COMPUTATIONAL COMPLEXITY AND GÖDEL'S INCOMPLETENESS THEOREM

ACM SIGACT News, No. 9 (April 1971), pp. 11–12

G. J. Chaitin
IBM World Trade, Buenos Aires

Abstract

Given any simply consistent formal theory F of the state complexity $L(S)$ of finite binary sequences S as computed by 3-tape-symbol Turing machines, there exists a natural number $L(F)$ such that $L(S) > n$ is provable in F only if $n < L(F)$. On the other hand, almost all finite binary sequences S satisfy $L(S) > L(F)$. The proof resembles Berry's paradox, not the Epimenides nor Richard paradoxes.

Computational complexity has many points of view, and many points of contact with other fields. The purpose of this note is to show that a strong version of Gödel's classical incompleteness theorem follows very naturally if one considers the limitations of formal theories of computational complexity.

The state complexity $L(S)$ of a finite binary sequence S as computed by 3-tape-symbol Turing machines is defined to be the number of states that a 3-tape-symbol Turing machine must have in order to compute S. This concept is a variant of the descriptive or information complexity. Note that there are $(6n)^{3n}$ n-state 3-tape-symbol Turing machines. (The 6 is because there are six operations: tape left, tape right, halt, write 0, write 1, write blank.) Thus only finitely many finite binary sequences S have a given state complexity n, that is, satisfy $L(S) = n$.

Any simply consistent formal theory F of the state complexity of finite binary sequences will have the property that $L(S) > n$ is provable only if true, unless the methods of deduction of the theory are extremely weak. For if $L(S) > n$ isn't true then there is an n-state 3-tape-symbol Turing machine that computes S, and as this computation is finite, by carrying it out step by step in F it can be proved that it works, and thus that $L(S) \leq n$.

Suppose that there is at least one finite binary sequence S such that $L(S) > n$ is a theorem of F. Then there is a $(\lfloor \log_2 n \rfloor + 1 + c_F)$-state 3-tape-symbol Turing machine that computes a finite binary sequence S satisfying $L(S) > n$. Here c_F is independent of n and depends only on F. How is the Turing machine constructed? Its first $\lfloor \log_2 n \rfloor + 1$ states write the number n in binary notation on the Turing machine's tape. The remaining c_F states then do the following. By checking in order each finite string of letters in the alphabet of the formal theory F (the machine codes the alphabet in binary) to see if it is a proof, the machine generates each theorem provable in F. As each theorem is produced it is checked to see if it is of the form $L(S) > n$. The first such theorem encountered provides the finite binary sequence S computed by the Turing machine.

Thus we have shown that if there were finite binary sequences which in F can be shown to be of state complexity greater than n, then there would be a $(\lfloor \log_2 n \rfloor + 1 + c_F)$-state 3-tape-symbol Turing machine that computes a finite binary sequence S satisfying $L(S) > n$. In other words, we would have

$$n < L(S) \leq \lfloor \log_2 n \rfloor + 1 + c_F$$

which implies

$$n < \lfloor \log_2 n \rfloor + 1 + c_F.$$

As this is impossible for

$$n \geq L(F) \approx c_F + \log_2 c_F,$$

we conclude that $L(S) > n$ can be proved in F only if $n < L(F)$. Q.E.D.[1]

Why does this resemble Berry's paradox of "the least natural number not nameable in fewer than 10000000 characters"? Because it may be paraphrased as follows. "The finite binary sequence S with the first proof that S cannot be described by a Turing machine with n states or less" is a $(\log_2 n + c_F)$-state description of S.

As a final comment, it should be mentioned that an incompleteness theorem may also be obtained by considering the time complexity of infinite computations, instead of the descriptive complexity of finite computations. But this is much less interesting, as the resulting proof is, essentially, just one of the classical proofs resembling Richard's paradox, and requires that ω-consistency be hypothesized.

Brief Bibliography of Gödel's Theorem

Davis, M. (Ed.) *The Undecidable,* Raven Press, 1965.

Weyl, H. *Philosophy of Mathematics and Natural Science,* Princeton University Press, 1949, pp. 219–236.

Kleene, S. C. *Introduction to Metamathematics,* Van Nostrand, 1950, pp. 204–213.

[1][I couldn't verify the original argument, but I got $n < L(F) = 2c_F + 2$, by rather loose arguments. Readers' comments will be passed on to Mr. Chaitin.—ed.]

Turing, A. M. "Solvable and Unsolvable Problems,"
in *Science News,* No. 31, Penguin Books, 1954, pp. 7–23.

Nagel, E., Newman, J. R. *Gödel's Proof,*
New York University Press, 1958.

Davis, M. *Computability and Unsolvability,*
McGraw-Hill, 1958, pp. 120–129.

Cohen, P. J. *Set Theory and the Continuum Hypothesis,*
Benjamin, 1966, pp. 39–46.