Power from Random Strings

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

Abstract

We show that sets consisting of strings of high Kolmogorov complexity provide examples of sets that are complete for several complexity classes under probabilistic and non-uniform reductions. These sets are provably not complete under the usual many-one reductions.

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 the usual Kolmogorov complexity measure K*, Levin's time-bounded Kolmogorov complexity* Kt *[27], a space-bounded Kolmogorov measure* KS*, and the time-bounded Kolmogorov complexity measure* KT *that was introduced in [4], respectively.*

Our main results are:

1. R_{KS} and R_{Kt} are complete for PSPACE *and* EXP*, respectively, under* P/poly*-truth-table reductions.*

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

3. $\mathrm{PSPACE} = \mathrm{ZPP}^{R_{\mathrm{KS}}} \subseteq \mathrm{P}^{R_{\mathrm{K}}}$.

4. The Discrete Log, Factoring, and several lattice problems are solvable in $\mathrm{BPP}^{R_{\mathrm{KT}}}$.

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.

We will focus on the set $R_{\mathrm{K}} = \{x : \mathrm{K}(x) \geq |x|/2\}$ 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 $\epsilon|x|$, for $0 < \epsilon < 1$.

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 intractable sets that are not complete for any of the standard complexity classes. For instance, Buhrman and Mayordomo [9] 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. In [4] it is argued that R_{Kt}, a set in EXP, is not complete under polynomial-time many-one or truth-table reductions.

As another example of this phenomenon, Cai and Kabanets [21] studied the Minimum Circuit Size Problem (MCSP), a set in NP which is closely related to R_{KT}. 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 [23] 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 [24], although it had long been known [29] that Turing reductions can be used. It should be emphasized that Kummer's reduction, in fact a conjunctive truth-table reduction, is *not* feasible and asks *many* queries. It can be shown that R_{K} is not complete for polynomial time conjunctive truth-table reductions.

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 [10] 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

*`{allender,mkoucky,detlef}@cs.rutgers.edu`, Rutgers University. Partially supported by NSF grant CCR-0104823.

†`buhrman@cwi.nl`, CWI and University of Amsterdam.

‡`dieter@cs.wisc.edu`, University of Wisconsin. Partially supported by NSF Career award CCR-0133693.

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 [27] 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 [5] 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 [38], 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 *deterministic* polynomial time Turing reductions. The main tool to prove this is a polynomial time algorithm to construct a string of high Kolmogorov complexity, using R_{K} as an oracle.

For R_{KT}, a set in coNP, we do not obtain completeness results but we show that R_{KT} is hard under BPP-reductions for the well-known candidate NP-intermediate problems: Discrete Log, Factoring, and several lattice problems.

1.1 The connection to derandomization

There is an obvious connection between Kolmogorov complexity and derandomization. For any randomized algorithm $\mathcal{A}$ having small error probability, and any input x, the coin flip sequences r resulting in a wrong answer are atypical, and hence 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 Kolmogorov measure μ. If we can generate a coin flip sequence r belonging 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 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 [35]. 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. [6] and by Impagliazzo and Wigderson [18] can be cast as an application with $\mu = $ KT, a time-bounded Kolmogorov measure introduced in [4], which essentially measures the circuit complexity of the Boolean function defined by the string described. The observation from [22] that the construction of [18] 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 pseudorandom generators [37]. Other connections are surveyed in [4].

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 pseudorandom generator secure against A that is computable within $\mathcal{C}$. However, we argue that no pseudorandom 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 [6] and [18], as well as the uniform ones in [16] and [19].

1.2 The structure of complexity classes

The tools of reducibility and 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 [26] 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 [26] 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 [40, 1] (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.

We mentioned two notable examples that run counter to this trend [9, 21]. 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.

2 Resource-Bounded K-Complexity

The computational model that we use throughout this paper is the multi-tape Turing machine with random-access to its tapes. Our ideas work in any general model of computation. Except for some statements in *this* section (e.g., Theorems 3 and 5), all our results hold verbatim for models such as sequential-access Turing machines.

We assume that the reader is familiar with Kolmogorov complexity: $\mathrm{K}(x) = \min\{|d| : U(d) = x\}$ (for some fixed universal Turing machine U). For background, consult [28] (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 [27].

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 [4], where it was shown to be related to the "easy witness" method of Impagliazzo, Kabanets, and Wigderson [20, 17], and the "natural proofs" framework of Razborov and Rudich [33].

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 $\log\log|x| \leq \mathrm{KT}(x) \leq |x| + O(\log|x|)$. Also observe that KT can be generalized to obtain KT^A for oracles A, by giving U access to the oracle.

