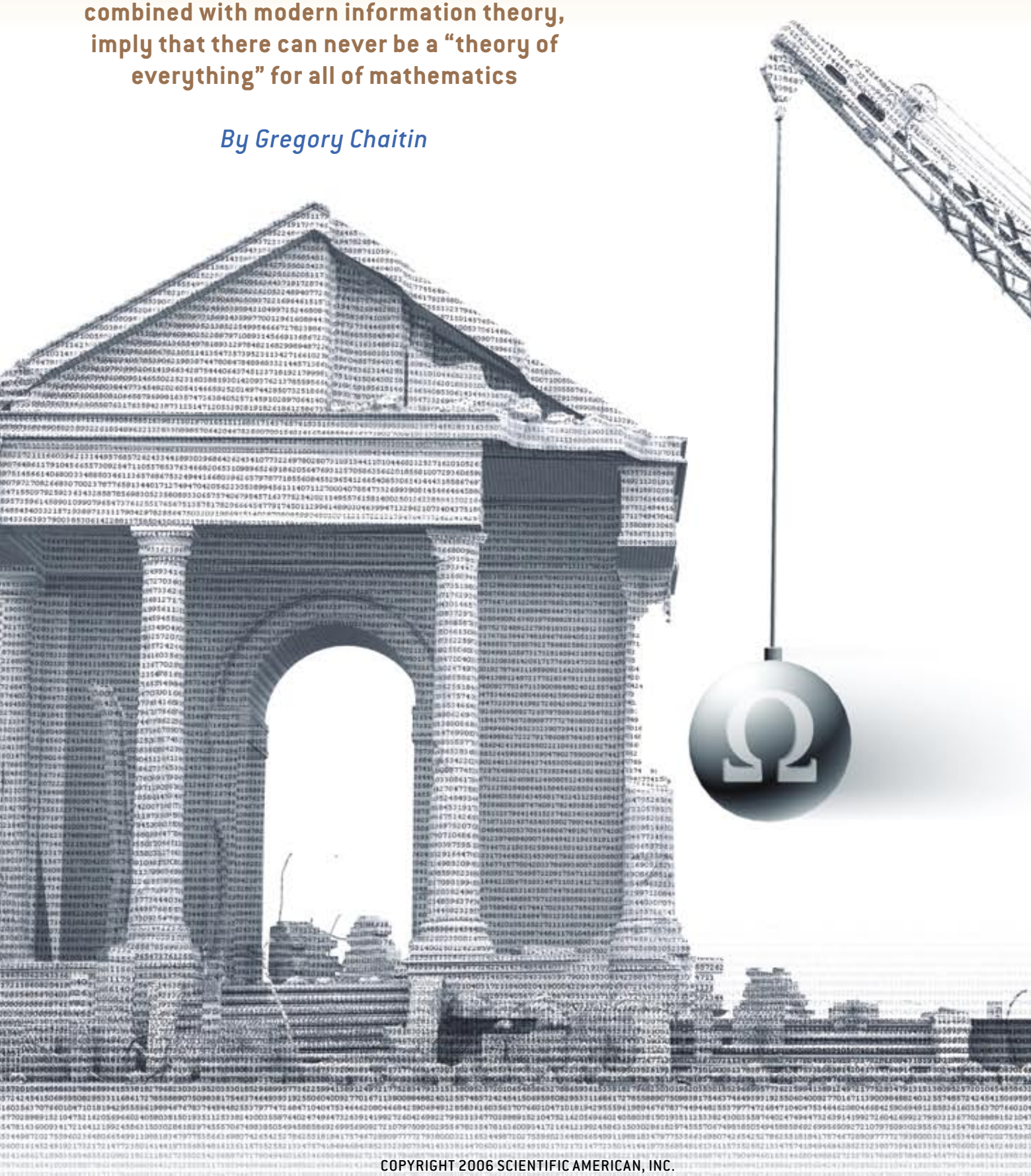
Ideas on complexity and randomness originally suggested by Gottfried W. Leibniz in 1686, combined with modern information theory, imply that there can never be a "theory of everything" for all of mathematics

