GÖDEL'S THEOREM AND CREATIVITY

GRAHAM PRIEST
University of Queensland

1. Introduction

Contemporary cognitive science is motivated by the thought that mental activity is, in some sense—possibly a literal one—computational. It is a traditional complaint against this view that computers cannot be creative since they are merely rule-governed devices. The argument is heard less often than it used to be, since AI has got computers to do things that seem pretty creative (such as managing investment portfolios and organising wars); but it is unlikely to disappear whilst the nature of human creativity is not understood. Notwithstanding any of this, I think that the argument is pretty worthless as it stands: the notion of creativity is so vague that it is not at all clear what is being claimed when it is said that computers cannot be creative, and so not clear whether or not this is true. If the argument is to be worth discussing, there must, therefore, be some way of making it much more precise.

This might be attempted in many ways, by focusing on various kinds of, or topics of, creativity. One approach that offers some hope is to focus on the creativity involved in establishing new mathematical results. This is not because this is a *particularly* creative activity. Rather, it is because we have a reasonably well articulated understanding of what proof in mathematics is, and of what its properties are. An argument on just these lines was, in fact, given by J. R. Lucas in 1961 in a now (in)famous paper (Lucas, 1961). There, he argued on the basis of Gödel's Incompleteness Theorem that a mind will always be able to prove mathematical results that a machine cannot. Over the last 30 years the paper has occasioned a considerable literature,[1] mainly critical; and one might have thought the issue closed. But at the Turing conference at the University of Sussex in 1990 in a paper entitled 'Minds, Machines and Gödel: a retrospect' Lucas still stoutly defended his argument against the literature. Moreover, Roger Penrose (1989) has recycled the argument as the dialectical centre-piece of his book *The Emperor's New Mind,* which has stirred up the hornet's nest of critics again.[2] Since the argument has refused to lie down and die quietly, I think it worth thinking through it again from first principles. That is the purpose of this paper.

I will start with what I take to be a fair statement of Lucas' argument; I will then evaluate it. In the process, this will require a number of important clarifications, in particular, of what it is to give a proof. Both the original paper and many subsequent

[1] For a reasonably comprehensive but by no means complete bibliography see Boyer (1983).
[2] See *Brain and Behavioural Sciences*, **13**: 643–705.

T. Dartnall (ed.), Artificial Intelligence and Creativity, 107–115.

discussions are flawed by confusions concerning this notion. I do not intend this to be a scholarly paper, and so will not discuss the large literature. In fact, many of the standard objections to the argument can be seen to fail, or to be beside the point, once the argument is spelled out carefully. On the other hand, I make no claim to originality for a number of the points I shall make. What originality this paper has, it seems to me, is in spelling out the argument more clearly than is commonly done, with the attendant benefits of this.

2. Lucas' argument

So, what is Lucas' argument? It can be put very simply as follows. Take any mind, M, and computer, C:

> There is a mathematical truth of which C cannot give a proof
> but of which M can.
> Hence M is not C.

The argument is an instance of the indiscernibility of identicals (or rather, its converse, the difference of discernibles), and whilst one might have certain doubts about the unrestricted validity of this form of argument when the discerning property is intensional, these are of no concern here: the property of having the ability to do so and so is quite extensional. Hence there is no problem about the validity of the argument, and any problem must reside in the premise.

Is this true? Well, obviously not. The minds of dogs, newborn babes and mathematical illiterates cannot give any kind of mathematical proof. Of course, Lucas never meant his argument to apply to *any* mind. He has in mind, here, a mind of reasonable mathematical sophistication. So let us assume henceforth that M is of this kind, whilst noting that, if the argument works, mathematicians are not computers, but the rest of you still may be.

3. Output and ability

Given that M is a mind of this kind, why should we suppose the premise to be true? The answer is that it is supposed to follow from an application of Gödel's Theorem. We will look at the details in a moment, but first there is the preliminary question of how, given M and C, we are to decide what they can and cannot do. (Note that we cannot simply replace 'can' with 'does' in the argument, since there is no reason to suppose that M will actually prove the statement.) Fortunately we do not need to address the issue as far as M is concerned, but for C there is no escaping the question. The strategy adopted by Lucas is never spelled out very clearly, but in effect, I take it, it comes to this. We suppose C set in motion and demonstrate that the proof of the formula in question will not be given (or, if the machine is non-deterministic, will never be given whichever of the non-determined paths is followed); we conclude from this that C cannot give a proof of the formula.

