Power from Random Strings

Eric Allender* Harry Buhrman† Michal Koucký‡
Dieter van Melkebeek§ Detlef Ronneburger¶

May 14, 2002

Abstract

We consider sets of strings with high Kolmogorov complexity, mainly in resource-bounded settings but also in the traditional recursion-theoretic sense. We present efficient reductions, showing that these sets are hard and complete for various complexity classes.

In particular, in addition to the usual Kolmogorov complexity measure K, we consider the time-bounded Kolmogorov complexity measure KT that was introduced in [All01], as well as a space-bounded measure KS, and Levin's time-bounded Kolmogorov complexity Kt [Lev84]. Let $R_{\mathrm{K}}, R_{\mathrm{KT}}, R_{\mathrm{KS}}, R_{\mathrm{Kt}}$ be the sets of strings x having complexity at least $|x|/2$, according to each of these measures. Our main results are:

- R_{KS} and R_{Kt} are complete for PSPACE and EXP, respectively, under P/poly-truth-table reductions.
- $\mathrm{EXP} = \mathrm{NP}^{R_{\mathrm{Kt}}}$.
- $\mathrm{PSPACE} = \mathrm{ZPP}^{R_{\mathrm{KS}}} \subseteq \mathrm{P}^{R_{\mathrm{K}}}$.
- The Discrete Log is in $\mathrm{BPP}^{R_{\mathrm{KT}}}$.

Our hardness results for EXP and PSPACE rely on nonrelativizing proof techniques.

Our techniques also allow us to show that all recursively-enumerable sets are reducible to R_{K} via P/poly-truth-table reductions.

Our hardness result for PSPACE gives rise to fairly natural problems that are complete for PSPACE under $\leq^{\mathrm{p}}_{\mathrm{T}}$ reductions, but not under $\leq^{\log}_{\mathrm{m}}$ reductions.

In spite of the EXP- and PSPACE-completeness of R_{Kt} and R_{KS}, it remains unknown if either of these problems is in logspace.

1 Introduction

Much recent work in derandomization can be viewed as an attempt to understand and exploit the interplay between the two common meanings of the phrase "random string": a string picked at random (according to some distribution), and a string with high Kolmogorov complexity (in some sense). In this paper, we further investigate the relationship between these two notions. We apply recent advances in derandomization to obtain fundamentally new types of complete sets for several standard complexity classes. The sets consist of random strings with respect to various Kolmogorov measures.

*Rutgers University, allender@cs.rutgers.edu
†CWI and University of Amsterdam, buhrman@cwi.nl
‡Rutgers University, mkoucky@paul.rutgers.edu
§University of Wisconsin, dieter@cs.wisc.edu
¶Rutgers University, detlef@paul.rutgers.edu

We will focus on the set $R_{\mathrm{K}} = \{x \ : \ \mathrm{K}(x) > |x|/2\}$ of strings with high traditional Kolmogorov complexity, as well as various resource-bounded variants R_μ for $\mu = \mathrm{KT}, \mathrm{KS}, \mathrm{Kt}$. See Section 2 for the definitions of KT, KS, and Kt. The choice of $|x|/2$ as a quantification of "high complexity" is rather arbitrary. Our results hold for any reasonable bound ranging from $|x|^\epsilon$ to $\epsilon|x|$, $0 < \epsilon < 1$.

The sets R_μ of Kolmogorov random strings are good examples of sets with a lot of information content that is difficult to access. Time- and space-bounded versions of Kolmogorov random strings have been studied as possible examples of sets that are intractable without being complete for any of the standard complexity classes. For instance, Buhrman and Mayordomo [BM97] studied the time t-incompressible strings (for $t(n) = 2^{n^2}$) and showed that this set is in $\mathrm{EXP} - \mathrm{P}$, but is *not* complete under polynomial-time Turing reducibility.

As yet another example of this phenomenon, Cai and Kabanets [KC00] studied the Minimum Circuit Size Problem (MCSP), which (as observed in [All01]) is closely related to KT complexity. They present evidence that MCSP is not in P, but is also not likely to be NP-complete under $\leq^{\mathrm{p}}_{\mathrm{m}}$ reductions. Also, Ko [Ko91] showed for a variant of this set that there are relativized worlds where it is neither in P nor coNP-complete with respect to polynomial-time Turing reductions.

When no resource bounds are present, the set of Kolmogorov random strings R_{K} is easily seen to be co-r.e. and not decidable, but the complement of the halting problem is *not* reducible to R_{K} via a many-one reduction. It was shown only recently by Kummer that a truth-table reduction is sufficient [Kum96], although it had long been known [Mar66] that Turing reductions can be used. It should be emphasized that Kummer's reduction is *not* feasible and asks *many* queries. Indeed, it is easy to see that any long string that is used as a query in a polynomial-time truth-table reduction must have small Kolmogorov complexity, and thus one might conjecture that reductions to the set R_{K} must involve exponentially-many queries.

These results suggest that the sets of resource-bounded random strings are not complete for the complexity class they naturally live in. Buhrman and Torenvliet [BT01] gave some evidence that this is not the complete picture. They showed that for the *conditional* version of space bounded Kolmogorov complexity the set of random strings is hard for PSPACE under NP reductions. However, their result had the major drawback that it needed conditional Kolmogorov complexity, used NP reductions, and, moreover, their proof technique could not be used beyond PSPACE.

Using very different techniques, we provide much stronger results in the same direction: the set of random strings *can* be exploited by efficient reductions. For instance, we show that the set R_{Kt} of strings with high complexity using Levin's time-bounded Kolmogorov notion Kt [Lev84] is complete for EXP under truth-table reductions computable by polynomial-size circuits. Thus we obtain natural examples that witness the difference in power of various reducibilities.

In some instances, we are also able to provide completeness results under uniform reductions. By making use of multiple-prover interactive proofs for EXP ([BFL91]) we show that R_{Kt} is complete for EXP under NP-Turing reductions.

Of greater interest is the fact that the set R_{KS} of strings with high space-bounded Kolmogorov complexity is complete for PSPACE under ZPP-Turing reductions. Our proofs rely on the existence of complete sets for PSPACE that are both downward-self-reducible and random-self-reducible [TV02], and hence our proofs do not relativize. It remains unknown if the results themselves hold relative to all oracles.

For the unbounded case we even show that PSPACE is reducible to R_{K} under *uniform* polynomial time reductions. The main tool here in addition to the previous results, is the fact that we can construct, in polynomial time, a string of high Kolmogorov complexity using R_{K} as an oracle.

1.1 The connection to derandomization

There is an obvious connection between Kolmogorov complexity and derandomization. For any randomized algorithm $\mathcal{A}$ with a small probability of error and any input x, the coin flip sequences r that result in a wrong answer are atypical, and therefore have short descriptions of some kind.

