

A note on the complexity of time-bounded Kolmogorov random strings

Stephen Fenner
Department of Computer Science
and Engineering
University of South Carolina
fenner@engr.sc.edu

Tarsem S. Purewal Jr.
Computer Science Department
The College of Charleston
purewals@cofc.edu

Abstract

A result due to Allender *et al.* states that the set of time-bounded Kolmogorov random strings, R_{Kt} , is not hard for EXP under uniform polynomial-time truth table reductions [1]. Using similar techniques, we show that R_{Kt} is not complete for EXP under polynomial-time Turing reductions using a fixed polynomial number of queries. In addition, we show that under the hypothesis that $R_{Kt} \in \bigcap_d \text{DTIME}(2^{n/2}/n^d)$, R_{Kt} is not complete for EXP under Turing reductions. While this does not completely resolve the issue, it does give the tightest time-completeness tradeoff that is known for this set, as it can be shown that $R_{Kt} \in \text{DTIME}(2^{n/2}n^3)$.

1 Introduction

The complexity of the set of Kolmogorov random strings in the traditional sense is well understood. Specifically this set, denoted R_K , is an undecidable set that has been shown to be co-c.e. and complete under truth-table reductions [2]. When a resource bound is applied to the definition, however, the picture becomes much less clear. While upper bounds on the associated sets of random strings are often readily apparent for these decidable sets, they have resisted categorization in terms of completeness under common notions of reducibility such as polynomial-time Turing reductions.

Allender *et al.* studied this phenomenon in some detail, and manage to show that some of these sets are complete under various non-traditional notions of reducibility [1]. For example, they show that the set R_{Kt} is complete for EXP under non-uniform polynomial-time truth-table reductions. They also obtain a related negative result — it turns out that R_{Kt} is not complete for EXP under uniform polynomial-time truth-table reductions.

The question of whether or not R_{Kt} can be shown to be complete for EXP under polynomial-time Turing reductions remains open, however. In this note, we obtain some results that are relevant to this problem. Using techniques similar to those used to obtain Allender *et al.*'s negative result, we show that R_{Kt} is not complete for EXP under polynomial-time Turing reductions where the machine is only allowed to make a polynomial number of queries. Furthermore, we give the following time-completeness tradeoff: if $R_{Kt} \in \bigcap_d \text{DTIME}(2^{n/2}/n^d)$, then R_{Kt} is not complete for EXP under Turing reductions. It is known that $R_{Kt} \in \text{DTIME}(2^{n/2}n^3)$.

2 Preliminaries

We will use the following definitions and notations.

Definition 1 (Polynomial-Time Turing Reducibility). *Let A and B be languages. $A \leq_T^p B$ if and only if there exists a polynomial-time oracle Turing machine M where $x \in A$ iff M^B accepts x .*

Definition 2 (Bounded Polynomial-Time Turing Reducibility). Let A and B be languages and let f be a computable function. $A \leq_{T[f(n)]}^p B$ if and only if there exists a polynomial-time oracle Turing machine M where $x \in A$ iff M^B accepts x and for all strings x , M makes at most $f(|x|)$ oracle queries before halting.

Definition 3 (E). $E = \text{DTIME}(2^{cn})$.

Definition 4 (EXP). $\text{EXP} = \bigcup_i \text{DTIME}(2^{n^i})$.

We use the following characterization of time-bounded Kolmogorov complexity originally due to Levin [3]. See Allender *et al.* [1] for more details.

Definition 5. Let U be a universal Turing machine. Then $\text{Kt}_U(x) = \min\{|d| + \log(t) : U(d) \text{ outputs } x \text{ in } t \text{ steps}\}$.

In this paper, we assume a fixed universal machine and suppress the U . We can then define the set of time-bounded Kolmogorov random strings as follows.

Definition 6. $R_{\text{Kt}} = \{w : \text{Kt}(w) \geq |w|/2\}$.

Allender *et al.* give an upper-bound on the complexity of R_{Kt} [1].

Theorem 7 (Allender, et al.). $R_{\text{Kt}} \in E$.

They also show that R_{Kt} is not complete under polynomial-time truth table reductions. We prove this theorem to illustrate the technique that they use.

Theorem 8 (Allender, et al. [1]). R_{Kt} is not complete for EXP under \leq_{tt}^p reductions.

Proof. Let $L \subseteq 0^*$ such that $L \in \text{EXP} - P$, which is guaranteed to exist by the time hierarchy theorem. Suppose $L \leq_{tt}^p R_{\text{Kt}}$. We'll reach a contradiction by showing that $L \in P$.

