Reprinted from JOURNAL OF COMPUTER AND SYSTEM SCIENCES
All Rights Reserved by Academic Press, New York and London

Vol. 39, No. 1, August 1989
*Printed in Belgium*

# Some Consequences of the Existence of Pseudorandom Generators

ERIC W. ALLENDER*

*Department of Computer Science, Rutgers University*
*New Brunswick, New Jersey 08903*

Received June 2, 1988

This paper introduces a type of generalized Kolmogorov complexity and uses it as a tool to explore the consequences of several assumptions about the existence of secure pseudorandom generators. It is shown that if secure generators exist, then there are fast deterministic simulations of probabilistic algorithms; the nature of the simulations and the class of probabilistic algorithms for which simulations can be exhibited depends on the notion of "security" which is assumed. One goal of the investigation begun here is to show that many important questions in complexity theory may be viewed as questions about the Kolmogorov complexity of sets in P. © 1989 Academic Press, Inc.

## 1. INTRODUCTION

A pseudorandom generator is an efficient routine which takes a short input (the seed) and produces a long (pseudorandom) output. Since the pseudorandom output is produced efficiently from a short input, the output of a pseudorandom generator consists of strings of low generalized Kolmogorov complexity. If the pseudorandom generator is secure, then feasible adversaries are unable to distinguish truly random input from pseudorandom input. Thus the existence of pseudorandom generators says something about the ability of feasible processes to distinguish strings of high and low generalized Kolmogorov complexity.

Building on the intuition in the preceding paragraph, this paper examines several hypothesis about the security of pseudorandom generators and derives for each hypothesis a necessary condition in terms of Kolmogorov complexity. Using these conditions, a number of new results are proved which relate the security of pseudorandom generators to the existence of fast deterministic simulations of probabilistic computations.

The following two hypotheses are common in work relating to pseudorandom number generation:

* Supported in part by a 1986 Rutgers University Research Council Summer Fellowship.

0022-0000/89 $3.00

Hypothesis 1. *There exist generators which are secure against probabilistic polynomial time statistical tests.*

Hypothesis 2 (The strong hypothesis). *There exist generators which are secure against* P/*poly statistical tests.*

In his fundamental paper [39], Yao considered both hypotheses. Most of the results of [39] were stated in terms of Hypothesis 1. However, in order to prove results about probabilistic complexity classes, the stronger hypothesis was needed. Yao's result, as improved by [12], states that

$$\text{Hypothesis } 2 \Rightarrow \forall \varepsilon > 0 \ \text{BPP} \subseteq \text{DTIME}(2^{n^{\varepsilon}}).$$

It is not known if Hypothesis 2 can be replaced by Hypothesis 1 in this result. However, a result with a somewhat similar flavor is proved in Section 6:

$$\textbf{Result: } \text{Hypothesis } 1 \Rightarrow \forall \varepsilon > 0 \ \text{BPTIME}(2^{O(n^2)}) \subseteq \text{DTIME}(2^{2^{\varepsilon n^2}}).$$

One of the major thrusts of this investigation involves the consideration of the existence of generators which are secure against probabilistic adversaries which are allowed *more* than polynomial time. There are a number of reasons for doing this:

1. Certain relationships between Kolmogorov complexity and pseudorandom generators become clearer when strong notions of security are used.

2. Proof techniques which are useful for studying pseudorandom generators under polynomial-time security assumptions seem to be different from the techniques which are useful for studying pseudorandom generators which are secure against more powerful adversaries.

3. Little is known about which security assumptions are reasonable and which are too strong. By examining strong assumptions about security, we can hope to identify which assumptions are reasonable.

4. In some cryptographic applications of pseudorandom generators, it may be advisable to consider adversaries who have more than polynomial-bounded resources.

To this end, we consider new hypotheses, Hypothesis 3 and Hypothesis 4, which (informally) say that there exists some $\varepsilon > 0$ and some pseudorandom generator which is secure against all statistical tests computable by probabilistic $2^{\varepsilon n}$ time-bounded machines, or circuits of size $2^{\varepsilon n}$, respectively. We show that Hypothesis $3 \Rightarrow \text{RTIME}(2^{O(2^n)}) = \text{DTIME}(2^{O(2^n)})$, and Hypothesis $4 \Rightarrow \text{RP} = \text{P}$.

If one believes that probabilistic computation is more powerful than deterministic computation, then these results can be taken as evidence that Hypotheses 3 and 4 are unlikely to be true. On the other hand, it should be noted that some researchers have conjectured that $\text{RP} = \text{P}$ [9].

Although Hypothesis 3 is very strong, we note that it is probably not strong enough to imply Hypothesis 2.

Since the proofs of these results use intermediate lemmas about Kolmogorov complexity, we are able to prove a number of other results about the structure of complexity classes, under the assumption that secure generators exist. For example, we show that Hypothesis 3 implies that every dense set in RP has an infinite P-printable subset. Thus, for instance, our results show that there is an infinite P-printable set of primes, assuming that very secure pseudorandom generators exist. (Note in this regard that it has only recently been proved that there is an infinite set of primes in P [31].) Also, we show that Hypothesis 1 implies a certain immunity property for all dense sets in RP.

In order to make efficient use of Kolmogorov complexity as a tool in proving these results, we define, for each set L, a function $K_L(n)$ which measures the complexity of the simplest strings of length $n$ in L. This definition is built on the notions of generalized Kolmogorov complexity proposed by Hartmanis [18] and Levin [26].

The general technique used to prove the results mentioned above consists of two parts. First, one shows that if secure pseudorandom generators exist, then $K_L(n)$ grows slowly for every dense set L in P. Then, one shows that if $K_L(n)$ grows slowly for all dense sets in P, then there are fast deterministic simulations of probabilistic computations.

Once a connection has been established between probabilistic complexity classes on the one hand, and the rate of growth of $K_L(n)$ for dense sets L in P on the other hand, it is a small step to relate the complexity of sets in NP and NE to the rate of growth of $K_L(n)$ for *all* sets L in P (not just the dense sets). Thus, for example, we can show that if there are sets in NE which require doubly exponential time to recognize determinisitically, then there are sets L in P for which $K_L(n)$ is large.

These observations raise some interesting questions. For instance, note that every string in a sparse set in P has a short description (namely, its index in that set). Thus there is some reason to suspect that $K_L(n)$ might grow more *slowly* for sparse sets in P than for dense sets. That is, if $M$ is a polynomial-time machine which "singles out" a few strings of length $n$, it might seem reasonable to guess that those strings must be relatively simple.

On the other hand, either this intuition is wrong, or some popular conjectures are false, since results in this paper show that one of the following three things must happen:

1. Hypothesis 2 is false.
2. There are fast deterministic simulations for all sets in NE.
3. There is some non-dense set L in P and some $\varepsilon$ such that $K_L(n) > n^{\varepsilon}$ i.o., but $K_L(n) < n^{\delta}$ for all $\delta$ and all dense L in P (and even for all L in P/poly).

In Section 2, we present some basic definitions and establish notation. In Section 3, we review concepts and definitions related to the theory of pseudorandom generators. In Section 4, the notion of generalized Kolmogorov complexity is reviewed, and the function $\mathrm{K_L}(n)$, which measures the Kolmogorov complexity of the set L, is defined.

In Section 5, results are proved which relate the security of pseudorandom generators to the Kolmogorov complexity of sets in P and P/poly. In Section 6, we present results which relate the Kolmogorov complexity of sets in P to a number of open problems in complexity theory. For example, we show that

1. $(\forall \mathrm{L} \in \mathrm{P}, \mathrm{K_L}(n) = O(\log n)) \Rightarrow \mathrm{E} = \mathrm{NE}$
2. $(\forall \mathrm{L} \in \mathrm{P}, \mathrm{K_L}(n) \neq \omega(\log n)) \Rightarrow \mathrm{DTIME}(t(n)) \not\subseteq \mathrm{NP}$, for any time-constructible superpolynomial $t$
3. $(\forall \varepsilon > 0, \forall \mathrm{L} \in \mathrm{P}, \mathrm{K_L}(n) \neq \omega(n^{\varepsilon})) \Rightarrow \forall \varepsilon > 0, \mathrm{DTIME}(2^{n^{\varepsilon}}) \not\subseteq \mathrm{NP}$.
4. $\mathrm{RP} \not\subseteq \mathrm{DTIME}(2^{n^{\varepsilon}}) \Rightarrow \exists \delta, \exists \mathrm{L} \in \mathrm{P/poly}$, L is a.e. dense and $\mathrm{K_L}(n) = \omega(n^{\delta})$.

Section 7 contains results relating immunity properties of RP to the existence of secure pseudorandom generators. Finally, in Section 8 we discuss open problems and conclusions.

## 2. Definitions and Preliminaries

In this section, we present some basic definitions, and we review concepts related to pseudorandom generators.

All sets considered in this paper are subsets of $\{0, 1\}^*$. For all strings $x$, $|x|$ denotes the length of $x$. For a set $S$, $|S|$ denotes the cardinality of S. For a set $\mathrm{L} \subseteq \Sigma^*$, $\mathrm{L}^{=n}$ denotes $\mathrm{L} \cap \Sigma^n$. We make use of the usual correspondence between $\{0, 1\}^*$ and the positive integers; namely, the string $x$ will denote the integer whose binary representation is $1x$. Thus, for example, we may write $|x| = \lfloor \log x \rfloor$. We will use a one–one pairing function mapping $\{0, 1\}^* \times \{0, 1\}^*$ onto $\{0, 1\}^*$, and for inputs $x$ and $y$, we will denote the output of the pairing function by $\langle x, y \rangle$.

The census $c_L(n)$ of a set L is the number of strings in L of lenth $n$. We say L is *sparse* if $c_{\mathrm{L}}(n) = n^{O(1)}$. L is a *tally set* if $\mathrm{L} \subseteq 0^*$. An important class of sparse sets are the P-*printable* sets. A set L is P-printable if there is a deterministic Turing machine which, on input $n$, runs in time polynomial in $n$ and prints out a list of all the elements of L of length at most $n$. P-printable sets were defined in [19]; a number of results relating to P-printable sets may be found in [3].

The density $d_{\mathrm{L}}(n)$ of L is $c_{\mathrm{L}}(n)/2^n$. We will say that L is *dense* if for some $k$, L has density $\geqslant n^{-k}$ i.o. We will also have occasion to refer to sets which satisfy a stronger density requirement: we say that L is *a.e. dense* if L is infinite and for some $k$ and for all large $n$, $\mathrm{L}^{=n} \neq \varnothing \Rightarrow d_{\mathrm{L}}(n) \geqslant n^{-k}$. Notice that a.e. dense sets may contain *no* strings of many lengths $n$, however if L is a.e. dense and contains some strings of length $n$, then it contains many strings of length $n$.

