COMMENTS AND CRITICISM

WHAT GÖDEL'S INCOMPLETENESS RESULT DOES AND DOES NOT SHOW

In a recent paper, Storrs McCall[1] adds another link to a chain of attempts to enlist Kurt Gödel's incompleteness result as an argument for the thesis that human reasoning cannot be construed as being carried out by a computer. McCall's paper is undermined by a technical oversight. My concern however is not with the technical point. The argument from Gödel's result to the no-computer thesis can be made without following McCall's route; it is then straighter and more forceful. Yet the argument fails in an interesting and revealing way. And it leaves a remainder: if some computer does in fact simulate all our mathematical reasoning, then, in principle, we cannot fully grasp how it works. Gödel's result also points out a certain essential limitation of self-reflection. The resulting picture parallels, not accidentally, Donald Davidson's view of psychology as a science that in principle must remain "imprecise," not fully spelled out. What is intended here by 'fully grasp', and how all this is related to self-reflection, will become clear at the end.

I should add that the full implications and the significance of Gödel's result are often misunderstood in other important respects. The result is rightfully conceived as revealing an essential limitation of formal deductive systems: in any deductive system that satisfies the non-negotiable requirement of effectiveness (the existence of an effective procedure for deciding whether any purported proof is a proof), and some minimal adequacy conditions (the capacity to represent certain elementary arithmetic, or combinatorial notions), there are sentences that are neither provable nor refutable in the system. What this formulation hides, and what the proofs current in most textbooks miss, is the constructive nature of the original proof, which does not appeal to the truth concept. It is by virtue of this feature that the result has undermined Hilbert's finitistic program in the foundation of mathematics. Although Gödel's theorem is an independence result, it is not a radical one, like the independence of

[1] "Can a Turing Machine Know that the Gödel Sentence Is True?" this JOURNAL, XCVI, 10 (October 1999): 525-32. The two previous most noted attempts have been John Lucas, "Minds, Machines and Gödel," *Philosophy,* XXXVI (1961): 112-27, and Roger Penrose in *The Emperor's New Mind* (New York: Oxford, 1989) and *Shadows of the Mind* (New York: Oxford, 1994).

0022-362X/00/9708/462–70

the continuum hypothesis; rather the opposite: we know, by good reasons as any, the truth value of the independent sentence. And it is this "weak" character which endows Gödel's result with its foundational significance and with its force against Hilbert's program. Finally, while the theorem has profound effects on the foundations of mathematics and on our view of formal systems, it has had very little effect on mathematical practice. These and other related points are beyond the scope of the present reflection; I hope to address them elsewhere.

Let me first refine, for clarification, what I called above the *no-computer thesis*. The claim is that no computer program that produces, or searches for, proofs can achieve what is achievable by mathematicians who engage in proving theorems by mathematical reasoning. This thesis is of great interest in itself, and I shall not be concerned here with the broader implications it might have in the philosophy of mind. The argument for the thesis consists in the following two moves. (I) A theorem prover (that is, a computer program that proves theorems) can be seen as finding theorems in some formal deductive system. (II) For any formal deductive system there is a sentence (for example, the so-called Gödel sentence that says of itself that it is unprovable in the system) which cannot be proved in the system but whose truth can be inferred by mathematical reasoning. (I) is not problematic. A theorem prover generates a recursively enumerable set of sentences (the general reader may ignore this technical detail without missing much), and the closure of this set under logical implications can be seen as the set of theorems of some formal deductive system (we can actually assume that the theorem prover generates the closure of this set). My concern here will be with (II). At this point a minimal sketch of Gödel's proof may not be redundant, given that some elementary misunderstandings can still be found in the philosophical literature.[2]

Usually, Gödel's result is stated for number-theoretic systems, and its proof uses an encoding of sentences, sentence parts, and proofs by natural numbers: the so-called Gödel numbers. This makes it possible to express within a formal theory of numbers the basic syntactic, and