By Gregory Chaitin



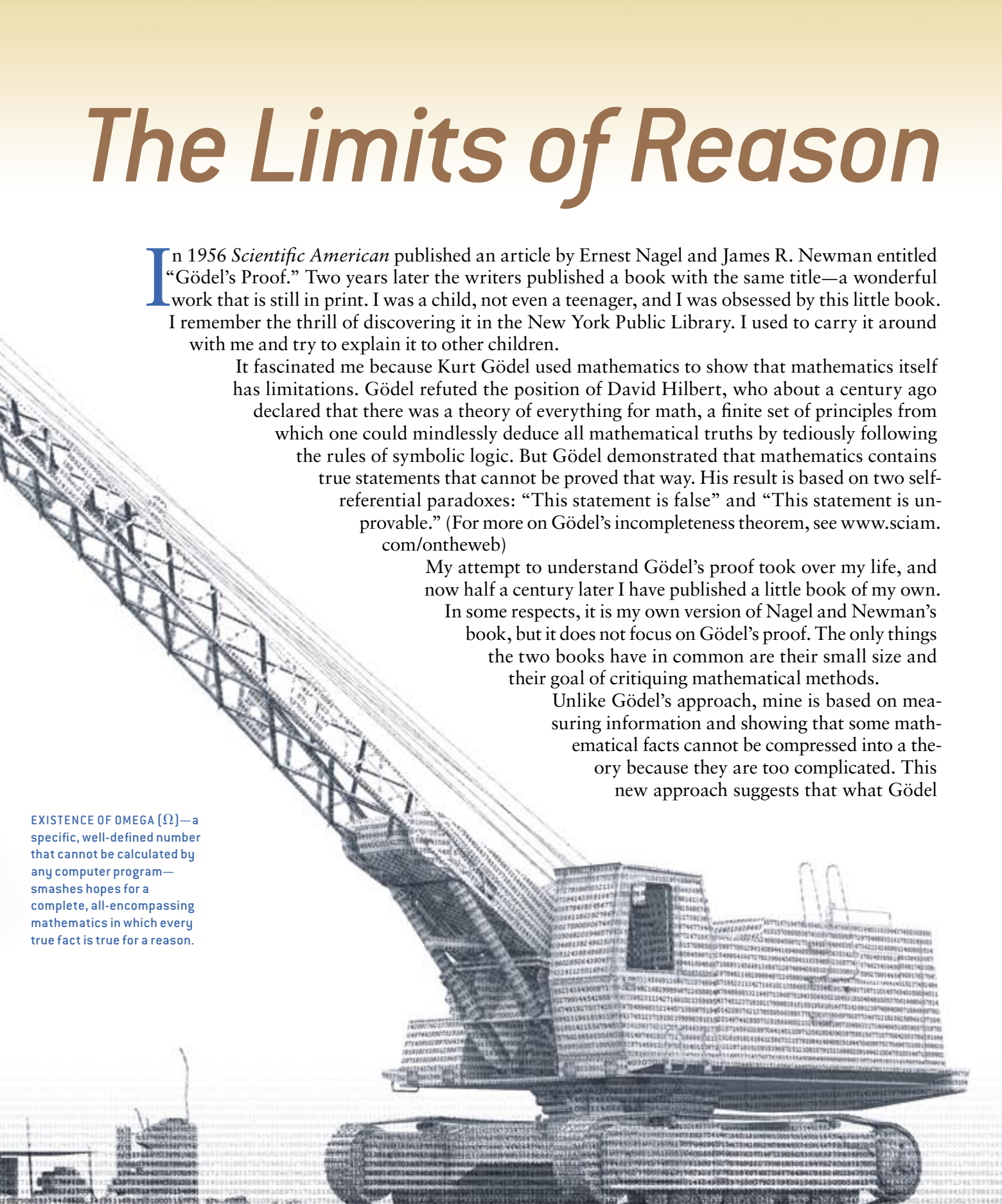
The Limits of Reason

In 1956 *Scientific American* published an article by Ernest Nagel and James R. Newman entitled "Gödel's Proof." Two years later the writers published a book with the same title—a wonderful work that is still in print. I was a child, not even a teenager, and I was obsessed by this little book. I remember the thrill of discovering it in the New York Public Library. I used to carry it around with me and try to explain it to other children.

It fascinated me because Kurt Gödel used mathematics to show that mathematics itself has limitations. Gödel refuted the position of David Hilbert, who about a century ago declared that there was a theory of everything for math, a finite set of principles from which one could mindlessly deduce all mathematical truths by tediously following the rules of symbolic logic. But Gödel demonstrated that mathematics contains true statements that cannot be proved that way. His result is based on two self-referential paradoxes: "This statement is false" and "This statement is unprovable." (For more on Gödel's incompleteness theorem, see www.sciam.com/ontheweb)

My attempt to understand Gödel's proof took over my life, and now half a century later I have published a little book of my own. In some respects, it is my own version of Nagel and Newman's book, but it does not focus on Gödel's proof. The only things the two books have in common are their small size and their goal of critiquing mathematical methods.

Unlike Gödel's approach, mine is based on measuring information and showing that some mathematical facts cannot be compressed into a theory because they are too complicated. This new approach suggests that what Gödel



EXISTENCE OF OMEGA (Ω)—a specific, well-defined number that cannot be calculated by any computer program—smashes hopes for a complete, all-encompassing mathematics in which every true fact is true for a reason.

discovered was just the tip of the iceberg: an infinite number of true mathematical theorems exist that cannot be proved from any finite system of axioms.