There are two points that should be made about this strategy straight away. The first is that what is actually given as output by a computer depends, of course, on its input (in the case of a Turing Machine, the initial state of the tape). Hence, in the context of the argument, 'computer' should be taken to mean 'device + input'; and what the argument proves, if it works, is therefore that there is no machine-with-input that is identical to M.

The second point concerns the fact that the argument moves from what the machine *doesn't* do, to what it *can't* do. If it were a person that were in question here the inference would be hotly contested by compatibilists (soft determinists). Philosophers of this stripe point out that the fact that someone does not do something does not entail that they cannot (i.e. that they do not have the ability), and argue that this is so even if it is, in fact, determined that they do not do it. Since what is at issue here is whether computers may be minds, can we not make out the same case for C?

There are two possible strategies for reply here. The first is to confront the arguments for compatibilism head on. The second is to argue that even if compatibilism is true in general, there is something specific about the computer case which rules it out: the machine is, after all, doing everything it can. I do not think it appropriate to take up the first strategy here; and I do not know how to pursue the second successfully. So I intend to leave the issue there, assume that in this case *cannot* follows from *does not*, and simply note the weak point of the argument.

4. Gödel's theorem

We must now address the central question of what the formula is, of which C, supposedly, never gives a proof but of which M can. This is to be delivered by an application of Gödel's famous Incompleteness Theorem, although how, never gets spelled out very carefully. Let us start with Gödel's Theorem itself. Statements of this come in various shapes and sizes. The relevant one in the present context is as follows.[3]

> Let T be an axiomatic theory that can represent all recursive functions. Then there is a formula, φ, such that (i) if T is consistent φ is not in T and (ii) if the axioms and rules of T are intuitively correct, we can establish φ to be true by an intuitively correct argument.

The statement of the theorem contains various technical notions. To understand the discussion it is not necessary to have a complete grip on them, but it is necessary to have a reasonable understanding, so I will spend a little time explaining them.

A theory is a set of sentences of some (formal) language. In the case in question we must suppose the language to be a language with numerals, function symbols or predicates for addition, multiplication, etc. To be a theory the set must be closed

[3] I take it essentially from Priest (1987, chapter 3); a proof of the theorem in this form can also be found there.

under deducibility. That is, any logical consequence of its members must also be a member. It is pertinent to inquire what notion of deducibility is in question here. In fact, it is sufficient to assume that it validates little more than *modus ponens*, substitutivity of identicals and some simple quantifier inferences.

To say that the theory can *represent all recursive functions* is to say that certain arithmetic facts are in the theory. It suffices, as Gödel showed, to suppose that the theory contains statements of the basic properties of addition and multiplication.

To say that the theory is *axiomatic* is to say that there is a decidable set of axioms such that the members of the theory are exactly the logical consequences of the axioms. If this is true then the members of the set can be effectively generated (by applying the rules systematically to the axioms). In the jargon of recursion theory, they are *recursively enumerable* (re). Conversely, as Craig's Theorem shows, any theory that is recursively enumerable has a decidable set of axioms, and is therefore axiomatic. Hence, being axiomatic and being recursively enumerable are the same thing.

5. ... and computers

Gödel's theorem tells us that under certain conditions a true formula is not provable in a certain theory. But this does not yet give us what is required. We need a formula for which C cannot give a proof. How do we get this? Lucas hoped to obtain this in virtue of the close connection between computations and axiom systems.