By considering different notions of Kolmogorov complexity, less-obvious (and more-useful) connections can be exposed. Assume that the coin flip sequences that result in wrong answers have small μ-complexity, for some some Kolmogorov measure μ. If we can generate a coin flip sequence r that belongs to the set R_μ of strings with high μ-complexity, then we obtain an upper bound on the complexity of $\mathcal{A}$, by running the randomized algorithm on (x, r). If we can at least check membership in R_μ, we can reduce the error probability to zero by picking a coin flip sequence r uniformly at random until we get one in R_μ (of which there are many), and then running $\mathcal{A}$ on (x, r). The first interesting application of this approach is due to Sipser [Sip83]. His proof that BPP lies in the polynomial-time hierarchy uses $\mu = \mathrm{KD}^{poly}$, the polynomial-time bounded distinguishing complexity. The corresponding set R_μ lies in coNP. The hardness versus randomness tradeoffs by Babai et al. [BFNW93] and by Impagliazzo and Wigderson [IW97] can be cast as an application with $\mu = \mathrm{KT}$, a time-bounded Kolmogorov measure introduced in [All01], which essentially measures the circuit complexity of the Boolean function defined by the string described. The observation from [KvM99] that the construction of [IW97] relativizes with respect to any oracle A can be viewed in terms of a Kolmogorov measure which we denote as KT^A. This interpretation plays a crucial role in Trevisan's recent construction of extractors out of pseudo-random generators [Tre01]. Other connections are surveyed in [All01].

Our main technique is to use relativizing hardness versus randomness tradeoffs in the contrapositive. Such results state that if there exists a computational problem in a certain complexity class $\mathcal{C}$ that is hard when given oracle access to A, then there exists a pseudo-random generator secure against A that is computable within $\mathcal{C}$. However, we argue that no pseudo-random generator computable in $\mathcal{C}$ can be secure against R_μ. Thus we conclude that every problem in $\mathcal{C}$ is easy given oracle access to R_μ, i.e., $\mathcal{C}$ reduces to R_μ. For our results, we exploit the nonuniform hardness versus randomness tradeoffs in [BFNW93] and [IW97], as well as the uniform ones in [HILL99] and [IW98].

1.2 The structure of complexity classes

The tools of reducibility and completeness (in particular NP-completeness) are responsible for most of the success that complexity theory has had in proving (or providing evidence for) intractability of various problems. Although it has been known since the work of Ladner [Lad75] that, if P is not equal to NP, then there are intractable problems in NP that are not NP-complete, there are not many interesting candidates for this status. Certainly the sets constructed in [Lad75] are quite artificial (constructed by putting huge empty segments inside a standard complete problem such as SAT). Similarly, although there are a great many notions of reducibility that have been considered, and for many of these notions it is known that more powerful reducibilities provide more complete sets [Wat87, AAI+02] (at least for large complexity classes such as EXP), almost all of the known constructions proceed by diagonalization and do not produce very "natural" languages.

There are two notable examples that run counter to this trend, and that are relevant to the research we report on here.

1. Buhrman and Mayordomo [BM97] study time-bounded Kolmogorov complexity K^t for

time bounds $t(n) \geq 2^{n^2}$, and show that the set $R^t = \{x : \mathrm{K}^t(x) \geq |x|\}$ lies in $\mathrm{EXP} - \mathrm{P}$ and is not complete for EXP under polynomial-time Turing reducibility $\leq_{\mathrm{T}}^{\mathrm{p}}$.

2. Kabanets and Cai [KC00] study the set MCSP defined as $\{(f, s) : f$ is a string of length 2^n interpreted as the truth-table of a Boolean function, and f is computed by a Boolean circuit with at most s gates$\}$. Kabanets and Cai show that if MCSP is in P/poly, then there are no secure pseudorandom generators, thus providing evidence of intractability. On the other hand, they show that some "natural" approaches that one might use to show that MCSP is NP-complete would have some dramatic consequences if they were to succeed (such as proving that $\mathrm{EXP} \not\subseteq \mathrm{P/poly}$ and $\mathrm{BPP} \neq \mathrm{EXP}$).

Although these examples seem to have little to do with each other, we present a framework that shows both of these problems are variations on a single theme. Also, the techniques of resource-bounded Kolmogorov complexity provide us with natural examples of computational problems witnessing that certain well-studied notions of reducibility yield different classes of PSPACE-complete sets.

1.3 Outline of the rest of the paper

In Section 2 we present background and definitions regarding resource-bounded Kolmorogov complexity. In Section 3 we present the statement of our main results, although most of the proofs are postponed until Sections 4 and 5. We conclude with open problems in Section 6.

2 Resource-Bounded Kolmogorov Complexity

We assume that the reader is familiar with Kolmogorov complexity: $\mathrm{K}(x) = \min\{|d| : U(d) = x\}$ (for some fixed universal multitape Turing machine U). For background, consult [LV93] (but note that the function $\mathrm{K}(x)$ is called $C(x)$ there).

Several notions of time-bounded Kolmogorov complexity have been studied. One of the most useful was introduced by Levin [Lev84].

Definition 1 (Levin) *Let U be a universal Turing machine. Define* $\mathrm{Kt}(x)$ *to be* $\min\{|d| + \log t : U(d) = x$ *in at most t steps*$\}$.

The elements of $\{0,1\}^*$ can be enumerated in order of increasing $\mathrm{Kt}(x)$, and Levin observed that this ordering yields essentially the fastest way to search for accepting computations of nondeterministic Turing machines.

Another notion of time-bounded Kolmogorov complexity was introduced in [All01], where it was shown to be related to the "easy witness" method of Impagliazzo, Kabanets, and Wigderson [Kab01, IKW01], and the "natural proofs" framework of Razborov and Rudich [RR97].

Definition 2 *Let U be a universal Turing machine. Define* $\mathrm{KT}(x) = \min\{|d| + t :$ *for all $i \leq |x|$, $U(d, i) = x_i$, and $U(d, |x| + 1) = *$, in at most t steps*$\}$.

The significant difference between the KT and Kt measures lies in the exponential difference in the weight that is given to the time bound in the two measures; the fact that the description d in KT describes the string x bit-wise is necessary in order to allow sublinear running times. The properties of the measure Kt would be essentially unchanged if the bit-wise convention were employed in that definition as well. Note that $\log |x| \leq \mathrm{KT}(x) \leq |x| + O(\log |x|)$.

Let us further define a space-bounded analog of KT complexity:

Definition 3 *Let U be a universal Turing machine. Define* $\mathrm{KS}(x) = \min\{|d| + s : U(d) = x$ *in space at most* $s\}$.

