Electronic Colloquium on Computational Complexity, Revision 01 of Report No. 083 (2001)

A computably enumerable undecidable set with low prefix complexity: a simplified proof

Nikolai K. Vereshchagin*
Moscow Lomonossov University
Vorobjevy Gory, Moscow 119899
Email: ver@mccme.ru

Keywords: prefix complexity, *KP*-trivial set.

Let *KP* denote prefix complexity. The goal of the paper is to give a simple proof for the following

Theorem 1 (Solovay, Calude and Coles). *There is a computably enumerable undecidable set A such that* $KP(A_{1:n}) \leq KP(n) + O(1)$. *(Here* $A_{1:n}$ *stands for the prefix of length n of the characteristic sequence of A.)*

Solovay [9] proved the statement without enumerability requirement and Calude and Coles [1] added this requirement. Both Solovay's and Calude and Coles' proofs are rather involved (the latter one is 8 pages long). In the present paper we propose a simplified proof of Solovay–Calude–Coles theorem. Essentially the same proof as ours appears in the paper [4], it is attributed there to Downey, Hirschfeldt, and Nies (Theorem 50 on page 37). Our work was done independently of [4]. Both our and Downey–Hirschfeldt–Nies's arguments are similar to those of Kučera and Terwijn [5] used in construction of an undecidable Martin-Löf random set. In the cited paper, Downey gives a second proof for the result, which is based on the original proof of Solovay [9].

We will recall now all relevant definitions and then present our proof.

Let Ξ denote the set of all binary strings and $|x|$ stand for the length of string x. Given a partially computable function $\psi : \Xi \to \Xi$ let $K_\psi(x) = \min\{|p|\colon \psi(p) = x\}$.

Definition ([6, 3, 2]). A partial function $\psi : \Xi \to \Xi$ is a *prefix* function if for any p such that $\psi(p)$ is defined $\psi(q)$ is undefined for any proper prefix q of p.

Theorem 2. *(see [7, Th. 3.1.1]) The class of partially computable prefix functions has an optimal function. This means that there is a partially computable prefix function* ϕ *such that for any other partially computable prefix function* ψ *there is c such that* $K_\phi(x) \leq K_\psi(x) + c$ *for any x.*

*Work was done while visiting LIM, Université de Provence.

ISSN 1433-8092

Choose any optimal partially computable prefix function ϕ and define *prefix complexity* $KP(x)$ of x as $K_\phi(x)$.

Prefix complexity is closely related to universal semimeasures defined as follows.

A (total) function $P : \Xi \to [0;1]$ is called *a discrete semimeasure* if

$$\sum_{x\in\Xi} P(x) \leq 1.$$

It is called *enumerable* if the set $\{\langle r, x\rangle \mid r$ is a positive rational number and $x \in \Xi,\ r < P(x)\}$ is computably enumerable. For a definition of computably (= recursively) enumerable and decidable (= recursive) sets see [8]. An enumerable discrete semimeasure P is called *universal* if it *dominates* any other enumerable discrete semimeasure P', that is there is a constant such that $P'(x) \leq cP(x)$ for any x.

Theorem 3 ([6]). *The function $m(x) = 2^{-KP(x)}$ is a universal enumerable discrete semimeasure.*

Both $KP(x)$ and $m(x)$ are defined on binary strings. When we apply $KP(x)$ and $m(x)$ to natural numbers we actually mean their binary representations.

A simplified proof of Theorem 1. By Theorem 3, the inequality $KP(A_{1:n}) \leq KP(n) + O(1)$ is equivalent to the inequality $m(A_{1:n}) \geq m(n)/c$ for some constant c. And to prove the latter inequality it suffices to find an enumerable distributon q and another constant c' such that $q(A_{1:n}) \geq m(n)/c'$.

We describe an algorithm enumerating the set A. Fix an enumeration of all programs $p_1, p_2, \ldots$ such that the function $(p_k, n) \mapsto p_k(n)$ is partially computable. Fix an enumeration of the set $\{\langle r, x\rangle \mid x \in \Xi,\ r < m(x)\}$, where m is a universal enumerable discrete semimeasure. The algorithm enumerating A starts an enumeration of this set and, in parallel, an enumeration of the graph of the function $(p_k, n) \mapsto p_k(n)$.

In order to ensure that A is undecidable we will associate with every program p_k a number n_k such that $p_k(n_k)$ is either undefined, or defined and different from $A(n_k)$ (where $A(n_k)$ is 1 if $n_k \in A$ and 0 otherwise). To do so we start with $n_k = 2k$ (say) and with $A = \emptyset$. Once in the enumeration of the graph of the function $(p_k, n) \mapsto p_k(n)$ we find, for some k, that $p_k(n_k) = 0$ then we enumerate n_k into A.

