Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 127 (2022)

Kolmogorov Complexity Characterizes Statistical Zero Knowledge*

Eric Allender
Rutgers University, NJ, USA

Shuichi Hirahara
National Institute of Informatics, Japan

Harsha Tirumala
Rutgers University, NJ, USA

Abstract

We show that a decidable promise problem has a non-interactive statistical zero-knowledge proof system if and only if it is randomly reducible via an honest polynomial-time reduction to a promise problem for Kolmogorov-random strings, with a superlogarithmic additive approximation term. This extends recent work by Saks and Santhanam (CCC 2022). We build on this to give new characterizations of Statistical Zero Knowledge SZK, as well as the related classes $\mathsf{NISZK_L}$ and $\mathsf{SZK_L}$.

2012 ACM Subject Classification Complexity Classes; Problems, reductions and completeness; Circuit complexity

Keywords and phrases Kolmogorov Complexity, Interactive Proofs

Funding *Eric Allender*: Supported in part by NSF Grants CCF-1909216 and CCF-1909683.
Shuichi Hirahara: Supported in part by JST, PRESTO Grant Number JPMJPR2024, Japan
Harsha Tirumala: Supported in part by NSF Grants CCF-1909216 and CCF-1909683.

1 Introduction

In this paper, we give the first non-trivial characterization of a computational complexity class in terms of reducibility to the Kolmogorov random strings.

Some readers may be surprised that this is possible. After all, the set of Kolmogorov random strings is undecidable, and undecidable sets typically do not figure prominently in complexity-theoretic investigations.[1] But what does it mean to be reducible to the Kolmogorov-random strings? Let us consider the prefix-free Kolmogorov complexity K (which is one of the most-studied types of Kolmogorov complexity), and recall that different universal Turing machines U give a slightly different Kolmogorov measure K_U. Then if we say "A is reducible to the K-random strings" we probably mean that A is reducible to the K_U random strings, no matter which universal machine U we are using. But it turns out that the class of languages that can be solved in polynomial time with an oracle that returns $K_U(q)$ for any query q—*regardless* of which universal machine U is used—is a complexity class that contains NEXP and lies in $\mathsf{EXPSPACE}$ [27, 13, 35].[2] There has been substantial interest in obtaining a precise understanding of which problems can be reduced in this way to the Kolmogorov complexity function under different notions of reducibility [2, 3, 9, 7, 8, 12, 13, 14, 24, 27, 36, 35, 38, 40, 53], but until now, no previously studied

* A preliminary version of this work appeared as [19].

[1] We do wish to highlight the recent work of Ilango, Ren, and Santhanam [44], who related the existence of one-way functions to the *average case* complexity of computing Kolmogorov complexity.

[2] More specifically, it is shown in [13] that all decidable sets with this property lie in $\mathsf{EXPSPACE}$, and it is shown in [27] that there are no undecidable sets with this property. Hirahara shows in [36] that every set in $\mathsf{EXP}^{\mathsf{NP}}$ (and hence in NEXP) has this property.

ISSN 1433-8092

complexity class has been characterized in this way, with the exception of P [8, 53]. (The characterizations of P obtained in this way can be viewed as showing that certain limited polynomial-time reductions are useless when using the Kolmogorov complexity function as an oracle.)

Faced with this lack of success, it was proposed in [3, Open Question 4.8] that a more successful approach might be to consider reductions to *approximations* to the Kolmogorov complexity function. Saks and Santhanam [53] took the first significant step in this direction, by showing the following results:

▶ **Theorem 1** (Saks & Santhanam [53]). **1.** *Although (by the work of Hirahara [36]) every language in* $\mathsf{EXP}^{\mathsf{NP}}$ *is reducible in deterministic polynomial time to any function that differs from* K *by at most an additive* $O(\log n)$ *term, no decidable language outside of* P *is reducible to all approximations to* K *that differ by an error margin* $e(n) = \omega(\log n)$ *via an "honest" deterministic polynomial-time nonadaptive reduction.*

2. *Although (by the work of Hirahara [35]) every language in* NEXP *is reducible via randomized nonadaptive reductions to any function that differs from* K *by at most an additive* $O(\log n)$ *term, no decidable language outside of* $\mathsf{AM} \cap \mathsf{coAM}$ *is reducible to all approximations to* K *that differ by an error margin* $e(n) = \omega(\log n)$ *via an "honest" probabilistic polynomial-time nonadaptive reduction.*

3. *No decidable language outside of* SZK *is randomly m-reducible to each* $\omega(\log n)$ *approximation to the* K*-random strings.*[3]

This is not the first time that the complexity class SZK (for *Statistical Zero Knowledge* has arisen in the context of investigations relating to Kolmogorov complexity. In particular, SZK and its "non-interactive" subclass NISZK have been studied in connection with a version of time-bounded Kolmogorov complexity, which in turn is studied because of its connection with the Minimum Circuit Size Problem (MCSP) [11, 14]. These problems lie at the heart of what has come to be called *meta-complexity*: the study of the computational difficulty of answering questions about complexity.