We shall have occasion to make statements of the form "$g = O(f)$" or "$g = \omega(f)$", where $g(n)$ is undefined for some $n$. The assertion "$g = O(f)$" will thus mean there exists some $c$ such that for all large $n$, $g(n)$ defined $\Rightarrow g(n) < cf(n)$.

Similarly, "$g \neq \omega(f)$" means that there exists some $c$ and there are infinitely many $n$ such that $g(n)$ is defined, and $g(n) < cf(n)$.

The reader is expected to be familiar with deterministic, nondeterministic, and probabilistic Turing machines, and with complexity classes such as RP, BPP, NP, etc. For background, the reader is directed to [33 or 21].

A function $t\colon \mathbf{N} \to \mathbf{N}$ is said to be *time-constructible* if there is some Turing machine which, on all inputs of length $n$, runs for exactly $t(n)$ steps.

We let $E = \mathrm{DTime}(2^{O(n)})$, and $\mathrm{NE} = \mathrm{NTime}(2^{O(n)})$. We will also need to refer to complexity classes defined in terms of probabilistic machines which run in exponential time. We let BPTime(T($n$))) denote the class of languages L for which there exists some probabilistic Turing machine M which runs in time T($n$) such that $x \in \mathrm{L} \Rightarrow \mathrm{Prob}(\mathrm{M}\text{ accepts }x) > \frac{3}{4}$ and $x \notin \mathrm{L} \Rightarrow \mathrm{Prob}(\mathrm{M}\text{ accepts }x) < \frac{1}{4}$. Similarly, we define RTime(T($n$)) to be the class of languages L for which there exists some probabilistic Turing machine M which runs in time T($n$) such that $x \in \mathrm{L} \Rightarrow \mathrm{Prob}(\mathrm{M}\text{ accepts }x) > \frac{3}{4}$ and $x \notin \mathrm{L} \Rightarrow \mathrm{Prob}(\mathrm{M}\text{ accepts }x) = 0$.

## 3. Pseudorandom Generators

A pseudorandom generator is an algorithm which runs in polynomial time, takes a string of length $n$ (the *random seed*) as input, and produces a string of length $n^k$ (the *pseudorandom sequence*) as output, where $k > 1$. In order to be acceptable for many purposes, the pseudorandom sequences produced by a pseudorandom generator should satisfy certain statistical properties, and it should be difficult for any adversary, given the initial segment of a pseudorandom sequence, to predict which bit will be produced next by the generator. Important papers which have dealt with pseudorandom generators from a complexity-theoretic point of view include [35, 39, 17, 11, 27, 12, 29].

The following definitions are essentially those of [39]. A $T(n)$ *statistical test* is a probabilistic algorithm which runs in time $T(n)$. Given any generator $g$ and statistical test $A$, we will be interested in how the behavior of $A$ on pseudorandom strings compares with the behavior of $A$ on truly random strings. Suppose $g$ takes seeds of length $n$ and produces pseudorandom sequences of length $n^k$. Define $P_{A,n^k}(\mathrm{R})$ to be the probability that $A$ accepts $x$, where $x$ is a string of length $n^k$ (all strings of length $n^k$ being equally likely). Define $P_{A,n^k}(\mathrm{PS}g)$ to be the probability that $A$ accepts $g(s)$, where $s$ is a seed of length $n$ (all seeds of length $n$ being equally likely). We say that $g$ passes the $(T(n), e(n))$ test $A$ if $|P_{A,n^k}(\mathrm{R}) - P_{A,n^k}(\mathrm{PS}g)| \leqslant e(n)$ for all large enough $n$. That is, $g$ passes the $(T(n), e(n))$ test $A$ if $A$ behaves roughly the same on $g$'s pseudorandom output as on truly random strings. (*Warning*: a possible point of confusion is that $n$ is the length of the *seed*, i.e., $n$ is the length of the input to the generator $g$, rather than the length of the input to the test $A$. By adopting this convention, calculations are simplified, and certain lemmas are easier to state.)

We say that $g$ is $(T(n), e(n))$ secure if it passes every $(T(n), e(n))$ statistical test, and $g$ is polynomially secure if it is $(n^k, n^{-k})$ secure for every $k$.

There are other notions of security which have been studied. For example, in [11, 39] a *next bit test* was also considered. A generator $g$ fails a $(T(n), e(n))$ next bit test $A$ if $A$ is a probabilistic algorithm which runs in time $T(n)$ and if there are infinitely many $n$ such that $A$, when given an initial segment of a pseudorandom string of length $n^k$, can predict the next bit with success rate $>\frac{1}{2}+e(n)$. It was shown in [39] that a generator $g$ is polynomially secure iff it passes all $(n^k, n^{-k})$ next bit tests. See also [12] for a nice proof of this fact.

Pseudorandom generators are closely linked to one-way functions: i.e., functions which are easy to compute but are hard to invert. Purported one-way functions were used in [35, 39, 11] to construct pseudorandom generators. Levin [27] showed that the existence of polynomially secure pseudorandom generators is in fact equivalent to the existence of a certain type of one-way function; see [12] for nice proofs of some of Levin's results.

Boppana and Hirschfeld [12] introduced the notion of an *extender*, which is closely related to the notion of one-way functions. An extender is a function computable in polynomial time which takes input of length $n$ and produces output of length $n+1$. Thus an extender is a pseudorandom generator whose pseudorandom sequences are only one bit longer than the seed. The notion of security for extenders is defined in exactly the same way as for generators. That is, an extender $g$ is $(T(n), e(n))$ secure if it passes every $(T(n), e(n))$ statistical test, and $g$ is polynomially secure if it is $(n^k, n^{-k})$ secure for every $k$. It is shown in [12] that polynomially secure generators exist iff polynomially secure extenders exist.

The technique whereby a secure extender is transformed into a secure generator is required in a number of the proofs in this paper. To simplify those proofs, it is useful to set up some machinery here. What follows is a generalization of a technique used in [12, 39, 11].

DEFINITION. Let $x$ be any string of length $n$. Define head$(x)$ to be the first bit of $x$, and tail$(x)$ to be the suffix of $x$ of length $n-1$. Let $g$ be an extender. Define $b_i = \text{head} \circ g \circ [\text{tail} \circ g]^{i-1}(x)$. For any $r \in \mathbf{N}$ define $g_r(x) = b_1(x)\, b_2(x) \cdots b_r(x)$. In other words, $g_r(x)$ is the sequence of $r$ bits which results by computing $g(x)$, removing the first bit and applying $g$ to the result, and repeating the process $r$ times.

Given a language L, an extender $g$, and a number $r$, we denote by T(L, $g$, $r$) the statistical test which, on input $x$ of length $n+1$, does as follows:

```
Begin
    Probabilistically choose $i \in \{0, ..., r-1\}$
    Probabilistically choose $z \in \Sigma^{r-i-1}$
    Let $y = g_i(x)$, and let $b = \text{head}\ (x)$
    Accept iff $zby \in \text{L}$
End
```

Correction: This should be $y = g_i(\text{tail}(x))$

LEMMA 1. $P_{\mathrm{T}(\mathrm{L},g,r),n+1}(\mathrm{R}) - P_{\mathrm{T}(\mathrm{L},g,r),n+1}(\mathrm{PS}g) = [d_{\mathrm{L}}(r) - \mathrm{Prob}(g_r(y) \in \mathrm{L} \mid y \in \Sigma^n)]/r$.

*Proof.* Let $p_i$ denote the probability that $\mathrm{T}(\mathrm{L}, g, r)$ accepts $y \in \Sigma^{n+1}$, given that $i$ is chosen in step 1, where all $y$ are equally likely. Clearly, $P_{\mathrm{T}(\mathrm{L},g,r),n+1}(\mathrm{R}) = \sum_i p_i/r$.

Similarly, let $q_i$ denote the probability that $\mathrm{T}(L, g, r)$ accepts $\underline{f(x)} \in \Sigma^{n+1}$, given that $i$ is chosen in step 1, where all $x \in \Sigma^n$ are equally likely. Again, clearly $P_{\mathrm{T}(\mathrm{L},g,r),n+1}(\mathrm{PS}g) = \sum_i q_i/r$.

Should be $g(x)$

The crucial fact to note is that $p_{i+1} = q_i$ for $0 \leqslant i \leqslant r-2$, and $p_o = d_{\mathrm{L}}(r)$ and $q_{r-1} = \mathrm{Prob}(g_r(y) \in \mathrm{L} \mid y \in \Sigma^n)$. Thus $P_{\mathrm{T}(\mathrm{L},g,r),n+1}(\mathrm{R}) - P_{\mathrm{T}(\mathrm{L},g,r),n+1}(\mathrm{PS}g) = (\sum_i(p_i - q_i))/r = (p_0 - q_{r-1})/r$. The result follows. ▌

Note that, in the proof of Lemma 1, it was assumed that the number $i$ could be chosen in the range $\{0, ..., r-1\}$ with all $i$ being equally likely to be chosen. Depending on the model of a probabilistic Turing machine being used, that may or may not be possible. However, on all reasonable models of probabilistic Turing machine, it is possible to choose $i$ so that all $i$ are *approximately* equally likely to be chosen, so that, on such models of computation, the statement of Lemma 1 is true modulo some insignificant terms which we find it convenient to ignore. The interested reader may verify that all theorems and corollaries proved in this paper remain true without this simplifying assumption.

We remark that, although presented as a probabilistic algorithm, $\mathrm{T}(\mathrm{L}, r, g)$ can also be implemented as a circuit, where the size of the circuit is determined by the complexity of computing $g_r$ and by the complexity of determining membership in L. (The probabilistic aspects can be removed by "hardwiring" in the optimal choices for $i$ and $z$. For examples of arguments of this sort, see [12].)

Since secure generators exist iff secure extenders exist iff certain kinds of one-way functions exist, hypotheses about pseudorandom generators can be expressed using any of these notions. *To simplify the statement of certain results, this paper expresses all such hypotheses in terms of extenders.*

Up to this point, we have considered only statistical tests computable by probabilistic algorithms. A number of papers dealing with pseudorandom generators have used a much stronger notion of security (e.g., [11, 39, 12]). A *strong $T(n)$ statistical test* is an algorithm computed by a circuit of size $T(n)$. A generator is strongly polynomially secure if it passes all strong $(n^k, n^{-k})$ statistical tests. All of the results mentioned so far are also true in the context of strong statistical tests. For instance, a generator is secure against next bit tests computed by poly-size circuits iff it is strongly polynomially secure, and strongly polynomially secure generators exist iff strongly polynomially secure extenders exist.