Simultaneously, we define a lower semicomputable distribution q on $\{0,1\}^*$ such that $q(A_{1:n}) \geq m(n)/2$ for all n. To this end, for each pair $\langle r, n\rangle$ enumerated so far into the set $\{\langle r, n\rangle \mid r < m(n)\}$ we enumerate the pair $\langle r/2, A_{1:n}\rangle$ into the set $\{\langle r, x\rangle \mid r < q(x)\}$ (for the current value of $A_{1:n}$). This obviously will ensure the inequality $q(A_{1:n}) \geq m(n)/2$. The problem however is that the function q defined by our process may not satisfy the inequality $\sum_y q(y) \leq 1$. In other words, it may be not a distribution.

Now comes the crucial point. To force q to be a distribution we will sometimes change n_k for some k. For any particular k the value of n_k will be changed only a finite number of times, thus changing n_k will not disturb the undecidability of A.

More specifically, we keep true the following invariant

$$\sum_{i \geq n_k} m'(i) \leq 2^{-k} \qquad \text{for all } k \text{ such that } p_k(n_k) \text{ has not yet been defined,}$$

where $m'(i)$ denotes the best currently known lower bound for $m(i)$. To this end, once we see that for some k with $p_k(n_k)$ not yet defined this inequality is false we assign n_k a greater value different from all current n_i's and such that the inequality becomes true (note that on any step only finitely many $m'(i)$ are different from 0). Every n_k may be changed only finitely many times: once n_k has become so great that $\sum_{i \geq n_k} m(i) \leq 2^{-k}$ it remains unchanged forever.

Note that enumerating n_k into A implies changing prefixes of the characteristic function of A (of length n_k and greater) and thus forces to increase q on changed prefixes. Thus we need to show that $\sum_y q(y) \leq 1$. The sum of $q(y)$ over all prefixes y of the characteristic sequence of A is at most $1/2$ as $q(z) \leq m(|z|)/2$ for any z. However, since A has been changed (infinitely) many times, $q(y)$ may be non-zero also for prefixes y of the previous values of the characteristic sequence of A. For any such y there is a step t such that y was a prefix of the characteristic sequence of A on step t but not on step $t+1$. In other words, n_k was added in A on step t for some n_k not greater than $|y|$. Let A^t denote the value of A before adding n_k in A. The invariant implies that the sum of $q(y)$ (on step t) over all prefixes of the characteristic function of A^t of length n_k or more is at most 2^{-k-1}. On later steps $q(y)$ remains unchanged for all such y's. Hence the limit value of the sum of $q(y)$ over all prefixes of the characteristic function of A^t of length n_k or more is at most 2^{-k-1}. Observe now that for any k only one n_k may be added to A (we enumerate n_k into A only when we have found that $p_k(n_k)$ is defined and in this case we do not change n_k any more). Hence the sum of $q(y)$ over all y that are not prefixes of the characteristic function of A is at most $\sum_{k=1}^{\infty} 2^{-k-1} = 1/2$. □

References

[1] C. S. Calude, R. J. Coles. Program-size complexity of initial segments and domination relation reducibility, in J. Karhumäki, H. A. Maurer, G. Păun, G. Rozenberg (eds.). *Jewels Are Forever*, Springer-Verlag, Berlin, 1999, 225-237.

[2] G. J. Chaitin. A theory of program size formally identical to information theory. J. Assoc. Comp. Mach., 22:329–340, 1975.

[3] P. Gács. On the symmetry of algorithmic information. Soviet Math. Dokl., 15:1477–1480, 1974.

[4] R. Downey. Some computability-theoretical aspects of reals and randomness. Available from `http://www.mcs.vuw.ac.nz/research/maths-pubs.shtml`.

[5] A. Kučera and S. A. Terwijn. Lowness for the class of random sets. Journ. Symb. Logic., 64(4) (1999) 1396-1402.

[6] L.A. Levin. Laws of information conservation (non-growth) and aspects of the foundation of probability theory. Problems Inform. Transmission, 10:206–210, 1974.

[7] M. Li, P. Vitányi. An Introduction to Kolmogorov complexity and its applications. Second edition. Springer Verlag, 1997.

[8] P. Odifreddy. Classical recursion theory. North-Holland, 1989.

[9] R. Solovay. Lecture notes on algorithmic complexity. Unpublished, UCLA, 1975.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'