$\mathrm{KT}^A(x)$ essentially measures the circuit size of x relative to A in the following sense. 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 $\mathrm{SIZE}^A(x)$ denote the number of wires in the smallest circuit with oracle A that computes the Boolean function represented by x. We have:

Theorem 3 ([4]) $\mathrm{SIZE}^A(x) = O(\mathrm{KT}^A(x)^3)$*, and* $\mathrm{KT}^A(x) = O(\mathrm{SIZE}^A(x) \cdot (\log \mathrm{SIZE}^A(x) + \log|x|))$.

We also need a space-bounded analog of KT complexity:

Definition 4 *Let U be a universal Turing machine. Define* $\mathrm{KS}(x) = \min\{|d| + s : U(d) = x$ *in space at most s*$\}$.

The following theorem is credited to Ronneburger in [4]. It states that 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 $\mathrm{DSPACE}(n)$.

Theorem 5 *Let A, A' be complete under linear-time reductions for* E *and* $\mathrm{DSPACE}(n)$*, respectively. Then* $\mathrm{Kt}(x) = \Theta(\mathrm{KT}^A(x) + \log|x|)$*, and* $\mathrm{KS}(x) = \Theta(\mathrm{KT}^{A'}(x) + \log|x|)$.

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

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

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 [4] 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. [4] 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 [25].

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 Complexity of R_{Kt}, R_{KS}, and R_{K}

Improving greatly the results in [10], we show that strings of high Kolmogorov complexity are very useful as oracles. 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 pseudorandom 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, we will exploit pseudorandom generators G_f that may take more time to compute than the randomized procedure they try to fool. They yield our hardness results for R_{Kt}, R_{KS}, and R_{K}. In the next section, we will use pseudorandom generators G_f that are more efficiently computable and obtain hardness results for R_{KT}.

We first describe the derandomization tools we will use. Then we present some hardness results in terms of nonuniform (P/poly) reductions that apply to many complexity classes. Finally, we consider some special cases where we are able to strengthen our results to provide uniform reductions.

3.1 Tools

It will be useful to recall the definition of PSPACE-*robustness* [6]. 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 use several related constructions that build a pseudorandom generator G_f out of a function f. They are all based on the Nisan-Wigderson paradigm [31]. The authors of [6] construct, for any $\epsilon > 0$, a variant $G_f^{\mathrm{BFNW}} : \{0,1\}^{n^\epsilon} \mapsto \{0,1\}^n$ 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 the Boolean function f on inputs of size at most n^ϵ. Moreover, if f is PSPACE-robust, then there is a constant c independent of ϵ, such that each bit of $G_f^{\mathrm{BFNW}}(x)$ is computable in time $n^{\epsilon c}$ with oracle access to f. The following hardness versus randomness tradeoff holds.

Theorem 8 ([6, 22]) *Let f be a Boolean function, $\epsilon > 0$, and $G_f^{\mathrm{BFNW}} : \{0,1\}^{n^\epsilon} \mapsto \{0,1\}^n$ be the pseudorandom generator described above. Let T be a set and $p(n)$ 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 large 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.*

In [19], Impagliazzo and Wigderson reexamine the approach of [6]. For any $\epsilon > 0$, their pseudorandom generator $G_f^{\mathrm{IW98}} : \{0,1\}^{n^\epsilon} \mapsto \{0,1\}^n$ is also computable in space $O(n^\epsilon)$ with oracle access to f on inputs of size at most n^ϵ. It satisfies the following uniform version of Theorem 8.

Theorem 9 ([19]) *Let $f : \{0,1\}^* \mapsto \{0,1\}^*$ be a random and downward self-reducible function, $\epsilon > 0$, and $G_f^{\mathrm{IW98}} : \{0,1\}^{n^\epsilon} \mapsto \{0,1\}^n$ be the pseudorandom generator described above. Let T be a set and $p(n)$ be a polynomial. If $|\Pr_{r\in U_n}[r \in T] - \Pr_{x\in U_{n^\epsilon}}[G_f^{\mathrm{IW98}}(x) \in T]| \geq 1/p(n)$, 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 [6], as provided by Impagliazzo and Wigderson [18] (see also [36]). Given access to a Boolean function f, [18] constructs a pseudorandom generator $G_f^{\mathrm{IW97}} : \{0,1\}^{O(\log n)} \mapsto \{0,1\}^n$ with the following property (see also [22]):

Theorem 10 ([18, 22]) *For any $\epsilon > 0$, there exist constants $c, c' > 0$ such that the following holds. Let A be a set and $n > 1$ be an integer. Let $f : \{0,1\}^{c\log n} \mapsto \{0,1\}$ be a Boolean function that cannot be computed by oracle circuits of size $n^{c\epsilon}$ with oracle A. Then $G_f^{\mathrm{IW97}} : \{0,1\}^{c'\log n} \mapsto \{0,1\}^n$ satisfies: $|\Pr_{r\in U_n}[C^A(r) = 1] - \Pr_{x\in U_{c'\log n}}[C^A(G_f^{\mathrm{IW97}}(x)) = 1]| < 1/n$, for any oracle circuit C^A of size at most n.*

For x of size $c'\log n$, $G_f^{\mathrm{IW97}}(x)$ is computable in time polynomial in n given access to f on inputs of length $c\log n$.