Observe that the Kt, KT and KS measures can be generalized to obtain Kt^A, KT^A and KS^A for oracles A, by giving U access to an oracle. It turns out that this is useful in clarifying the relationship between these complexity measures. The following theorem is credited to Ronneburger in [All01]:

Theorem 4 *The following statements hold:*

1. *Let A be complete for* E *under linear-time reductions. Then there is a $k \in \mathbb{N}$ such that* $\mathrm{KT}^A(x)/k \leq \mathrm{Kt}(x) \leq k\mathrm{KT}^A(x)$. *In other words,* $\mathrm{Kt} = \Theta(\mathrm{KT}^A)$.
2. *Let B be complete for* $\mathrm{DSPACE}(n)$ *under linear-time reductions. Then there is a $k' \in \mathbb{N}$ such that* $\mathrm{KT}^B(x)/k' \leq \mathrm{KS}(x) \leq k'\mathrm{KT}^B(x)$. *In other words,* $\mathrm{KS} = \Theta(\mathrm{KT}^B)$.

That is, one can view Kt and KS complexities as merely variations of KT complexity; Kt complexity is KT complexity relative to E and KS is KT relative to $\mathrm{DSPACE}(n)$.

If x is a string of length 2^n, then x can be viewed as the truth-table of an n-ary Boolean function. Let $\mathrm{SIZE}(x)$ denote the number of gates in the smallest circuit computing the Boolean function represented by x. As observed in [All01], $\mathrm{SIZE}(x) = O(\mathrm{KT}(x) \log \mathrm{KT}(x))$ and $\mathrm{KT}(x) = O(\mathrm{SIZE}^2(x) + \log |x|)$. For relativized computation, the correspondence is nearly as close.

Theorem 5 $\mathrm{SIZE}^A(x) = O(\mathrm{KT}^A(x)^3)$ *and* $\mathrm{KT}^A(x) = O(\mathrm{SIZE}^A(x)^2 + \log |x|)$.

In this paper we focus our attention mainly on sets of strings with high Kolmogorov complexity.

Definition 6 *For any any Kolmogorov complexity measure μ (such as* $\mathrm{K}, \mathrm{Kt}, \mathrm{KS}, \mathrm{KT}$*), define* $R_\mu = \{x \ : \ \mu(x) \geq |x|/2\}$.

The bound of $|x|/2$ in the definition of R_μ is arbitrary; all of our results hold for any reasonable bound in the range $|x|^\epsilon$ to $\epsilon|x|$, $0 < \epsilon < 1$.

It is obvious that R_{K} is co-r.e. $R_{\mathrm{Kt}} \in \mathrm{E}$, $R_{\mathrm{KS}} \in \mathrm{DSPACE}(n)$, and $R_{\mathrm{KT}} \in \mathrm{coNP}$. It is observed in [All01] that none of these sets lie in AC^0. No other upper or lower bounds are known for the resource-bounded sets R_μ. As stated in the introduction, earlier results had indicated that these sets would *not* be complete for the complexity classes in which they reside. [All01] points out that R_{Kt} is not complete for EXP under $\leq^{\mathrm{p}}_{\mathrm{tt}}$ reductions, and essentially identical observations show that R_{KS} is not complete for PSPACE under $\leq^{\log}_{\mathrm{T}}$ reducibility. For completeness, we include this argument below.

Theorem 7 R_{KS} *is not hard for* PSPACE *under* $\leq^{\log}_{\mathrm{T}}$ *reductions.*

Proof. Note first of all that $\leq^{\log}_{\mathrm{T}}$ reducibility coincides with $\leq^{\log}_{\mathrm{tt}}$ reducibility [LL76].

Let T be a subset of 0^* that is in PSPACE but not in L. Suppose $T \leq^{\log}_{\mathrm{tt}} R_{\mathrm{KS}}$, and let f be the query generator of such a reduction. Note that $\mathrm{KS}(f(0^n)) = O(\log n)$. Thus each of the strings y that the reduction queries on input 0^n has $\mathrm{KS}(y) = O(\log n)$. Hence, the only strings y for which the query can receive a "yes" answer are of length $O(\log n)$, and for such queries the answer is computable directly in space $O(\log n)$. Hence all of the queries can be answered in space $O(\log n)$ and it follows that $T \in \mathrm{L}$, contrary to our choice of T. □

3 Main Results

Some evidence that the set of resource bounded random strings could be useful was given in [BT01]. There it was shown that the set of strings with high conditional linear space complexity is hard for PSPACE under NP reductions. Improving greatly the results in [BT01], we show that strings of high Kolmogorov complexity are very useful as oracles. First, we present some hardness results in terms of nonuniform (P/poly) reductions that apply to many large complexity classes. In the next subsection we consider some special cases where we are able to strengthen our results to provide uniform reductions.

3.1 Nonuniform Hardness Results

It will be useful to recall the definition of PSPACE-*robustness* [BFNW93]. A language A is PSPACE-robust if $\mathrm{PSPACE}^A = \mathrm{P}^A$. Clearly, the complete sets for many large complexity classes (such as PSPACE, EXP, EXPSPACE, EEXPTIME, EEXPSPACE, ...) have this property.

We will refer to the *density* of language L as the fraction of all strings of length n that belong to L.

Theorem 8 *Let A be any* PSPACE*-robust set. Let L be a set of polynomial density such that for every $x \in L$, $\mathrm{KT}^A(x) > |x|^\gamma$, for some constant $0 < \gamma < 1$. Then A is reducible to L via $\leq_{\mathrm{tt}}^{\mathrm{P/poly}}$ reductions.*

Corollary 9
R_{Kt} is complete for EXP *under $\leq_{\mathrm{tt}}^{\mathrm{P/poly}}$ reductions.*
R_{KS} is complete for PSPACE *under $\leq_{\mathrm{tt}}^{\mathrm{P/poly}}$ reductions.*

The hardness results apply not only to R_{Kt} and R_{KS}, but to any dense set containing no strings of low resource-bounded Kolmogorov complexity (including the Buhrman-Mayordomo set and R_{K}).

A similar proof shows that, in the recursion-theoretic setting, the (very-high-complexity) truth-table reductions of [Kum96] can be replaced by reductions produced by small, but nonuniform, circuit families.

Theorem 10 *Any recursively enumerable set is $\leq_{\mathrm{tt}}^{\mathrm{P/poly}}$ reducible to R_{K}, hence, r.e. $\subseteq \mathrm{P}^{R_{\mathrm{K}}}/\mathrm{poly}$.*

3.2 Uniform Hardness Results

For the special cases of EXP and PSPACE, we are able to show hardness results under uniform notions of reducibility.

Theorem 11 $\mathrm{EXP} = \mathrm{NP}^{R_{\mathrm{Kt}}}$.

Theorem 12 $\mathrm{PSPACE} = \mathrm{ZPP}^{R_{\mathrm{KS}}}$.

Again, these hardness results apply not only to R_{Kt} and R_{KS}, but to any dense set containing no strings of low resource-bounded Kolmogorov complexity.

It is an interesting question if the ZPP-Turing reducibility in this PSPACE-completeness result can be improved to hardness under deterministic poly-time Turing reducibility. Although we are not yet able to answer this question, the following theorem may be indicative.