Complexity and Scientific Laws

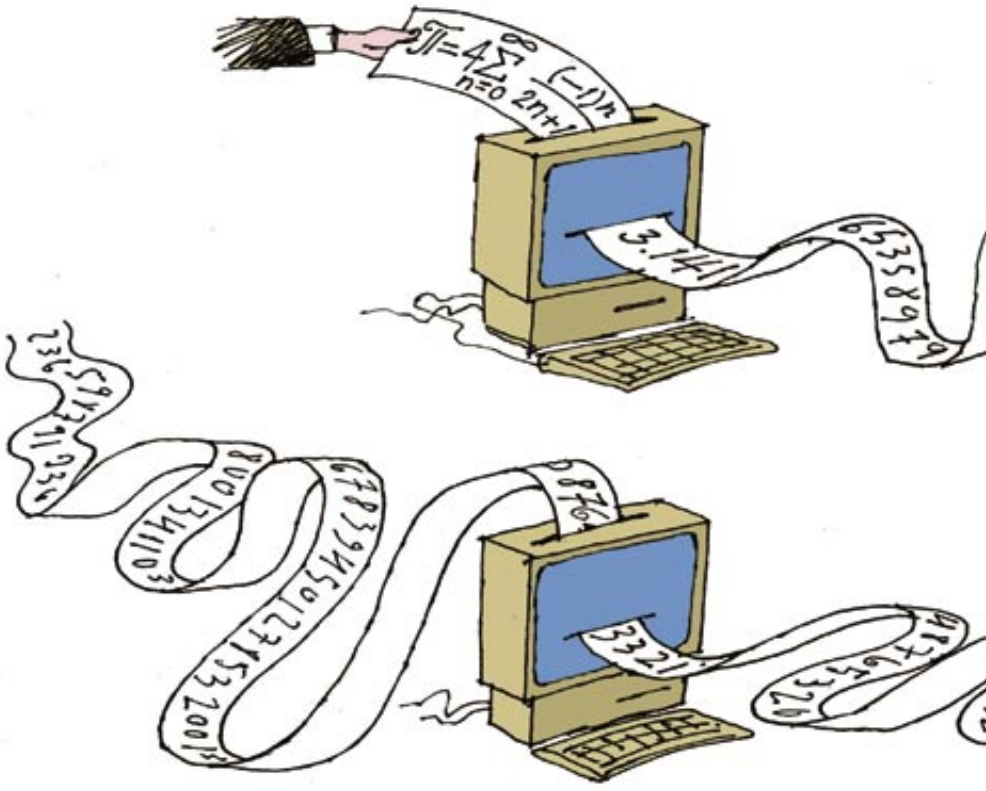
MY STORY BEGINS in 1686 with Gottfried W. Leibniz's philosophical essay *Discours de métaphysique* (*Discourse on Metaphysics*), in which he discusses how one can distinguish between facts that can be described by some law and those that are lawless, irregular facts. Leibniz's very simple and profound idea appears in section VI of the *Discours,* in which he essentially states that a theory has to be simpler than the data it explains, otherwise it does not explain anything. The concept of a law becomes vacuous if arbitrarily high mathematical complexity is permitted, because then one can always construct a law no matter how random and patternless the data really are. Conversely, if the only law that describes some data is an extremely complicated one, then the data are actually lawless.

Today the notions of complexity and simplicity are put in precise quantitative terms by a modern branch of mathematics called algorithmic information theory. Ordinary information theory quantifies information by asking how many bits are needed to encode the information. For example, it takes one bit to encode a single yes/no answer. Algorithmic information, in contrast, is defined by asking what size computer program is necessary to generate the data. The minimum number of bits—what size string of zeros and ones—needed to store the program is called the algorithmic information content of the data. Thus, the infinite sequence of numbers 1, 2, 3, … has very little algorithmic information; a very short computer program can generate all those numbers. It does not matter how long the program must take to do the computation or how much memory it must use—just the length of the program in bits counts. (I gloss over the question of what programming language is used to write the program—for a rigorous definition, the language would have to be specified precisely. Different programming languages would result in somewhat different values of algorithmic information content.)

To take another example, the number pi, 3.14159…, also has only a little algorithmic information content, because a relatively short algorithm can be programmed into a computer to compute digit after digit. In contrast, a random number with a mere million digits, say 1.341285…64, has a much larger amount of algorithmic information. Because the number lacks a defining pattern, the shortest program for outputting it will be about as long as the number itself:

```
Begin
Print "1.341285…64"
End
```