3.2 Nonuniform Hardness Results

Our first result illustrates our main technique. We will refer to the *density* of language L as the fraction of all strings of length n that belong to L.

Theorem 11 *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.*

Proof. Let f be the characteristic function of A. Consider $G_f^{\mathrm{BFNW}} : \{0,1\}^{n^\epsilon} \mapsto \{0,1\}^n$, 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, given access to A. 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, so by Theorem 8, f is $\leq_{\mathrm{tt}}^{\mathrm{P/poly}}$ reducible to L. □

Corollary 12
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}).

In the recursion-theoretic setting, the (very-high-complexity) truth-table reductions of [24] can be replaced by reductions produced by small, but nonuniform, circuit families. The proof relies on the fact that any recursively enumerable set can be decided with polynomial advice.

Theorem 13 *Any recursively enumerable set is $\leq_{\mathrm{tt}}^{\mathrm{P/poly}}$ reducible to R_{K}, hence, r.e. $\subseteq \mathrm{P}^{R_{\mathrm{K}}}$/poly.*

Proof. Let f be the characteristic function of a recursively-enumerable set. Consider $G_f^{\mathrm{BFNW}} : \{0,1\}^{n^{1/2}} \mapsto \{0,1\}^n$. 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 8, f is $\leq_{\mathrm{tt}}^{\mathrm{P/poly}}$ reducible to R_{K}. □

3.3 Uniform Hardness Results

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

Theorem 14 $\mathrm{EXP} = \mathrm{NP}^{R_{\mathrm{Kt}}}$.

Theorem 15 $\mathrm{PSPACE} = \mathrm{ZPP}^{R_{\mathrm{KS}}}$.

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.

For the proofs of Theorems 14 and 15, we will need the next lemma. Several authors have observed that having access to the truth table of a hard function can be used to obtain upper bounds on BPP [31, 18]. Similarly, using oracle access to strings of high Kolmogorov complexity we obtain the following derandomizations.

Lemma 16 *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. $\mathrm{MA}^L \subseteq \mathrm{NP}^L$*, if L contains a string of every length.*
2. $\mathrm{BPP}^L \subseteq \mathrm{ZPP}^L$*, if L is of at least polynomial density.*

Proof. 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 $\chi_f \in L$, $\mathrm{KT}^A(\chi_f) \geq m^\gamma$. Thus by Theorem 3, 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^{IW97}, as stated in Theorem 10, to stretch $O(\log n)$ random bits into n pseudorandom bits that are indistinguishable from random by any oracle circuit of size n with 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 under linear-time reductions for E and $\mathrm{DSPACE}(n)$ respectively, Theorem 5 yields the following corollary.

Corollary 17 $\mathrm{MA}^{R_{\mathrm{Kt}}} = \mathrm{NP}^{R_{\mathrm{Kt}}}$ *and* $\mathrm{BPP}^{R_{\mathrm{KS}}} = \mathrm{ZPP}^{R_{\mathrm{KS}}}$.

Proof of Theorem 14. The inclusion $\mathrm{NP}^{R_{\mathrm{Kt}}} \subseteq \mathrm{EXP}$ is trivial, so we focus on the other inclusion. A consequence of Corollary 12 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 [5]. 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 from Corollary 17. □

To prove a uniform hardness result for PSPACE, as stated in Theorem 15, we can use the technique of [19] and the following theorem of [38].

Theorem 18 ([38]) *There exists a problem in* $\mathrm{DSPACE}(n)$ *that is hard for* PSPACE*, random self-reducible, and downward self-reducible.*

Proof of Theorem 15. Let $f \in \mathrm{DSPACE}(n)$ be a function that is downward and random self-reducible and that is hard for PSPACE, as guaranteed by Theorem 18. First, we will show that $f \in \mathrm{BPP}^{R_{\mathrm{KS}}}$. Consider $G_f^{\mathrm{IW98}} : \{0,1\}^{n^{1/2}} \mapsto \{0,1\}^n$. From the properties of G_f^{IW98} it follows that all the strings in the range of G_f^{IW98} have small space-bounded Kolmogorov complexity, in particular, $\mathrm{KS}(y) \leq O(|y|^{1/2})$, for any $y = G_f^{\mathrm{IW98}}(z)$, for some z. Hence, R_{KS} distinguishes the output of G_f^{IW98} from random strings and by Theorem 9, 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 17 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 Theorem 15, if there was a deterministic method to construct 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 19 $\mathrm{BPP}^{R_{\mathrm{K}}} = \mathrm{P}^{R_{\mathrm{K}}}$.