Note that since $L \leq_{tt}^p R_{\text{Kt}}$, there exists a polynomial-time Turing machine A that decides L given the results of queries that are generated prior to the execution of L by a query generator Turing machine Q . Q runs in time $O(n^c)$ for some constant c . If the description of Q has length $|Q|$, then the only information needed to output a queried string q_i is a description of Q , the length of the input on which q_i is queried, and i . It follows that the Kt complexity of q_i is bound above by

$$\text{Kt}(q_i) \leq \log(n) + \log(i) + \log(n^c) + O(1) = O(\log(n)).$$

We can modify A so that when it needs the result of a query q_i it does the following. First, it checks to see if $|q_i| > 2(|Q| + (c+2)\log(n))$. If it is, then $q_i \notin R_{\text{Kt}}$ by definition, and A uses “no” as the result of the query. If it is smaller, then A simulates the E algorithm for R_{Kt} on input q_i . Since $|q_i| = O(\log(n))$, the simulation only requires a polynomial number of steps, and therefore A can compute L in polynomial time. \square

3 Results

Theorem 9. For all polynomials p , R_{Kt} is not complete for EXP under $\leq_{T[p(n)]}^p$ reductions.

Proof. Fix $l \geq 1$. Then the hierarchy theorem guarantees that there exists an $L \subseteq 0^*$ such that $L \in \text{DTIME}(2^{n^{l+2}}) - \text{DTIME}(2^{n^{l+1}})$.

Suppose that $L \leq_{T[n^l]}^p R_{\text{Kt}}$. Let A be a polynomial-time oracle Turing machine deciding L that runs in time n^c . Also note that A makes at most n^l queries to R_{Kt} where l is independent of c . This machine can be modified to output a given query q_i if it is given the length of the input on which q_i is queried and the answers to the previous $i-1$ queries. It follows that the Kt complexity of q_i is bound above by

$$\text{Kt}(q_i) \leq \log(n) + \log(i) + (i-1) + \log(n^c) + O(1) = O(n^l)$$

Now we can modify A to decide L without an oracle for R_{Kt} . Whenever A needs to make a query q_i it first checks to see if $|q_i| > 2(|A| + (c + l + 1) \log(n) + n^l)$. If it is, the answer to the query is “no.” Otherwise, it computes whether or not $q_i \in R_{\text{Kt}}$ directly using the E algorithm.

By definition, the E algorithm for R_{Kt} runs in time 2^{sn} for some constant s . By using the above bound on the length of each query, it can be shown that the running time of this machine on a particular query q_i is bound above by 2^{sn^l} . Then the overall running time of the modified version of A is $O(2^{sn^l}) = O(2^{n^{l+1}})$. Thus, we obtain a contradiction to $L \notin \text{DTIME}(2^{n^{l+1}})$. \square

Theorem 10. *If $R_{\text{Kt}} \in \bigcap_d \text{DTIME}(2^{n/2}/n^d)$, then R_{Kt} is not complete for EXP under \leq_T^p reductions.*

Proof. Suppose otherwise. Let L be any tally set in EXP – E and let A be a polynomial-time Turing reduction from L to R_{Kt} running in time n^c . We begin by claiming that all query strings of length at least n cannot be in R_{Kt} . If not, let q_i be the first query string of length at least n that is contained in R_{Kt} . We can obtain an upper-bound on its Kt complexity by modifying A so that it computes all previous queries directly using some $\text{DTIME}(2^{n/2}/n^d)$ algorithm for R_{Kt} where d will be defined later, and including binary representations of i and n as an input. Therefore, we have

$$\text{Kt}(q_i) = |A| + \log n + \log i + \log(n^c(2^{n/2}/n^d)) \leq |A| + (1 + 2c - d) \log n + \frac{n}{2}$$

By picking $d \geq 1 + 2c + |A|$, we get an upper-bound is smaller than $n/2$, contradicting q_i in R_{Kt} . It follows that for sufficiently large n , any query string of length n or greater will not be in R_{Kt} .

Therefore A will only need to run the E algorithm for R_{Kt} on strings of length less than n . Hence the modified version of A will run in time $O(2^{n/2})$. Therefore $L \in \text{E}$, which is a contradiction. \square

References

- [1] E. Allender, H. Buhrman, Michal Koucky, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35:1467–1493, 2006.
- [2] M. Kummer. On the complexity of random strings. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 25–36, 1996.
- [3] L. Levin. Randomness conservation inequalities; information and independence in mathematical theories. *Information and Control*, 61(1):15–37, 1984.