However, in some cases, the strong notion of security is necessary in order to achieve the desired results. For instance, Yao shows in [39] that if strongly polynomially secure pseudorandom generators exist, then every language in RP has subexponential time complexity. His proof relies heavily on the strong hypothesis of security.

Little is known about how secure it is possible for an extender to be. Clearly, if

P = NP, then no extender is polynomially secure, since if P = NP it would be possible to recognize the range of the extender in polynomial time. For essentially the same reason, no extender computable in time $n^k$ is $(2^n n^k, \frac{1}{2})$ secure. When less than exponential time is allowed, however, the bounds on security which we are able to prove plummet quickly. The following proposition seems to be the best bound which is known.

PROPOSITION 2. *No extender computable in time $n^k$ is $(2^{\varepsilon n}, 2^{-[(1-\varepsilon)n+k\log n]})$ secure, for any $\varepsilon > 0$.*

*Proof.* Let $g$ be any extender computable in time $n^k$. Consider the $2^{\varepsilon n}$ test $A$ which, on input $y$ of length $n+1$ probabilistically guesses a set $S \subseteq \Sigma^n$ of size $r = 2(2^{\varepsilon n}/n^k)$, and computes $g(x)$ for all $x \in S$, accepting if $g(x) = y$ for some $x$:

$$\begin{aligned} P_{A,n+1}(R) &= \sum_{y \in \Sigma^{n+1}} \frac{1}{2^{n+1}} \frac{1}{\binom{2^n}{r}} |\{S: y \in g(S)\}| \\ &= \frac{1}{2^{n+1}} \frac{1}{\binom{2^n}{r}} \sum_S |g(S)| \\ &\leqslant \frac{1}{2^{n+1}} \frac{1}{\binom{2^n}{r}} \sum_S r \\ &= \frac{r}{2^{n+1}} \end{aligned}$$

$$P_{A,n+1}(\mathrm{PS}g) \geqslant \frac{\binom{2^n-1}{r-1}}{\binom{2^n}{r}} = \frac{r}{2^n}.$$

Thus $P_{A,n+1}(\mathrm{PS}g) - P_{A,n+1}(R) \geqslant r/2^{n+1} = 2^{-[(1-\varepsilon)n+k\log n]}$. ▮

Using similar analysis, one can show that no extender $g$ is strongly $(2^{\varepsilon n}, 2^{-(1-\varepsilon)n})$ secure. (The circuit can have elements of range($g$) encoded directly in it; this elimates the need to compute $g$, and thus eliminates the factor of $k \log n$.) It would be interesting to know if there can be extenders which meet the security bounds, or if there is any strategy which is significantly better than the naïve strategy outlined above. Results presented later in this paper indicate that, in particular, it would be interesting to know if it is possible for there to be extenders which are strongly $(2^{\varepsilon n}, 2^{-\varepsilon n})$ secure for some $\varepsilon > 0$. (Proposition 2 shows that, for this to happen, it must be that $\varepsilon < \frac{1}{2}$.) If such secure extenders exist, then RP = P (Corollary 31).

We can now state the four hypotheses about security of pseudorandom generators which we will use in the rest of the paper:

HYPOTHESIS 1. *There exist extenders computable in polynomial time which are polynomially secure.*

HYPOTHESIS 2. *There exist extenders computable in polynomial time which are strongly polynomially secure.*

HYPOTHESIS 3. *There exists an extender computable in polynomial time which is* $(2^{\varepsilon n}, 2^{-\varepsilon n})$ *secure, for some* $\varepsilon > 0$.

HYPOTHESIS 4. *There exists an extender computable in polynomial time which is strongly* $(2^{\varepsilon n}, 2^{-\varepsilon n})$ *secure, for some* $\varepsilon > 0$.

We should mention that other notions of pseudorandomness have been considered in the literature. Ko [22] and Wilber [38] consider infinite pseudorandom sequences. Longpré [29] compares the notion of infinite pseudorandom sequences to the notion of pseudorandom generators considered here. Pseudorandom generators which are secure relative to circuits of restricted depth were considered in [32, 6].

## 4. TIME-BOUNDED KOLMOGOROV COMPLEXITY

In this section we review concepts related to Kolmogorov complexity and define a new measure of the time-bounded Kolmogorov complexity of a set.

There have been many attempts to give a rigorous definition for the informal concept of "randomness." One approach which has met with some success is to consider a string of length $n$ to be *random* if it has no description of length less than $n$. This is the approach of Kolmogorov complexity (see, e.g., [24, 13]).

Kolmogorov complexity provides a framework for studying the complexity of finite objects. This theory has proved useful in simplifying counting arguments in proving lower bounds (e.g., [28]) and has appealing parallels to information theory [14]. Nonetheless, a shortcoming of Kolmogorov complexity is that it is based on recursion theory and fails to take time and space complexity into consideration. That is, a string has small Kolmogorov complexity if it can be constructed (effectively) from only a few bits of information, even if the construction process requires an enormous amount of resources. A number of researchers have considered variants of Kolmogorov complexity which deal explicitly with issues of time and space complexity [1, 20, 22, 30, 36, 37]. We will be most concerned with the definitions proposed by Hartmanis [18] and Levin [26].

Let $M_v$ be a Turing machine. Following [18], define $K_v[s(n), t(n)]$ to be $\{x \in \{0, 1\}^* : \exists y \in \{0, 1\}^*, |y| \leqslant s(|x|)$ and $M_v$ prints $x$ on input $y$ in $\leqslant t(|x|)$ steps$\}$. That is, $K_v[s(n), t(n)]$ is the set of strings which can be "recovered" by $M_v$ in time $t(n)$ from strings of length $\leqslant s(n)$. It was shown in [18] that there is a machine $M_u$ (called a *universal* Turing machine) such that for all $v$ there is a constant $c$ such that for all $s$ and $t$ $K_v[s(n), t(n)] \subseteq K_u[s(n)+c, ct(n) \log t(n)+c]$. We will let $K[s(n), t(n)]$ denote $K_u[s(n), t(n)]$. Many recent papers deal with this notion of generalized Kolmogorov complexity, including [3, 7, 23, 29].

A much older notion of time-bounded Kolmogorov complexity was used by Levin in [25] and was defined more formally in [26]. We restate that definition here. Let $M_u$ be the universal Turing machine considered above. Define $\mathrm{Kt}(x)$ to be $\min\{|p| + \log t\colon M_u \text{ prints } x \text{ in } \leqslant t \text{ steps on input } p\}$. Similarly, we define $\mathrm{Kt}(x \mid y)$ to be $\min\{|p| + \log t\colon M_u \text{ prints } x \text{ in } \leqslant t \text{ steps on input } \langle p, y\rangle\}$.

The following proposition, which is implicit in [25], illustrates the nature of the measure Kt.

PROPOSITION 3. *Let* $\mathrm{L} \in \mathrm{NP} - \mathrm{DTIME}(t(n))$, *and let* $Q$ *be a linear-time predicate such that* $x \in \mathrm{L} \Leftrightarrow \exists y Q(x, y)$. *Then there exists an infinite sequence* $x_1, x_2, \ldots$ *of strings in* L *such that* $\forall x_i \forall y\ (Q(x_i, y) \Rightarrow \mathrm{Kt}(y \mid x) > \log \sqrt{t(|x_i|)})$.

*Proof.* Assume that no such infinite sequence exists. That is, for all large $x$ in L $\exists y(Q(x, y) \wedge \mathrm{Kt}(y \mid x) \leqslant \log \sqrt{t(|x|)})$. Then on input $x$ a witness for $x$ can be found, if one exists, by running $M_u(\langle p, x\rangle)$ for $\sqrt{t(|x|)}$ steps, for all strings $p$ of length at most $\log \sqrt{t(|x|)}$, which can clearly be done in time $t(|x|)$. ▮

The definition due to Hartmanis has the advantage that the time to construct a string and the size of the description are *both* specified, so that some finer distinctions can be made than using Levin's Kt function. On the other hand, in this paper we find it convenient to have a function which measures the complexity of a string; Levin's definition is much better suited to this task than the definition of Hartmanis.

The following easy proposition shows that the two notions are closely related.

PROPOSITION 4. $\mathrm{Kt}(x) \leqslant r(|x|) \Rightarrow x \in \mathrm{K}[r(n), 2^{r(n)}] \Rightarrow \mathrm{Kt}(x) \leqslant 2r(|x|)$.

Some recent papers (e.g., [3, 7]) have considered sets which are subsets of $\mathrm{K}[s(n), t(n)]$ for small $s$ and $t$. Note that this approach tends to equate the Kolmogorov complexity of a language L with the complexity of the *most* complex strings in L. One goal of this paper is to show that there are reasons to be interested in the complexity of the *least* complex strings in L. To that end, we define the following measure of the complexity of strings in a set L.

DEFINITION. Let $\mathrm{L} \subseteq \{0, 1\}^*$. Define $\mathrm{K_L}(n)$ to be equal to $\min\{\mathrm{Kt}(x)\colon x \in \mathrm{L}^{=n}\}$ (If $\mathrm{L}^{=n} = \varnothing$, then $\mathrm{K_L}(n)$ is undefined.)

The rest of this paper can be viewed as an investigation into the properties of $\mathrm{K_L}$, for sets L in P.

Note that there is a conflict between the notion of pseudorandomness and the notion of Kolmogorov randomness. If $x$ is a pseudorandom sequence of length $n^k$ produced from a seed of length $n$, then $x$ is a string of very low generalized Kolmogorov complexity. Many of the results in this paper are proved by taking advantage of this conflict.

## 5. Pseudorandomness and Kolmogorov Complexity

In this section, we investigate hypotheses about the security of pseudorandom generators and derive necessary conditions, in terms of generalized Kolmogorov complexity, for these hypotheses to be true.

Lemma 5. *Let $r$ be any time-constructible function, such that $r(n) < 2^n$. If there is a set $\mathrm{L} \in \mathrm{DTIME}(n^k)$ such that $\mathrm{K}_{\mathrm{L}}(r(n)) \neq O(n)$, then no extender computable in time $n^l$ is $(t(n), e(n))$ secure, where $t(n) \geqslant r(n)\, n^l + r(n)^k$ and $e(n) < d_{\mathrm{L}}(r(n))/r(n)$.*

*Proof.* Let $g$ be an extender computed in time $n^l$. Assuming the existence of the set L in the hypothesis of the theorem, we will present a $(t(n), e(n))$ test which $g$ fails.