The proof of this lemma 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 (personal communication), 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 we then use to derandomize the BPP computation.

Proof of Lemma 19. For any $m = n^{O(1)}$, we first show how to construct a string z of length m such that $\mathrm{K}(z) \geq |z|/2$ via a polynomial-time computation with access to oracle R_{K}. We will use the following property referred to as symmetry of information in [28, Section 2.8]: There exists a constant c_1 such that for any strings z and y, $\mathrm{K}(zy) \geq \mathrm{K}(z) + \mathrm{K}(y|z) - c_1 \log |zy|$.

We build z inductively. We start with z 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 $\mathrm{K}(y|z) \geq |y|$ holds for most y, and for any such y, $\mathrm{K}(zy) \geq \mathrm{K}(z) + \mathrm{K}(y|z) - c_1 \log |zy| \geq |z|/2 + |y| - c_1 \log |zy| \geq |zy|/2$. Set z to be zy.

Note that any function $f : \{0,1\}^{\log m} \mapsto \{0,1\}$ that is computable by a circuit C of size $m^{1/2}$ with oracle R_{K} has Kolmogorov complexity at most $O(m^{1/2} \log m)$. (We define the Kolmogorov complexity of f to be the Kolmogorov complexity of its characteristic sequence.) This is because 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 $m^{1/3}$, we only need to specify how many of these strings are in R_{K}. This can be done using $m^{1/2} + 1$ bits. Hence, f can be described by these $m^{1/2} + 1$ bits plus the description of the circuit C plus some constant. Thus, $\mathrm{K}(f) \leq O(m^{1/2} \log m)$.

It follows that the function f with characteristic sequence z requires R_{K} oracle circuits of size at least $m^{1/2}$. By Theorem 10, we can use z to derandomize $\mathrm{BPP}^{R_{\mathrm{K}}}$ computations. □

Using Theorem 15 (relative to R_{K} instead of R_{KS}) and Lemma 19, we obtain:

Theorem 20 $\mathrm{PSPACE} \subseteq \mathrm{P}^{R_{\mathrm{K}}}$.

Let us mention one other consequence of our PSPACE-completeness result. Watanabe and Tang showed in [41] that $\leq_{\mathrm{T}}^{\mathrm{p}}$ and $\leq_{\mathrm{m}}^{\mathrm{p}}$ reducibilities yield different classes of PSPACE-complete sets if and only if $\leq_{\mathrm{T}}^{\mathrm{BPP}}$ and $\leq_{\mathrm{m}}^{\mathrm{p}}$ reducibilities yield different PSPACE-complete sets. Using similar techniques, we are able to prove the following.

Theorem 21 *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* $\leq_{\mathrm{tt}}^{\mathrm{p}}$ *reductions, but not under* $\leq_{\mathrm{T}}^{\log}$ *reductions, or*
- $R_{\mathrm{KS}} \cup T$ *is complete for* PSPACE *under* $\leq_{\mathrm{T}}^{\mathrm{p}}$ *reductions, but not under* $\leq_{\mathrm{m}}^{\mathrm{p}}$ *reductions.*

Proof. By Theorem 15 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 $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions, then the second condition of the theorem holds.

On the other hand, if QBF $\leq_{\mathrm{m}}^{\mathrm{p}} R_{\mathrm{KS}} \cup T$, then we can apply a standard argument of [7]. Berman showed that if a length-decreasing self-reducible set A is $\leq_{\mathrm{m}}^{\mathrm{p}}$-reducible to a tally set, then $A \in \mathrm{P}$. A similar argument applied to the $\leq_{\mathrm{m}}^{\mathrm{p}}$ reduction from $A = \mathrm{QBF}$ to $R_{\mathrm{KS}} \cup T$ yields a $\leq_{\mathrm{tt}}^{\mathrm{p}}$ 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 $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete but not $\leq_{\mathrm{m}}^{\log}$-complete for PSPACE. It was already known (using the techniques of [40]) that $\leq_{\mathrm{T}}^{\mathrm{p}}$ and $\leq_{\mathrm{m}}^{\log}$ reducibilities provide different classes of PSPACE-complete sets, but the preceding theorem provides fairly "natural" examples of sets witnessing the difference.

4 Complexity of R_{KT}

The previous section paints an illuminating picture about the hardness of sets with high Kt, KS, and K complexity for PSPACE, EXP, and larger complexity classes. In this section, we explore what these techniques have to say about the hardness of R_{KT}, a set in coNP. We are not able to show completeness of R_{KT} for coNP, but we can show the hardness of R_{KT} under randomized polynomial-time reductions for problems that are thought to be NP-intermediate: Discrete Log, Integer Factorization, and certain lattice problems.