Allender [2] proposed an intriguing research program towards the $\mathsf{P} = \mathsf{BPP}$ conjecture. The class P can be characterized as the class of languages reducible to the set of Kolmogorov-random strings under polynomial-time disjunctive truth-table reductions [8]. Similarly, he conjectured that BPP can also be characterized by polynomial-time truth-table reductions to the set of Kolmogorov-random strings, and envisioned that such a completely new characterization of complexity classes would give us new insights into BPP, especially from the perspective of computability theory. However, his conjecture was refuted by Hirahara [36] under a plausible complexity-theoretic assumption.

In this paper, we show that SZK, NISZK and their logspace variants SZK_L and $\mathsf{NISZK}_\mathsf{L}$ can be characterized by reductions to approximations to the Kolmogorov complexity function. More specifically, we define a promise problem $\widetilde{R}_K$ whose YES instances are strings of high Kolmogorov complexity, and whose NO instances are strings with significantly lower Kolmogorov complexity, and we show the following:

[3] Although the statement of this theorem in [53] does not mention "honesty," the proof requires that the approximation error be $\omega(\log n)$, where n is the *input* size, rather than the *query* size [54]. The proof of [53, Theorem 39] shows that, under this assumption, all queries on an input x can be assumed to have the same length, greater than $|x|$. (See Lemma 6 for a similar result.) An earlier version of our paper [18] mistakenly interpreted this as holding when the approximation error is a function of the *query* size, and consequently our main theorems were stated without assuming "honesty".

1. A decidable promise problem is randomly reducible to $\widetilde{R}_K$ via an honest polynomial time reduction if and only it is in NISZK. **(Theorem 15)**
2. A decidable promise problem is randomly reducible to $\widetilde{R}_K$ via an honest logspace or NC^0 reduction if and only it is in $\mathsf{NISZK_L}$. **(Theorem 33)**
3. Analogous characterizations of SZK and $\mathsf{SZK_L}$ are given in terms of probabilistic honest nonadaptive reductions. **(Theorems 29 and 35)**

We hope that our new characterization of these complexity classes will improve our understanding of zero knowledge interactive proof systems in the future. Zero knowledge interactive proof systems have many applications in cryptographic protocols, and they have been studied very widely. We refer the reader to the excellent survey by Vadhan for more background [56]. For our purposes, the complexity classes of interest to us (SZK, NISZK, $\mathsf{SZK_L}$, and $\mathsf{NISZK_L}$) can be defined in terms of their complete problems. But first, we need to define some basic notions and provide some background.

2 Preliminaries

We assume familiarity with basic complexity classes such as P, L, and AC^0; we view these as classes of *functions*, as well as of *languages*. We also will refer to the class of functions computed in NC^0, where each output bit depends on at most $O(1)$ input bits. For circuit complexity classes such as NC^0, and AC^0, by default we assume that the circuit families are "First-Order-uniform" as discussed in [5, 22, 45]. This coincides with Dlogtime-uniform AC^0, and what one might call "Dlogtime-uniform AC^0-uniform" NC^0. (We refer the reader to [58] for more background on circuit uniformity.) When we need to refer to *nonuniform* circuit complexity, we will be explicit.

All of these classes give rise to restrictions of Karp reducibility $\leq_{\mathrm{m}}^{\mathsf{P}}$, such as $\leq_{\mathrm{m}}^{\mathsf{L}}, \leq_{\mathrm{m}}^{\mathsf{AC}^0}$, and $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$. We will also discuss *projections* ($\leq_{\mathrm{m}}^{\mathsf{proj}}$), which are $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$ reductions in which each output bit depends on at most one input bit. Thus projections are computed by circuits consisting of constants, wires, and NOT gates.

For any class of functions $\mathcal{C}$ and type of reducibility r (such as m-reducibility, truth-table reducibility, Turing reducibility, or other notions considered in this paper) if there is some $\epsilon > 0$ such that all queries made by the $\leq_r^{\mathcal{C}}$ reduction on inputs of length n have length at least n^ϵ, the reduction is said to be "honest", and we use the notation $\leq_{hr}^{\mathcal{C}}$ to denote this.

