pleteness here is part and parcel of TNT; it is an essential part of the nature of TNT and cannot be eradicated in any way, whether simple-minded or ingenious. What's more, this problem will haunt any formal version of number theory, whether it is an extension of TNT, a modification of TNT, or an alternative to TNT. The fact of the matter is this: the possibility of constructing, in a given system, an undecidable string via Gödel's self-reference method, depends on three basic conditions:

(1) That the system should be rich enough so that all desired statements about numbers, whether true or false, can be *expressed* in it. (Failure on this count means that the system is from the very start too weak to be counted as a rival to TNT, because it can't even express number-theoretical notions that TNT can. In the metaphor of the *Contracrostipunctus,* it is as if one did not have a phonograph but a refrigerator or some other kind of object.)

(2) That all general recursive relations should be *represented* by formulas in the system. (Failure on this count means the system fails to capture in a theorem some general recursive truth, which can only be considered a pathetic bellyflop if it is attempting to produce all of number theory's truths. In the *Contracrostipunctus* metaphor, this is like having a record player, but one of low fidelity.)

(3) That the axioms and typographical patterns defined by its rules be recognizable by some terminating decision procedure. (Failure on this count means that there is no method to distinguish valid derivations in the system from invalid ones—thus that the "formal system" is not formal after all, and in fact is not even well-defined. In the *Contracrostipunctus* metaphor, it is a phonograph which is still on the drawing board, only partially designed.)

Satisfaction of these three conditions guarantees that any consistent system will be incomplete, because Gödel's construction is applicable.

The fascinating thing is that any such system digs its own hole; the system's own richness brings about its own downfall. The downfall occurs essentially because the system is powerful enough to have self-referential sentences. In physics, the notion exists of a "critical mass" of a fissionable substance, such as uranium. A solid lump of the substance will just sit there, if its mass is less than critical. But beyond the critical mass, such a lump will undergo a chain reaction, and blow up. It seems that with formal systems there is an analogous critical point. Below that point, a system is "harmless" and does not even approach defining arithmetical truth formally; but beyond the critical point, the system suddenly attains the capacity for self-reference, and thereby dooms itself to incompleteness. The threshold seems to be roughly when a system attains the three properties listed above. Once this ability for self-reference is attained, the system has a hole which is tailor-made for itself; the hole takes the features of the system into account and uses them against the system.

The Passion According to Lucas

The baffling repeatability of the Gödel argument has been used by various people—notably J. R. Lucas—as ammunition in the battle to show that there is some elusive and ineffable quality to human intelligence, which makes it unattainable by "mechanical automata"—that is, computers. Lucas begins his article "Minds, Machines, and Gödel" with these words:

> Gödel's theorem seems to me to prove that Mechanism is false, that is, that minds cannot be explained as machines.[1]

Then he proceeds to give an argument which, paraphrased, runs like this. For a computer to be considered as intelligent as a person is, it must be able to do every intellectual task which a person can do. Now Lucas claims that no computer can do "Gödelization" (one of his amusingly irreverent terms) in the manner that people can. Why not? Well, think of any particular formal system, such as TNT, or TNT+G, or even TNT+G_ω. One can write a computer program rather easily which will systematically generate theorems of that system, and in such a manner that eventually, any preselected theorem will be printed out. That is, the theorem-generating program won't skip any portion of the "space" of all theorems. Such a program would be composed of two major parts: (1) a subroutine which stamps out axioms, given the "molds" of the axiom schemas (if there are any), and (2) a subroutine which takes known theorems (including axioms, of course) and applies rules of inference to produce new theorems. The program would alternate between running first one of these subroutines, and then the other.

We can anthropomorphically say that this program "knows" some facts of number theory—namely, it knows those facts which it prints out. If it fails to print out some true fact of number theory, then of course it doesn't "know" that fact. Therefore, a computer program will be inferior to human beings if it can be shown that humans know something which the program cannot know. Now here is where Lucas starts rolling. He says that we humans can always do the Gödel trick on any formal system as powerful as TNT—and hence no matter what the formal system, we know more than it does. Now this may only sound like an argument about formal systems, but it can also be slightly modified so that it becomes, seemingly, an invincible argument against the possibility of Artificial Intelligence ever reproducing the human level of intelligence. Here is the gist of it:

Rigid internal codes entirely rule computers and robots; ergo . . .
Computers are isomorphic to formal systems. Now . . .
Any computer which wants to be as smart as we are has got to be able to do number theory as well as we can, so . . .

Among other things, it has to be able to do primitive recursive arithmetic. But for this very reason . . .
It is vulnerable to the Gödelian "hook", which implies that . . .
We, with our *human* intelligence, can concoct a certain statement of number theory which is true, but the *computer* is blind to that statement's truth (i.e., will never print it out), precisely because of Gödel's boomeranging argument.
This implies that there is one thing which computers just cannot be programmed to do, but which we can do. So we are smarter.

