The Mathematical Implications of Algorithmic Information Theory

The Limits of Mathematics. *By Gregory J. Chaitin, Springer-Verlag, Sing-apore, 1998, 148+xii pages, $39.95.*

In perusing the remainder table at my local—and now kaput—megabook-store, I ran across Gregory Chaitin's slender volume on the mathematical implications of algorithmic information theory. The book consists of four chapters and an appendix explaining the rudiments of Chaitin's own personal dialect of LISP. He sees a critical role for such a language in foundational studies, which he deems far from complete. The first three chapters consist of edited transcripts of distinct, self-contained, one-hour lectures given by the author between 1992 and 1996, while the fourth reproduces a 50-page article from the *Journal of Universal Computer Science*. Warning: 43 of those pages consist of computer printout demonstrating the capabilities of Chaitin's "new lisp."

Demolishing Hilbert's Logically Connected Dreams

Being self-contained expositions of closely related topics, the book's four chapters overlap one another to a considerable extent. The first is by far the easiest to read, in part because it begins at the very beginning, with Euclid's attempt to axiomatize* geometry,

---

BOOK REVIEW

By James Case

---

and Leibnitz's dream of a symbolic logic. It then proceeds to the partial realization of Leibnitz's dream in the works of de Morgan, Jevons, Boole, Peano, Cantor, Frege, Russell, and Whitehead, before launching into an explanation of Hilbert's plan for deducing all of mathematics from a single finite list of (presumably self-evident) axioms. Hilbert already knew that such a system need only imply the salient facts of arithmetic within the system $\mathbb{N}$ of natural numbers, since he and his 19th-century colleagues had worked long and hard to reconstruct all else (including real numbers and Euclidean geometry) from known properties of $\mathbb{N}$.

Hilbert's point of departure had been the so-called Frege–Russell thesis, according to which mathematics is nothing more than a particularly well developed branch of symbolic logic. Such a thesis takes for granted that every mathematical truth or falsehood can be expressed in a formal language consisting of (i) a fixed alphabet of *admissible symbols*, and (ii) *explicit rules of syntax* for combining those symbols into meaningful words and sentences. The totality of mathematical truths would consist of a finite and irreducible list of sentences—known as axioms and accepted without proof—together with theorems deduced from those axioms. A proof was to consist of a sequence of sentences couched in the formal language, each one of which was to be either an axiom or a sentence deducible from earlier sentences in the sequence via specific *rules of inference*. The most famous of the latter is of course Aristotle's rule of "modus ponens," which asserts that if *A* is true, and *A* implies *B*, then *B* is true as well. Hilbert originally spoke as if modus ponens were the only rule of inference that would ever be needed, but eventually acknowledged the need for more.

Hilbert sought a formal axiomatic system (FAS) that would be both *complete* and *consistent,* so that—for any meaningful statement *A*—it would always be possible to prove either *A* or ¬*A*, but never both. Therein lies the rub. Only three years after Hilbert published his preliminary findings, Gödel published (in 1931) his famous "incompleteness theorem," proving that the attempt must fail—a consistent FAS rich enough to fully describe addition and multiplication in $\mathbb{N}$ is necessarily incomplete. Both Hilbert and his student von Neumann were quick to express their amazement at Gödel's result—in part because it was true, but mainly because neither of them had even paused to consider that it might be.

In 1936—only five years after Gödel's bombshell—Alan Turing used his notion of a universal Turing machine (UTM) to prove an even more powerful incompleteness theorem. UTMs differ from ordinary TMs in that they are "programmable." Turing's method was more powerful than Gödel's because it destroyed not one but two of Hilbert's dreams. In addition to the hope (I) of finding a finite list of axioms from which all mathematical truths can be deduced—a hope that (in Chaitin's opinion) Hilbert fully expected to realize—he entertained the related hope (II) of solving the *entscheidungsproblem*, or "decision problem," by producing a "fully automatic procedure" for deciding whether a given proposition (sentence) is true or false.

Hilbert's dreams are logically dependent—since (I) implies (II)—but seem not to be logically equivalent, since (II) does not appear to imply (I). That (I) implies (II) follows from the fact that, with only finitely many axioms to work with, the totality $\wp$ of all valid proofs is at most countably infinite. Hence, the elements of $\wp$ can be arranged in a sequence, according to length. The

Gregory Chaitin, the main architect of algorithmic information theory. Chaitin finds it discouraging, the reviewer writes, "that the combined weight of the three incompleteness theorems [Gödel and Turing] has had little effect on the working habits of mathematicians."

truth or falsehood of a given proposition A can then be established by a search through the final sentences of that sequence, one by one, until either A or $\neg A$ is found. Hilbert never claimed that the entscheidungsproblem would have a *practical* solution, or that his exertions would provide anything more than closure to an investigation that had already lasted more than two millennia. Chaitin is particularly impressed with the ease and naturalness of the method by which Turing demolished Hilbert's related dreams.