The test is simply this: on input $y$ of length $n+1$, run $\mathrm{T}(\mathrm{L}, r(n), g)$, where $\mathrm{T}(\mathrm{L}, r(n), g)$ is the test presented in Lemma 1. The time required to run the test is $r(n) + r(n)\, n^l + r(n)^k$, and thus this is a $t(n)$ test.

Note that for all $x \in \Sigma^n$, $\mathrm{Kt}(g_{r(n)}(x)) \leqslant n + \log(r(n)\, n^l) + O(1) = n + \log r(n) + O(\log n) < 3n$ for all large $n$. Since $\mathrm{K}_{\mathrm{L}}(r(n)) \neq O(n)$, there are infinitely many $n$ such that $g_{r(n)}(\Sigma^n) \cap \mathrm{L} = \varnothing$. I.e., there are infinitely many $n$ such that $\mathrm{Prob}(g_{r(n)}(y) \in \mathrm{L} \mid y \in \Sigma^n) = 0$. By Lemma 1, we have that for any such $n$,

$$P_{\mathrm{T}(\mathrm{L},\, g,\, r(n)),\, n+1}(\mathrm{R}) - P_{\mathrm{T}(\mathrm{L},\, g,\, r(n)),\, n+1}(\mathrm{PS}g) = d_{\mathrm{L}}(r(n))/r(n) > e(n).$$

Thus $g$ fails the $(t(n), e(n))$ test. ▮

Corollary 6. Hypothesis 1 $\Rightarrow$ *for all a.e. dense sets* $\mathrm{L} \in \mathrm{P}$, $\forall k$, $\mathrm{K}_{\mathrm{L}}(n^k) = O(n)$.

Corollary 7. Hypothesis 3 $\Rightarrow$ *for all a.e. dense sets* $\mathrm{L} \in \mathrm{P}$, $\exists \varepsilon > 0$, $\mathrm{K}_{\mathrm{L}}(2^{\varepsilon n}) = O(n)$.

*Proof.* To see that Corollary 6 is true, assume that there is a set L in $\mathrm{DTIME}(n^k)$ and numbers $s$ and $r$ such that for all large $n$, $\mathrm{L}^{=n} \neq \varnothing \Rightarrow d_{\mathrm{L}}(n) > 1/n^s$ and $\mathrm{K}_{\mathrm{L}}(n^r) \neq O(n)$. Let $g$ be any extender computable in time $n^l$. It suffices to show that $g$ is not polynomially secure. This is immediate, since it follows directly from the lemma that $g$ is not $(n^r n^l + n^{rk}, 1/n^{rs+r})$ secure.

Similarly, to see that Corollary 7 is true, assume that there is a set L in $\mathrm{DTIME}(n^k)$ such that for all large $n$, $\mathrm{L}^{=n} \neq \varnothing \Rightarrow d_{\mathrm{L}}(n) > 1/n^s$, and for all $\varepsilon > 0$, $\mathrm{K}_{\mathrm{L}}(2^{\varepsilon n}) \neq O(n)$. Let $g$ be an extender computable in time $n^l$, and let $\delta > 0$. It suffices to show that $g$ is not $(2^{\delta n}, 2^{-\delta n})$ secure. Note that if $\varepsilon$ is chosen small enough, then $2^{\delta n} > 2^{\varepsilon n} n^l + 2^{k \varepsilon n}$, and $2^{-\delta n} < 1/2^{s \varepsilon n + \varepsilon n}$. Now it follows from Lemma 5 that $g$ is not $(2^{\delta n}, 2^{-\delta n})$ secure. ▮

Given an a.e. dense set L in P, and assuming that secure pseudorandom generators exist, Corollaries 6 and 7 allow us to conclude that L contains simple strings of *many* lengths, but we cannot conclude that it contains simple strings of *all* lenths. That is, it would be more satisfying if Corollaries 6 and 7 could be improved so that one could conclude that $\mathrm{K}_{\mathrm{L}}(n) = O(n^{1/k})$ or $\mathrm{K}_{\mathrm{L}}(n) = O(\log n)$, respectively, but we do not know if such an improvement is possible. Although it is possible to improve these corollaries somewhat, to slightly expand the class of lengths about which some conclusions can be drawn, we are still only able to derive conclusions about a vanishingly small fraction of all lengths. On the other hand, the following lemma allows us to draw stronger conclusions from Hypotheses 2 and 4.

LEMMA 8. *Let $s$ be any real-valued monotone increasing function such that $\log n < s(n)$. If there is a set* L *of a.e. density $\geqslant 1/n^a$ which is accepted by a family of circuits of size $n^k$ such that $\mathrm{K}_{\mathrm{L}}(n) \neq O(s(n))$, then no extender computable in time $n^l$ is $(t(n), e(n))$, secure, for any $t(n) \geqslant (s^{-1}(n+1))^k + s^{-1}(n+1)\, n^l$ and $e(n) < 1/(s^{-1}(n+1))^{a+1}$.*

*Proof.* Assume that $s$ and L are as given in the hypothesis. Let $g$ be computable in time $n^l$, and let $t$ and $e$ satisfy the given bounds. We must build a $(t(n), e(n))$ test which $g$ fails for infinitely many $n$.

We are given that there are infinitely many $r$ such that $\mathrm{K}_{\mathrm{L}}(r) > 3s(r)$. For each such $r$, consider $n = \lfloor s(r) \rfloor$ and consider the behavior of the test T(L, $r$, $g$) on inputs of size $n$. The test can be computed by a circuit of size $r^k + rn^l$. Since $s$ (and hence $s^{-1}$) are monotone nondecreasing, it follows that $r \leqslant s^{-1}(n+1)$, and thus the test is a $t(n)$ test.

Notice that for any string $y$ of length $n$, $\mathrm{Kt}(g_r(y)) \leqslant (n + 2\log r) + \log(rn^k) + O(1) < 3n$ for all large $n$. Thus for all of these $n$, $\mathrm{Prob}(g_r(y) \in \mathrm{L} \mid y \in \Sigma^n) = 0$, and thus by Lemma 1, $P_{\mathrm{T}(\mathrm{L},\, g,\, r),\, n+1}(\mathrm{R}) - P_{\mathrm{T}(\mathrm{L},\, g,\, r),\, n+1}(\mathrm{PSg}) = d_{\mathrm{L}}(r)/r \geqslant 1/r^{a+1} \geqslant 1/(s^{-1}(n+1))^{a+1} > e(n)$. Thus $g$ fails the $(t(n), e(n))$ test. ▌

COROLLARY 9. Hypothesis $2 \Rightarrow \forall \varepsilon > 0$, $\mathrm{K}_{\mathrm{L}}(n) = O(n^{\varepsilon})$ *for all a.e. dense sets* L *in* P/*poly*.

COROLLARY 10. Hypothesis $4 \Rightarrow \mathrm{K}_{\mathrm{L}}(n) = O(\log n)$ *for all a.e. dense sets* L *in* P/*poly*.

Note that Corollary 6 tells us that, if Hypothesis 1 is true and L is an a.e. dense set in P, then $\mathrm{K}_{\mathrm{L}}(n) \neq \omega(n^{\varepsilon})$ for any $\varepsilon > 0$. The following lemma allows us to drop the "a.e." condition on L.

LEMMA 11. *Let $r$ be time-constructible and monotone increasing, $r(n) < 2^n$, and let $s$ be a monotone-increasing real-valued function such that $s(r(n)) = n$ for all $n \in \mathbf{N}$. If there is a set $\mathrm{L} \in \mathrm{DTIME}(n^k)$ such that $\mathrm{K}_{\mathrm{L}}(n) = \omega(s(n))$ and $d_{\mathrm{L}}(n) \geqslant n^a$* i.o., *then no extender computable in time $n^l$ is $(t(n), e(n))$ secure, for any $t(n) \geqslant r(n+1)\, n^l + r(n+1)^k$, and $e(n) < 1/r(n+1)^{a+2}$.*

*Proof.* Assume that L satisfies the conditions given above, and let $g$ be an extender computable in time $n^l$. Let $A$ be the statistical test which, on input $x$ of length

$n+1$, probabilistically chooses $b \in \{r(n), ..., r(n+1)\}$, and then runs $\mathrm{T}(\mathrm{L}, b, g)$, A runs in time $t(n)$. Clearly,

$$\begin{aligned} &P_{A,n+1}(\mathrm{R}) - P_{A,n+1}(\mathrm{PS}g) \\ &\quad = \frac{1}{r(n+1)-r(n)} \sum_b P_{\mathrm{T}(\mathrm{L},g,b),n+1}(\mathrm{R}) - P_{\mathrm{T}(\mathrm{L},g,b),n+1}(\mathrm{PS}g) \\ &\quad \geqslant \frac{1}{r(n+1)} \sum_b \frac{d_{\mathrm{L}}(b) - \mathrm{Prob}(g_b(y) \in L \mid y \in \Sigma^n)}{b}. \end{aligned}$$

By assumption, for all large $b$, $\mathrm{K}_{\mathrm{L}}(b) > 3s(b) \geqslant 3(s(r(n))) = 3n$. Note also that for any $y$ of length $n$, $\mathrm{Kt}(g_b(y)) \leqslant n + \log(bn^l) + O(1) \leqslant n + \log r(n+1) + O(\log n) < 3n$ for all large $n$. That is, for all large $b$, $\mathrm{Prob}(g_b(y) \in \mathrm{L} \mid y \in \Sigma^n) = 0$. Thus, by Lemma 1, for all large $n$,

$$P_{A,n+1}(\mathrm{R}) - P_{A,n+1}(\mathrm{PS}g) \geqslant \frac{1}{r(n+1)} \sum_b \frac{d_{\mathrm{L}}(b)}{b} \geqslant \frac{1}{r(n+1)^2} \sum_b d_{\mathrm{L}}(b).$$

Since $d_{\mathrm{L}}(b) \geqslant 1/b^a \geqslant 1/(r(n+1))^a$ i.o., we have that $\mathrm{P}_{A,n+1}(\mathrm{R}) - \mathrm{P}_{A,n+1}(\mathrm{PS}g) \geqslant 1/(r(n+1))^{a+2}$ for infinitely many $n$. ▮

Corollary 12. Hypothesis $1 \Rightarrow \forall \varepsilon > 0$, $\mathrm{K}_{\mathrm{L}}(n) \neq \omega(n^{\varepsilon})$ for all dense sets L in P.