Let us enjoy, with Lucas, a transient moment of anthropocentric glory:

> However complicated a machine we construct, it will, if it is a machine, correspond to a formal system, which in turn will be liable to the Gödel procedure for finding a formula unprovable-in-that-system. This formula the machine will be unable to produce as being true, although a mind can see it is true. And so the machine will still not be an adequate model of the mind. We are trying to produce a model of the mind which is mechanical—which is essentially "dead"—but the mind, being in fact "alive," can always go one better than any formal, ossified, dead system can. Thanks to Gödel's theorem, the mind always has the last word.[2]

On first sight, and perhaps even on careful analysis, Lucas' argument appears compelling. It usually evokes rather polarized reactions. Some seize onto it as a nearly religious proof of the existence of souls, while others laugh it off as being unworthy of comment. I feel it is wrong, but fascinatingly so—and therefore quite worthwhile taking the time to rebut. In fact, it was one of the major early forces driving me to think over the matters in this book. I shall try to rebut it in one way in this Chapter, and in other ways in Chapter XVII.

We must try to understand more deeply why Lucas says the computer cannot be programmed to "know" as much as we do. Basically the idea is that we are always *outside* the system, and from out there we can always perform the "Gödelizing" operation, which yields something which the program, from within, can't see is true. But why can't the "Gödelizing operator", as Lucas calls it, be programmed and added to the program as a third major component? Lucas explains:

> The procedure whereby the Gödelian formula is constructed is a standard procedure—only so could we be sure that a Gödelian formula can be constructed for every formal system. But if it is a standard procedure, then a machine should be able to be programmed to carry it out too. . . . This would correspond to having a system with an additional rule of inference which allowed one to add, as a theorem, the Gödelian formula of the rest of the formal system, and then the Gödelian formula of this new, strengthened, formal system, and so on. It would [illegible] out-Gödel the new machine, Gödelizing operator and all. This has, in fact, proved to be the case. Even if we adjoin to a formal system the infinite set of axioms consisting of the successive Gödelian formulae, the resulting system is still incomplete, and contains a formula which cannot be proved-in-the-system, although a rational being can, standing outside the system, see that it is true. We had expected this, for even if an infinite set of axioms were added, they would have to be specified by some finite rule or specification, and this further rule or specification could then be taken into account by a mind considering the enlarged formal system. In a sense, just because the mind has the last word, it can always pick a hole in any formal system presented to it as a model of its own workings. The mechanical model must be, in some sense, finite and definite: and then the mind can always go one better.[3]

Jumping Up a Dimension

A visual image provided by M. C. Escher is extremely useful in aiding the intuition here: his drawing *Dragon* (Fig. 76). Its most salient feature is, of course, its subject matter—a dragon biting its tail, with all the Gödelian connotations which that carries. But there is a deeper theme to this picture. Escher himself wrote the following most interesting comments. The first comment is about a set of his drawings all of which are concerned with "the conflict between the flat and the spatial"; the second comment is about *Dragon* in particular.

> I. Our three-dimensional space is the only true reality we know. The two-dimensional is every bit as fictitious as the four-dimensional, for nothing is flat, not even the most finely polished mirror. And yet we stick to the convention that a wall or a piece of paper *is* flat, and curiously enough, we still go on, as we have done since time immemorial, producing illusions of space on just such plane surfaces as these. Surely it is a bit absurd to draw a few lines and then claim: "This is a house". This odd situation is the theme of the next five pictures [including *Dragon*].[4]
>
> II. However much this dragon tries to be spatial, he remains completely flat. Two incisions are made in the paper on which he is printed. Then it is folded in such a way as to leave two square openings. But this dragon is an obstinate beast, and in spite of his two dimensions he persists in assuming that he has three; so he sticks his head through one of the holes and his tail through the other.[5]

This second remark especially is a very telling remark. The message is that no matter how cleverly you try to simulate three dimensions in two, you are always missing some "essence of three-dimensionality". The dragon tries very hard to fight his two-dimensionality. He defies the two-dimensionality of the paper on which he thinks he is [illegible]