Theorem 13 $\mathrm{PSPACE} \subseteq \mathrm{P}^{R_{\mathrm{K}}}$.

Let us mention one other consequence of our PSPACE-completeness result. Watanabe and Tang showed in [WT92] that $\leq^{\mathrm{p}}_{\mathrm{T}}$ and $\leq^{\mathrm{p}}_{\mathrm{m}}$ reducibilities yield different classes of PSPACE-complete sets if and only if $\leq^{\mathrm{BPP}}_{\mathrm{T}}$ and $\leq^{\mathrm{p}}_{\mathrm{m}}$ reducibilities yield different PSPACE-complete sets. Using similar techniques, we are able to prove the following.

Theorem 14 *There is a tally set T such that* $\mathrm{PSPACE} = \mathrm{P}^{R_{\mathrm{KS}} \cup T}$*, where at least one of the following holds:*

- R_{KS} *is complete for* PSPACE *under* $\leq^{\mathrm{p}}_{\mathrm{tt}}$ *reductions, but not under* $\leq^{\log}_{\mathrm{T}}$ *reductions, or*
- $R_{\mathrm{KS}} \cup T$ *is complete for* PSPACE *under* $\leq^{\mathrm{p}}_{\mathrm{T}}$ *reductions, but not under* $\leq^{\mathrm{p}}_{\mathrm{m}}$ *reductions.*

Proof. By Theorem 12 there is a probabilistic Turing machine M running in time n^k accepting QBF with oracle R_{KS}, with error probability less than 2^{2n}. Let S be the set $\{r \ : \ r$ is the lexicographically-first sequence of length n^k with the property that, for all strings x of length n $M^{R_{\mathrm{KS}}}(x, r)$ accepts if and only if $x \in \mathrm{QBF}\}$. By standard arguments, S consists of exactly one string of each length n^k, and $S \in \mathrm{PSPACE}$. We encode the "good" coin flip sequences of S in the tally set $T = \{0^{2n^k+i} \ :$ the i-th bit of the unique string of length n^k in S is $1\}$. Note that R_{KS} can only contain a finite number of strings in 0^*. It is immediate that $T \in \mathrm{PSPACE}$ and that QBF is in $\mathrm{P}^{R_{\mathrm{KS}} \cup T}$.

If $R_{\mathrm{KS}} \cup T$ is not complete for PSPACE under under $\leq^{\mathrm{p}}_{\mathrm{m}}$ reductions, then the second condition of the theorem holds.

On the other hand, if $\mathrm{QBF} \leq^{\mathrm{p}}_{\mathrm{m}} R_{\mathrm{KS}} \cup T$, then we can apply a standard argument of [Ber78]. Berman showed that if a length-decreasing self-reducible set A is $\leq^{\mathrm{p}}_{\mathrm{m}}$-reducible to a tally set, then $A \in \mathrm{P}$. A similar argument applied to the $\leq^{\mathrm{p}}_{\mathrm{m}}$ reduction from $A = \mathrm{QBF}$ to $R_{\mathrm{KS}} \cup T$ yields a $\leq^{\mathrm{p}}_{\mathrm{tt}}$ reduction from QBF to R_{KS}. This, combined with Theorem 7, shows that the first condition of the theorem holds. □

Note that in either case, we obtain a set that is $\leq^{\mathrm{p}}_{\mathrm{T}}$-complete but not $\leq^{\log}_{\mathrm{m}}$-complete for PSPACE. It was already known (using the techniques of [Wat87]) that $\leq^{\mathrm{p}}_{\mathrm{T}}$ and $\leq^{\log}_{\mathrm{m}}$ reducibilities provide different classes of PSPACE-complete sets, but the preceding theorem provides fairly "natural" examples of sets witnessing the difference.

We are not able to prove that R_{KT} is complete for coNP under any notion of feasible reducibility. On the other hand, we are able to show that some apparently-intractable problems are reducible to R_{KT}.

Theorem 15 *The Discrete Logarithm problem is solvable in* $\mathrm{BPP}^{R_{\mathrm{KT}}}$.

Corollary 16 *The Discrete Logarithm problem is solvable in* $\mathrm{BPP}^{\mathrm{MCSP}}$.

4 Reductions from Complete Problems

In this and the next section we give proofs of our results from Section 3. Our main technique is to use relativizing hardness-randomness tradeoffs in the contrapositive. In particular we will argue that an appropriate set R_μ of Kolmogorov random strings can be used to distinguish the output of a pseudo random generator G_f (based on a function f) from truly random strings. This in turn will enable us to efficiently reduce f to R_μ. In this section, the function f will always be a complete function for a standard complexity class.

4.1 Tools

Let us first review the main technical tools we are going to use.

The authors of [BFNW93] show, for any $\epsilon > 0$, how to construct from a function f a pseudo-random generator $G_f^{\mathrm{BFNW}} : \{0,1\}^{n^\epsilon} \to \{0,1\}^n$ such that for any x of size n^ϵ, the function $G_f^{\mathrm{BFNW}}(x)$ is computable in space $O(n^\epsilon)$ given access to f for inputs of size at most n^ϵ. Moreover, for some constant c independent of ϵ, if f is PSPACE-robust, then each bit of $G_f^{\mathrm{BFNW}}(x)$ is computable in time $n^{\epsilon c}$ with oracle access to f.

Theorem 17 ([BFNW93, KvM99]) *Let f be a function, $\epsilon > 0$, and $G_f^{\mathrm{BFNW}} : \{0,1\}^{n^\epsilon} \to \{0,1\}^n$ be the pseudo-random generator described above. Let T be a set and $p(n)$ a polynomial. If for all large n $|\Pr_{r \in U_n}[r \in T] - \Pr_{x \in U_{n^\epsilon}}[G_f^{\mathrm{BFNW}}(x) \in T]| \geq 1/p(n)$, then there exists a polynomial size oracle circuit family $\{C_n\}_{n \in \mathbb{N}}$ with oracle T that computes f and queries T non-adaptively.*

Impagliazzo and Wigderson reexamine the approach of [BFNW93] and obtain a uniform version of Theorem 17. Their result immediately implies:

Theorem 18 ([IW98]) *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a random and downward self-reducible function, $\epsilon > 0$, and $G_f^{\mathrm{BFNW}} : \{0,1\}^{n^\epsilon} \to \{0,1\}^n$ be the pseudo-random generator described above. Let T be a set and $p(n)$ be a polynomial. If $|\Pr_{r \in U_n}[r \in T] - \Pr_{x \in U_{n^\epsilon}}[G_f^{\mathrm{BFNW}}(x) \in T]| \geq 1/p(n)$, for all n large enough, then there exists a probabilistic, polynomial time Turing machine with oracle T that on input x outputs $f(x)$ with probability at least $2/3$.*

