logical rigour. I think this objection is beside the point but it also raises an interesting issue. If the objection provides a general answer to Penrose's position then it must assert that *no* cognitive system capable of grasping elementary arithmetic and the Gödel argument can be consistent. This would be an astonishing result for which I can see no plausibility whatsoever, although it would be a delicious irony if the very thought processes requisite for mathematics – the bastion of logical correctness – were such as to guarantee that these processes were actually inconsistent. Curiously, there is a sort of proof that the formal system which, we are supposing, corresponds to our mathematical belief generator must be inconsistent, so long as we *believe* that it is consistent (which of course, we do). This is because the computationalist assumption is, in essence, that our beliefs are the theorems of a formal system (implemented, as it were, by T, as above). If we believe our mathematical belief generator is consistent then we must believe that we (via *it*) cannot prove that, say, 0=1. Let's formalize this crudely[4] as Bel(~Prov(0=1)), where 'Prov' is our provability predicate (see Boolos and Jeffrey (1980) or Smullyan (1992) for more on this class of predicate). The existence of such a predicate with regard to our own beliefs about mathematical proof seems uncontroversial, for the technical demands on provability predicates are minimal. But we are supposing that our mathematical beliefs correspond to the theorems of some appropriate formal system S_T (which could be implemented by the Turing machine T that we described above). So to say Bel(~Prov(0=1)) is to say that ~Prov(0=1) is a theorem of S_T, or $\vdash_{ST}$ ~Prov(0=1). But according to Löb's theorem (the reasoning behind which humans can obviously appreciate), provability predicates meet the following condition: if $\vdash_{ST}$ Prov(X) ⊃ X then $\vdash_{ST}$ X. By elementary logic, $\vdash_{ST}$~Prov(0=1) implies that $\vdash_{ST}$ Prov(0=1) ⊃ 0=1 and hence $\vdash_{ST}$ 0=1. So S_T must be inconsistent after all (or else, as Penrose contends, our mathematical belief generator cannot be algorithmically modelled). It is difficult to know exactly what to make of this argument – it is strangely unclear, for example, which side of the debate it favours. It *could* be taken as a new argument against computationalism or, no less remarkably, be taken to show that computationalism about the mind (or, at least, about belief generation) guarantees that any believer who takes himself to be consistent must be wrong! I believe, though, that any challenge this argument might pose will be avoided by the solution to be proposed below.[5]

[4] It is crude at least insofar as we should distinguish formulas from their numerical codes, but this does not affect the argument and makes everything typographically clearer.

[5] One might rightly complain that the argument seems to trade on ignoring the distinction between believing a mathematical proposition and believing a mathematical proposition on the basis of proof. Such an observation is beside the point however. For the argument appears to show that adding the axiom ~Prov(0=1) to S_T makes S_T inconsistent. Yet that is exactly our position: we believe that we are consistent in our mathematical beliefs and cannot seriously believe that this last belief actually makes us inconsistent! Furthermore, what prevents the possibility of a mind (perhaps rather more extensive than our own) understanding its own algorithm sufficiently to see that its associated formal system has a model?

On the other hand, if the ‘inconsistent system reply’ is restricted to *human* thought processes then it simply doesn’t address the real point of the argument. We would need to see some sort of reason, grounded in a theory of cognition rather than based on the magic of ‘pure logic’, why inconsistency has to arise as human cognition becomes more complex before this reply could regain our interest, and I can see no ground for such an assertion of ‘universal cognitive incompetence’.

Taking another tack, we could assert that P2 could be false because we cannot know that A underlies our mathematical belief generator even though in fact it does. But why should this be so? What prevents us from getting this knowledge? Would a researcher die if he or she got a little too close to this unattainable knowledge?! Or again, we could doubt that A is even understandable (this is Putnam’s (1994) route: ‘... there is an obvious lacuna: the possibility of a program we could write down but not succeed in understanding is overlooked! This is the mathematical fallacy on which the whole book rests.’) But what grounds could we give for seriously entertaining this possibility, especially as a *principled* answer to Penrose’s argument? It won’t do to say that it must be impossible to know A just because that would violate Gödel’s theorem (there is no *magic* in this theorem that can suddenly freeze our minds). We would want to know how the *world* enforces the limitation that Gödel’s theorem points to here. In sum, we should not rest content with a miraculous selection from a set of possible errors that always and inescapably undermines Penrose’s argument.

§3. Dynamic Cognition.

I would like to suggest that focussing on how the world works (or at least might work) could provide the proper answer to Penrose without having to succumb to the Gödel mysticism that simply outlaws certain sorts of knowledge prior to all investigation (it is ironic that Penrose’s critics are guilty of a failing – Gödel mysticism – which they so enjoy scornfully attributing to Penrose and Lucas). I think the fundamental problem is that the Penrose argument neglects the dynamic qualities of cognition. Of course, Penrose acknowledges that the Gödel sentence for any particular system can be deduced in a ‘broader’ system. This does not matter since he is engaged in a general *reductio ad absurdum* form of argument which begins with the assumption that a certain specifiable algorithm underlies our mathematical belief generator. Penrose is quite right to complain that it is then illicit to ‘change’ the algorithm in mid-argument (see Penrose 1994, pp. 80-82).