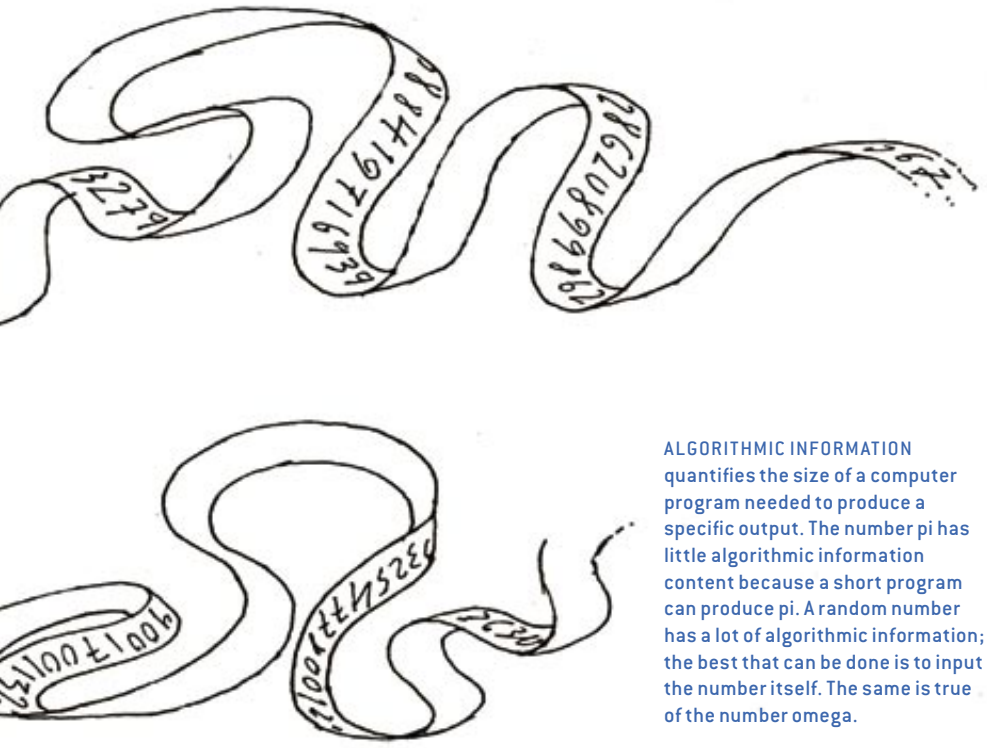
(All the digits represented by the ellipsis are included in the program.) No smaller program can calculate that se-



KENN BROWN; CONCEPT BY DUSAN PETRICIC (*preceding pages*); DUSAN PETRICIC (*above*)

Overview/*Irreducible Complexity*

- Kurt Gödel demonstrated that mathematics is necessarily incomplete, containing true statements that cannot be formally proved. A remarkable number known as omega reveals even greater incompleteness by providing an infinite number of theorems that cannot be proved by any finite system of axioms. A "theory of everything" for mathematics is therefore impossible.
- Omega is perfectly well defined [*see box on opposite page*] and has a definite value, yet it cannot be computed by any finite computer program.
- Omega's properties suggest that mathematicians should be more willing to postulate new axioms, similar to the way that physicists must evaluate experimental results and assert basic laws that cannot be proved logically.
- The results related to omega are grounded in the concept of algorithmic information. Gottfried W. Leibniz anticipated many of the features of algorithmic information theory more than 300 years ago.



ALGORITHMIC INFORMATION quantifies the size of a computer program needed to produce a specific output. The number pi has little algorithmic information content because a short program can produce pi. A random number has a lot of algorithmic information; the best that can be done is to input the number itself. The same is true of the number omega.

quence of digits. In other words, such digit streams are incompressible, they have no redundancy; the best that one can do is transmit them directly. They are called irreducible or algorithmically random.

How do such ideas relate to scientific laws and facts? The basic insight is a software view of science: a scientific theory is like a computer program that predicts our observations, the experimental data. Two fundamental principles inform this viewpoint. First, as William of Occam noted, given two theories that explain the data, the simpler theory is to be preferred (Occam's razor). That is, the smallest program that calculates the observations is the best theory. Second is Leibniz's insight, cast in modern terms—if a theory is the same size in bits as the data it explains, then it is worthless, because even the most random of data has a theory of that size. A useful theory is a compression of the data; comprehension is compression. You compress things into computer programs, into concise algorithmic descriptions. The simpler the theory, the better you understand something.

Sufficient Reason

DESPITE LIVING 250 years before the invention of the computer program, Leibniz came very close to the modern idea of algorithmic information. He had all the key elements. He just never connected them. He knew that everything can be represented with binary information, he built one of the first calculating machines, he appreciated the power of computation, and he discussed complexity and randomness.

If Leibniz had put all this together, he might have questioned one of the key pillars of his philosophy, namely, the principle of sufficient reason—that everything happens for a reason. Furthermore, if something is true, it must be true for a reason. That may be hard to believe sometimes, in the confusion and chaos of daily life, in the contingent ebb and flow of human history. But even if we cannot always see a reason (perhaps because the chain of reasoning is long and subtle), Leibniz asserted, God can see the reason. It is there! In that, he agreed with the ancient Greeks, who originated the idea.

Mathematicians certainly believe in reason and in Leibniz's principle of sufficient reason, because they always try to prove everything. No matter how much evidence there is for a theorem, such as millions of demonstrated examples, mathematicians demand a proof of the general case. Nothing less will satisfy them.