Turing began by observing that the list $\wp$ must surely contain—for every UTM program p—a proof either that p halts or that it doesn't. Therein lies the connection between Gödel's impossibility theorem, Hilbert's entschei-dungsproblem, and Turing's own "halting problem." Although Turing's 1936 paper never referred to the latter by name, Chaitin discerns unmistakable evidence of its presence on every page. In suitable hands, the problem leads directly to Turing's results, then back to Gödel's, and finally on to his own. It does so by means of a second (complete, denumerable) list L containing all possible UTM programs, and the L' of UTM programs designed to compute real numbers d with infinite decimal expansions. Following Cantor, Turing wrote the elements of L' in a column, beside the numbers they compute:

$$p_1 \sim d_1 = .\underline{d_{11}}d_{12}d_{13}d_{14}d_{15}d_{16}...$$
$$p_2 \sim d_2 = .d_{21}\underline{d_{22}}d_{23}d_{24}d_{25}d_{26}...$$
$$p_3 \sim d_3 = .d_{31}d_{32}\underline{d_{33}}d_{34}d_{35}d_{36}...$$
$$p_4 \sim d_4 = .d_{41}d_{42}d_{43}\underline{d_{44}}d_{45}d_{46}...$$
$$p_5 \sim d_5 = .d_{51}d_{52}d_{53}d_{54}\underline{d_{55}}d_{56}...$$

He then proceeded to argue that any number $d = .d_{11}d_{22}d_{33}...$ that differs from d_1 in the first decimal, from d_2 in the second, and so on, is an uncomputable real number. The computable reals form a denumerable set, while the uncomputable reals are as numerous as the reals themselves.

If the halting problem were solvable—as it would be if Hilbert's entscheidungsproblem were solvable—one could be sure that every $p_k \in L'$ would eventually print out a kth digit. So the program p^*, which runs p_k until it prints out a kth digit d_{kk}, then changes d_{kk} to δ_{kk}, then runs p_{k+1}, etc., would compute the uncomputable real $d = .d_{11}d_{22}d_{33}...$ digit by laborious digit! Hence, Hilbert's entscheidungs-problem is unsolvable. And since (I) would imply (II), Gödel's theorem becomes a consequence of Turing's.

Both Hilbert and his student von Neumann were quick to express their amazement at Gödel's result— in part because it was true, but mainly because neither of them had even paused to consider that it might be.

Chaitin's mathematical biography is remarkable. It begins with his enroll-ment—while still in high school—in a programming course for talented teenag-ers. The instructor encouraged the students to compete in producing the shortest programs capable of performing the routine tasks assigned. Though many tried hard to win, the none had the faintest idea how to decide whether or not the winning program was in fact the shortest capable of performing the assigned task.

Chaitin continued to dwell on the matter and, by the time he entered college, had begun (independently of other early workers in the field, mainly Kolmogorov and Solomonoff) to develop algorithmic information theory, a subject of which he was to become the main architect. His 1965 paper on gedanken experiments on automata—written while he was still in high school—continues to be of interest. He subsequently gravitated to IBM, where he has worked for more than thirty years, in a variety of capacities, and made numerous contributions to (among other things) the development of RISC technology.

Some Things Are True for No Reason at All

Much of Chaitin's work concerns the "halting probability" Ω that a program p chosen at random from the set $\wp$ of all UTM programs will eventually halt. In other words,

$$\Omega = \Sigma_{p \text{ halts}} 2^{-|p|} .$$

Here $|p|$ denotes the length of the program p, measured in bits of Shannon information, and $\wp$ must—for technical reasons—be restricted to programs that are not extensions of other programs.

Ω is more than an uncomputable number. It showed the world how very uncomputable a number can sometimes be. It is so uncom-putable that the length $|p_N|$ of the shortest program p_N capable of computing Ω_N—the first N binary digits of Ω—must satisfy $|p_N| > N - c$, where c is a constant independent of N. So $|p_N| = o(N)$, meaning that the quantity of information obtained by computing Ω_N barely exceeds the quantity with which the machine must be supplied in order to perform the computation. Unless c is large in comparison with a particular N, the (binary) digits of Ω_N can as easily be looked up in a table.

For large N, Ω_N is essentially no easier to compute than it would be if the individual digits had been determined by N tosses of a fair coin, a situation Chaitin describes as "algorithmic randomness." Whereas numbers like π, e, and $\sqrt{2}$ can easily be recovered from short (easily remembered) programs for computing them, no such program exists for computing Ω. Any program for doing so consumes (asymptotically speaking) as much space in main memory as the binary representation of the number itself. As Chaitin puts it, "some things are true for no reason at all," as if nature itself had determined them by tossing coins. Logic is powerless to help store and retrieve such numbers efficiently.