The preceding two theorems provide the key derandomization techniques that are required to prove our completeness results. They are stated in the contrapositive of their original formulations since that is the way we will use them. However, some of our completeness results (namely for EXP and PSPACE) involve uniform reductions that make use of randomness. These reductions can then be further derandomized by applying hardness versus randomness tradeoffs in the standard way. We will make use of the strengthening of the results of [BFNW93], as provided by Impagliazzo and Wigderson [IW97] (see also [STV01]). Given access to a Boolean function f, [IW97] constructs a pseudo-random generator $G_f^{\mathrm{IW}} : \{0,1\}^{O(\log n)} \to \{0,1\}^n$ with the following property (see also [KvM99]):

Theorem 19 ([IW97], [KvM99]) *For any $\epsilon > 0$, there exists a constant $c > 0$ such that the following holds. Let A be a set and $n > 1$ be an integer. Let $f : \{0,1\}^{\lceil \log n \rceil} \to \{0,1\}$ be a Boolean function that cannot be computed by oracle circuits of size n^ϵ with oracle A. Then $G_f^{\mathrm{IW}} : \{0,1\}^{c\lceil \log n \rceil} \to \{0,1\}^n$ satisfies: $|\Pr_{r \in U_n}[C^A(r) = 1] - \Pr_{x \in U_{c\lceil \log n \rceil}}[C^A(G_f^{\mathrm{IW}}(x)) = 1]| < 1/n$, for any oracle circuit C^A of size at most n.*

For x of size $c\lceil \log n \rceil$, function $G_f^{\mathrm{IW}}(x)$ is computable in time polynomial in n given access to function f on inputs of length $\lceil \log n \rceil$.

4.2 Proofs

The next proof illustrates our proof technique.

Proof of Theorem 8: Let A, L and γ be as in the statement of the theorem and let f be the characteristic function of A. Consider $G_f^{\mathrm{BFNW}} : \{0,1\}^{n^\epsilon} \to \{0,1\}^n$, where we choose ϵ

as follows. We know that every bit of G_f^{BFNW} is computable in time $n^{\epsilon c}$, for some constant c independent of ϵ. We may assume that $c > 1$, so set $\epsilon = \gamma/2c$. We claim that any string in the range of G_f^{BFNW} has small KT^A complexity. Let $y = G_f^{\mathrm{BFNW}}(x)$, for some $x \in \{0,1\}^{n^\epsilon}$. On input x, every bit of $G_f^{\mathrm{BFNW}}(x)$ is computable in time $n^{\gamma/2}$ with access to oracle A. Hence, $\mathrm{KT}^A(y) \leq 2|x| + O(n^{\gamma/2}) + O(1) \leq O(n^{\gamma/2})$. It follows, that L distinguishes the output of G_f^{BFNW} from random strings, so by Theorem 17, f is $\leq_{\mathrm{tt}}^{\mathrm{P/poly}}$ reducible to L. □

Using the fact that any recursively enumerable set can be decided with polynomial advice, we can prove the analog in the recursion theoretic setting.

Proof of Theorem 10: The proof is similar to the proof of Theorem 8. Let f be the characteristic function of a recursively-enumerable set. Consider $G_f^{\mathrm{BFNW}} : \{0,1\}^{n^{1/2}} \to \{0,1\}^n$. We claim that any string in the range of G_f^{BFNW} has small Kolmogorov complexity. Let $y = G_f^{\mathrm{BFNW}}(x)$, for some $x \in \{0,1\}^{n^{1/2}}$. To compute y given x, we need to query f on inputs of size at most $n^{1/2}$. If we know the *number* l of strings z of length at most $n^{1/2}$ on which $f(z) = 1$, then we can determine exactly which strings z these are (by carrying out the enumeration until all have appeared). This number l is less than $2^{n^{1/2}+1}$ and hence can be specified with $O(n^{1/2})$ bits. String y can be fully described by giving x, l, and programs for f and G_f^{BFNW}, thus $\mathrm{K}(y) \leq O(n^{1/2})$. It follows that R_{K} distinguishes the output of G_f^{BFNW} from random strings. By Theorem 17, f is $\leq_{\mathrm{tt}}^{\mathrm{P/poly}}$ reducible to R_{K}. □

Many authors have observed that having access to the truth table of a hard function can be used to obtain upper bounds on BPP [NW94, IW97]. Similarly, using oracle access to strings of high Kolmogorov complexity we obtain the following derandomizations which we will use in the proofs of Theorems 11 and 12.

Lemma 20 *Let A be any oracle and L be a set such that $L \in \mathrm{P}^A/\mathrm{poly}$ and for every $x \in L$, $\mathrm{KT}^A(x) > |x|^\gamma$, for some constant $0 < \gamma < 1$.*

1. *If L contains a string of every length then $\mathrm{MA}^L \subseteq \mathrm{NP}^L$,*
2. *If L is of at least polynomial density then $\mathrm{BPP}^L \subseteq \mathrm{ZPP}^L$.*

Proof of Lemma 20:
1. The NP-machine M guesses a string $\chi_f \in L$ of length m, for $m = n^c$, and interprets it as the truth-table of a Boolean function f on inputs of size $\log m$. Since $\chi_f \in L$, $\mathrm{KT}^A(\chi_f) \geq m^\gamma$. As the circuit size of a function is lower bounded by the root of the KT-complexity of its truth-table (Theorem 5), f requires circuits of size at least $\Omega(m^{\gamma/2})$ even when the circuit has access to the oracle A. If, instead of A, the circuit has access to L, then (because of our assumption that L is reducible to A by circuits of size m^k for some k), the size required to compute f is at least $\Omega(m^{\gamma/4k})$.

The function f is of sufficient hardness to construct the generator G_f^{IW}, as stated in Theorem 19, to stretch $O(\log n)$ random bits into n^c pseudorandom bits that are indistinguishable from random by any polynomially bounded computation with oracle access to L. Thus we can fully derandomize the probabilistic part of the MA-computation.

2. We use a similar argument as above. Here we randomly choose a candidate for χ_f. As L is dense, the result follows immediately. □

Applying the previous lemma with oracles A and A', complete for E and DSPACE(n) respectively, Theorem 4 yields the following corollary.

Corollary 21

1. $\mathrm{MA}^{R_{\mathrm{Kt}}} = \mathrm{NP}^{R_{\mathrm{Kt}}}$,
2. $\mathrm{BPP}^{R_{\mathrm{KS}}} = \mathrm{ZPP}^{R_{\mathrm{KS}}}$.

We can immediately apply this corollary to prove Theorem 11.