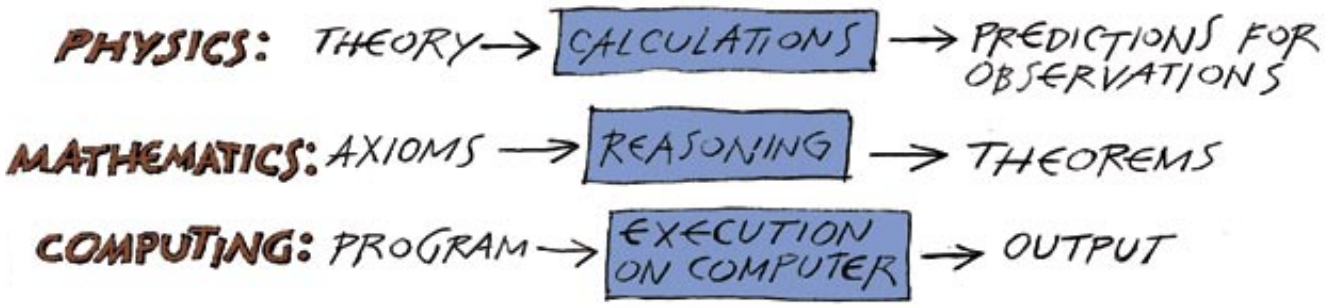
And here is where the concept of algorithmic information can make its surprising contribution to the philosophical discussion of the origins and limits of knowledge. It reveals that certain mathematical facts are true for no rea-

How Omega Is Defined

To see how the value of the number omega is defined, look at a simplified example. Suppose that the computer we are dealing with has only three programs that halt, and they are the bit strings 110, 11100 and 11110. These programs are, respectively, 3, 5 and 5 bits in size. If we are choosing programs at random by flipping a coin for each bit, the probability of getting each of them by chance is precisely $1/2^3$, $1/2^5$ and $1/2^5$, because each particular bit has probability $1/2$. So the value of omega (the halting probability) for this particular computer is given by the equation:

$$\text{omega} = 1/2^3 + 1/2^5 + 1/2^5 = .001 + .00001 + .00001 = .00110$$

This binary number is the probability of getting one of the three halting programs by chance. Thus, it is the probability that our computer will halt. Note that because program 110 halts we do not consider any programs that start with 110 and are larger than three bits—for example, we do not consider 1100 or 1101. That is, we do not add terms of .0001 to the sum for each of those programs. We regard all the longer programs, 1100 and so on, as being included in the halting of 110. Another way of saying this is that the programs are self-delimiting; when they halt, they stop asking for more bits. —*G.C.*



PHYSICS AND MATHEMATICS are in many ways similar to the execution of a program on a computer.

son, a discovery that flies in the face of the principle of sufficient reason.

Indeed, as I will show later, it turns out that an infinite number of mathematical facts are irreducible, which means no theory explains why they are true. These facts are not just computationally irreducible, they are logically irreducible. The only way to "prove" such facts is to assume them directly as new axioms, without using reasoning at all.

The concept of an "axiom" is closely related to the idea of logical irreducibility. Axioms are mathematical facts that we take as self-evident and do not try to prove from simpler principles. All formal mathematical theories start with axioms and then deduce the consequences of these axioms, which are called theorems. That is how Euclid did things in Alexandria two millennia ago, and his treatise on geometry is the classical model for mathematical exposition.

In ancient Greece, if you wanted to convince your fellow citizens to vote with you on some issue, you had to reason with them—which I guess is how the Greeks came up with the idea that in mathematics you have to prove things rather than just discover them experimentally. In contrast, previous cultures in Mesopotamia and Egypt apparently relied on experiment. Using reason has certainly been an extremely fruitful approach, leading to modern mathematics and mathematical physics and all that goes with them, including the technology for building that highly logical and mathematical machine, the computer.

So am I saying that this approach that science and mathematics has been following for more than two millennia crashes and burns? Yes, in a sense I am. My counterexample illustrating the limited power of logic and reason, my source of an infinite stream of unprovable mathematical facts, is the number that I call omega.

The Number Omega

THE FIRST STEP on the road to omega came in a famous paper published precisely 250 years after Leibniz's essay. In a 1936 issue of the *Proceedings of the London Mathematical Society,* Alan M. Turing began the computer age by presenting a mathematical model of a simple, general-purpose, programmable digital computer. He then asked, Can we determine whether or not a computer program will ever halt? This is Turing's famous halting problem.

Of course, by running a program you can eventually discover that it halts, if it halts. The problem, and it is an extremely fundamental one, is to decide when to give up on a program that does not halt. A great many special cases can be solved, but Turing showed that a general solution is impossible. No algorithm, no mathematical theory, can ever tell us which programs will halt and which will not. (For a modern proof of Turing's thesis, see www.sciam.com/ontheweb) By the way, when I say "program," in modern terms I mean the concatenation of the computer program and the data to be read in by the program.

The next step on the path to the number omega is to consider the ensemble of all possible programs. Does a program chosen at random ever halt? The probability of having that happen is my omega number. First, I must specify how to pick a program at random. A program is simply a series of bits, so flip a coin to determine the value of each bit. How many bits long should the program be? Keep flipping the coin so long as the computer is asking for another bit of input. Omega is just the probability that the machine will eventually come to a halt when supplied with a stream of random bits in this fashion. (The precise numerical value of omega depends on the choice of computer programming language, but omega's surprising properties are not affected by this choice. And once you have chosen a language, omega has a definite value, just like pi or the number 3.)