[2] Daniel C. Dennett, for example, in chapter 13 of *Brainstorms* (Cambridge: MIT, 1986) (I thank Ihsan Dogramachi for calling my attention to this chapter), speaks of Gödel sentences of Turing machines, including (on page 265) the Gödel sentence of the universal Turing machine. This makes no sense. A Gödel sentence is defined with respect to a deductive system, or with respect to a theorem prover, which a Turing machine is in general not. "The Gödel sentence of the universal Turing machine" is either meaningless, or—if by stretch of imagination the "universal Turing machine" is supposed to cover all deductive systems—a false, or even contradictory sentence.

proof-theoretic, notions of the system itself. But the result can be stated and proved with respect to any system that can handle its own syntax—for example, some theory of finite strings. Let T (referred to also as 'theory') be a system based on a first-order language, whose intended interpretation is over the universe of natural numbers. (The universe may also include other objects, provided that the subclass of natural numbers can be defined by a suitable formula in the language.) The language has the usual vocabulary: names for addition and multiplication, a name, $\underline{0}$, for the number 0, and a name, $s(\ )$, for the successor function. Standard names for other numbers are obtained, in the usual way, by iterating $s(\ )$; for example, the name, $\underline{4}$, for the number 4 is $s(s(s(s(\underline{0}))))$. The axioms should be sufficient for deriving certain basic properties, which enable one to represent various syntactic notions inside the system, where formulas and proofs are "identified with" their Gödel numbers. This, it turns out, is a rather weak requirement, which is satisfied by systems considerably weaker than Peano's arithmetic.[3] Under these conditions, there is a formula: $Proof_T(y,x)$, which says that y is a proof, in T, of x, such that the following holds (where 'T $\vdash$ ...' means that, in T, ... is provable, $\underline{n}$ is the standard name of n, and proofs and sentences are identified with their Gödel numbers).

(1) *If* n is a proof, in T, of m, then T $\vdash Proof_T(\underline{n},\underline{m})$

(2) *If* n is not proof, in T, of m, then T $\vdash \neg Proof_T(\underline{n},\underline{m})$

At the heart of Gödel's proof is the fixed-point theorem, also known as the diagonal theorem, which says that, for any given formula, $\phi(x)$, with one free variable, there is a sentence α such that the following biconditional is provable in T:

$$\alpha \leftrightarrow \phi(\text{`}\alpha\text{'})$$

where 'α' is the standard name of α's Gödel number. (Moreover, the proof of the theorem yields an actual construction of α, as well as of the proof of the biconditional). We can think of α as a sentence that, referring to itself via its Gödel number, says of itself that it has the property $\phi(\ )$. Letting $\phi(x)$ be the formula $\neg\exists y Proof_T(y,x)$, we get a sentence γ that says of itself that it is not provable (in T):

[3] For example, it is sufficient that the following be provable: all true inequalities $\underline{m} \neq \underline{n}$, all true equalities $\underline{m} + \underline{n} = \underline{k}$, and similarly for multiplication; also, for each n, the formulas: $x < \underline{n} \leftrightarrow x = \underline{0} \vee x = \underline{1} \vee ... \vee x = \underline{n\text{-}1}$ and $x < \underline{n} \vee x = \underline{n} \vee \underline{n} < x$ (where the ordering, $<$, is expressed either through some formula, or is named by a relation symbol of the vocabulary).