Let us take some computational device, say a Turing Machine (though in virtue of Church's Thesis, essentially the same points will hold of any computational device). We can think of a computational state as a pair comprising the non-blank part of the tape (with the square being scanned marked in some way) and the machine state. Starting from some initial state, the computational state is modified by the application of certain effective rules to generate a potentially infinite set of states. By their nature the set of states generated is re. Or, to be slightly more precise, if we were to code the states arithmetically, the set of codes would be re.[4]

It should be noted that the existence of connectionist machines casts some doubt on this conclusion. Provided such a machine is set up as a discrete state system, as they normally are (and indeed, must be, if they are to be implemented on standard machines) then the conclusion holds. In theory, at least, however, they could be set up as analog machines, with outputs being continuous functions of inputs over real-valued time. In this case, there is no reason to suppose that the sequence of output states is re. Indeed, it is not even clear what this would mean anymore. Such a possibility therefore poses a very radical challenge to Lucas' argument. However, an analog notion of computation might threaten Church's Thesis itself, and would therefore occasion a radically novel situation that I do not wish to go into here.

[4] If the device is a non-deterministic Turing Machine then there is no reason why the set of actual states generated is re. But in this case we consider a suitable deterministic Turing Machine that generates all the possible states of the non-deterministic machine in some systematic order.

Leaving this issue aside, we can conclude that the set of computational states is re. But we cannot apply anything like Gödel's Theorem yet. To do this we need to know about what proofs the machine can give, that is, to look at its output. So let us suppose that the machine is hooked up to some output device. For want of a better word, let us call this a mouth. We can suppose that every time the machine state takes (a) certain determined value(s) some part of the tape which can be effectively determined is output through the mouth. The sequence of outputs can be effectively culled from an re set, and so is re too.

Now, of these outputs some will be proofs. Let the set of theorems of which these are the proofs be T. We are at last in a position to see whether we can apply Gödel's Theorem to this T. First, let us consider whether the general conditions are satisfied. That is, can T represent all recursive functions; is it deductively closed; and is it re?

6. Representability and deductive closure

Is there, for a start, any reason to suppose that T can represent all recursive functions? For all we have said so far, of course not. Recall, however, that we are in the process of attempting to show that C is not M, where M is a mind of reasonable mathematical sophistication. Now a mind of such sophistication can clearly establish the basic properties of addition and multiplication. Hence, if C cannot do the same it is not M. We may therefore suppose, without loss of generality, that T can represent all recursive functions.

Next, we turn to deductive closure. Is T deductively closed? Again, for all we have said so far, no. We can argue, as before, however, that we can restrict ourselves to the case where T is deductively closed: the set of theorems that M can establish is deductively closed, and so if T is not deductively closed C is not M. It might be doubted that the set of theorems a mind can establish is deductively closed. The set of theorems any human mind *will* establish is, of course, finite, and so not deductively closed. But the set of theorems a human mind *can* establish would seem deductively closed, at least in principle: for example, if a mind can establish α and $\alpha \rightarrow \beta$ then it can establish β. The principle is that the mind is given sufficient time and secondary memory space—which is a principle we also have to apply to the computer, of course.

7. Recursive enumerability

Finally let us turn to the crucial question of whether T is re. The output of C is re. To get at T we need to disentangle the set of things given as proofs from the rest of the output, and then extract T from this. For T to be re both of these procedures must be effective. If they are not, there is no reason why T should itself be re. An re set—such as the natural numbers—can have non-re subsets—e.g. the set of codes of true arithmetical statements.

Let us start with the question of whether we can effectively disentangle proofs from the rest of the output. To answer this we have to address the question of what, exactly, it is to give a proof. In fact, a number of things might be meant by this. Let us consider them in turn, and see whether any of them will do what is required.

First, the obvious sense in which M can give a proof is an intensional one. M produces a statement with the intention that it be understood in a certain way, namely as establishing a certain mathematical statement as true. Now, it is not clear that a computer can have intensional states in the same way. But someone who denied this would have a much more fundamental objection against the identity of M and C. So suppose that it can. Is there an effective procedure for telling when the output of the computer is being given as a proof in this sense? Clearly not; the computer might not be giving a proof at all: it might be joking, lying or just doodling. And there is certainly no *effective* procedure for ruling out these cases. If this is what is meant by giving a proof the argument therefore folds here.