We point out that the set R_{KT} seems closely related to the set MCSP defined in [21] as $\{(x, s) : \mathrm{SIZE}(x) \leq s\}$. Although we do not know of efficient reductions between R_{KT} and MCSP in either direction, all our hardness results for R_{KT} also hold for MCSP. Based on a connection with natural proofs, [21] showed that if MCSP is in P then, for any $\epsilon > 0$, there is a randomized algorithm running in time $2^{n^{\epsilon}}$ that factors Blum integers well on the average. Our results imply that if MCSP is in P then there is a randomized polynomial-time algorithm that factors arbitrary integers.

4.1 Tools

In order to prove hardness results for R_{KT}, we need to apply hardness versus randomness tradeoffs for pseudorandom generators G_f of lower computational complexity than in Section 3. We will use the cryptographic pseudorandom generators that developed out of the seminal work by Blum and Micali [8], and by Yao [42]. In [16], it is shown how to construct such a pseudorandom generator $G_f^{\mathrm{HILL}} : \{0,1\}^n \mapsto \{0,1\}^{2n}$ out of any function f. The pseudorandom generator G_f^{HILL} is computable in polynomial time given access to f, and is secure provided f is one-way.

The known hardness versus randomness tradeoffs for G_f^{HILL} differ in two relevant respects from those used in Section 3. First, breaking G_f^{HILL} only lets us invert f on a nonnegligible fraction of the inputs, so not necessarily on all inputs. The implicit or explicit random self-reducibility of the problems considered in Section 3, allowed us there to make the transition from nonnegligible fractions to all. However, unlike for EXP and PSPACE, there are no NP-complete problems that are known to be random self-reducible. In fact, there provably are no non-adaptively random self-reducible NP-complete sets unless the polynomial-time hierarchy collapses [14].

Nevertheless, for some specific NP-intermediate problems h and polynomial-time computable functions f (where h may or may not coincide with f^{-1}), a worst-case to average-case connection *is* known. That is, inverting f on a nonnegligible fraction of the inputs allows one to compute h efficiently on any input. We are able to prove hardness of R_{KT} for such problems.

The second difference with Section 3 is that the uniform hardness versus randomness tradeoffs in [16] are as strong as the nonuniform ones. Therefore, unlike in Section 3, we only need to consider uniform reductions here.

We will apply G_f^{HILL} for functions f that take some additional parameters y besides the actual input x. In the case of the Discrete Log problem, for example, y will describe the prime p and the generator g of $\mathbb{Z}_p^*$ defining the basis for the logarithm. More precisely, we will consider functions of the form $f(y, x)$ that are length-preserving for every fixed y, i.e., $f_y : \{0,1\}^n \mapsto \{0,1\}^n$, where $f_y(x) = f(y, x)$. The function $f(y, x)$ will be computable uniformly in time polynomial in $|x|$ (so without loss of generality, $|y|$ is polynomially bounded in $|x|$). The hardness versus randomness tradeoff in [16] can be stated as follows.

Theorem 22 ([16]) *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* $|\Pr_{r \in U_{2n}, s}[M^L(y, r, s) = 1] - \Pr_{x \in U_n, s}[M^L(y, G_{f_y}^{\mathrm{HILL}}(x), s) = 1]| \geq 1/p(n)$*, 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 22 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 polynomial fraction of the inputs. We now argue that such a distinguisher exists in the case where L denotes R_{KT} or any set of polynomial density that contains no strings of small KT complexity.

Theorem 23 *Let L be a language of polynomial density such that, for some $\epsilon > 0$, for every $x \in L$,* $\mathrm{KT}(x) \geq |x|^{\epsilon}$.

Let $f(y,x)$ be computable uniformly in time polynomial in $|x|$. There exists a polynomial-time probabilistic oracle Turing machine N and polynomial q such that for any n and y: $\Pr_{x \in U_n, s}[f(y, N^L(y, f(y,x), s)) = f(y,x)] \geq 1/q(n)$, where s denotes the internal coin flips of N.

Proof. Let $G_y(x)$ denote $G^{\mathrm{HILL}}_{f_y}(x)$. As in [33], we use the construction of [15] to modify G_y and obtain a pseudorandom generator G'_y producing longer output. Specifically, the proof of Theorem 4.1 of [33] constructs $G'_y : \{0,1\}^n \mapsto \{0,1\}^{2^k}$, and shows that if $z = G'_y(x)$, then $\mathrm{SIZE}(z) = (n+k)^{O(1)}$. We can pick $k = O(\log n)$ such that for each x and polynomially bounded y, $\mathrm{KT}(G'_y(x)) < |G'_y(x)|^\epsilon$. Thus L is a statistical test that accepts many random strings of length $|x|^{O(1)}$, but rejects all pseudorandom strings. As in [33], this gives us a probabilistic oracle machine M using L that distinguishes G_y from the uniform distribution. We then apply Theorem 22. □

4.2 Hardness Results

We now apply Theorem 23 to obtain our hardness results for R_{KT}. As will be clear from the proofs, R_{KT} can be replaced by any suitably dense language containing no strings of KT-complexity less than n^ϵ for some $\epsilon > 0$, e.g., MCSP.