Corollary 13. Hypothesis $3 \Rightarrow \mathrm{K}_{\mathrm{L}}(n) \neq \omega(\log n)$ *for all dense sets* L *in* P.

Corollary 14. Hypothesis $3 \Rightarrow$ *every dense set in* P *has an infinite* P-*printable subset.*

*Proof.* It was shown in [3] that a set L is P-printable iff L is in P and for some $k$, $\mathrm{L} \subseteq \mathrm{K}[k \log n, n^k]$. Equivalently, L is P-printable iff L is in P and for some $k$, $\mathrm{Kt}(x) < k \log n$ for all $x \in \mathrm{L}$.

Now assume Hypothesis 3, and let L be a dense set in P. By Corollary 13, there is some $k$ such that for infinitely many $x \in \mathrm{L}$, $\mathrm{Kt}(x) < k \log n$. Let $A$ be the set of all strings $y$ such that $\mathrm{Kt}(y) < k \log n$. It is easy to check that A is in **P**, and thus $\mathrm{L} \cap \mathrm{A}$ is an infinite P-printable subset of L. ▮

## 6. The Power of Probabilistic and Nondeterministic Computation

In the previous section, hypotheses about the security of pseudorandom generators were shown to have consequences about the rate of growth of functions of the sort $\mathrm{K}_{\mathrm{L}}(n)$, where L is a sufficiently dense set in P. In this section, we show that the rate of growth of $\mathrm{K}_{\mathrm{L}}(n)$ is also closely related to various open questions about the relationships among deterministic, probabilistic, and nondeterministic complexity classes.

Hartmanis has raised the question of what can be said about the complexity of $\mathrm{K}[n/2, n^3]$. It seems unlikely that $\mathrm{K}[n/2, n^3]$ is in P, although there is little hope of proving that directly, since $K[n/2, n^3]$ is in NP. Given that there is some vague intuition that $\mathrm{K}[n/2, n^3]$ is not in P, it is tempting to suggest that behind that intuition, there is a feeling that polynomial-time machines cannot distinguish complex strings from easy ones. The belief in the existence of secure pseudorandom number generators is a manifestation of that feeling. The results in this section constitute an examination of the consequences of the hypothesis that, given a machine M which runs in polynomial time, if M accepts infinitely many complex strings, then M must also accept infinitely many noncomplex strings.

First we must introduce the notions of a.e. complexity and bi-immunity. Following [15], we say that a set $L$ is *a.e. t-complex* if every algorithm accepting $L$ requires more than time $t(|x|)$ for all large inputs $x$. It can be shown [15, 34] that if $T$ is a time-constructible function, and $t(n) \log t(n) = o(T(n))$, then there is a set L in $\mathrm{DTIME}(T(n))$ which is a.e. $t$-complex.

A set L is said to be *immune* with respect to a class $C$ of sets if L is infinite and has no infinite subset in $C$. $L$ is *bi-immune* with respect to $C$ if both L and its complement are immune with respect to $C$. As was pointed out in [15], the notions of bi-immunity and a.e. complexity are closely related. Specifically, if L is a.e. $t$-complex, then L is bi-immune with respect to $\mathrm{DTIME}(t(n))$. Furthermore, if $t$ is time-constructible and L is bi-immune with respect to $\mathrm{DTIME}(t(n))$, then L is a.e. $t$-complex. (On the other hand, it can be shown that there are non-time-constructible functions $t$ and sets L which are bi-immune with respect to $\mathrm{DTIME}(t(n))$, but are not a.e., $t$-complex.)

Combining the facts from the previous two paragraphs, we observe that if $T$ is a time-constructible function and $t(n) \log t(n) = o(T(n))$, then there is a set L in $\mathrm{DTIME}(T(n))$ which is bi-immune with respect to $\mathrm{DTIME}(t(n))$. The results below make use of this fact.

THEOREM 15. *If there is a set* $A \in \mathrm{NTIME}(n^{k-1})$ *which is immune with respect to* $\mathrm{DTIME}(2^{bs(n^k)})$ *for all b, then* $\exists \mathrm{L} \in \mathrm{DTIME}(O(n))$ *such that* $\mathrm{K_L}(n) = \omega(s(n))$.

*Proof.* Assume that every set L in $\mathrm{DTIME}(\mathrm{O}(n))$ has $\mathrm{K_L}(n) \neq \omega(s(n))$. It will suffice to show that for every $\mathrm{A} \in \mathrm{NTIME}(n^{k-1})$ there is some $b$ such that A has an infinite subset in $\mathrm{DTIME}(2^{bs(n^k)})$.

Let A be accepted by a nondeterministic machine M running in time $n^{k-1}$. Let $\mathrm{L} = \{\langle x, y\rangle : |\langle x, y\rangle| = n^k$ and $|x| = n$ and $y$ encodes an accepting computation of M on $x\} \in \mathrm{DTIME}(O(n))$. By assumption, there is some $c$ such that $\mathrm{K_L}(n) < cs(n)$ for infinitely many $n$. Let $b > 2c$. It is easy to verify that the following routine accepts an infinite subset of L in time $2^{bs(n^k)}$:

```
Begin
    On input $x$ of length $n$,
    For all $z$ of length $\leqslant cs(n^k)$, run $\mathrm{M}_u(z)$ for $2^{cs(n^k)}$ steps.
    If for some $z \exists y$, $M_u(z)$ prints out $\langle x, y\rangle \in L$, accept.
    Else, reject.
End ▮
```

Corollary 16. $\exists \varepsilon\ \mathrm{DTIME}(2^{n^\varepsilon}) \subseteq \mathrm{NP} \Rightarrow \exists \delta\ \exists \mathrm{L} \in \mathrm{DTIME}(O(n)), \mathrm{K_L}(n) = \omega(n^\delta)$.

*Proof.* There is an infinite $\mathrm{A} \in \mathrm{DTIME}(2^{n^\varepsilon})$ which has no infinite subset in $\mathrm{DTIME}(2^{n^\gamma})$ for any $\gamma < \varepsilon$. If A is in NP, then it is in $\mathrm{NTIME}(n^{k-1})$ for some $k$. Choosing $\delta < \varepsilon/k$ satisfies the conditions of Theorem 15. ▮

Corollary 17. *Let* $T(n)$ *be any time-constructible function such that* $\forall k$, $T(n) = \omega(n^k)$. $\mathrm{DTIME}(T(n)) \subseteq \mathrm{NP} \Rightarrow \exists \mathrm{L} \in \mathrm{DTIME}(O(n))$, $\mathrm{K_L}(n) = \omega(\log n)$. (*Equivalently, if every infinite set in* P *has an infinite* P-*printable subset, then* NP *does not contain any deterministic time class larger than* P.)

*Proof.* There is an infinite $\mathrm{A} \in \mathrm{DTIME}(T(n))$ with has no infinite subset in P. The corollary now follows directly from Theorem 15. ▮

Theorem 18. *If there is a dense set* $A \in \mathrm{RTIME}(n^{k-1})$ *which is immune with respect to* $\mathrm{DTIME}(2^{bs(n^k)})$ *for all b, then there is a dense* $\mathrm{L} \in \mathrm{DTIME}(O(n))$ *such that* $\mathrm{K_L}(n) = \omega(s(n))$.

*Proof.* The proof of this result is really the same as the proof of Theorem 15. It suffices to note that the set $\mathrm{L} = \{\langle x, y\rangle : |\langle x, y\rangle| = n^k$ and $|x| = n$ and $y$ encodes an accepting computation of M on $x\} \in \mathrm{DTIME}(O(n))$ has density $d_L(n^k) \geqslant d_A(n)/2$. ▮

Corollary 19. $\exists \varepsilon, \mathrm{DTIME}(2^{n^\varepsilon}) \subseteq \mathrm{RP} \Rightarrow \exists \delta, \exists \mathrm{L} \in \mathrm{DTIME}(O(n))$, L *is dense and* $\mathrm{K_L}(n) = \omega(n^\delta)$.

*Proof.* This is essentially the same as the proof of Corollary 16. It need only be noted that since there is a set $\mathrm{A} \in \mathrm{DTIME}(2^{n^\varepsilon})$ which is bi-immune with respect to $\mathrm{DTIME}(2^{n^\gamma})$ for all $\gamma < \varepsilon$, it must be the case that either A or its complement is dense. ▮

Corollary 20. *Let* $T(n)$ *be any time-constructible function such that* $\forall k$, $T(n) = \omega(n^k)$. $\mathrm{DTIME}(T(n)) \subseteq \mathrm{RP} \Rightarrow \exists\ \mathrm{L} \in \mathrm{DTIME}(O(n))$ *such that* L *is dense and* $\mathrm{K_L}(n) = \omega(\log n)$. (*Equivalently, if every dense set in* P *has an infinite* P-*printable subset, then* RP *does not contain any deterministic time class larger than* P.)

The preceding four corollaries have the form $\mathrm{K_L}(n) \neq \omega(s(n)) \Rightarrow \mathrm{DTIME}(t(n) \nsubseteq \mathrm{NP}$ (or RP), for various functions $s$ and $t$ and conditions on L. In an earlier version of this paper [4], it was remarked that even stronger conclusions could be seen to follow; namely under the given hypotheses, one can conclude that $\mathrm{DTIME}(t(n)) \nsubseteq \mathrm{NP} \cup \mathrm{coNP}$, since it can be shown that $\mathrm{DTIME}(t(n)) \subseteq \mathrm{NP} \cup \mathrm{coNP} \Rightarrow$

$\mathrm{DTIME}(t(n)) \subseteq \mathrm{NP}$. In the meantime, it has been shown that a slightly stronger result holds; namely $\mathrm{DTIME}(t(n)) \leqslant^{p}_{1-tt} \mathrm{NP} \Rightarrow \mathrm{DTIME}(t(n)) \subseteq \mathrm{NP}$ [5]. Somewhat surprisingly, no further strengthening along these lines can be expected, since there is an oracle relative to which $\mathrm{DTIME}(t(n)) \leqslant^{p}_{2-tt} \mathrm{NP}$ and $\mathrm{DTIME}(t(n)) \nsubseteq \mathrm{NP}$ [5].

