W H A T   I S . . .

a Halting Probability?

Cristian S. Calude and G. J. Chaitin

Turing's famous 1936 paper "On computable numbers, with an application to the *Entscheidungsproblem*" defines a computable real number and uses Cantor's diagonal argument to exhibit an uncomputable real.

Roughly speaking, a computable real is one that can be calculated digit by digit, one for which there is an algorithm for approximating as closely as one may wish. All the reals one normally encounters in analysis are computable, like π, $\sqrt{2}$, and e. But they are much scarcer than the uncomputable reals because, as Turing points out, the computable reals are countable, whilst the uncomputable reals have the power of the continuum. Furthermore, any countable set of reals has measure zero, so the computable reals have measure zero. In other words, if one picks a real at random in the unit interval with uniform probability distribution, the probability of obtaining an uncomputable real is unity. One may obtain a computable real, but that is infinitely improbable.

But how about individual examples of uncomputable reals? We will show two: H and the halting probability Ω, both contained in the unit interval. Their construction was anticipated in 1927 by Émile Borel, who "defined" a real number $0 < B < 1$ in the following way. The Nth digit b_N of B answers the Nth question in an enumeration of all possible yes/no questions that one can write in French. Borel argued that knowledge of every point of the continuum would have strange consequences, and B is a case in point. The number B is an oracle that "would give answers to all past, present, and future enigmas of science, history, and curiosity."

Cristian S. Calude is professor of computer science at the University of Auckland. His email address is `cscalude@gmail.com`.

G. J. Chaitin is emeritus researcher at the IBM Watson Research Center and honorary professor at the University of Buenos Aires. His email address is `gjchaitin@gmail.com`.

Let's change this into something more mathematical, more realistic: an oracle for the halting problem H.

Systematically enumerate all programs with no input (say, lexicographically). The Nth bit h_N of the real number H tells us whether or not the Nth computer program ever halts. H is an oracle for the halting problem, which would be extremely useful to have, as we will show later.

However, this oracle H for the halting problem is extremely wasteful, because N instances of the halting problem are not N bits of mathematical information, they are only $\log_2 N$ bits of mathematical information. One only needs to know *how many* of these N programs halt—a number expressed in less than or equal to $1 + \log_2 N$ bits—to be able to determine which ones halt. Indeed, one just runs in parallel all programs till exactly the known number of programs that halt have stopped: all the remaining programs won't halt.

Using a slightly more sophisticated version of this idea, we finally arrive at the **halting probability** Ω, which is defined by the following formula:

$$0 < \Omega = \sum_{\text{program } p \text{ halts}} 2^{-(\text{size of } p \text{ in bits})} < 1.$$

This is the probability that a computer program whose bits are generated one by one by independent tosses of a fair coin will eventually halt. There are actually many halting probabilities, not one, because the precise numerical value of $\Omega = \Omega_L$ depends on the choice L of programming language for the programs p.

Nota Bene: For this definition of a halting probability to work properly, the computer programming language L that the programs p are written in has to satisfy certain requirements:

- The programs p have to be self-delimiting (no extension of a valid program is a valid program) or the sum won't converge. This enables one to construct large programs by concatenating (abutting) subroutines.
- The language L has to be universal and concise. More precisely, if any other language L' can program something in K bits, then L can do it in $\leq K + c_{L'}$ bits.

For historical reasons, such a programming language L is often referred to as a *universal self-delimiting Turing machine,* and we shall do so below.

What are the key properties of a halting probability Ω?

By writing Ω in binary one can show that given the first N bits $\omega_1 \ldots \omega_N$ of

$$\Omega = \sum_{K=1,2,3,\ldots} \omega_K / 2^K,$$

one can solve the halting problem for all programs up to N bits in size. This implies that the bits of Ω are algorithmically and logically irreducible:

- The smallest program for computing the first N bits of Ω has $\geq N - O(1)$ bits. (This algorithmic irreducibility property is normally called (*algorithmic*) *randomness.*)
- Any formal axiomatic theory, like ZFC, that enables one to prove what are the values of the first N bits of Ω must have $\geq N - O(1)$ bits of axioms. (This is *logical irreducibility.*)

In other words, the bits of Ω are the most concise oracle for solving the halting problem, the best possible compression of all the answers to individual instances of the halting problem.

The philosophical and epistemological significance of Ω is primarily due to the following surprising fact:

> *The bits $\omega_1\omega_2\omega_3\ldots$ of the halting probability Ω provide a perfect simulation within pure mathematics, where all truths are necessary, of an infinite stream of contingent, accidental yes/no facts.*

For instance, Ω provides us with a natural example of what É. Borel termed an *absolutely normal real number.* This means that if Ω is written in any base $b \geq 2$, then all blocks of K base-b digits will occur with equal limiting relative frequency b^{-K}.

It is an immediate corollary of Ω's algorithmic and logical irreducibility that Ω is uncomputable and that, in fact, no algorithm can compute more than finitely many scattered bits of Ω. Some of these bits have actually been determined. For a natural choice of the programming language L, the first 40 bits of $\Omega = \Omega_L$ are[1]

0001000000010000101001110111000011111010.

If we knew the first 7,780 bits,[2] which is less than one quarter of this note's size in bits, we would know whether the Riemann hypothesis is correct.

Now for some more advanced results.

Although uncomputable, Ω is the most "computable" among all algorithmically random reals: any Ω_L is *left-computable,* i.e., the least upper bound of a computable sequence of rationals. The set $\{\Omega_L\}$ of all halting probabilities of universal self-delimiting Turing machines L coincides with the set of all left-computable algorithmically random reals.

Can a halting probability be formally proved algorithmically random in Peano arithmetic (PA)? The answer depends on the representation: if $\Omega = \Omega_L$ is given by a self-delimiting Turing machine L whose universality (and conciseness) is proved in PA, then PA proves that Ω is algorithmically random. Every Ω can be written as $\Omega = \Omega_L$, for some L satisfying the above requirements, so it is provably random in PA.[3]

Further Reading. For the intellectual history leading up to Ω, including the role played by Leibniz, see the second author's *Meta Math!*. For a technical treatment, see the first author's *Information and Randomness*, or *An Introduction to Kolmogorov Complexity and Its Applications* by M. Li and P. Vitányi. For halting probabilities for versions of a real programming language, LISP, and a plethora of other nonstandard halting probabilities and their applications, see the second author's *Information-Theoretic Incompleteness.* For some of the latest results, see *Computability and Randomness* by A. Nies and *Algorithmic Randomness and Complexity* by R. G. Downey and D. Hirschfeldt (forthcoming). For discussions of the philosophical impact and additional historical material, see the second author's *Thinking about Gödel and Turing.*

[1] *C. S. Calude, M. J. Dinneen, Exact approximations of omega numbers,* Internat. J. Bifur. Chaos ***17****(6) (2007), 1937–54.*

[2] *C. S. Calude, E. Calude, M. J. Dinneen, A new measure of the difficulty of problems,* J. Mult.-Valued Logic Soft Comput. ***12*** *(2006), 285–307. In fact, this number can be lowered to less than 5,000 bits.*

[3] *C. S. Calude, N. J. Hay, Every computably enumerable random real is provably computably enumerable random,* Logic Jnl. IGPL ***17*** *(2009), 325–350.*