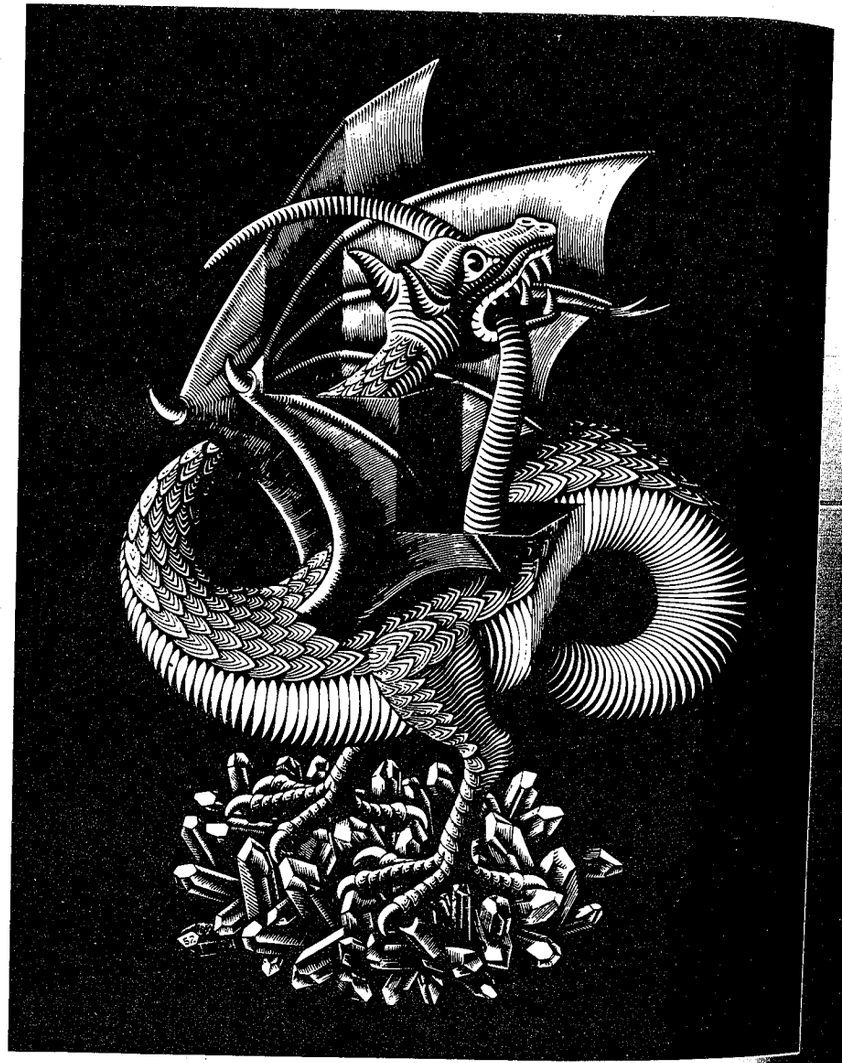

FIGURE 76. Dragon, *by M. C. Escher (wood-engraving, 1952).*

know it as we do. We could, in fact, carry the Escher picture any number of steps further. For instance, we could tear it out of the book, fold it, cut holes in it, pass it through itself, and photograph the whole mess, so that it again becomes two-dimensional. And to that photograph, we could once again do the same trick. Each time, at the instant that it becomes two-dimensional—no matter how cleverly we seem to have simulated three-

Now with this wonderful Escherian metaphor, let us return to the program versus the human. We were talking about trying to encapsulate the "Gödelizing operator" inside the program itself. Well, even if we had written a program which carried the operation out, that program would not capture the essence of Gödel's method. For once again, we, outside the system, could still "zap" it in a way which it couldn't do. But then are we arguing with, or against, Lucas?

The Limits of Intelligent Systems

Against. For the very fact that we cannot write a program to do "Gödelizing" must make us somewhat suspicious that we ourselves could do it in every case. It is one thing to make the argument in the abstract that Gödelizing "can be done"; it is another thing to know how to do it in every particular case. In fact, as the formal systems (or programs) escalate in complexity, our own ability to "Gödelize" will eventually begin to waver. It must, since, as we have said above, we do *not* have any algorithmic way of describing how to perform it. If we can't tell *explicitly* what is involved in applying the Gödel method in all cases, then for each of us there will eventually come some case so complicated that we simply can't figure out how to apply it.

Of course, this borderline of one's abilities will be somewhat ill-defined, just as is the borderline of weights which one can pick up off the ground. While on some days you may not be able to pick up a 250-pound object, on other days maybe you can. Nevertheless, there are no days whatsoever on which you can pick up a 250-ton object. And in this sense, though everyone's Gödelization threshold is vague, for each person, there are systems which lie far beyond his ability to Gödelize.