Alternatively, we might interpret the notion of giving a proof simply as the *outputting* of a proof (whatever intention—if any—is behind it). After all, M does this too. But in this case the output (which might just be a sequence of 1s and 0s) has no intrinsic meaning at all; neither, therefore, does the question of how one can recognise a proof in the output. Such a question makes sense only relative to some scheme for decoding the output. We can get around this problem as follows. M will, presumably, speak some language and so can give a proof of the formula in question in that language, or even in some canonical part of it, say a certain formal language, L. If C cannot even output strings which are, syntactically, sentences of L then it cannot do what M does. Hence C is not M. Thus, we may restrict our attention to the case where C outputs sentences of L, and take the decoding scheme to be the normal semantics of L. Can we now effectively recognise a proof in L when one is output? This depends on still further disambiguations.

A proof is a deductive argument; but obviously not all deductively valid arguments are proofs ($0 = 1$; hence $1 = 0$). Suppose we take a proof to be a sound deductive argument, i.e. a valid argument with true premises. In this case there is no way that recognition of a proof can be effective. For even assuming that the question of whether or not an inference is valid is decidable, the question of whether or not a premise is true is certainly not. Hence, again, if we use this notion of proof the argument folds.

A final sense of the notion of proof (and the only one that, as far as I can see, is capable of advancing the argument further) is that according to which a proof is a sequence of statements that appears to us (or, rather to an L speaker, say M) to be sound. This, plausibly, *can* be recognised effectively. Henceforth I shall interpret proof (and cognate notions such as consistency) in this way. We may now take it that the set of proofs is re. Assuming that L is such that we can effectively determine the theorem from the proof, T is re too. The assumption is not toothless. It rules out referring to the conclusions by names such as 'Euclid's Prime Theorem'. But provided we take it, as we may, that all statements are spelled

out (e.g. $\forall x \in \mathrm{N} \exists y \in \mathrm{N}(y > x \wedge \mathrm{prime}(y))$), and not just named, the assumption is satisfied.

8. Consistency

Having seen that the general conditions of Gödel's Theorem are satisfied on one (and only one) understanding of proof, we must now see whether the particular conditions for parts (i) and (ii) of it are satisfied (on the same understanding). Let us take things in that order. To establish that φ is not provable, we need to establish that T is consistent. Is there any reasons to suppose so? Clearly not. Nor will it help to argue that we can restrict ourselves to those computers where T is consistent, on the grounds that an inconsistent machine cannot be M. Real minds are frequently inconsistent in the sense of providing proofs of inconsistent things. Most people have produced a proof that 0=1 when doing elementary algebra; and nearly everybody has applied the algorithm for adding two numbers (which is a proof of sorts) and got the wrong answer.

Are there any avenues of repair here? One argument Lucas uses in his original paper is that if T is deductively closed and inconsistent then it contains a proof of everything; and therefore C cannot be M because people, even if inconsistent, do not offer proofs of everything. This reply, however, will not work. There is no reason to suppose that the logic in question is explosive. It may well be paraconsistent; in which case triviality does not follow from inconsistency. (As I observed, to prove Gödel's Theorem, we need to make very few assumptions about what the logic in question is.)

Another move that Lucas makes in his original paper is to argue that although people may be inconsistent, they are not essentially so, in the sense that when they discover that they have proved inconsistent things they will take back at least one of the proofs. However, this does not really help, since there is no reason to suppose that C may not be essentially consistent in the same sense. It might be suggested that we grant this, but take for T the set of all those theorems that are proved and never taken back, which is consistent. Whether or not this is so, the argument now collapses. Even granting that taking back is something that can be effectively recognised when it occurs, there is no effective way of telling when a theorem is *going* to be taken back. There is therefore no longer any reason to suppose that T is re.

The only way, it seems to me, that offers any hope of getting T to be consistent is to suppose that M (and so any C which is supposed to be M) is not only a mathematical mind but an ideal mathematical mind, that never makes mistakes of any kind: either of memory, inference, judgment or output. But this is sufficient to destroy the argument. After all, the only candidate for a mind of this kind is God's. So at best, we have a (theo)logical proof that God is not a computer.

But I am skeptical of even this repair. I doubt that even the ideal mathematical mind is (mathematically) consistent. Simple Peano Arithmetic may be consistent

(we hope); but once one passes beyond these bounds contradictions are wont to arise. Take Berry's Paradox, for example. There is only a finite number of numerical descriptions of some preassigned length, say 100 words. Hence there must be numbers that are not so described. In particular, the least such is not so described. But we have just so described it. This and many similar logical paradoxes threaten consistency even for an ideal mind.

