CDMTCS Research Report Series

Program-Size Complexity Computes the Halting Problem

G. J. Chaitin, A. Arslanov, C. Calude
IBM Research Division, New York, USA,
University of Auckland, New Zealand

CDMTCS-001
September 1995

Centre for Discrete Mathematics and
Theoretical Computer Science

PROGRAM-SIZE COMPLEXITY COMPUTES THE HALTING PROBLEM

Solutions by G. J. Chaitin[1], A. Arslanov[2] and C. Calude[3]

Can the halting problem be solved if one could compute program-size complexity?[4] [5] The answer is **yes** and here are two different proofs.

1. *Solution by G. J. Chaitin* (26 July 1995)[6]

LEMMA.[7] If an n-bit program p halts, then the time t it takes to halt satisfies $H(t) \leq n + c$. So if p has run for time T without halting, and T has the property that $t \geq T \Longrightarrow H(t) > n + c$, then p will never halt.

Consider the r.e. set of all true upper bounds on H: the set of all true upper bounds $\{H(x) \leq k\}$ is recursively enumerable. Imagine enumerating this set, and keep track of the time. Assuming that H is computable, compute $H(x)$ for each n-bit string x. Then enumerate $\{H(x) \leq k\}$ until we get the best possible upper bound on $H(x)$ for all n-bit strings x. Let $\beta(n)$ be defined to be the time it takes to enumerate enough of the set of all true upper bounds on program-size complexity until one obtains the correct value of $H(x)$ for all n-bit strings x. If one is given n and $\beta(n)$ or any number greater than $\beta(n)$, one can use this to determine an n-bit bit string x_{max} with maximum possible complexity $H(x_{max}) = n + H(n) + O(1)$. Thus any number $k \geq \beta(n)$ has

$$n + H(n) - c' < H(x_{max}) \leq H(k) + H(n) + c''$$

and

$$H(k) > n - c' - c''.$$

Thus we can use $\beta(n)$, which is computable from H, with the LEMMA to solve the halting problem as follows: an n-bit program p halts iff it halts before time $\beta(n + c + c' + c'')$.

2. *Solution by Asat Arslanov and Cristian Calude* (27 July 1995)[8]

Let A^* be the set of strings over the alphabet A, and let $p(x)$ be the place of x in A^* ordered quasi-lexicographically. Fix an acceptable gödelization $(\varphi_x)_{x \in A^*}$ of all partial recursive functions from strings to strings, and let W_x be the domain of (φ_x). Let $(C_x)_{x \in A^*}$ be an enumeration of all Chaitin computers (partial recursive string functions with prefix-free domains), $U(0^{p(x)}1y) = C_x(y)$ be a fixed universal Chaitin computer, and H its complexity.

We shall use the following completeness criterion (due to M. Arslanov):[9]

an recursively enumerable set X is Turing equivalent to the halting problem iff there is a Turing computable in X function f without fixed-points, i.e. $W_x \neq W_{f(x)}$, for all x,

---

[1]IBM Research Division, P. O. Box 704, Yorktown Heights, NY 10598, USA, chaitin @ watson.ibm.com.

[2]Computer Science Department, The University of Auckland, Private Bag 92019, Auckland, New Zealand, asat@cs.auckland.ac.nz.

[3]Computer Science Department, The University of Auckland, Private Bag 92019, Auckland, New Zealand, cristian@cs.auckland.ac.nz.

[4]The problem was discussed during the Summer School **Chaitin Complexity and Applications** held in the Romanian city Mangalia, at the Black Sea, in the period 26 June – 7 July 1995.

[5]For basic algorithmic information theory see G. J. Chaitin, *Algorithmic Information Theory,* Cambridge University Press, 1987 or C. Calude, *Information and Randomness—An Algorithmic Perspective,* Springer-Verlag, 1994.

[6]With thanks for stimulating discussions to Cris Calude and George Markowsky.

[7]Cf. G. J. Chaitin, Computing the Busy Beaver function, in T. M. Cover and B. Gopinath (eds.), *Open Problems in Communication and Computation,* Springer-Verlag, 1987, 108-112.

[8]With thanks for stimulating discussions to Greg Chaitin, Cristian Grozea and George Markowsky.

[9]Cf. R. I. Soare, *Recursively Enumerable Sets and Degrees,* Springer-Verlag, 1987, p. 88.

for the set $X = \{H(x) \leq k\}$.

FACT 1. There is a Chaitin computer $C = C_w$ acting as a choice function for non-empty r.e. sets, i.e. if W_x is non-empty, then $C(0^{p(x)}1)$ is defined and belongs to W_x.

FACT 2. There is a recursive function g such that $\varphi_{g(x)}(y) = C(0^{p(x)}1)$, for all strings x, y.

FACT 3. The function $F(y)$ defined to be the *minimum (in quasi-lexicographical order) string* x *such that* $H(x) > |0^{p(w)}10^{p(y)}1|$ is computable in H, total (as H is unbounded), and for every y,

$$F(y) \neq C(0^{p(y)}1).$$

Otherwise, the equalities

$$F(y) = C(0^{p(y)}1) = C_w(0^{p(y)}1) = U(0^{p(w)}10^{p(y)}1),$$

justify the inequality

$$H(F(y)) \leq |0^{p(w)}10^{p(y)}1|,$$

which contradicts the construction of F.

FACT 4. The function f defined by $W_{f(x)} = \{F(x)\}$ is computable in H, or, equivalently, computable in $X = \{H(x) \leq k\}$, and has no fixed-points.

Indeed, if $W_x = W_{f(x)}$, then W_x is not empty, so by FACT 1 and FACT 4, we deduce the equality $C(0^{p(x)}1) = F(x)$, which contradicts FACT 3.

3. COMMENT. Combining LEMMA with the information-theoretic Busy Beaver function[10]

$$\Sigma(n) = \max\{x \mid H(x) \leq n\}$$

one gets a constant $c > 0$ such that if an n-bit program p halts, then p halts in time less than $\Sigma(n+c)$.[11] However, the function Σ cannot be bounded by any recursive function! The difficulty might be also explained by the fact that Σ grows as fast as the least time necessary for all programs of length less than n that halt on U to stop.[12] The above solutions show that the non-recursive bound can in fact be replaced by a bound recursive in H.

Furthermore, Σ is computable in H. Indeed, the formula

$$\Sigma(n) = \max\{U(p) \mid |p| \leq n\},$$

proves that Σ is computable relative to the halting problem which, in turn, is computable from H.

4. COMMENT. After finishing this note it has come to our attention the paper *On the Complexity of Random Strings, Extended Abstract* [13] by M. Kummer in which problems related to those discussed here are studied.

[10] See note 5.
[11] This idea was discussed in Mangalia by Greg Chaitin, George Markowsky and Cris Calude.
[12] Cf. G. J. Chaitin, Information-theoretic limitations of formal systems, *J. Assoc. Comput. Mach.* 21(1974), 403-424.
[13] Manuscript, August 1995, 11 pp.