Proof of Theorem 11: The inclusion $\mathrm{NP}^{R_{\mathrm{Kt}}} \subseteq \mathrm{EXP}$ is trivial, so we focus on the other inclusion. A consequence of Corollary 9 is that $\mathrm{EXP} \subseteq \mathrm{MA}^{R_{\mathrm{Kt}}}$. To see this let A be any language in EXP. Then A is accepted by a 2-prover interactive proof system with provers computable in EXP [BFL91]. The $\mathrm{MA}^{R_{\mathrm{Kt}}}$ protocol is as follows. Merlin sends Arthur the poly-size circuits that, when given access to the oracle R_{Kt}, compute the answers given by the two provers for A. Arthur then executes the MIP protocol, simulating the provers' answers by executing the circuits and querying the oracle R_{Kt}. The theorem now follows immediately from Corollary 21. □

To prove a uniform hardness result for PSPACE, as stated in Theorem 12, we can use the technique of [IW98] and the following theorem of [TV02].

Theorem 22 ([TV02]) *There exists a problem for* DSPACE(n) *that is hard for* PSPACE*, random self-reducible, and downward self-reducible.*

Proof of Theorem 12: Let $f \in \mathrm{DSPACE}(n)$ be a function that is downward and random self-reducible and hard for PSPACE, as guaranteed by Theorem 22. First, we will show that $f \in \mathrm{BPP}^{R_{\mathrm{KS}}}$. Consider $G_f^{\mathrm{BFNW}} : \{0,1\}^{n^{1/2}} \to \{0,1\}^n$. From the properties of G_f^{BFNW} it follows that all the strings in the range of G_f^{BFNW} have small space-bounded Kolmogorov complexity, in particular, $\mathrm{KS}(y) \leq O(|y|^{1/2})$, for any $y = G_f^{\mathrm{BFNW}}(z)$, for some z. Hence, R_{KS} distinguishes the output of G_f^{BFNW} from random strings and by Theorem 18, there is a probabilistic procedure with oracle R_{KS} that on input x outputs $f(x)$ with probability at least 2/3.

Hence, $\mathrm{PSPACE} \subseteq \mathrm{BPP}^{R_{\mathrm{KS}}}$. We can use Corollary 21 to derandomize the $\mathrm{BPP}^{R_{\mathrm{KS}}}$ computation to obtain $\mathrm{PSPACE} \subseteq \mathrm{ZPP}^{R_{\mathrm{KS}}}$. □

We would obtain a *deterministic* polynomial-time reduction of PSPACE to R_{KS} in the preceding theorem, if there was a deterministic method to obtain a string of high KS complexity using R_{KS} as an oracle. We do not know if this is possible. On the other hand, if one considers Kolmogorov complexity without resource bounds, we are able to achieve this.

Lemma 23 $\mathrm{BPP}^{R_{\mathrm{K}}} = \mathrm{P}^{R_{\mathrm{K}}}$.

The proof of this theorem shows how to use R_{K} as an oracle to incrementally build up a string z in R_{K}. The construction works block-wise, and is inspired by a construction of Buhrman and Vereshchagin, who use the oracle $\{x \ : \ \mathrm{K}(x) \geq |x|\}$ to build a string x with $\mathrm{K}(x) \geq |x|$ in a bit-wise fashion. Our construction obtains a string $z \in R_{\mathrm{K}}$ which then is used to derandomize the BPP computation.

Proof of Lemma 23: We prove the lemma in two steps. First, we will show that for any $m = n^{O(1)}$, we can find a string f of length m such that $\mathrm{K}(f) \geq |f|/2$, via a polynomial-time computation with access to oracle R_{K}. Then, we will use f to derandomize $\mathrm{BPP}^{R_{\mathrm{K}}}$.

We use the following claim: there exists a constant c_1 such that for any strings z and y, $\mathrm{K}(zy) \geq \mathrm{K}(z) + \mathrm{K}(y|z) - c_1 \log |zy|$. This claim immediately follows from [LV93], Section 2.8.

We search for f inductively. We start with a string z, initially equal to the empty string. Assume (inductively) that $\mathrm{K}(z) \geq |z|/2$. Try all strings y of length $2c_1 \log m$, and use the oracle R_{K} to see if $\mathrm{K}(zy) \geq |zy|/2$. We are guaranteed to find such a y, since for most y it holds that $\mathrm{K}(y|z) \geq |y|$, and for any such y $\mathrm{K}(zy) \geq \mathrm{K}(z) + \mathrm{K}(y|z) - c_1 \log |zy| \geq |z|/2 + |y| - c_1 \log |zy| \geq |z|/2 + 2c_1 \log m - c_1 \log m = |zy|/2$.

Let M be a probabilistic oracle Turing Machine that with oracle R_{K} computes a language from $\mathrm{BPP}^{R_{\mathrm{K}}}$. M takes input x and a random string r of length polynomial in $|x|$. For input x of length n, we can construct in polynomial time an oracle circuit C_x such that $C_x^{R_{\mathrm{K}}}(r) = M^{R_{\mathrm{K}}}(x, r)$. Denote the size of C_x by m; (m is polynomial in n.) We claim and prove later that any function $f : \{0,1\}^{\lceil \log m \rceil} \to \{0,1\}$ that is computable by a circuit of size $m^{1/2}$ with oracle R_{K} has Kolmogorov complexity at most $O(m^{1/2} \log m)$. (We consider the Kolmogorov complexity of a function f as the Kolmogorov complexity of its truth-table.) By the first part of the proof, in $\mathrm{P}^{R_{\mathrm{K}}}$ we can find a truth-table of function $f : \{0,1\}^{\lceil \log m \rceil} \to \{0,1\}$ such that $\mathrm{K}(f) \geq |f|/2$. Since f cannot be computed by circuits of size $O(m^{1/2})$, by Theorem 19, it can be used to estimate the acceptance probability of C_x in polynomial time with oracle access to R_{K}. Thus, to conclude $\mathrm{BPP}^{R_{\mathrm{K}}} = \mathrm{P}^{R_{\mathrm{K}}}$ we only need to show the previous claim.

A circuit of size at most $m^{1/2}$ makes oracle queries of size at most $m^{1/2}$. Since R_{K} is a recursively-enumerable set, to compute the characteristic function of R_{K} for all strings of size at most $m^{1/2}$, we only need to specify how many of these strings are in R_{K}. This can be described by $m^{1/2} + 1$ bits. Hence, any function $f : \{0,1\}^{\lceil \log m \rceil} \to \{0,1\}$, that is computable by an oracle circuit of size $m^{1/2}$ with oracle R_{K} can be described by $m^{1/2} + 1$ bits plus the description of the circuit plus some constant. Thus, for any such f, $\mathrm{K}(f) \leq O(m^{1/2} \log m)$. □

Proof of Theorem 13: Using Theorem 12 (relative to R_{K} instead of R_{KS}) and Lemma 23, we immediately obtain the desired result $\mathrm{PSPACE} \subseteq \mathrm{P}^{R_{\mathrm{K}}}$. □

5 Reductions from the Discrete Log Problem

The previous section paints an illuminating picture about completeness of sets with high resource-bounded Kolmogorov complexity and completeness in PSPACE and EXP and larger complexity classes. In this section we explore what these techniques have to say about KT complexity and sets in NP and coNP.