Being a probability, omega has to be greater than 0 and less than 1, because some programs halt and some do not. Imagine writing omega out in binary. You would get something like 0.1110100.... These bits after the decimal point form an irreducible stream of bits. They are our irreducible mathematical facts (each fact being whether the bit is a 0 or a 1).

Omega can be defined as an infinite sum, and each N-bit program that halts contributes precisely $1/2^N$ to the sum [*see box on preceding page*]. In other words,

DUSAN PETRICIC

THE AUTHOR

GREGORY CHAITIN is a researcher at the IBM Thomas J. Watson Research Center. He is also honorary professor at the University of Buenos Aires and visiting professor at the University of Auckland. He is co-founder, with Andrei N. Kolmogorov, of the field of algorithmic information theory. His nine books include the nontechnical works *Conversations with a Mathematician* (2002) and *Meta Math!* (2005). When he is not thinking about the foundations of mathematics, he enjoys hiking and snowshoeing in the mountains.

each *N*-bit program that halts adds a 1 to the *N*th bit in the binary expansion of omega. Add up all the bits for all programs that halt, and you would get the precise value of omega. This description may make it sound like you can calculate omega accurately, just as if it were the square root of 2 or the number pi. Not so—omega is perfectly well defined and it is a specific number, but it is impossible to compute in its entirety.

We can be sure that omega cannot be computed because knowing omega would let us solve Turing's halting problem, but we know that this problem is unsolvable. More specifically, knowing the first *N* bits of omega would enable you to decide whether or not each program up to *N* bits in size ever halts [*see box on page 80*]. From this it follows that you need at least an *N*-bit program to calculate *N* bits of omega.

Note that I am not saying that it is impossible to compute some digits of omega. For example, if we knew that computer programs 0, 10 and 110 all halt, then we would know that the first digits of omega were 0.111. The point is that the first *N* digits of omega cannot be computed using a program significantly shorter than *N* bits long.

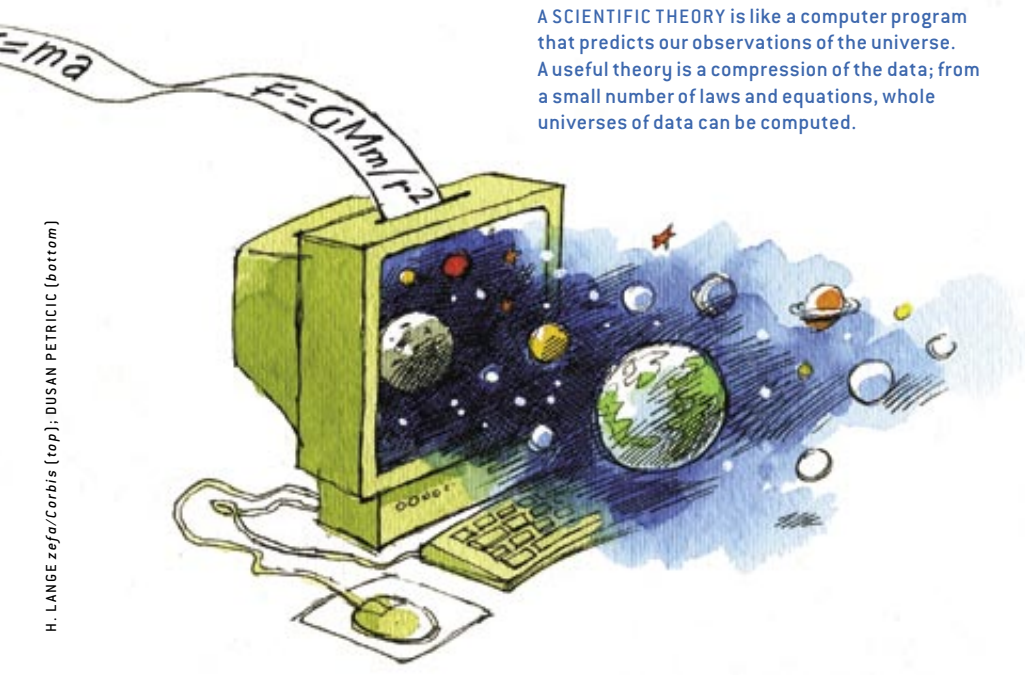
Most important, omega supplies us with an infinite number of these irreducible bits. Given any finite program, no matter how many billions of bits long, we have an infinite number of bits that the program cannot compute. Given any finite set of axioms, we have an infinite number of truths that are unprovable in that system.

Because omega is irreducible, we can immediately conclude that a theory of everything for all of mathematics cannot exist. An infinite number of bits of omega constitute mathematical facts (whether each bit is a 0 or a 1) that cannot be derived from any principles simpler than the string of bits itself. Mathematics therefore has infinite complexity, whereas any individual theory of everything would have only finite complexity and could not capture all the richness of the full world of mathematical truth.