(3) $T \vdash \gamma \leftrightarrow \exists y Proof_T(y, \ulcorner \gamma \urcorner)$

Now, if n is any proof (in T) of γ, then, combining it with the proof of the biconditional, we get a proof of $\neg\exists y Proof_T(y,x)$; on the other hand, (1) gives us a proof of $Proof_T(\underline{n}, \ulcorner \gamma \urcorner)$. Since these two sentences contradict each other (in first-order logic), we get a proof of a contradiction. Therefore, if T is consistent, γ is not provable in it. But the unprovability of γ is formally stated as $\neg\exists y Proof_T(y, \ulcorner \gamma \urcorner)$; hence, we have just inferred $\neg\exists y Proof_T(y, \ulcorner \gamma \urcorner)$ from the assumption that T is consistent. Combining this with the biconditional in (3), we infer γ. Now the claim that T is consistent can be formalized in T, say, as the sentence that says that there is no proof of $\underline{0} \neq \underline{0}$: $\neg\exists y Proof_T(y, \ulcorner \underline{0} \neq \underline{0} \urcorner)$. Call this sentence Con(T). Then the argument we have just given amounts to a derivation (outside T) of:

(4) $Con(T) \rightarrow \neg\exists y Proof_T(y, \ulcorner \gamma \urcorner)$

If T is minimally adequate, this derivation can be formalized in it, making (4) a theorem of T. ('Minimally adequate' is still rather weak, considerably less than Peano's arithmetic.) Assuming from now on that T is minimally adequate, it follows that, if T is consistent, Con(T) is not provable in it; otherwise, we would get from (4) a proof of $\neg\exists y Proof_T(y, \ulcorner \gamma \urcorner)$, hence a proof of γ. This is the content of Gödel's second incompleteness theorem.

The reason we accept the Gödel sentence, γ, as true is that it is implied by the consistency of T. Rather than γ, let us therefore focus on Con(T) as the unprovable sentence. (Indeed, γ is provably equivalent to Con(T): we have just indicated how to get γ from Con(T) and, in the other direction, Con(T) follows from the unprovability of any sentence, in particular from $\neg\exists y Proof_T(y, \ulcorner \gamma \urcorner)$.) The basis of all philosophical take-offs from Gödel's result is thus the following simple fact: for any minimally adequate formal deductive system, T, if T is consistent then its being consistent is expressible in T but unprovable in it.

Before proceeding, a short remark on McCall's oversight. Gödel's theorem has also a second part, proved by similar arguments, which says that, if T is ω-consistent, then $\neg\gamma$ is not provable in it; ω-consistency is the following condition that is stronger than consistency: whenever $\exists x \alpha(x)$ is provable (where 'x' ranges over the natural numbers), it is not the case that $\neg\alpha(\underline{n})$ is provable for all n. McCall observes that it is unknown whether ω-consistency can be replaced in the second part by consistency. In fact, it is not difficult to show that it *cannot,* unless T is not ω-consistent. But all this is neither here nor there, for the use of ω-consistency does not make for a new or

improved argument. We can express in the language of T the statement that T is ω-consistent and get an analogous conditional to (4): ω-Con(T) $\rightarrow \neg\exists y Proof_T(y, \ulcorner\neg\gamma\urcorner)$. This, again, is provable in T, under mild adequacy assumptions. The situation is similar to that of the first part. If T is ω-consistent, this fact cannot be proved in it. Since consistency is a weaker and more natural property than ω-consistency, the argument for the no-computer thesis is better if it is based on the unprovability of Con(T). Let me now turn to this argument.

Any deductive system, T, that formalizes mathematical reasoning must leave something outside: its own consistency, expressed as Con(T), cannot be derived in it. As we remarked above, in principle, a computer that proves theorems generates proofs in some formal system. (This is true of the standard notion of computers. It may fail of computers that incorporate physical elements that, by virtue of physical theory, compute nonrecursive functions. I shall not pursue this possibility here.) If the computer can "know" only what it can prove, then it cannot know that it is consistent (that is, never produces a contradiction), something that we qua mathematicians seem to know. We may know it, for example, on the basis of a soundness argument, by appealing to the concept of truth: all the axioms of T are true, in the intended interpretation of the language in question, and all the inference rules preserve truth; therefore, whatever we prove from the axioms by applying inference rules, must be true, a fortiori, noncontradictory.