We first consider the Discrete Logarithm Problem, which takes as input a triple (p, g, z) where p is a prime number, $0 < g < p$, and $0 < z < p$, and outputs a positive integer i such that $g^i \equiv z \bmod p$, or 0 if there is no such i. We have:

Theorem 24 *The Discrete Logarithm Problem is computable in* $\mathrm{BPP}^{R_{\mathrm{KT}}}$.

Proof. On input (p, g, z), we first check if p is prime [2].

Let n be the number of bits in p, and y denote the pair (p, g). Consider the function $f(y,x) = g^x \bmod p$. Theorem 23 with $L = R_{\mathrm{KT}}$ gives 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 $g^i \equiv g^x \bmod p$.

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 at random 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 $z \equiv g^i \bmod p$.

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. □

We are not able to improve our reduction from a BPP-reduction to a ZPP-reduction. I.e., we know of no analog of Lemma 16 for KT-complexity. 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, we do have ZPP-like behavior, since we can check whether the number i we obtain 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 proof of Theorem 24 relies on the random self-reducibility of the Discrete Log Problem. In the terminology of Section 4.1, this allows us to consider f^{-1} as the problem h that reduces to R_{KT}. The next problem we consider is a first example where h differs from f^{-1}.

Theorem 25 *Integer Factorization is computable in* $\mathrm{ZPP}^{R_{\mathrm{KT}}}$.

Proof. Consider Rabin's candidate one-way function $f(y,x) = x^2 \bmod y$, where $0 \leq x < y$. Theorem 23 with $L = R_{\mathrm{KT}}$ gives us a probabilistic polynomial-time procedure with oracle access to R_{KT} that, for any fixed y, finds an inverse of $f(y,x)$ for a fraction at least $1/|y|^{O(1)}$ of the x's with $0 \leq x < y$. Rabin [32] showed how to use such a procedure and some randomness to efficiently find a nontrivial factor of y in case y is composite. This leads to a $\mathrm{BPP}^{R_{\mathrm{KT}}}$ algorithm for factoring. Since primality is in P [2], we can avoid errors and obtain the promised $\mathrm{ZPP}^{R_{\mathrm{KT}}}$ factoring algorithm. □

Since Ajtai's seminal paper [3], several worst-case to average-case connections have been established for lattice problems. We will exploit these next.

We first review some lattice terminology. We refer to [11] for more background. Given a set B of n linearly independent vectors $b_1, \ldots, b_n$ over $\mathbb{R}^n$, the set $L(B) = \{\sum_{i=1}^n z_i b_i \mid z_i \in \mathbb{Z}\}$ is called the lattice spanned by the basis B. We consider the usual notion of length of a vector v, $|v| = \sqrt{\sum v_i^2}$, and define the length of a set of vectors as the length of a longest vector in that set. For a lattice $L(B)$, $\lambda_i(B)$, $1 \leq i \leq n$, denotes the length of a shortest set of i linearly independent vectors from $L(B)$. The covering radius of $L(B)$, $\rho(B)$, is defined as the smallest ρ such that the union of all spheres of radius ρ centered at the points of $L(B)$ covers the entire space $\mathbb{R}^n$.

Applying our technique to Ajtai's worst-case to average-case connections and their subsequent improvements and extensions [13, 30], we obtain the following hardness results for R_{KT}.

Theorem 26 *For every $\epsilon > 0$, we can solve each of the following problems in* $\mathrm{BPP}^{R_{\mathrm{KT}}}$*: Given a basis $B \subseteq \mathbb{Z}^n$ (and a vector $t \in \mathbb{Z}^n$), find*

- *(Shortest Independent Vector Problem) a set of n linearly independent vectors in $L(B)$ of length at most $n^{3+\epsilon} \cdot \lambda_n(B)$,*

- *(Shortest Basis Problem) a basis for $L(B)$ of length at most $n^{3.5+\epsilon} \cdot \lambda_n(B)$,*
- *(Length of Shortest Vector Problem) a value $\tilde{\lambda}$, s.t. $\lambda_1(B) \leq \tilde{\lambda} \leq \omega(n^{3.5}\log n) \cdot \lambda_1(B)$,*
- *(Unique Shortest Vector Problem) a shortest vector in $L(B)$ in case $\lambda_2(B) > n^{4+\epsilon} \cdot \lambda_1(B)$,*
- *(Closest Vector Problem) a vector $u \in L(B)$ such that $|u - t| \leq n^{3.5+\epsilon} \cdot \lambda_n(B)$, and*
- *(Covering Radius Problem) a value $\tilde{\rho}$, s.t. $\rho(B) \leq \tilde{\rho} \leq \omega(n^{2.5}\log n) \cdot \rho(B)$.*

The approximation factors in Theorem 26 represent the current state-of-the-art but are likely to be improved in the future.