A *promise problem* A is a pair of disjoint sets (Y_A, N_A) of YES instances and NO instances, respectively. A *solution* to a promise problem is any set B such that $Y_A \subseteq B$ and $N_A \subseteq \overline{B}$. A *don't-care instance* of A is any string that is not in $Y_A \cup N_A$. A *language* can be viewed as a promise problem that has no don't-care instances.

We say that a promise problem $A = (Y, N)$ is *decidable* if Y and N are decidable sets.[4] Note that the property of being a decidable promise problem is not the same as having a decidable solution: If $A = (Y, N)$ is decidable, then the set Y is a solution to A, and thus every decidable promise problem has a decidable solution, but the converse need not hold. For instance, if $B = (Y', N')$ with $Y' \subseteq Y$ and $N' \subseteq N$, then any solution to A is also a solution to B, and thus B has a decidable solution. Since there are uncountably many subsets of Y and N for any nontrivial promise problem, clearly not every promise problem with a decidable solution is decidable according to our definition. For complexity classes such as SZK, every promise problem in the class is $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$ reducible to a decidable promise problem,

[4] Such promise problems have also been called *totally decidable promise problems* [31].

and thus our main theorems (which are stated in terms of decidable promise problems) have wide applicability.

When defining reductions between two promise problems A and B, there are two options. Either

- for every solution S to B there is a reduction from A to S, or
- there is a reduction that correctly decides A when given any solution S for B as an oracle.

As it turns out, these two notions are equivalent [34, 50]. Thus we shall always use the second approach, when defining notions of reducibility between promise problems.

We assume that the reader is familiar with Kolmogorov complexity; more background on this topic can be found in references such as [48, 29]. Briefly, $K_U(x|y) = \min\{|d| : U(d,y) = x\}$, and $K_U(x) = K_U(x|\lambda)$ where λ denotes the empty string.[5] Although this definition depends on the choice of the Turing machine U, we pick some "universal" machine U' and define $K(x|y)$ to be $K_{U'}(x|y)$; for every machine U, there is a constant c such that $K(x|y) \le K_U(x|y) + c$. One important non-trivial fact regarding Kolmogorov complexity is known as *symmetry of information*:

▶ **Theorem 2.** *(Symmetry of Information)*

$$K(x,y) = K(x) + K(y|x) \pm O(\log(K(x,y))).$$

Let $\widetilde{R}_K$ be the promise problem $(Y_{\widetilde{R}_K}, N_{\widetilde{R}_K})$ where $Y_{\widetilde{R}_K}$ contains all strings y such that $K(y) \ge |y|/2$ and the NO instances $N_{\widetilde{R}_K}$ consists of those strings y where $K(y) \le |y|/2 - e(|y|)$ for some approximation error term $e(n)$, where $e(n) = \omega(\log n)$ and $e(n) = n^{o(1)}$. All of our theorems hold for any $e(n)$ in this range. We will sometimes assume that $e(n)$ is computable in AC^0, which is true for most approximation terms of interest.

Since the approximation error $e(n)$ is superlogarithmic, it is worth noting that $\widetilde{R}_K$ can be defined equivalently either in terms of prefix-free or plain Kolmogorov complexity (because these two measures are within an additive logarithmic term of each other).

Any *language* that is reducible to $\widetilde{R}_K$ via any of the reducibilities that we consider is decidable, by a theorem of [27]. However, it is not known whether this carries over in any meaningful way to promise problems.

The reader may wonder about the justification for the threshold $K(y) \ge |y|/2$ in the definition of $\widetilde{R}_K$. The following proposition indicates that, for large error bounds $e(n)$, using a larger threshold reduces to $\widetilde{R}_K$. Later, we show a related result for smaller thresholds.

▶ **Proposition 3.** *Let $A = (Y, N)$ be the promise problem where $Y = \{y : K(y) \ge t(|y|)\}$ for some AC^0-computable threshold $t(n) \ge \frac{n}{2}$, and where $N = \{y : K(y) \le t(|y|) - |y|^\epsilon\}$ for some $1 > \epsilon > 0$. Then $A \le_{\mathrm{m}}^{\mathsf{proj}} \widetilde{R}_K$.*

Proof. Let $\delta = \frac{\epsilon}{2}$. Given an instance y of length n (for all large n), in AC^0 we can find the least integer $i < n$ such that $2t(n) - n + 5\log n + (2(2n)^\delta - n^\epsilon) \le i \le 2t(n) - n - 6\log n$.