There is now a philosophical temptation to argue that our access to the concept of truth gives us an edge over any computer. The thought appears attractive, given the traditional distinction between syntax and semantics and the view that sees the computer as carrying out mere syntactic manipulations. But this line fails. In the same sense that computers can prove, they can handle semantic notions. The argument from soundness can be formalized in a richer system, in which we can define a truth predicate for the language of T; the clauses of the truth definition are deducible as theorems in the second system, and so is the general statement that every sentence provable in T is true, hence noncontradictory. A computer that proves theorems in the richer system "knows" that T is consistent.

The appeal to the truth concept is therefore misplaced. Still, the argument seems to carry through. As a rule, mathematicians take the consistency of the framework within which they work for granted. (This is different from physics where *faute de mieux* contradictions are tolerated as temporary anomalies, to be resolved later.) If T is obtained by formalizing a framework of mathematical reasoning, those who accept the framework accept, at least implicitly, that T is consis-

tent; that is, they accept Con(T). This acceptance is not merely some state ascribed to the mathematician for philosophical convenience. Had it been so, one might have stipulated "by definition" that the theorem prover that works in T "knows" that T is consistent. In principle, the acceptance can be manifested by using Con(T) in mathematical practice; a mathematician will reject as false any hypothesis he is trying to establish, if he finds that the hypothesis implies the inconsistency of the system within which he is working. Which is what a theorem prover is incapable of. Observe, moreover, that the mathematician's ability derives from a certain capacity for self-reflection. Reflecting on one's mathematical reasoning and realizing that it can be formalized as T, one takes Con(T) as something that can be legitimately used in mathematical reasoning.

We can extend T by adding Con(T) as an additional axiom. But then the consistency of *that* system is beyond *its* reach, and so on. We thus have a systematic way of extending any acceptable formal system to a stronger acceptable system. That at any single stage we can view ourselves as generating proofs in some formal system does not invalidate the no-computer argument, for the fact remains that no single system, hence no single computer, can faithfully capture what we can do. Or does it?

A closer look reveals the hole in this line of thought. Our alleged edge over the computer that generates proofs in T consists of our knowing that T is consistent. But how do we know this? If I believe that my mathematical reasoning is immune to contradiction, and *if I recognize that the formal system* T *represents faithfully my reasoning*, then indeed I should accept Con(T). But T may represent my reasoning without me recognizing this fact. Call a formal system (theory) *recognizably valid* if it can be recognized as a theory whose theorems should be accepted, because it represents faithfully our mathematical reasoning (or because it derives from some such representations). Then, a recognizably valid T cannot capture our full mathematical reasoning; because we shall reason that T is consistent, and *this* reasoning is something T cannot deliver. Yet, for all we know, there may be a system, call it T*, which is not recognizably valid and which in fact represents, in a very complicated way, all our mathematical reasoning. A computer that generates T*-proofs simulates all our doings in mathematics, in all the systems we can create. Suppose we are told that a certain complicated theorem prover, consisting of millions of instructions, produces exactly all that human mathematical thinking is capable of. Whoever believes it will probably conclude that the algorithm is consistent (cannot produce a contradiction). But what grounds can we have for such a belief? Say we believe it on the

authority of some superior being (create, dear reader, your own favorite story). Or say we believe it on the inductive force of empirical evidence: somehow we got this program (again, create your own scenario), which has proved all major mathematical theorems known to us, and whose output, as far as we checked it, contained no errors. But such a belief does not derive from *mathematical* reasoning; hence it is not a counterexample to the computer's alleged ability to simulate human mathematical reasoning.

One should note here that the ability to form beliefs on someone's word, or on empirical evidence, does not by itself constitute a principled difference between humans and computers. For a computer can be provided with channels through which it is continuously fed environmental input, which can affect its deductive processes. It is not clear to what extent the no-computer argument can be extended to this case (by using a suitable version of the diagonal theorem). In any case, this goes beyond the topic of Gödel's incompleteness result. Our present discussion does however cover the case of a mathematical community, whose members exchange proofs and ideas. For we can model such a community as a network of intercommunicating computational modules, a system that constitutes one giant computer program. The particulars of the program do not matter, as long as the total output is recursively enumerable, which it will be if each module behaves according to some algorithm. The same goes for a single human, who can be construed as performing several tasks in parallel; there will be several subroutines that form one overall program.[4] None of this affects the arguments.