Proof of Theorem 26. In order to construct the reduction for SIVP, we will use the following worst-case to average-case connection.

Theorem 27 ([3, 13]) *For any $\epsilon > 0$, there exist functions $q(n) = n^{O(1)}$ and $m(n) = n\log^2 q$ with q a power of 2 such that the following holds. For every $c > 0$, if there is probabilistic algorithm A, such that, with probability at least $1/n^c$ (over $M \in \mathbb{Z}_q^{n\times m}$ and over the random choices of A), $A(M)$ outputs a nonzero vector $u \in \mathbb{Z}^m$ such that $|u| \leq m$ and $Mu \equiv 0 \bmod q$, then there exists a* BPP^A-*algorithm A' that given any basis $B \subseteq \mathbb{Z}^n$, outputs with high probability a set of n linearly independent vectors in $L(B)$ of length at most $n^{3+\epsilon} \cdot \lambda_n(B)$.*

Let $q(n)$ and $m(n)$ be as in Theorem 27. Consider the collection of functions $f_{q,m} : \mathbb{Z}_q^{n\times m} \times \{0,1\}^m \mapsto \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$ defined by $f_{q,m}(M, v) = (M, Mv \bmod q)$. Letting $y = (q, m)$ and $x = (M, v)$, the function $f_y(x)$ is computable uniformly in time $|x|^{O(1)}$. By Theorem 23, there exists a polynomial-time probabilistic algorithm N with oracle access to R_{KT} that computes a preimage of $f_{q,m}(M, v)$ for at least a $1/n^{O(1)}$ fraction of the inputs (M, v).

For any fixed M, f maps to at most q^n different values. Thus there can be at most q^n vectors v, such that $f(M, v) \neq f(M, v')$ for all $v' \neq v$. In other words, for a fraction at least $1 - q^n/2^m > 1 - O(1/2^n)$ of the vectors v, there is a $v' \neq v$ with $f(M, v') = f(M, v)$. Therefore, if we pick M and v uniformly at random, and run N to invert $f(M, Mv \bmod q)$, with probability at least $1/n^{O(1)}$ we obtain a vector $v' \in \mathbb{Z}^m$ such that $v' \neq v$ and $Mv \equiv Mv' \bmod q$.

Setting $u = v - v'$ yields a nonzero vector in $\mathbb{Z}^m$ satisfying $|u| \leq m$ and $Mu \equiv 0 \bmod q$. This vector u is the output of the algorithm A on input M. Applying Theorem 27 to A yields the BPP$^{R_{\mathrm{KT}}}$ algorithm for SIVP.

The reductions for SBP, Unique-SVP, and CVP follow from the one for SIVP by arguments given in [13, 12]. The results for LSVP and CRP are obtained in a similar way based on the variants of the candidate one-way function f and the worst-case to average-case connection corresponding to Theorem 27 presented in [30]. □

4.3 R_{KT} versus NP

Given our inability to prove that R_{KT} is coNP-complete, one may wonder whether R_{KT} is in NP. If it is, then this would provide a dense combinatorial property in NP that is useful against P/poly, contrary to a conjecture of Rudich [34]. We can also show the following.

Theorem 28 *If R_{KT} is in* NP*, then* MA = NP.

Proof. It is shown in [17] 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 ones in [17]) can not be taken as evidence that $R_{\mathrm{KT}} \notin$ NP, since many conjecture that MA is equal to NP. However, it does show that proving R_{KT} in NP would require nonrelativizing proof techniques.

5 Open Problems

KT-complexity was introduced in [4] 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 $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions. This deserves further investigation.

Is there some sense in which KT complexity yields NP-complete sets? For instance, what can one say about $\{(x, y, i) : \mathrm{KT}(x|y) \leq i\}$? The result of [39] is intriguing in this light; in [39] 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 AC$^0[\oplus]$? Is PSPACE $\subseteq$ P$^{R_{\mathrm{KS}}}$? Does P$^{R_{\mathrm{K}}}$ contain any decidable set outside PSPACE? Does it contain *every* decidable set? We were able to show that for $\leq_{\mathrm{btt}}^{\mathrm{p}}$ and $\leq_{\mathrm{ctt}}^{\mathrm{p}}$ reductions every decidable set that reduces to R_{K} is in P, and for $\leq_{\mathrm{dtt}}^{\mathrm{p}}$ reductions that such decidable sets are in P/poly.