Let $z = y0^i$. Then $K(z) \le K(y) + 2\log i + O(1)$. Similarly, $K(y) \le K(z) + 2\log i + O(1)$, and hence $K(z) \ge K(y) - 2\log i - O(1)$.

Thus if $y \in Y$, then $K(z) \ge t(n) - 2\log i - O(1) > (t(n) - \frac{n}{2}) + \frac{n}{2} - 3\log n \ge \frac{n+i}{2} = \frac{|z|}{2}$. And if $y \in N$, then $K(z) \le t(n) - n^\epsilon + 2\log i + O(1) < (t(n) - \frac{n}{2}) + \frac{n}{2} - n^\epsilon + 2\log i + O(1) \le \frac{n+i}{2} - (n+i)^\delta = \frac{|z|}{2} - |z|^\delta < \frac{|z|}{2} - e(|z|)$.

[5] This is actually the definition of so-called "plain" Kolmogorov complexity, although the letter K is traditionally used for the "prefix-free" Kolmogorov complexity. These two measures differ by at most a logarithmic term, and our theorems hold for either measure. For simplicity, we have presented the simpler definition.

Thus $y \in Y$ implies $z \in Y_{\widetilde{R}_K}$ and $y \in N$ implies $z \in N_{\widetilde{R}_K}$. ◀

Randomized reductions play a central role in the results that we will be presenting. Here is the basic definition:

▶ **Definition 4.** *A promise problem $A = (Y, N)$ is $\leq_{\mathrm{m}}^{\mathsf{RP}}$-reducible to $B = (Y', N')$ with threshold θ if there is a polynomial p and a deterministic Turing machine M running in time p such that*

- *$x \in Y$ implies $\Pr_{r\in\{0,1\}^{p(|x|)}}[M(x, r) \in Y'] \geq \theta$.*
- *$x \in N$ implies $\Pr_{r\in\{0,1\}^{p(|x|)}}[M(x, r) \in N'] = 1$.*

If there is some $\epsilon > 0$ such that, for every x and every r of length $p(|x|)$, $M(x, r)$ has length $\geq |x|^\epsilon$, then we say that M computes an "honest" reduction, and we write $A \leq_{\mathrm{hm}}^{\mathsf{RP}} B$.

Randomized reductions were introduced by Adleman and Manders, as a probabilistic generalization of $\leq_{\mathrm{m}}^{\mathsf{P}}$ reducibility[6] [1]. They used the threshold $\theta = \frac{1}{2}$. One of the most important applications of randomized reductions is the theorem of Valiant and Vazirani [57], where they showed that SAT reduces to Unique Satisfiability (USAT) via a randomized reduction, with threshold $\theta = \frac{1}{4n}$.[7] The reader may expect that—as is so often the case with probabilistic notions in computational complexity theory—the choice of threshold is arbitrary, and can be changed with no meaningful consequences. However, this does not appear to be true; we refer the reader to the work of Chang, Kadin, and Rohatgi [28] for a discussion of this point. As they point out, different thresholds are appropriate in different situations. If $A \leq_{\mathrm{m}}^{\mathsf{RP}} B$ with threshold $\frac{1}{4n}$ (for instance), where the set $\mathrm{OR}_B = \{(x_1, \ldots, x_k) : \exists i, x_i \in B\} \leq_{\mathrm{m}}^{\mathsf{P}} B$, then it is indeed true that $A \leq_{\mathrm{m}}^{\mathsf{RP}} B$ with threshold $1 - \frac{1}{2^n}$ [28]. But Chang, Kadin, and Rohatgi point out that it is far from clear that USAT has this property. We are concerned here with problems that are $\leq_{\mathrm{hm}}^{\mathsf{RP}}$-reducible to $\widetilde{R}_K$; just as in the case with randomized reductions to USAT, we must be careful about which threshold θ we choose. For the remainder of this paper, we will use the threshold $\theta = 1 - \frac{1}{n^{\omega(1)}}$. (For a discussion of why we select this threshold, see Remark 17.)

The following proposition is the counterpart to Proposition 3, for thresholds smaller than $\frac{n}{2}$.