It might be replied that there is something wrong with this proof. I do not think this is the case, though I shall not enter into the discussion here. Let me just say that if there is, we have not yet found it (at least to most people's satisfaction). Alternatively, it might be argued that the phenomenon of logical paradoxes is irrelevant in the present context since we may take L to be non-self-referential. Nothing, however, could be further from the truth. The proof of Gödel's Theorem makes notorious use of self-reference. Indeed, the very theorem that is claimed to be unprovable is, intuitively, a logical paradox. Roughly, the sentence in question, φ, says of itself that it is not provable. Now suppose that it is false. Then it is provable, and so true. Hence it is true, and so unprovable. But we have just proved this; hence it is provable. Thus, it would seem, we cannot apply Gödel's Theorem to infer that φ is unprovable, since the theory in question is inconsistent; and the inconsistency is precisely $\varphi \wedge \neg\varphi$.[5] In any case, there is therefore no hope of trying to argue that the logical paradoxes are an irrelevant phenomenon that may be safely cordoned off.

9. The mathematician's proof

Next, and finally, we can turn to the question of whether φ is indeed provable by M, in other words, whether part (ii) of the theorem can be applied. In order for this to be so we require that the axioms and rules of T be intuitively correct. Are they? First, what are they?[6] With a bit of rational reconstruction, we can always suppose that the sole rule of inference is *modus ponens,* which is certainly intuitively correct. What of the axioms? Given any proof in L that C comes up with we can effectively pick out its ultimate premises, just as we picked out its ultimate conclusion. By our assumption of what a proof is, each ultimate premise is intuitively correct. This does not quite give us what we want, however. Since the collection of proofs is re, the collection of ultimate premises is re, but need not be decidable, and so need not be a decidable set of axioms for T. However, Craig's Theorem (Craig, 1953) shows us how to construct a decidable and logically equivalent set. If α is the nth member of the enumeration, we simply take $\alpha \wedge \ldots \wedge \alpha$ (with n conjuncts) as an axiom. Clearly, if α is intuitively correct, so is this. So T does have a set of axioms

[5] For a further discussion of all these issues see Priest, 1987.

[6] It is sometimes suggested that Lucas, J. R.' argument fails since M may not be able to determine what the axioms of T are, and so may not be able to 'formulate its own Gödel sentence', φ. This is an *ignoratio.* It is sufficient for the argument that φ exists; it is not necessary that M can determine that φ is the Gödel sentence in question.

all of which are intuitively correct, and we can apply the second part of Gödel's Theorem to establish that α is provable by M.[7]

10. Conclusion

We have now considered the whole of Lucas' argument. By carefully spelling it out and trimming it we have seen how a number of its problems can be avoided. But, as we have also seen, in the last analysis it fails: however one spells out the notion of proof concerned, the argument breaks down somewhere. Machine creativity breathes again.

Acknowledgements

I would like to thank the editor, Terry Dartnall, for his helpful comments on an earlier draft of the paper.

References

Boyer, D.: 1983, Lucas, Gödel and Astaire, *Philosophical Quarterly*, **33**: 147–59.

Craig, W.: 1953, On Axiomatizability within a System, *Journal of Symbolic Logic*, **18**: 30–2.

Lucas, J. R.: 1961, Minds, Machines and Gödel, *Philosophy*, **36**: 112–27.

Penrose, R., R.: 1989, *The Emperor's New Mind: concerning computers, minds and the laws of physics*, Oxford University Press, Oxford.

Priest, G.: 1987, *In Contradiction*, Nijhoff, The Hague.

[7] It is sometimes argued that M may not be able to establish φ on the ground that M may not be able to establish that T is sound. (Usually, the point is put in terms of consistency since people fail to distinguish between these notions.) However, if T is defined in the way required for the rest of the argument to work (as the set of theorems whose proofs are intuitively correct), the soundness of T can be established in an intuitively correct way quite trivially, as follows: all the axioms are true, all the rules of inference are valid; so (by recursion) all the theorems are true.