It was already observed in [KC00] that if MCSP is in P/poly, then secure pseudorandom generators do not exist. Although we know of no implication (in either direction) between the hypotheses "MCSP is in P/poly" and "R_{KT} is in P/poly," the proof of [KC00] also suffices to show that secure pseudorandom generators do not exist if R_{KT} is in P/poly.

We do not know if R_{KT} is complete for coNP or hard for any familiar subclass of coNP under any feasible notion of reducibility. However, we are able to show that there is a probabilistic polynomial-time Turing machine that solves the Discrete Logarithm Problem with bounded error, when given an oracle for R_{KT}. More precisely, when given as input a triple (p, g, z) where p is an n-bit prime number, $0 < g < p$, and $0 < z < p$, the machine outputs with probability at

least 2/3 a positive integer i such that $g^i \equiv z \pmod{p}$, or 0 if there is no such i. As will be clear from the proof, R_{KT} can be replaced by any suitably dense language containing no strings of KT-complexity less than n^ϵ for some $\epsilon > 0$. In particular, the Discrete Log is in $\mathrm{BPP}^{\mathrm{MCSP}}$.

Kabanets and Cai observe in [KC00] that $\mathrm{BPP} \subseteq \mathrm{ZPP}^{R_{\mathrm{KT}}}$, but we are not able to show that $\mathrm{BPP}^{R_{\mathrm{KT}}}$ is contained in $\mathrm{ZPP}^{R_{\mathrm{KT}}}$. (I.e., we know of no analog of Lemma 20 for KT-complexity.) Thus we are not able to improve our reduction from a BPP-reduction to a ZPP-reduction. (We note that, for inputs (p, g, x) such that x is in the orbit of g, which is the usual class of inputs for which the discrete log is of interest, the algorithm we present does exhibit ZPP-like behavior, since with high probability we obtain a number i that we can deterministically check satisfies $g^i \equiv x \bmod p$. However, when x is not in the orbit of g, we obtain no *proof* of this fact – merely strong evidence.)

The hardness versus randomness tradeoff we will use to establish this result is the construction by [HILL99] of a pseudo-random generator $G_f^{\mathrm{HILL}} : \{0,1\}^n \to \{0,1\}^{2n}$ out of a (supposedly one-way) function $f : \{0,1\}^n \to \{0,1\}^n$. G_f^{HILL} is computable in polynomial time given oracle access to f.

We will apply their result with f^{-1} being the Discrete Log function. Since the Discrete Log really is a family of functions parameterized by the prime p and the generator g, we will consider functions f that are similarly parameterized. More specifically, we will consider a function $f(y, x)$ that is length-preserving for every fixed y, i.e. $x \mapsto f_{|x|}(y, x)$ for $f_n(y, \cdot) : \{0,1\}^n \to \{0,1\}^n$. The function $f(y, x)$ will be computable uniformly in time polynomial in $|x|$ (so wlog. y is polynomially bounded in $|x|$). The result in [HILL99] can be stated as follows.

Theorem 24 ([HILL99]) *Let $f(y, x)$ be computable uniformly in time polynomial in $|x|$. For any oracle L, polynomial-time probabilistic oracle Turing machine M, and polynomial p, there exists a polynomial-time probabilistic oracle Turing machine N and polynomial q such that the following holds for any n and y: If $|\Pr_{r \in U_{2n}, s}[M^L(y, r, s) = 1] - \Pr_{x \in U_n, s}[M^L(y, G^{\mathrm{HILL}}_{f_n(y,\cdot)}(x), s) = 1]| \geq 1/p(n)$, then $\Pr_{x \in U_n, s}[f(y, N^L(y, f(y, x), s)) = f(y, x)] \geq 1/q(n)$, where s denotes the internal coin flips of M or N, respectively.*

Theorem 24 states that if there exists a distinguisher with access to an oracle L that distinguishes the output of G_f^{HILL} from the uniform distribution, then oracle access to L suffices to invert f on a nonnegligible fraction of the inputs. We now argue that such a distinguisher exists in case $L = R_{\mathrm{KT}}$.

Lemma 25 *Let L be a language of polynomial density such that, for some c, for every $x \in L$, $\mathrm{KT}(x) \geq |x|^{1/c}$. Let $G(y, x)$ be a function computable uniformly in time polynomial in $|x|$ such that for any y, $G_n(y, \cdot) : \{0,1\}^n \to \{0,1\}^{2n}$. Then there is a polynomial-time probabilistic oracle machine M and a polynomial p such that for any n and y, $|\Pr_{r \in U_{2n}, s}[M^L(y, r, s) = 1] - \Pr_{x \in U_n, s}[M^L(y, G(y, x), s) = 1]| \geq 1/p(n)$.*

Proof. Similar to [RR97], we use the construction of [GGM86] to build out of G a pseudo-random generator G' with larger stretching. Specifically, the proof of Theorem 4.1 of [RR97] constructs $G'(y, x)$ such that $G'_n(y, \cdot) : \{0,1\}^n \to \{0,1\}^{2^k}$, and shows that if $z = G'(y, x)$, then the circuit size $\mathrm{SIZE}(z) = (n + k)^{O(1)}$. We can pick $k = O(\log n)$ such that for each x and polynomially bounded y, $\mathrm{KT}(G'(y, x)) < |G'(y, x)|^{1/c}$. Thus L is a statistical test that accepts almost all random strings of length $|x|^{O(1)}$ but rejects all pseudorandom strings. As in [RR97], this gives us a probabilistic oracle machine using L that distinguishes $G(y, \cdot)$ from the uniform distribution. □

Theorem 24 and Lemma 25 allow us to show that Discrete Log is computable in $\mathrm{BPP}^{R_{\mathrm{KT}}}$.

Proof of Theorem 15: On input (p, g, z), we first check (in BPP) if p is prime.

Let n be the number of bits in p, and y denote the pair (p, g). Consider the function $f(y, x) = g^x \bmod p$. Lemma 25 with $L = R_{\mathrm{KT}}$ and Theorem 24 give us a polynomial-time probabilistic oracle Turing machine N such that for some polynomial q and all p and g, with probability at least $1/q(n)$ over randomly chosen x and s, $N^{R_{\mathrm{KT}}}(p, g, g^x, s)$ produces an output i such that $g^i = g^x$.

Now we make use of the self-reducibility properties of the Discrete Log. In particular, on our input (p, g, z), we choose many more than $q(n)$ values v and run algorithm N on input $(p, g, zg^v \bmod p)$. If z is in the orbit of g, then with high probability at least one of these trials will return a value u such that $g^u = zg^v \bmod p$, which means that we can pick $i = u - v$ and obtain $z = g^i \bmod p$.