▶ **Proposition 5.** *Let $A = (Y, N)$ be the promise problem where $Y = \{y : K(y) \geq t(|y|)\}$ for some polynomial-time computable threshold $t(n) \leq \frac{n}{2}$, and where $N = \{y : K(y) \leq t(|y|) - |y|^\epsilon\}$ for some $1 > \epsilon > 0$. Then $A \leq_{\mathrm{hm}}^{\mathsf{RP}} \widetilde{R}_K$.*

Proof. Given an instance y of length n (for all large n), in polynomial time we can find the least integer $i < n$ such that $2t(n) - 2n^\epsilon + 2e(3n) + 4\log n \leq i \leq 2t(n) - e(n) - 2c\log n$ (for a constant c that will be picked later).

Pick a random string r of length n. Let $z = yr0^i$. Then $K(z) \leq K(y) + 2\log i + |r|$. Also, by symmetry of information, $K(z) \geq K(yr0^i|y0^i) + K(y0^i) - c'\log n$ (for some fixed constant c', and hence with probability at least $1 - \frac{1}{n^{\omega(1)}}$, $K(z) \geq (n - \frac{e(n)}{2}) + K(y) - c\log n$ (for some fixed c, which is the constant c that we use above in defining i).

Thus if $y \in Y$, then with high probability $K(z) \geq t(n) + (n - \frac{e(n)}{2}) - c\log n > n + \frac{i}{2} = \frac{|z|}{2}$. And if $y \in N$, then $K(z) \leq (t(n) - n^\epsilon) + 2\log i + |r| \leq n + \frac{i}{2} - e(3n) \leq \frac{|z|}{2} - e(|z|)$.

Thus $y \in Y$ implies $z \in Y_{\widetilde{R}_K}$ (with probability $\geq 1 - \frac{1}{n^{\omega(1)}}$), and $y \in N$ implies $z \in N_{\widetilde{R}_K}$. ◀

---

[6] We assume that the reader is familiar with Karp reducibility $\leq_{\mathrm{m}}^{\mathsf{P}}$.

[7] Recently, there have also been several papers showing that certain meta-complexity-theoretic problems are NP-complete under randomized reductions, including [10, 37, 41, 42, 43, 49, 51].

We will also need the following lemma, which states that short queries to $\widetilde{R}_K$ can be replaced by (longer) padded queries. Since $\widetilde{R}_K$ is defined so as to distinguish between strings of length n having Kolmogorov complexity $\geq n/2$ and those with complexity $\leq n/2-\omega(\log n)$, the idea is to pad the (short) query with a string that has complexity around half of its length — with some room to adjust for the difference needed to preserve the Yes and No instances.

▶ **Lemma 6** (Query padding). *Let $\widetilde{R}_K(g)$ denote the parameterized version of $\widetilde{R}_K$ with Yes instances y satisfying $K(y) \geq |y|/2$ and No instances satisfying $K(y) \leq |y|/2 - g(|y|)$. If $g(n) = \omega(\log n)$ is nondecreasing and computable in AC^0 and $A \leq^{\mathsf{RP}}_{\mathrm{hm}} \widetilde{R}_K(g)$, then for some $\delta > 0$, $A \leq^{\mathsf{RP}}_{\mathrm{hm}} \widetilde{R}_K(2g(n^\delta)/3)$ via a reduction in which all queries on input x have the same length.*

Proof. If $A \leq^{\mathsf{RP}}_{\mathrm{hm}} \widetilde{R}_K(g)$ via a reduction computable in time $p(n)$ where each query has length at least n^ϵ, consider the reduction that replaces each query q of length k by queries of the form $qy = qr0^{\frac{m-k}{2}-a(n)}$ where $m = p(n)$ and $r \in \{0,1\}^{\frac{m-k}{2}+a(n)}$ is sampled uniformly at random. (Here, $a(n)$ is a function that will be specified below.) Pick δ so that $p(n)^\delta < n^\epsilon$. We recall that by the Symmetry of Information theorem :

$$K(q) + K(y|q) - s\log m \leq K(qy) \leq K(q) + K(y|q) + s\log m$$

for some constant $s > 0$.

Case 1 : $q \in Y_{\widetilde{R}_{K(g)}}$

Thus $K(q) \geq \frac{k}{2}$, and hence, if we set $b(n) = (\log(g(n^\epsilon)/\log n))\log n = \omega(\log n)$, then with probability at least $1 - \frac{1}{n^{\omega(1)}}$

$$K(qy) \geq K(q) + K(y|q) - s\log m \geq \frac{k}{2} + \frac{m-k}{2} + a(n) - b(n) - s\log m$$

where the second inequality holds with probability $1 - \frac{1}{n^{\omega(1)}}$ since there are at most $\frac{1}{n^{\omega(1)}}$ fraction of $y \in \{0,1\}^{\frac{m-k}{2}+a(n)}$ satisfying $K(y|q) \leq \frac{(m-k)}{2} + a(n) - b(n)$. Setting $a(n) = g(n^\epsilon)/4$ gives $K(qy) \geq \frac{m}{2}$ with probability at least $1 - \frac{1}{n^{\omega(1)}}$ for all large n.