This notion is illustrated in the *Birthday Cantatatata.* At first, it seems obvious that the Tortoise can proceed as far as he wishes in pestering Achilles. But then Achilles tries to sum up all the answers in a single swoop. This is a move of a different character than any that has gone before, and is given the new name 'ω'. The newness of the name is quite important. It is the first example where the old naming scheme—which only included names for all the natural numbers—had to be transcended. Then come some more extensions, some of whose names seem quite obvious, others of which are rather tricky. But eventually, we run out of names once again—at the point where the answer-schemas

$$\omega,\ \omega^{\omega},\ \omega^{\omega^{\omega}},\ \ldots$$

are all subsumed into one outrageously complex answer schema. The altogether new name 'ϵ_0' is supplied for this one. And the reason a new name is needed is

There Is No Recursive Rule for Naming Ordinals

Now offhand you might think that these irregularities in the progression from *ordinal* to *ordinal* (as these names of infinity are called) could be handled by a computer program. That is, there would be a program to produce new names in a regular way, and when it ran out of gas, it would invoke the "irregularity handler", which would supply a new name, and pass control back to the simple one. But this will not work. It turns out that the irregularities themselves happen in irregular ways, and one would need also a second-order program—that is, a program which makes new programs which make new names. And even this is not enough. Eventually, a third-order program becomes necessary. And so on, and so on.

All of this perhaps ridiculous-seeming complexity stems from a deep theorem, due to Alonzo Church and Stephen C. Kleene, about the structure of these "infinite ordinals", which says:

> There is no recursively related notation-system
> which gives a name to every constructive ordinal.

What "recursively related notation-systems" are, and what "constructive ordinals" are, we must leave to the more technical sources, such as Hartley Rogers' book, to explain. But the intuitive idea has been presented. As the ordinals get bigger and bigger, there are irregularities, and irregularities in the irregularities, and irregularities in the irregularities in the irregularities, etc. No single scheme, no matter how complex, can name all the ordinals. And from this, it follows that no algorithmic method can tell how to apply the method of Gödel to all possible kinds of formal systems. And unless one is rather mystically inclined, therefore one must conclude that any human being simply will reach the limits of his own ability to Gödelize at some point. From there on out, formal systems of that complexity, though admittedly incomplete for the Gödel reason, will have as much power as that human being.

Other Refutations of Lucas

Now this is only one way to argue against Lucas' position. There are others, possibly more powerful, which we shall present later. But this counter-argument has special interest because it brings up the fascinating concept of trying to create a computer program which can get outside of itself, see itself completely from the outside, and apply the Gödel zapping-trick to itself. Of course this is just as impossible as for a record player to be able to play records which would cause it to break.

But—one should not consider TNT defective for that reason. If there is a defect anywhere, it is not in TNT, but in our expectations of what it should be able to do. Furthermore, it is helpful to realize that *we* are equally by C. H. Whitely, when he proposed the sentence "Lucas cannot consistently assert this sentence." If you think about it, you will see that (1) it is true, and yet (2) Lucas cannot consistently assert it. So Lucas is also "incomplete" with respect to truths about the world. The way in which he mirrors the world in his brain structures prevents him from simultaneously being "consistent" and asserting that true sentence. But Lucas is no more vulnerable than any of us. He is just on a par with a sophisticated formal system.

An amusing way to see the incorrectness of Lucas' argument is to translate it into a battle between men and women . . . In his wanderings, Loocus the Thinker one day comes across an unknown object—a woman. Such a thing he has never seen before, and at first he is wondrous thrilled at her likeness to himself; but then, slightly scared of her as well, he cries to all the men about him, "Behold! I can look upon her face, which is something *she* cannot do—therefore women can never be like me!" And thus he proves man's superiority over women, much to his relief, and that of his male companions. Incidentally, the same argument proves that Loocus is superior to all other males, as well—but he doesn't point that out to them. The woman argues back: "Yes, you can see my face, which is something I can't do—but I can see *your* face, which is something *you* can't do! We're even." However, Loocus comes up with an unexpected counter: "I'm sorry, you're deluded if you think you can *see* my face. What you women do is not the same as what we men do—it is, as I have already pointed out, of an inferior caliber, and does not deserve to be called by the same name. You may call it 'womanseeing'. Now the fact that you can 'womansee' my face is of no import, because the situation is not symmetric. You see?" "I woman-see," womanreplies the woman, and womanwalks away . . .

Well, this is the kind of "heads-in-the-sand" argument which you have to be willing to stomach if you are bent on seeing men and women running ahead of computers in these intellectual battles.

Self-Transcendence—A Modern Myth

It is still of great interest to ponder whether we humans ever can jump out of ourselves—or whether computer programs can jump out of themselves. Certainly it is possible for a program to modify itself—but such modifiability has to be inherent in the program to start with, so that cannot be counted as an example of "jumping out of the system". No matter how a program twists and turns to get out of itself, it is still following the rules inherent in itself. It is no more possible for it to escape than it is for a human being to decide voluntarily not to obey the laws of physics. Physics is an overriding system, from which there can be no escape. However, there is a lesser ambition which it is possible to achieve: that is, one can certainly jump from a subsystem of one's brain into a wider subsystem. One can step out of ruts on occasion. This is still