On the other hand, if none of the trials is successful, then with high probability z is not in the orbit of g and the algorithm should return 0. □

To summarize our knowledge about R_{KT}, it lies in coNP, and it is at least as hard as the discrete logarithm problem. Is it in NP ∩ coNP? If so, then it would provide a dense combinatorial property in NP that is useful against P/poly, contrary to a conjecture of Rudich [Rud97]. Is there any stronger evidence that R_{KT} is not in NP?

Theorem 26 *If R_{KT} is in* NP*, then* MA = NP*.*

Proof. It is shown in [IKW01] that if an NP machine can, on input of length n, find the truth table of a function of size $n^{O(1)}$ with large circuit complexity, then MA = NP. Certainly this is easy if R_{KT} is in NP. □

This observation (similar to observations in [IKW01]) can not be taken as evidence that $R_{\mathrm{KT}} \notin$ NP, since many people conjecture that MA is equal to NP.

However it does show that this would require nonrelativizing proof techniques.

6 Open Problems

KT-complexity was introduced in [All01] as a tool for summarizing some recent progress in the field of derandomization, and for describing the theory of natural proofs from the standpoint of Kolmogorov complexity. In this paper, we provide additional motivation for studying KT complexity and its variants. It provides natural and interesting examples of apparently intractable problems in NP, PSPACE, and EXP that are not complete under the more familiar notions of reducibility and hence constitute a fundamentally new class of complete problems. It is worth pointing out that variants of these sets (such as $\{x : \mathrm{Kt}(x) \geq |x|/3\}$) appear to be incomparable to R_{Kt} under $\leq^{\mathrm{p}}_{\mathrm{m}}$ reductions. This deserves further investigation.

Is there some sense in which KT complexity yields NP-complete sets? For instance, what can one say about $\{(x, y, i) \mid \mathrm{KT}(x|y) \leq i\}$? The result of [VV83] is intriguing in this light; in [VV83] Vazirani and Vazirani presented a language in NP that is complete under probabilistic reductions that is not known to be complete under deterministic reductions. Their problem superficially seems to be related to time-bounded Kolmogorov complexity.

Other intriguing open questions are: Is the EXP-complete set R_{Kt} in P? Is it in L? Is PSPACE $\subseteq \mathrm{P}^{R_{\mathrm{KS}}}$?

Acknowledgments

This work was supported by National Science Foundation grant number CCR-0104823. We acknowledge illuminating discussions with Lance Fortnow, as well as helpful conversations of the first author with Paul Beame, Henry Cohn, Chris Umans, and Ramarathnam Venkatesan during a visit to Microsoft Research. We also acknowledge fruitful discussions with Steven Rudich, Valentine Kabanets, Manindra Agrawal, and Kolya Vereshchagin.

References

[AAI+02] M. Agrawal, E. Allender, R. Impagliazzo, R. Pitassi, and S. Rudich. Reducing the complexity of reductions. *Computational Complexity*, 2002. to appear; a preliminary version appeared in STOC 1997.

[All01] E. Allender. When worlds collide: Derandomization, lower bounds, and kolmogorov complexity. In *Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS)*, volume 2245 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2001.

[Ber78] P. Berman. Relationship between density and deterministic complexity of NP-complete languages. In *International Conference on Automata, Languages, and Programming (ICALP)*, volume 62 of *Lecture Notes in Computer Science*, pages 63–71. Springer, 1978.

[BFL91] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[BFNW93] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[BM97] H. Buhrman and E. Mayordomo. An excursion to the Kolmogorov random strings. *Journal of Computer and System Sciences*, 54:393–399, 1997.

[BT01] H. Buhrman and L. Torenvliet. Randomness is hard. *SIAM Journal on Computing*, 30:1485–1501, 2001.

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[HILL99] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28:1364–1396, 1999.

[IKW01] R. Impagliazzo, V. Kabanets, and A. Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. In *IEEE Conference on Computational Complexity*, pages 2–12, 2001.

[IW97] R. Impagliazzo and A. Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *ACM Symposium on Theory of Computing (STOC)*, pages 220–229, 1997.

[IW98] R. Impagliazzo and A. Wigderson. Randomness vs. time: de-randomization under a uniform assumption. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 734–743, 1998.

[Kab01] V. Kabanets. Easiness assumptions and hardness tests: Trading time for zero error. *Journal of Computer and System Sciences*, 63(2):236–252, 2001.

[KC00] V. Kabanets and J.-Y. Cai. Circuit minimization problem. In *ACM Symposium on Theory of Computing (STOC)*, pages 73–79, 2000.

[Ko91] K.-I Ko. On the complexity of learning minimum time-bounded turing machines. *SIAM Journal on Computing*, 20:962–986, 1991.

[Kum96] M. Kummer. On the complexity of random strings. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1046 of *Lecture Notes in Computer Science*, pages 25–36. Springer, 1996.

[KvM99] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. In *ACM Symposium on Theory of Computing (STOC)*, pages 659–667, 1999.

[Lad75] R. E. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, 22:155–171, 1975.

[Lev84] L. A. Levin. Randomness conservation inequalities; information and independence in mathematical theories. *Information and Control*, 61:15–37, 1984.

[LL76] R. Ladner and N. Lynch. Relativization of questions about log space reducibility. *Mathematical Systems Theory*, 10:19–32, 1976.

[LV93] M. Li and P. Vitanyi. *Introduction to Kolmogorov Complexity and its Applications.* Springer-Verlag, August 1993.

[Mar66] D.A. Martin. Completeness, the recursion theorem and effectively simple sets. *Proc. Am. Math. Soc.*, 17:838–842, 1966.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.

[RR97] A. A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55:24–35, 1997.

[Rud97] S. Rudich. Super-bits, demi-bits, and $\tilde{\text{np}}$/qpoly-natural proofs. In *RANDOM: International Workshop on Randomization and Approximation Techniques in Computer Science*, volume 1269 of *Lecture Notes in Computer Science*. Springer, 1997.

[Sip83] M. Sipser. A complexity theoretic approach to randomness. In *ACM Symposium on Theory of Computing (STOC)*, pages 330–335, 1983.

[STV01] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62:236–266, 2001.

[Tre01] L. Trevisan. Construction of extractors using pseudo-random generators. *Journal of the ACM*, 48(4):860–879, July 2001.

[TV02] L. Trevisan and S. Vadhan. Pseudorandomness and average-case complexity via uniform reductions. In *IEEE Conference on Computational Complexity*, 2002. To appear.

[VV83] U. V. Vazirani and V. V. Vazirani. A natural encoding scheme proved probabilistic polynomial complete. *Theoretical Computer Science*, 24:291–300, 1983.

[Wat87] O. Watanabe. A comparison of polynomial time completeness notions. *Theoretical Computer Science*, 54:249–265, 1987.

[WT92] O. Watanabe and S. Tang. On polynomial-time Turing and many-one completeness in PSPACE. *Theoretical Computer Science*, 97:199–215, 1992.