Case 2 : $q \in N_{\widetilde{R}_{K(g)}}$

We have $K(q) \leq \frac{k}{2} - g(k) \leq \frac{k}{2} - g(n^\epsilon)$ and need to show that $K(qy) \leq \frac{m}{2} - 2g(m^\delta)/3$.

$$\begin{aligned} K(qy) &\leq K(q) + K(y|q) + s\log m \leq \frac{k}{2} - g(n^\epsilon) + \left(\frac{m-k}{2} + g(n^\epsilon)/4\right) + O(\log m) \\ &< \frac{m}{2} - g(n^\epsilon) + g(n^\epsilon)/3 < \frac{m}{2} - 2g(m^\delta)/3. \end{aligned}$$

◀

▶ **Corollary 7.** *For any of the honest probabilistic reductions to $\widetilde{R}_K$ that we consider in this paper, we may assume without loss of generality that, for each input x, all queries made by the reduction on input x have the same length.*

Proof. If A is reducible to $\widetilde{R}_K$ using some approximation error $e(n)$ with $e(n) = \omega(\log n)$ and $e(n) = n^{o(1)}$, then, by Lemma 6, it is also reducible to $\widetilde{R}_K$ using approximation error $\frac{2e(n^\delta)}{3}$, which also is $\omega(\log n)$ and $n^{o(1)}$ via a reduction with the desired characteristics. ◀

We will also need a "two-sided error" version of random reducibility, analogous to the relationship between RP and BPP.

▶ **Definition 8.** *A promise problem $A = (Y, N)$ is $\leq_{\mathrm{m}}^{\mathsf{BPP}}$-reducible to $B = (Y', N')$ with threshold $\theta > \frac{1}{2}$ if there is a polynomial p and a deterministic Turing machine M running in time p such that*

- *$x \in Y$ implies $\Pr_{r\in\{0,1\}^{p(|x|)}}[M(x, r) \in Y'] \geq \theta$.*
- *$x \in N$ implies $\Pr_{r\in\{0,1\}^{p(|x|)}}[M(x, r) \in N'] \geq \theta$.*

Similar to the definition of $\leq_{\mathrm{hm}}^{\mathsf{RP}}$, we say that $A \leq_{\mathrm{hm}}^{\mathsf{BPP}} B$ if M is honest.

The complexity classes SZK (Statistical Zero Knowledge) and NISZK (Non-Interactive Statistical Zero Knowledge) are defined in terms of interactive proof protocols (with a *Prover* interacting with a probabilistic polynomial-time *Verifier*, together with a *Simulator* that can produce a distribution on transcripts that is statistically close to the distribution on messages that would be exchanged by the prover and the verifier on YES instances. (See, e.g. [56, 33].) But for our purposes, it will suffice (and be simpler) to present alternative definitions of these classes, in terms of their standard complete problems.

▶ **Definition 9** (Promise-EA). *Let a circuit $C\colon \{0,1\}^m \to \{0,1\}^n$ represent a probability distribution X on $\{0,1\}^n$ induced by the uniform distribution on $\{0,1\}^m$. We define Promise-EA to be the promise problem*

$$Y_{\mathsf{EA}} = \{(C, k) \mid H(X) > k + 1\}$$
$$N_{\mathsf{EA}} = \{(C, k) \mid H(X) < k - 1\}$$

where $H(X)$ denotes the entropy of X.

▶ **Theorem 10** ([33]). *EA is complete for NISZK under honest $\leq_{\mathrm{m}}^{\mathsf{P}}$ reductions.*

We will actually take this as a definition; we say that (Y, N) is in NISZK if and only if $(Y, N) \leq_{\mathrm{m}}^{\mathsf{P}} \mathsf{EA}$.

▶ **Definition 11** (Promise-SD). *SD (Statistical Difference) is the promise problem*

$$Y_{\mathsf{SD}} = \left\{ (C, D) \;\middle|\; \Delta(C, D) > \frac{2}{3} \right\},$$
$$N_{\mathsf{SD}} = \left\{ (C, D) \;\middle|\; \Delta(C, D) < \frac{1}{3} \right\}.$$

where $\Delta(C, D)$ denotes the statistical distance between the distributions represented by the circuits C and D.