The following results relate the $\mathrm{K}_{\mathrm{L}}$ complexity of dense sets in P/poly to the deterministic complexity of sets in RP. Note, in this regard, that there is nothing interesting to be said about the $\mathrm{K}_{\mathrm{L}}$ complexity of *non-dense* sets in P/poly, since a set in P/poly can consist of an infinite sequence of Kolmogorov-random strings.

THEOREM 21. *If* $\mathrm{RTIME}(n^k) \nsubseteq \mathrm{DTIME}(t(n))$, *then there is an a.e. dense set* $\mathrm{L} \in \mathrm{P}/poly$ *such that* $\mathrm{K}_{\mathrm{L}}(n)$ *defined* $\Rightarrow \mathrm{K}_{\mathrm{L}}(n) > \log \sqrt{t(n^{1/k})}$.

*Proof.* Let $\mathrm{A} \in \mathrm{RTIME}(n^k)$–$\mathrm{DTIME}(t(n))$, and let $Q$ be a linear-time predicate such that $x \in \mathrm{A} \Leftrightarrow \exists y, |y| = |x|^k \wedge Q(x, y)$, and furthermore $x \in A \Rightarrow$ at least half of the strings $y$ of length $|x|^k$ satisfy $Q(x, y)$. By Proposition 3, there is an infinite sequence $x_1, x_2, \ldots$ of strings in A such that $\forall x_i, \forall y, (Q(x_i, y) \Rightarrow \mathrm{Kt}(y) \geqslant \mathrm{Kt}(y \mid x) > \log \sqrt{t(|x_i|)})$. Asssume without loss of generality that no two strings in the sequence are of the same length. Let $\mathrm{L} = \{y \colon \exists i\, |y| = |x_i|^k \wedge Q(x_i, y)\}$. L is easily seen to be a.e. dense, and L is in P/poly, since it can easily be recognized using a family of circuits which has the sequence of $x_i$'s hardwired in. Also, note that $\mathrm{L}^{=m} \neq \varnothing \Rightarrow m = n^k$ for some $k$ and $\mathrm{K}_{\mathrm{L}}(m) = \mathrm{K}_{\mathrm{L}}(n^k) > \log \sqrt{t(n)} = \log \sqrt{t(m^{1/k})}$. ▮

COROLLARY 22. $\mathrm{RP} \nsubseteq \mathrm{DTIME}(2^{n^{\varepsilon}}) \Rightarrow \exists \delta, \exists \mathrm{L} \in \mathrm{P}/\mathrm{poly}$, L is a.e. dense and $\mathrm{K}_{\mathrm{L}}(n) = \omega(n^{\delta})$ (and thus $\mathrm{K}_{\mathrm{L}}(n) \neq O(n^{\delta})$).

*Proof.* Let $A \in \mathrm{RTIME}(n^k)$–$\mathrm{DTIME}(2^{n^{\varepsilon}})$. By Theorem 21 there is an a.e. dense set $\mathrm{L} \in \mathrm{P}/\mathrm{poly}$ such that $\mathrm{K}_{\mathrm{L}}(n) > (n^{\varepsilon/k})/2$ everywhere that $\mathrm{K}_{\mathrm{L}}(n)$ is defined. Choosing $\delta < \varepsilon/k$ satisfies the claim. ▮

COROLLARY 2.3. $\mathrm{RP} \neq \mathrm{P} \Rightarrow \exists \mathrm{L} \in \mathrm{P}/\mathrm{poly}$, L *is a.e. dense and* $\mathrm{K}_{\mathrm{L}}(n) = \omega(\log n)$ (*and thus* $\mathrm{K}_{\mathrm{L}}(n) \neq O(\log n)$).

*Proof.* Let $A \in \mathrm{RTIME}(n^k) - \mathrm{P}$, and $Q$ be the linear-time predicate associated with A, as in the proofs of Proposition 3 and Theorem 21. Since $A \notin \mathrm{DTIME}(n^l)$ for all $l$, it follows that there is an infinite sequence $x_1, x_2, \ldots$ of strings in A such that $\forall l, \forall y, (Q(x_l, y) \Rightarrow \mathrm{Kt}(y \mid x_l) > (l/2k) \log n)$. Letting $\mathrm{L} = \{y \colon \exists l\, |y| = |x_l|^k \wedge Q(x_l, y)\}$, it is easy to verify that L is a dense set in P/poly, and for all constants $c$ and all large enough $n$, $\mathrm{K}_{\mathrm{L}}(n)$ defined $\Rightarrow \mathrm{K}_{\mathrm{L}}(n) > c \log n$. ▮

Whereas the results so far in this section have dealt with conditions of the form $\mathrm{K}_{\mathrm{L}}(n) = \omega(r(n))$, the results which follow consider the condition $\mathrm{K}_{\mathrm{L}}(n) = O(r(n))$.

THEOREM 24. *Let* $r(2^n)$ *be a time-constructible function. If, for all* $\mathrm{L} \in \mathrm{P}$, $\mathrm{K}_{\mathrm{L}}(n) = O(r(n)$, *then* $\mathrm{NE} \subseteq \mathrm{DTIME}(2^{r(2^{O(n)})})$.

*Proof.* Let $A \in NE$ be accepted by a NE machine $M$ which runs in time $2^{cn}$, and let $L \in P$ be the set $\{y\colon |y| = m^c$ and $y$ encodes an accepting computation of $M$ on input $m\}$. (Recall that we make no distinction between a number and the string which encodes it.) By assumption, there exists some $k$ such that, for all large $m$, $K_L(m^c) < kr(m^c)$ if $L^{=m^c} \neq \emptyset$. Thus the following deterministic algorithm runs in time $2^{r(2^{O(n)})}$ and decides membership in A; on input $m$ of length $n$, search through the set of all strings $y$ such that $\mathrm{Kt}(y) < kr(m^c)$. If some string $y \in L$ such that $|y| = m^c$ is found, halt and accept; else halt and reject. ▌

COROLLARY 25. *If* $E \neq NE$, *then there is some* $L \in P$ *such that* $K_L(n) \neq O(\log n)$.

COROLLARY 26. *If* $NE \not\subseteq \bigcap_{\varepsilon > 0} \mathrm{DTIME}(2^{2^{\varepsilon n}})$, *then there is some* $L \in P$ *and some* $\varepsilon > 0$ *such that* $K_L(n) \neq O(n^{\varepsilon})$.

Some discussion is called for, comparing these results to the results in the previous section, particularly Corollaries 9 and 10. Since it is often conjectured that nondeterministic computations cannot be simulated deterministically significantly faster than by the naïve search strategy, it follows by Theorem 24 that it is often conjectured that there is no function $r(n) = o(n)$ such that for all $L \in P$, $K_L(n) = O(r(n))$. On the other hand, if secure pseudorandom generators exist, then Corollaries 9 and 19 say that sufficiently *dense* sets L in P (or even P/poly) must have $K_L(n)$ be quite small for many $n$.

The preceding results allow us to draw certain conclusions from assumptions about the growth of functions of the form $K_L(n)$ for sets L in P. Unfortunately, however, some of the results in Section 5 give information only about functions of the form $K_L(n^k)$ or $K_L(2^{\varepsilon n})$. The next few results deal with functions of the sort.

THEOREM 27. *If* $\mathrm{NTIME}(2^{O(n^2)}) \not\subseteq \bigcap_{\varepsilon > 0} \mathrm{DTIME}(2^{2^{\varepsilon n^2}})$ *then there is some* $L \in \mathrm{DTIME}(O(n))$ *and some* $k \in \mathbf{N}$ *such that* $K_L(n^k) \neq O(n)$.

*Proof.* Let $A \in \mathrm{NTIME}(2^{cn^2}) - \mathrm{DTIME}(2^{2^{\varepsilon n^2}})$, and let $M$ be a nondeterministic Turing machine accepting A in the given time bound. Let $L = \{y\colon \exists x, x^{c \log x} \leqslant |y| \leqslant (x+1)^{c(\log x + 1)}$ and some prefix of $y$ encodes an accepting computation of M on $x\}$. Clearly, $L \in \mathrm{DTIME}(O(n))$. Note that for all $x \in A$, there is a string encoding an accepting computation of M on $x$ in L, since $x^{c \log x} \leqslant 2^{c|x|^2}$. We must show that for some sufficiently large value of $k$, $K_L(n^k) \neq O(n)$.

Assume that $K_L(n^k) < bn$ for all large values of $n$ at which $K_L(n^k)$ is defined. Then consider the following routine for accepting A:

```
Begin
    On input $x$ of length $n$,
    let $m$ be the least such that $x^{c \log x} \leqslant m^k < (x+1)^{c(\log x + 1)}$
    Search through all strings $y$ such that $K_L(y) \leqslant bm$.
    If some string $y \in L$ of length $m^k$ is found, halt and accept, else reject.
End
```

The time required to run this routine is $2^{2bm} \leqslant 2^{2b((x+1)^{c\log x+1})^{1/k}}$ which is less than $2^{2^{\varepsilon\lfloor |x| \rfloor^2}} = 2^{2^{\varepsilon n^2}}$, assuming $k$ was chosen sufficiently large. ▮

We remark that the time bound $2^{O(n^2)}$ was chosen only in order to simplify the exposition. Similar results can be proved for NTIME($2^{O(s(n))}$) for any sufficiently "nice" superlinear function $s(n)$.

Corollary 28. *If* RTIME($2^{O(n^2)}$) $\not\subseteq \bigcap_{\varepsilon>0}$ DTIME($2^{2^{\varepsilon n^2}}$) *then there is some a.e. dense set* $L \in$ DTIME($O(n)$) *and some* $k \in \mathbf{N}$ *such that* $K_L(n^k) \neq O(n)$.

Theorem 29. *If* NTIME($2^{O(2^n)}$) $\neq$ DTIME($2^{O(2^n)}$), *then there is a set* L *in* DTIME($O(n)$) *such that for all* $\varepsilon > 0$, $K_L(2^{\varepsilon n}) \neq O(n)$.

*Proof.* The proof of this result is quite similar to the proof of Theorem 27. Let $A \in$ NTIME($2^{c2^n}$) $-$ DTIME($2^{O(2^n)}$), and let $M$ be a nondeterministic Turing machine accepting A in the given time bound. Let $L = \{y\colon \exists x 2^{cx} \leqslant |y| \leqslant 2^{c(x+1)}$ and some prefix of $y$ encodes an accepting computation of M on $x\}$. Clearly, $L \in$ DTIME($O(n)$). We must show that for all $\varepsilon > 0$, $K_L(2^{\varepsilon n}) \neq O(n)$.