But real cognition is carried on by a system that is constantly changing and some of these changes are the result of the very cognitive activities in which it is engaged. Our mathematical judgements are not and have not been produced by some one, final algorithm but rather our current judgements are produced by our *current* algorithm. And it is the acquisition of mathematical knowledge which is itself one of the forces which alters the very algorithm which generates it. Certainly, it is evident that as children learn to understand the nature of mathematics the very procedures of reasoning about mathematics change in that process. In particular, we can suppose that the acquisition of a knowledge of our mathematical algorithm sufficiently detailed to enable the generation of its Gödel sentence would alter the algorithm (and, no doubt, many other factors affect the algorithm as well). To be sure, each such algorithm has its own Gödel sentence but we cannot *know* the algorithm which underlies our current mathematical propensities in sufficient depth to run into the Gödel trap without in that very process changing the algorithm.

This is not a mathematical fact or a misappreciation of the nature of the *reductio* form of argument. In the abstract, Platonic world of pure mathematical objects there is a Gödel sentence for the algorithm which at any given time underlies our mathematical capacities (if there is such an algorithm – I am only arguing here that Penrose's argument fails, not that computationalism succeeds). But our mathematical knowledge does not spring from that ethereal and altogether hypothetical world of Plato's Forms – it springs from the concrete, brain-bound systems that are in place and acting *right now*. And given that coming to understand our own algorithm would of necessity change that algorithm we can only know, as it were, yesterday's algorithm. We can know, perhaps, that yesterday we could not have accepted a certain sentence, S, which *today* we know to be true. Unfortunately (or not), gathering the knowledge required to come to this conclusion about S precludes S from being relevant to today's algorithm.

I have heard it argued that the fact that our algorithm can change shows directly that we are not Turing machines, for they cannot alter their algorithm. Well, of course we are *not* Turing machines (where, for example, is our infinite tape). The claim here is just that our mathematical belief generator, at any time, can be emulated by a Turing machine in, for example, the way described above. There may be however an *algorithmic* description of the proposed mechanism of change in our mathematical algorithm and this might suggest that the Penrose argument could be replayed at the level of this 'meta-algorithm'. But, in the first place, such a meta-algorithm is outside the scope of our mathematical belief generators. Secondly, and more important, its existence will not affect the central point of my suggested reply, which is that the acquisition of knowledge of G(A) must in the process alter A so that G(A) is no longer inaccessible to my *current* algorithm. The worry can be expressed as follows. Suppose that I can know that G(A) is a truth that cannot be believed by someone whose beliefs are generated by algorithm A (so long as A is consistent). If I can know that my current algorithm is, say, B and that by the meta-algorithm of algorithm generation B leads to A then I could know that *tomorrow* I will not be able to believe G(A) even though it is true. I could write this reasoning down and read it tomorrow and thus come to believe (on the basis of what could reasonably be regarded as proof) that G(A) is true, contrary to our supposition that A will underlie my mathematical belief system tomorrow. But according to my suggestion, it is impossible to come to know one's current algorithm (in such detail) without altering that algorithm. So we simply won't have the proper input to our meta-algorithm and thus the problem does not arise.

This suggestion accepts the basic structure of Penrose's argument. If our cognition was the result of a static, final algorithm then the argument would have merit (it would be attackable only on the grounds of 'reverse Gödelian mysticism' discussed above – technically entertainable perhaps but metaphysically very unsatisfactory). But there is not the slightest reason to think that our cognitive algorithms are static and much reason to suppose they are plastic, and that one of the major determinants of algorithmic change is the acquisition of knowledge by way of these very algorithms. Our knowledge of ourselves is thus always one step behind our true natures, but that fact cannot refute computationalism. In fact, it is what a computationalist who allows that computations must be formed and participate in a dynamic world would expect.

William Seager
University of Toronto at Scarborough

Bibliography

Boolos, George (1990) ‘On Seeing the Truth of the Gödel Sentence’ in *Behavioral and Brain Sciences*, 13 (4), pp. 655-6.

Boolos, G. and Jeffrey Richard (1980) *Computability and Logic*, 2nd edition, Cambridge University Press.

Lucas, John (1961) ‘Minds, Machines and Gödel’, *Philosophy*, 36, pp. 120-24. Reprinted in A. Anderson (ed.) *Minds and Machines*, Englewood Cliffs: Prentice Hall, 1964.

Penrose, Roger (1989) *The Emperor’s New Mind*, Oxford University Press.

Penrose, Roger (1994) *Shadows of the Mind*, Oxford University Press.

Penrose, Roger (1997) ‘On Understanding Understanding’, *International Studies in the Philosophy of Science*, 11 (1), pp. 7-20.

Putnam, Hilary (1994) ‘The Best of All Possible Brains?’, *The New York Times*, Nov. 20, § 7.

Smullyan, Raymond (1992) *Gödel’s Incompleteness Theorems*, Oxford University Press.

Wang, Hao (1993) ‘On Physicalism and Algorithmism: Can Machines Think?’, in *Philosophia Mathematica*, 1 (3).