▶ **Theorem 12** ([52]). *SD is complete for SZK under honest $\leq_{\mathrm{m}}^{\mathsf{P}}$ reductions.*

Thus we will define SZK to be the class of promise problems (Y, N) such that $(Y, N) \leq_{\mathrm{m}}^{\mathsf{P}} \mathsf{SD}$.

We will also be making use of a restricted version of the NISZK-complete problem EA:

▶ **Definition 13** (Promise-EA′). *We define Promise-EA' to be the promise problem*

$$Y_{\mathsf{EA}'} = \{C \mid H(X) > n - 2\}$$
$$N_{\mathsf{EA}'} = \{C \mid |\mathrm{Supp}(X)| < 2^{n-n^{\epsilon}}\}$$

where C is a circuit $C\colon \{0,1\}^m \to \{0,1\}^n$ representing a probability distribution X on $\{0,1\}^n$ induced by the uniform distribution on $\{0,1\}^m$, and $\mathrm{Supp}(X)$ denotes the support of X, and ϵ is some fixed constant, $0 < \epsilon < 1$.

▶ **Lemma 14.** *EA' is complete for NISZK under honest $\leq_{\mathrm{m}}^{\mathsf{P}}$ reductions.*

Proof. Lemma 3.2 in [33] shows that the following promise problem A is complete for NISZK: All instances are of the form $(C, 1^s)$, where C is a circuit with m inputs and n outputs, representing a distribution (also denoted C) on $\{0,1\}^n$. $(C, 1^s)$ is a YES instance if C has statistical distance at most 2^{-s} from the uniform distribution on $\{0,1\}^n$. $(C, 1^s)$ is in the set of NO instances if the support of C has size at most 2^{n-s}. Furthermore, the reduction g from EA to A has the property that the parameter s is at least n^ϵ for some constant $\epsilon > 0$. Also, it is observed in Lemma 4.1 of [33] that the mapping $(C, 1^s) \mapsto (C, n-3)$ (i.e., the mapping that leaves the circuit C unchanged) is a reduction from A to EA. Combining these two results from [33] completes the proof of the lemma. ◀

3 A New Characterization of NISZK

We are now ready to present the characterization of NISZK by reductions to the set of Kolmogorov-random strings.

▶ **Theorem 15.** *The following are equivalent, for any decidable promise problem A:*
1. $A \in \mathsf{NISZK}$.
2. $A \leq^{\mathsf{RP}}_{\mathrm{hm}} \widetilde{R}_K$.
3. $A \leq^{\mathsf{BPP}}_{\mathrm{hm}} \widetilde{R}_K$.

Proof. In order to show that $A \in \mathsf{NISZK}$ implies $A \leq^{\mathsf{RP}}_{\mathrm{hm}} \widetilde{R}_K$, it suffices to reduce the NISZK-complete problem EA' to $\widetilde{R}_K$ (by Lemma 14).

Corollary 18 of [14] states that every promise problem in NISZK reduces to the problem of computing the time-bounded Kolmogorov complexity KT via a probabilistic reduction that makes at most one query along any computation path. Here we observe that the same approach can be used to obtain a $\leq^{\mathsf{RP}}_{\mathrm{hm}}$ reduction to $\widetilde{R}_K$.

Consider a probabilistic reduction that takes an instance C of EA' and constructs a string y that is the concatenation of t random samples from C (i.e., $y = C(r_1)C(r_2)\ldots C(r_t)$ for uniformly chosen random strings $r_1, \ldots, r_t$, for some polynomially-large t). Lemma 16 of [14] shows that, with probability exponentially close to 1, if C is a YES instance of EA', then the time-bounded Kolmogorov complexity $\mathsf{KT}(y)$ is greater than a threshold θ of the form $\theta = t(n-2) - t^{1-\alpha}$ for some constant $\alpha > 0$. (Briefly, this is because a good approximation to the entropy of a sufficiently "flat" distribution can be obtained by computing the KT complexity of a string composed of many random samples from the distribution [16].)