Now in order to believe the consistency of some program on *mathematical* grounds, we have to understand how the program works, just as we understand that a certain algorithm performs addition, or just as we understand that Peano's axioms are true in the model of natural numbers and that the inference rules preserve truth. Yet, when it comes to T* such understanding is beyond our capacity; the system, or the computer, is too complex for us to grasp as an object of mathematical thought. This is not merely a question of length or

[4] Dennett, *op. cit.*, tries to defuse the implications of the no-computer thesis by arguing that human behavior may represent several Turing machines, running in parallel, such that the Gödel sentence that is unprovable by one is provable by another. The argument, it can now be seen, misfires. If the total output is recursively enumerable, we have in principle one program. And if it is not, then it remains for the functionalist to explain how mathematical reasoning, performed according to some functionalist model, produces a nonrecursively enumerable set. Dennett also argues that the same physical device can implement, according to different interpretations, many different programs. This is trivially true, but of no help to the functionalist, who is after all committed to there being a particular computational model that explains mental activity.

of our memory getting worn down, but a limit that exist in principle, since it is implied by Gödel's theorem. (It is similar to upper and lower bounds that are established in complexity theory.)

Physical limitations on our performance exist, of course, in general. Although we have a good grasp of the formal system of Peano's arithmetic and recognize it as valid, certain deductions in it are beyond us, because of memory wear, or shortage of time. Still we idealize and say that, were it not for these limitations, humans could prove each theorem of Peano's arithmetic. When it comes to T*, we can say that the theorems of T* coincide with what is humanly provable in principle, that is, disregarding physical limitations on human performance. But even under these idealized conditions, we could not, as a matter of principle, recognize T* as valid. This would be a structural inherent feature, rather than a matter of wear and tear.

And if such is the case, then we (qua mathematicians) are machines that are unable to recognize the fact that they are machines. As the saying goes: if our brains could figure out how they work, they would have been much smarter than they are. Gödel's incompleteness result provides in this case solid grounds for our inability, for it shows it to be a mathematical necessity. The upshot is hauntingly reminiscent of Spinoza's conception, on which humans are predetermined creatures, who derive their sense of freedom from their incapacity to grasp their own nature. A human, namely, Spinoza himself, may recognize this general truth; but a human cannot know *how* this predetermination works, that is, the full theory. Just so, we can entertain the possibility that all our mathematical reasoning is subsumed under some computer program; but we can never know how this program works. For if we knew we could diagonalize and get a contradiction.

The discussion above brings to the fore the role of mathematical self-reflection: A mathematician realizes by self-reflecting on his own reasoning that his inferences can be formalized by such-and-such deductive system. From which the mathematician can go on to infer that the system in question is consistent. The self-reflection is therefore itself part of mathematical reasoning. Gödel's result shows however that self-reflection cannot encompass the whole of our reasoning; that is, it cannot comprehend itself within its horizon. There is, indeed, prima facie plausibility to the general claim that we can reflect on only part of our rational apparatus; for the very act of self-reflection remains outside the picture it reveals. Gödel's result lends mathematical grounds to this intuitive plausibility.

Finally, note how the impossibility of full grasp of our own "reasoning program" accords, in spirit, with Davidson's claim that psychology must remain on a certain level of imprecise generalities. We may speculate how our reasoning works and we may confirm some general aspects of our speculation. But we cannot have a full detailed theory. The reason for the impossibility is the same, both in the case of mathematical reasoning and in the case of psychology, namely, the theoretician who constructs the theory is also the subject the theory is about.

HAIM GAIFMAN

Columbia University