This conclusion does not mean that proofs are no good, and I am certainly not against reason. Just because some things are irreducible does not mean we should give up using reasoning. Irreducible principles—axioms—have always been a part of mathematics. Omega just shows that a lot more of them are out there than people suspected.

So perhaps mathematicians should not try to prove everything. Sometimes they should just add new axioms. That is what you have got to do if you are faced with irreducible facts. The problem is realizing that they are irreducible! In a way, saying something is irreducible is giving up, saying that it cannot ever be proved. Mathematicians would rather die than do that, in sharp contrast with their physicist colleagues, who are happy to be pragmatic and to use plausible reasoning instead of rigorous proof. Physicists are willing to add new principles, new scientific laws, to understand new domains of experience. This raises what I think is an extremely interesting question: Is mathematics like physics?

GOTTFRIED W. LEIBNIZ, commemorated by a statue in Leipzig, Germany, anticipated many of the features of modern algorithmic information theory more than 300 years ago.



A SCIENTIFIC THEORY is like a computer program that predicts our observations of the universe. A useful theory is a compression of the data; from a small number of laws and equations, whole universes of data can be computed.

H. LANGE *zefa/Corbis (top)*; DUSAN PETRICIC *(bottom)*

Mathematics and Physics

THE TRADITIONAL VIEW is that mathematics and physics are quite different. Physics describes the universe and depends on experiment and observation. The particular laws that govern our universe—whether Newton's laws of motion or the Standard Model of particle physics—must be determined empirically and then asserted like axioms that cannot be logically proved, merely verified.

Mathematics, in contrast, is somehow independent of the universe. Results and theorems, such as the properties of the integers and real numbers, do not depend in any way on the particular nature of reality in which we find ourselves. Mathematical truths would be true in any universe.

Yet both fields are similar. In physics, and indeed in science generally, scientists compress their experimental observations into scientific laws. They then show how their observations can be deduced from these laws. In mathematics, too, something like this happens—mathematicians compress their computational experiments into mathematical axioms, and they then show how to deduce theorems from these axioms.

If Hilbert had been right, mathematics would be a closed system, without room for new ideas. There would be a static, closed theory of everything for all of mathematics, and this would be like a dictatorship. In fact, for mathematics to progress you actually need new ideas and plenty of room for creativity. It does not suffice to grind away, mechanically deducing all the possible consequences of a fixed number of basic principles. I much prefer an open system. I do not like rigid, authoritarian ways of thinking.

Another person who thought mathematics is like physics was Imre Lakatos, who left Hungary in 1956 and later worked on philosophy of science in England. There Lakatos came up with a great word, "quasi-empirical," which means that even though there are no true experiments that can be carried out in mathematics, something similar does take place. For example, the Goldbach conjecture states that any even number greater than 2 can be expressed as the sum of two prime numbers. This conjecture was arrived at experimentally, by noting empirically that it was true for every even number that anyone cared to examine. The conjecture has not yet been proved, but it has been verified up to 10^{14}.

I think that mathematics is quasi-empirical. In other words, I feel that mathematics is different from physics (which is truly empirical) but perhaps not as different as most people think.

I have lived in the worlds of both mathematics and physics, and I never thought there was such a big difference between these two fields. It is a matter of degree, of emphasis, not an absolute difference. After all, mathematics and physics coevolved. Mathematicians should not isolate themselves. They should not cut themselves off from rich sources of new ideas.

New Mathematical Axioms

THE IDEA OF CHOOSING to add more axioms is not an alien one to mathematics. A well-known example is the parallel postulate in Euclidean geometry: given a line and a point not on the line, there is exactly one line that can be drawn through the point that never intersects the original line. For centuries geometers wondered whether that result could be proved using the rest of Euclid's axioms. It could not. Finally, mathematicians realized that they could substitute different axioms in place of the Euclidean version, thereby producing the non-Euclidean geometries of curved spaces, such as the surface of a sphere or of a saddle.

Why Is Omega Incompressible?

I wish to demonstrate that omega is incompressible—that one cannot use a program substantially shorter than *N* bits long to compute the first *N* bits of omega. The demonstration will involve a careful combination of facts about omega and the Turing halting problem that it is so intimately related to. Specifically, I will use the fact that the halting problem for programs up to length *N* bits cannot be solved by a program that is itself shorter than *N* bits (see www.sciam.com/ontheweb).

My strategy for demonstrating that omega is incompressible is to show that having the first *N* bits of omega would tell me how to solve the Turing halting problem for programs up to length *N* bits. It follows from that conclusion that no program shorter than *N* bits can compute the first *N* bits of omega. (If such a program existed, I could use it to compute the first *N* bits of omega and then use those bits to solve Turing's problem up to *N* bits—a task that is impossible for such a short program.)

Now let us see how knowing *N* bits of omega would enable me to solve the halting problem—to determine which programs halt—for all programs up to *N* bits in size. Do this by performing a computation in stages. Use the integer *K* to label which stage we are at: $K = 1, 2, 3, \ldots$

At stage *K*, run every program up to *K* bits in size for *K* seconds. Then compute a halting probability, which we will call omega_K, based on all the programs that halt by stage *K*. Omega_K will be less than omega because it is based on only a subset of all the programs that halt eventually, whereas omega is based on *all* such programs.