Assume that $K_L(2^{\varepsilon n}) < bn$ for all large values of $n$ at which $K_L(2^{\varepsilon n})$ is defined. Then consider the following routine for accepting A:

```
Begin
    On input $x$ of length $n$,
    let $m$ be the least such that $2^{cx} \leqslant 2^{\varepsilon m} < 2^{c(x+1)}$
    Search through all strings $y$ such that $K_L(y) \leqslant bm$.
    If some string $y \in L$ of length $2^{\varepsilon m}$ is found, halt and accept, else reject.
End
```

The time required to run this routine is $2^{2bm} \leqslant 2^{2b(c(x+1)/\varepsilon)} = 2^{O(2^n)}$. ▮

Corollary 30. *If* RTIME($2^{O(2^n)}$) $\neq$ DTIME($2^{O(2^n)}$), *then there is an a.e. dense set* L *in* DTIME($O(n)$) *such that for all* $\varepsilon > 0$, $K_L(2^{\varepsilon n}) \neq O(n)$.

At this point, we are finally in a position to relate assumptions about the existence of secure pseudorandom generators to consequences concerning probabilistic complexity classes.

Corollary 31. 1. Hypothesis 1 $\Rightarrow$ BPTIME($2^{O(n^2)}$) $\subseteq \bigcap_{\varepsilon>0}$ DTIME($2^{2^{\varepsilon n^2}}$).

2. Hypothesis 2 $\Rightarrow$ BPP $\subseteq \bigcap_{\varepsilon>0}$ DTIME($2^{n^{\varepsilon}}$).

3. Hypothesis $3 \Rightarrow \mathrm{RTIME}(2^{O(2^n)}) = \mathrm{DTIME}(2^{O(2^n)})$.
4. Hypothesis $4 \Rightarrow \mathrm{RP} = \mathrm{P}$.

*Proof.* Part 4 follows from Corollaries 10 and 23. Part 3 follows from Corollaries 7 and 30. Part 2 is from [12]. It follows from Corollaries 6 and 28 that Hypothesis $1 \Rightarrow \mathrm{RTIME}(2^{O(n^2)}) \subseteq \bigcap_{\varepsilon > 0} \mathrm{DTIME}(2^{2^{\varepsilon n^2}})$, but the stronger claim of part 1 remains to be proved.

By standard translational methods, it suffices to show that if Hypothesis 1 is true, then for all $\varepsilon > 0$, every tally set in $\mathrm{BPTIME}(n^{O(\log n)})$ is in $\mathrm{DTIME}(2^{n^{\varepsilon \log n}})$.

Let $\varepsilon > 0$, and choose $k$ so that $1/k < \varepsilon$. Let $g$ be any secure generator which takes inputs of length $n$ and produces outputs of length $n^k$. Let $L$ be a tally set accepted by some $\mathrm{BPTIME}(b^{c \log n})$ machine M. The following deterministic algorithm determines membership in L:

```
Begin
    On input $0^n$
    find $m$ such that $n^{c \log n} \leqslant m^k < (n+1)^{c \log(n+1)}$.
    for all strings $x$ length $m$
        compute $g(x)$
        record if $g(x)$ is a computation causing M to accept $0^n$.
    If at least half of the strings $g(x)$ cause M to accept, halt and accept.
    Else, halt and reject.
End
```

The running time of this algorithm is $2^m n^{O(1)} \leqslant 2^{(n+1)^{(c \log(n+1))/k}} \leqslant 2^{2^{\varepsilon \log n}}$. If the algorithm is not correct, assume without loss of generality that it rejects infinitely many strings in L. That is, it must be the case that there are infinitely many strings $0^n$ such that a random sequence of coin flips of length $n^{c \log n}$ causes M to accept, but fewer than half of the pseudorandom strings of that length cause M to accept. That is, $g$ fails the following probabilistic polynomial-time statistical test $A$:

```
Begin
    On input $y$ such that $|y| = m^k$ for some $m$,
    let $n$ be the greatest integer such that $n^{c \log n} \leqslant m^k$.
    Accept iff $y$ is an accepting computation of M on $0^n$.
End
```

By the comments above, for infinitely many $m$, $P_{A, m^k}(R) > \frac{3}{4}$ and $P_{A, m^k}(\mathrm{PS}g) < \frac{1}{2}$. Thus $g$ is not polynomially secure, contrary to our choice of $g$. ∎

## 7. Immunity Properties of RP

In this section we show that, under the assumption that secure pseudorandom generators exist, sufficiently dense sets in RP satisfy certain immunity properties.

Corollary 32. Hypothesis 3 $\Rightarrow$ *every dense set in* RP *has an infinite* P-*printable subset.*

*Proof.* If A is a dense set in RTIME($n^k$), then the set $\text{L} = \{\langle x, y \rangle : y$ is a sequence of coin flips of length $n^k$ witnessing that $x \in A$, and $|y| = |x|^k\}$ is a dense set in DTIME($O(n)$). By Corollary 14, Hypothesis 3 implies that $L$ has an infinite P-printable subset, S. Clearly, the set $\{y : \langle x, y \rangle \in S\}$ is an infinite P-printable subset of A. ▮

Thus, for example, Hypothesis 3 implies that there is an infinite P-printable set of primes.

Hypothesis 3 is, of course, a very strong assumption. It turns out that a somewhat similar immunity property holds for RP, using the more commonly accepted Hypothesis 2.

Theorem 33. Hypothesis 2 $\Rightarrow$ *if* L *is a dense set in* RP, *then* L × L *has an infinite subset in* P.

*Proof.* Assume Hypothesis 2 holds, and let L be a set in RP such that for some $\varepsilon < 1$, $d_{\text{L}}(n) > 2^{n^\varepsilon}$ i.o. (Note that we are assuming much less than density here; the claim holds even for sets of such "moderate" density.)

As Yao has pointed out [39], if Hypothesis 2 holds, then for all $k$ there is a probabilistic polynomial-time machine $\text{M}_k$ accepting L which, on input $x \in \text{L}$ of length $n$, flips only $n^{1/k}$ coins and accepts with probability $> \frac{1}{2}$. (This machine M flips $n^{1/k}$ coins, and then applies a secure pseudorandom generator, and then uses the pseudorandom sequence to continue the computation.) In the same way, given any $\delta < 1$, one can construct a probabilistic polynomial-time machine $\text{M}_\delta$ accepting L which, on input $x \in L$ of length $n$, flips $n$ coins and accepts with probability $> 1 - 2^{-n^\delta}$. (This is accomplished by choosing $k$ large enough, and simulating $\text{M}_k$ on $n^{1-(1/k)}$ independent trials.)

Let $\varepsilon < \delta < 1$, let $Q(x, y)$ be the predicate which is true if $y$ is a sequence of coin flips causing $\text{M}_\delta$ to accept $x$, and let $S = \{\langle x, y \rangle : |x| = |y| \wedge Q(x, y) \wedge Q(y, x)\}$. Clearly, S is in P and S is a subset of L × L. It suffices to show that S is infinite; we show that for all $n$ such that $d_{\text{L}}(n) > 2^{-n^\varepsilon}$, there are strings $x$ and $y$ of length $n$ such that $\langle x, y \rangle \in \text{S}$.

Consider the square matrix with one row and one column for each element of L of length $n$, where position $\langle x, y \rangle$ is filled with a 1 or a 0 according to whether or not $Q(x, y)$ is true. Since $\text{M}_\delta$ accepts each $x$ in L with probability $> 1 - 2^{-n^\delta}$, there are at most $(2^{-n^{\delta}})\, 2^n$ strings $y$ of length $n$ such that $Q(x, y)$ is false, and thus each row in the matrix has at least

$$d_{\mathrm{L}}(n)\, 2^n - \frac{2^n}{2^{n^{\delta}}} > d_{\mathrm{L}}(n)\, 2^n - \frac{1}{2}\frac{2^n}{2^{n^{\epsilon}}} > \frac{1}{2}\, d_{\mathrm{L}}(n)\, 2^n$$

1's in it. That is, more than half of the entries in the matrix are filled with 1's. Thus there must be at least one pair $\langle x, y \rangle$ such that positions $\langle x, y \rangle$ and $\langle y, x \rangle$ are both filled with 1's. (In fact, there must be many such pairs) Thus $\langle x, y \rangle \in \mathrm{S}$. ▌

Note that, for all sets L, L × L has an infinite P-printable subset iff L does. Also, L × L has an infinite r.e. or recursive subset iff L does. Furthermore, it is easy to show that, if P = NP, then L × L has an infinite subset of P iff L does. However, there are oracles relative to which there are sets L such that L has no infinite subset in P, but L × L does. (*Sketch of proof.* Let A be an oracle consisting of one Kolmogorov-random string for each of a very sparse set of lengths, so that $(x \in A$, $y \in A$, and $x < y) \Rightarrow 2^{|x|} < |y|$. Let L be the set $\{w \colon \exists v\, |w| = |v|$ and $wv \in A$ or $vw \in A\}$. Clearly, L × L has an infinite subset in $\mathrm{P}^{\mathrm{A}}$. However, if S is a subset of L in $\mathrm{P}^{\mathrm{A}}$, accepted by a polynomial-time oracle Turing machine M, then there is a Turing machine which, given $n$, the elements of A of length $\leqslant \log 2n$, and a description of $M$, along with at most a constant number of extra bits of information, will print any given string in S. Thus if S is infinite, it contains strings of low Kolmogorov complexity, and thus is not a subset of L.) Thus, knowing something about the immunity properties of sets of the form L × L in RP seems to tell us little about the immunity properties of arbitrary sets in RP.

If it could be shown that Hypothesis 2 ⇒ that every dense set in RP has an infinite subset in P, it would follow that Hypothesis 2 implies $\mathrm{DTIME}(t(n)) \nsubseteq \mathrm{RP}$, for any time-constructible superpolynomial function $t(n)$, since for any such $t(n)$ there is a set in $\mathrm{DTIME}(t(n))$ which is bi-immune with respect to P.

Although the existence of sets L such that L × L is P-immune can be shown using standard diagonalization techniques, these techniques seem to require slightly more than exponential time. That is, it does not seem to be currently known whether or not there are sets L in $\mathrm{DTIME}(2^n)$ such that L × L is P-immune, although such sets can be shown to exist in $\mathrm{DTIME}(2^{O(n)})$.