Acknowledgments

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 thank Jin-Yi Cai and Daniele Micciancio for clarifying some issues regarding lattice problems. We also acknowledge fruitful discussions with Steven Rudich, Valentine Kabanets, Manindra Agrawal, and Kolya Vereshchagin.

References

[1] M. Agrawal, E. Allender, R. Impagliazzo, R. Pitassi, and S. Rudich. Reducing the complexity of reductions. *Comp. Compl.*, 10:117–138, 2001.

[2] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. Available at: http://www.cse.iitk.ac.in/primality.pdf, 2002.

[3] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.

[4] E. Allender. When worlds collide: Derandomization, lower bounds, and Kolmogorov complexity. In *FST&TCS*, volume 2245 of *LNCS*, pages 1–15, 2001.

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

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

[7] P. Berman. Relationship between density and deterministic complexity of NP-complete languages. In *ICALP*, volume 62 of *LNCS*, pages 63–71, 1978.

[8] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13:850–864, 1984.

[9] H. Buhrman and E. Mayordomo. An excursion to the Kolmogorov random strings. *JCSS*, 54:393–399, 1997.

[10] H. Buhrman and L. Torenvliet. Randomness is hard. *SIAM J. Comput.*, 30:1485–1501, 2001.

[11] J.-Y. Cai. Some recent progress on the complexity of lattice problems. In *CCC*, pages 158–179, 1999.

[12] J.-Y. Cai. On the average-case hardness of CVP. In *FOCS*, pages 308–319, 2001.

[13] J.-Y. Cai and A. P. Nerurkar. An improved worst-case to average-case connection for lattice problems. In *FOCS*, pages 468–477, 1997.

[14] J. Feigenbaum and L. Fortnow. On the random-self-reducibility of complete sets. *SIAM J. Comput.*, 22:994–1005, 1993.

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

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

[17] R. Impagliazzo, V. Kabanets, and A. Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. In *CCC*, pages 2–12, 2001.

[18] R. Impagliazzo and A. Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *STOC '97*, pages 220–229, 1997.

[19] R. Impagliazzo and A. Wigderson. Randomness vs. time: de-randomization under a uniform assumption. In *FOCS*, pages 734–743, 1998.

[20] V. Kabanets. Easiness assumptions and hardness tests: Trading time for zero error. *JCSS*, 63(2):236–252, 2001.

[21] V. Kabanets and J.-Y. Cai. Circuit minimization problem. In *STOC*, pages 73–79, 2000.

[22] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 2002. To appear; a preliminary version appeared in STOC '99.

[23] K.-I. Ko. On the complexity of learning minimum time-bounded turing machines. *SIAM J. Comput.*, 20:962–986, 1991.

[24] M. Kummer. On the complexity of random strings. In *STACS*, volume 1046 of *LNCS*, pages 25–36, 1996.

[25] R. Ladner and N. Lynch. Relativization of questions about log space reducibility. *MST*, 10:19–32, 1976.

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

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

[28] M. Li and P. Vitanyi. *Introduction to Kolmogorov Complexity and its Applications.* Springer, 1993.

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

[30] D. Micciancio. Improved cryptographic hash functions with worst-case / average-case connection. In *STOC*, pages 609–618, 2002.

[31] N. Nisan and A. Wigderson. Hardness vs. randomness. *JCSS*, 49:149–167, 1994.

[32] M. Rabin. Digitalized signatures and public-key functions as intractible as factorization. MIT Tech. Report TR-212, 1979.

[33] A. A. Razborov and S. Rudich. Natural proofs. *JCSS*, 55:24–35, 1997.

[34] S. Rudich. Super-bits, demi-bits, and $\mathrm{N\tilde{P}}$/qpoly-natural proofs. In *RANDOM*, volume 1269 of *LNCS*, 1997.

[35] M. Sipser. A complexity theoretic approach to randomness. In *STOC*, pages 330–335, 1983.

[36] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. *JCSS*, 62:236–266, 2001.

[37] L. Trevisan. Construction of extractors using pseudorandom generators. *J. ACM*, 48(4):860–879, 2001.

[38] L. Trevisan and S. Vadhan. Pseudorandomness and average-case complexity via uniform reductions. In *CCC*, pages 129–138, 2002.

[39] U. V. Vazirani and V. V. Vazirani. A natural encoding scheme proved probabilistic polynomial complete. *TCS*, 24:291–300, 1983.

[40] O. Watanabe. A comparison of polynomial time completeness notions. *TCS*, 54:249–265, 1987.

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

[42] A. Yao. Theory and applications of trapdoor functions. In *FOCS*, pages 80–91, 1982.