As *K* increases, the value of omega_K will get closer and closer to the actual value of omega. As it gets closer to omega's actual value, more and more of omega_K's first bits will be correct—that is, the same as the corresponding bits of omega.

And as soon as the first *N* bits are correct, you know that you have encountered every program up to *N* bits in size that will ever halt. (If there were another such *N*-bit program, at some later-stage *K* that program would halt, which would increase the value of omega_K to be greater than omega, which is impossible.)

So we can use the first *N* bits of omega to solve the halting problem for all programs up to *N* bits in size. Now suppose we could compute the first *N* bits of omega with a program substantially shorter than *N* bits long. We could then combine that program with the one for carrying out the omega_K algorithm, to produce a program shorter than *N* bits that solves the Turing halting problem up to programs of length *N* bits.

But, as stated up front, we know that no such program exists. Consequently, the first *N* bits of omega must require a program that is almost *N* bits long to compute them. That is good enough to call omega incompressible or irreducible. (A compression from *N* bits to almost *N* bits is not significant for large *N*.)

—*G.C.*

OMEGA represents a part of mathematics that is in a sense unknowable. A finite computer program can reveal only a finite number of omega's digits; the rest remain shrouded in obscurity.

Other examples are the law of the excluded middle in logic and the axiom of choice in set theory. Most mathematicians are happy to make use of those axioms in their proofs, although others do not, exploring instead so-called intuitionist logic or constructivist mathematics. Mathematics is not a single monolithic structure of absolute truth!

Another very interesting axiom may be the "P not equal to NP" conjecture. P and NP are names for classes of problems. An NP problem is one for which a proposed solution can be verified quickly. For example, for the problem "find the factors of 8,633," one can quickly verify the proposed solution "97 and 89" by multiplying those two numbers. (There is a technical definition of "quickly," but those details are not important here.) A P problem is one that can be solved quickly even without being given the solution. The question is—and no one knows the answer—can every NP problem be solved quickly? (Is there a quick way to find the factors of 8,633?) That is, is the class P the same as the class NP? This problem is one of the Clay Millennium Prize Problems for which a reward of $1 million is on offer.

Computer scientists widely believe that P is not equal to NP, but no proof is known. One could say that a lot of quasi-empirical evidence points to P not being equal to NP. Should P not equal to NP be adopted as an axiom, then? In effect, this is what the computer science community has done. Closely related to this issue is the security of certain cryptographic systems used throughout the world. The systems are believed to be invulnerable to being cracked, but no one can prove it.

Experimental Mathematics

ANOTHER AREA of similarity between mathematics and physics is experimental mathematics: the discovery of new mathematical results by looking at many examples using a computer. Whereas this approach is not as persuasive as a short proof, it can be more convincing than a long and extremely complicated proof, and for some purposes it is quite sufficient.

In the past, this approach was defended with great vigor by both George Pólya and Lakatos, believers in heuristic reasoning and in the quasi-empirical nature of mathematics. This methodology is also practiced and justified in Stephen Wolfram's *A New Kind of Science* (2002).

Extensive computer calculations can be extremely persuasive, but do they render proof unnecessary? Yes and no. In fact, they provide a different kind of evidence. In important situations, I would argue that both kinds of evidence are required, as proofs may be flawed, and conversely computer searches may have the bad luck to stop just before encountering a counterexample that disproves the conjectured result.

All these issues are intriguing but far from resolved. It is now 2006, 50 years after this magazine published its article on Gödel's proof, and we still do not know how serious incompleteness is. We do not know if incompleteness is telling us that mathematics should be done somewhat differently. Maybe 50 years from now we will know the answer. SA

MORE TO EXPLORE

For a chapter on Leibniz, see **Men of Mathematics.** E. T. Bell. Reissue. Touchstone, 1986.

For more on a quasi-empirical view of math, see **New Directions in the Philosophy of Mathematics.** Edited by Thomas Tymoczko. Princeton University Press, 1998.

Gödel's Proof. Revised edition. E. Nagel, J. R. Newman and D. R. Hofstadter. New York University Press, 2002.

Mathematics by Experiment: Plausible Reasoning in the 21st Century. J. Borwein and D. Bailey. A. K. Peters, 2004.

For Gödel as a philosopher and the Gödel-Leibniz connection, see **Incompleteness: The Proof and Paradox of Kurt Gödel.** Rebecca Goldstein. W. W. Norton, 2005.

Meta Math!: The Quest for Omega. Gregory Chaitin. Pantheon Books, 2005.

Short biographies of mathematicians can be found at **www-history.mcs.st-andrews.ac.uk/BioIndex.html**

Gregory Chaitin's home page is **www.umcs.maine.edu/~chaitin/**

KENN BROWN; CONCEPT BY DUSAN PETRICIC