2

Incompleteness Theorems and Day-to-Day Mathematics

Chaitin finds it discouraging that the combined weight of the three incompleteness theorems has had little effect on the working habits of mathematicians. Most continue—or so it seems to him—to behave as if the axioms enunciated before 1931 are still the only legitimate ones. He wonders why the search for fruitful new axioms has not begun in earnest. The Riemann hypothesis—for which there now exists extensive numerical evidence—seems an obvious candidate. Does the world yet know the full extent of its implications? Riemann himself discovered a number of them, and Chaitin suspects that other important ones may still await discovery. He observes that there is now a *Journal of Experimental Mathematics*, and expresses the hope that his colleagues will begin to comb its pages for hypotheses that could hold the key to the solution of outstanding problems. He points out, among other things, that physics could hardly progress without formulating new axioms (laws of nature) for new phenomena as they come to light—astronomy, mechanics, heat, sound, electricity, light, relativity, quantum effects, and now perhaps chaos—and wonders why mathematics should behave so very differently. Does seventy years of increasingly powerful incompleteness theorems not suggest that a search for plausible new axioms is long overdue?

> *Does seventy years of increasingly powerful incompleteness theorems not suggest that a search for plausible new axioms is long overdue?*

Chaitin speculates that the impact of incompleteness theorems on the day-to-day practice of mathematics may well have been delayed by mathematicians' unfamiliarity with the techniques and concerns of foundational studies involving Turing machines. Most current mathematicians have some familiarity with computers and computer programs, but earlier generations did not. He therefore deems it important to explore the interfaces between foundational studies and more traditional branches of mathematics, such as the solution of Diophantine equations. He observes that Julia Robinson, Hilary Putnam, and Martin Davis did important early work of this nature.

He also cites Yuri Matijasevič's breakthrough discovery of 1970: a high-order polynomial with integer coefficients dependent on a single parameter k that has solutions if and only if the kth program in a suitable enumeration $\{p_k\}$ of $\wp$ halts. This bears directly on the tenth of the 23 problems commended by Hilbert to the attention of 20th-century mathematics. The problem in question requires an algorithm that, by looking only at the (integer) coefficients, can determine whether a given polynomial equation has solutions. Matijasevič settled the issue by showing that a solution of Hilbert's tenth problem would imply the existence of a solution to Turing's halting problem, already known to be insoluble.

Chaitin's own contribution to this intriguing field is an exponential polynomial (i.e., a polynomial in which some variables occur in the exponents of other variables) with integer coefficients depending on a single parameter k that has finitely many solutions for a given value of k if a certain k-related bit (binary digit) of Ω is a one, and infinitely many solutions if that bit is a zero. Chaitin's exponential polynomial involves some seventeen thousand variables, and occupies more than two hundred pages when written out in full. Deciding how many solutions it has for a particular value of k is a problem as intractable as computing the digits of Ω.

Chaitin seems particularly anxious to interest other theoreticians in using his private version of LISP to explore the capabilities of "self-delimiting" universal Turing machines. It was realized during the 1950s—by John McCarthy, who invented the language to facilitate the study of artificial intelligence—that programming in LISP is equivalent to programming such a machine, and Chaitin thinks that the time has come to exploit McCarthy's penetrating insight fully. Turing machines have already taught the world a number of important lessons, and nobody knows how many more they could teach if it were actually possible to sit down at a console and program one. Accordingly, he has constructed a virtual UTM that relies on Mathematica software, is easy to program, and runs very quickly. It is intended to serve as a new foundation for algorithmic information theory.

The name LISP is an acronym, formed from the words "list processor." Chaitin professes to love (pure) LISP because "it's really set theory, and all mathematicians love set theory." Where set theory would write {A, B, C}, a LISP programmer writes (A B C). Parenthesized expressions can be nested to arbitrary depth, and strings of letters unseparated by blanks (words?) denote single objects, as do the strings CD and 123 in the LISP expression (A (B CD) 123). Chaitin has equipped his "new lisp" with "primitive functions" specifically adapted to the needs of algorithmic information theory, and designed a short course explaining its use. He has taught versions of the course at universities in Finland, Australia, and New Mexico, and offers to e-mail the requisite software to anyone interested in learning the material.

Repetitive though it is, I found the book (relatively) easy and (remarkably) stimulating to read. The spontaneity of the lecture hall has not been edited out of the print version, and the author's almost visceral involvement with his subject seems to radiate from every page. One hopes that a more systematic account of the material will eventually be forthcoming. Until then, anyone with even a passing interest in the subject should spend time with this book.

The Limits of Mathematics is the fourth of five popular books Chaitin has written on this subject. The fifth, entitled The Unknowable, *was published in 1999. A complete list of publications can be found at www.cs.umaine.edu/~chaitin/.*

James Case writes from Baltimore, Maryland.