## 8. Conclusions

A new measure of the complexity of a language L has been introduced in this paper. For any set L, $\mathrm{K}_{\mathrm{L}}$ is a function which measures, for each $n$, the time-bounded Kolmogorov complexity of the simplest strings in L of length $n$. We have shown that if secure pseudorandom generators exist, then $\mathrm{K}_{\mathrm{L}}$ grows slowly for all dense sets L in P or P/poly, and if hard sets exist in NE, then $\mathrm{K}_{\mathrm{L}}$ grows quickly for some sets L in P. If RP $\neq$ P, then $K_L$ grows quickly for some dense sets L in P/poly.

Using the functions $K_L$ as a tool, we have shown that different hypotheses about the existence of secure pseudorandom generators imply the existence of fast deterministic simulations of probabilistic algorithms, although the nature of the sort of speedup which can be proved varies according to the sort of security which is assumed.

In particular, we were able to show:

1. Hypothesis 1 $\Rightarrow$ $\mathrm{BPTIME}(2^{O(n^2)}) \subseteq \bigcap_{\varepsilon > 0} \mathrm{DTIME}(2^{2^{\varepsilon n^2}})$.
2. Hypothesis 2 $\Rightarrow$ $\mathrm{BPP} \subseteq \bigcap_{\varepsilon > 0} \mathrm{DTIME}(2^{n^\varepsilon})$.
3. Hypothesis 3 $\Rightarrow$ $\mathrm{RTIME}(2^{O(2^n)}) = \mathrm{DTIME}(2^{O(2^n)})$.
4. Hypothesis 4 $\Rightarrow$ RP = P.

This paper considered some extremely strong hypotheses about the security of pseudorandom generators. In particular, Hypothesis 4 is so strong that it might be possible to prove that it is false. One possible approach to this problem is outlined here: For any $k$ and $x$, let $E_k(x) = \{y: \mathrm{Kt}(x \mid y) \leqslant k \log n\}$; intuitively, $E_k(x)$ is the set of strings $y$ relative to which $x$ is easy. There seems to be some relationship between the Kt complexity of $x$ and the size of $E_k(x) \cap \Sigma^{|x|}$, since it is easy to see that $\Sigma^n \subseteq E_k(0^n)$ for large enough $k$, whereas for most strings $x$ of length $n$ (and hence for all "random" strings) there are fewer than $n^k$ strings of length $n$ in $E_k(x)$. If there exists an infinite sequence of strings $x$ such that $\mathrm{Kt}(x)$ grows faster than $\log x$, but $E_k(x)$ is somewhat large, then Hypotheses 4 is false. More formally:

THEOREM 34. *If* $\exists k, \forall l, \exists x, \mathrm{Kt}(x) > l \log n$ *and* $|E_k(x) \cap \Sigma^{|x|}| > 2^{|x|}/|x|^k$, *then Hypothesis* 4 *is false.*

*Proof.* Assume the hypothesis of the lemma is true, and for $l = 1, 2, \ldots$ let $x_l$ be the string whose existence is guaranteed for each $l$. Assume without loss of generality that $i < j \Rightarrow |x_i| < |x_j|$. Note that the set $L = \{y: \exists x, |x| = |y| \wedge y \in E_k(x)\}$ is in P/poly. By assumption, it is dense. Also, it is clear that L has no infinite P-printable subset, since if S is a P-printable set, then there is some $l$ such that, for all $x \in S$, $\mathrm{Kt}(x) < l \log |x|$. By Corollary 10, Hypothesis 4 is false. ∎

## ACKNOWLEDGMENTS

Stimulating discussions with Ron Book, Ker-I Ko, Dave Russo, Osamu Watanabe, Pekka Orponen, and Uwe Schöning led me to consider some of the questions addressed by this work. Thanks are especially due to Ron Book for organizing the informal workshop at Santa Barbara in the spring of 1986, where those discussions took place. Thanks are also due to Joel Seiferas and Gautam Das, for valuable comments, and to Mike Sipser, for making [12] available to me.

## REFERENCES

1. L. ADLEMAN, "Time, space, and randomness," Technical Report MIT/MCS/TM 131, 1979.
2. L. ADLEMAN AND M.-D. HUANG, Recognizing primes in random polynomial time, *in* "Proceedings, 19th Annual ACM Symposium on Theory of Computing, 1987," pp. 462–469.
3. E. ALLENDER AND R. RUBINSTEIN, P-printable sets, *SIAM J. Comput.* **17** (1988), 1193–1202.
4. E. ALLENDER, Some consequences of the existence of pseudorandom generators, preliminary version, *in* "Proceedings, 19th Annual ACM Symposium on Theory of Computing, 1987," pp. 151–159.
5. E. ALLENDER, in preparation.
6. M. AJTAI AND A. WIGDERSON, Deterministic simulation of probabilistic constant depth circuits, *in* "Proceedings, 26th IEEE Symposium on Foundations of Computer Science, 1985," pp. 11–19.
7. J. BALCÁZAR AND R. BOOK, Sets with small generalized Kolmogorov complexity, *Acta Inform.* **23** (1986), 679–688.
8. J. BALCÁZAR AND U. SCHÖNING, Bi-immune sets for complexity classes, *Math. Systems Theory* **18** (1981), 1–10.
9. C. BENNETT AND J. GILL, Relative to a random oracle, $P(A) \neq NP(A) \neq \text{Co-NP}(A)$ with probability 1, *SIAM J. Comput.* **10** (1981), 96–113.
10. L. BERMAN AND J. HARTMANIS, On isomorphisms and density of NP and other complete sets, *SIAM J. Comput.* **6** (1977), 305–323.
11. M. BLUM AND S. MICALI, How to generate cryptographically strong sequences of pseudo-random bits, *SIAM J. Comput.* **13** (1984), 850–864.
12. R. BOPPANA AND R. HIRSCHFELD, Pseudorandom generators and complexity classes, *in* "Advances in Computing Research, Vol. 5, Randomness and Computation" (S. Micali, Ed.) JAI, Greenwich, CT, 1988.
13. G. CHAITIN, On the length of programs for computing finite binary sequences, *J. Assoc. Comput. Mach.* **13** (1966), 547–569.
14. G. CHAITIN, A theory of program size formally identical to information theory, *J. Assoc. Comput. Mach.* **22**, 329–340.
15. J. GESKE, D. HUYNH, AND A. SELMAN, A hierarchy theorem for almost everywhere complex sets with application to polynomial complexity degrees, *in* "Proceedings, 4th Annual Symposium on Theoretical Aspects of Computer Science," Lecture Notes in Computer Science Vol. 247, pp. 125–135, Springer-Verlag, Berlin/New York, 1987.
16. S. GOLDWASSER AND J. KILLIAN, Almost all primes can be quickly certified, *in* "Proceedings, 18th Annuel ACM Symposium on Theory of Computing, 1986," pp. 316–329.
17. S. GOLDWASSER AND S. MICALI, Probabilistic encryption, *J. Comput. System Sci.* **28** (1984), 270–299.
18. J. HARTMANIS, Generalized Kolmogorov complexity and the structure of feasible computations, *in* "Proceedings, 24th IEEE Symposium on Foundations of Computing, 1983," pp. 439–445.
19. J. HARTMANIS AND Y. YESHA, Computation times of NP sets of different densities, *Theoret. Comput. Sci.* **34** (1984), 17–32.
20. L. HEMACHANDRA, Can P and NP manufacture randomness? manuscript, 1986.
21. J. E. HOPCROFT AND J. D. ULLMAN, "Introduction to Automata Theory, Languages, and Computation," Addison–Wesley, Reading, MA, 1979.
22. K.-I. KO, On the notion of infinite pseudorandom sequences, *Theoret. Comput. Sci.* **48** (1986), 9–33.
23. K.-I. KO, P. ORPONEN, U. SCHÖNING, AND O. WATANABE, What is a hard instance of a computational problem? *in* "Proceedings, Structure in Complexity Theory Conference," Lecture Notes in Computer Science Vol. 223, pp. 197–217, Springer-Verlag, Berlin/New York, 1986.
24. A. KOLMOGOROV, Three approaches to the quantitative definition of randomness, *Problems Inform. Transmission* **1** (1965), 1–7.
25. L. LEVIN, Universal sequential search problems, *Problems Inform. Transmission* **9** (1973), 265–266.
26. L. LEVIN, Randomness conservation inequalities; Information and independence in mathematical theories, *Inform. and Control* **61** (1984), 15–37.

27. L. Levin, One-way functions and pseudorandom generators, *in* "Proceedings, 17th Annual ACM Symposium on Theory of Computing, 1985," pp. 363–365.
28. M. Li and Y. Yesha, New lower bounds for parallel computation, *in* "Proceedings, 18th Annual ACM Symposium on Theory of Computing, 1986," pp. 177–187.
29. L. Longpré, "Resource Bounded Kolmogorov Complexity, a Link between Computational Complexity and Information Theory," Doctoral dissertation, Cornell University, 1986.
30. G. Peterson, Succinct representations, random strings and complexity classes, *in* "Proceedings, 21st IEEE Symposium on Foundations of Computer Science, 1980," pp. 86–95.
31. J. Pintz, W. Steiger, and E. Szemerédi, Two infinite sets of primes with fast primality tests, *in* "Proceedings, 20th Annual ACM Symposium on Theory of Computing, 1988," pp. 504–509.
32. J. Reif and J. Tygar, "Efficient Parallel Pseudo-random Number Generation," Tech. Report TR-07-84, Harvard University, 1984.
33. U. Schöning, "Complexity and Structure," Lecture Notes in Computer Science Vol. 211, Springer-Verlag, New York/Berlin, 1986.
34. J. Seiferas, M. Fischer, and A. Meyer, Separating nondeterministic time complexity classes, *J. Assoc Comput. Mach.* **25** (1978), 146–167.
35. A. Shamir, On the generation of cryptographically strong pseudorandom sequences, *ACM Trans. Comput. Systems* **1**, 38–44.
36. M. Sipser, A complexity theoretic approach to randomness, *in* "Proceedings 15th Annual ACM Symposium on Theory of Computing, 1983," pp. 330–335.
37. O. Watanabe, Generalized Kolmogorov complexity of computations, manuscript, 1986.
38. R. Wilber, Randomness and the density of hard problems, *in* "Proceedings, 24th IEEE Symposium on Foundations of Computer Science, 1983," pp. 335–342.
39. A. Yao, Theory and applications of trapdoor functions, *in* "Proceedings, 23rd IEEE Symposium on Foundations of Computer Science, 1982," pp. 80–91.

Printed by Catherine Press, Ltd., Tempelhof 41, B-8000 Brugge, Belgium