As in the argument of [14, Theorem 17], we can choose t to be an arbitrarily large polynomial n^k. Choosing k to be large enough (relative to $1/\alpha$, and also so that n^k is large relative to $|C|$), we have $\theta > n^k(n-3)$ for all large n, and hence for all large YES instances we have that, with probability exponentially close to 1, the string y satisfies $\mathsf{KT}(y) > n^k(n-3) = \ell - \ell^\delta$ for some $\delta < 1$, where $|y| = tn = \ell$. The focus of [14] was on the measure KT, but (as was previously observed in [4, Theorem 1]) the analysis in [14, Lemma 16] carries over unchanged to the setting of non-resource-bounded Kolmogorov complexity K. (That is, in obtaining the lower bound on $\mathsf{KT}(y)$, the probabilistic argument is just bounding the number of short descriptions, and not making use of the time required to build y from a description.) Thus, with high probability, the probabilistic routine, when given a YES instance of EA', produces a string y where $K(y) \geq |y| - |y|^\delta$.

On the other hand, if C is a NO instance, then the support of C has size at most 2^{n-n^ϵ}, and thus any string z in the support of C has $K(z|C) \leq n - n^\epsilon + O(1)$. Thus any string y of length $\ell = tn = n^{k+1}$ that is produced by M in this case has $K(y) \leq t(n-n^\epsilon) + |C| + O(1) = n^k(n-n^\epsilon) + |C| + O(1)$. Since $t = n^k$ was chosen to be large (with respect to the length of the input instance C), we may assume $|C| < n^{k+\epsilon} - 4n^k$. Thus if C is any large NO instance, we have $K(y) < n^k(n-4) = \ell - \ell^{\delta'}$ for some $\delta' > \delta$. To summarize, with probability 1, the probabilistic routine, when given a NO instance of EA', produces a string y where $K(y) \le |y| - |y|^{\delta'} \le (|y| - |y|^{\delta}) - |y|^{\rho}$ for some $\rho > 0$. We can now conclude that $\mathsf{EA}' \le^{\mathsf{RP}}_{\mathrm{hm}} \widetilde{R}_K$ by appealing to Proposition 3.

To complete the proof of the theorem, we need to show that if A is any decidable promise problem that has a randomized poly-time m-reduction ($\le^{\mathsf{BPP}}_{\mathrm{hm}}$) with error $1/n^{\omega(1)}$ to the promise problem $\widetilde{R}_K$ then $A \in \mathsf{NISZK}$. This was essentially shown by Saks and Santhanam [53, Theorem 39], but we present a complete argument here. Let M be the probabilistic machine that computes this $\le^{\mathsf{BPP}}_{\mathrm{hm}}$ reduction.

Let $y = f(x, r) \in \{0,1\}^m$ denote the output that M produces, where x is an instance of A and r denotes the randomness used in the reduction. By Corollary 7, we may assume that, for each x, all outputs of the form $f(x, r)$ have the same length. Given an $x \in \{0,1\}^n$, observe that there is a polynomial-sized circuit C_x such that $C_x(r) = f(x, r)$. According to the correctness of the reduction, we have

$$x \in Y_A \Rightarrow \Pr_r[M(x,r) \in Y_{\widetilde{R}_K}] \ge 1 - 1/n^{\omega(1)} \text{ and}$$

$$x \in N_A \Rightarrow \Pr_r[M(x,r) \in N_{\widetilde{R}_K}] \ge 1 - 1/n^{\omega(1)}.$$

In other words, if x is a YES instance, then $K(y) \ge |y|/2$ with probability at least $1 - 1/n^{\omega(1)}$ and if x is a NO instance, then $K(y) \le |y|/2 - e(|y|)$ with probability at least $1 - 1/n^{\omega(1)}$. (Recall that $e(n)$ is the error term in the approximation $\widetilde{R}_K$.) We will now show that there is an entropy threshold that separates these two distributions, which will provide an NISZK upper bound on resolving A.

▷ Claim 16. The following holds for all large strings x. If x is a YES instance, then the entropy of the distribution $C_x(r)$ is at least $m/2 - e(m)/2 + 1$ and if x is a NO instance, then the entropy of $C_x(r)$ is at most $m/2 - e(m)/2 - 1$.

We first show that if the claim holds, then $A \in \mathsf{NISZK}$. Let $k = m/2 - e(m)/2$. The reduction given above reduces membership in A to the Entropy Approximation (EA) problem on the circuit description C_x with threshold k. Given x, we can compute the map $x \mapsto C_x$ in time $n^{O(1)}$. Recall that EA is complete for NISZK. Since NISZK is closed under $\le^{\mathsf{P}}_{\mathrm{m}}$ reductions, we can conclude that $A \in \mathsf{NISZK}$.

Proof of Claim 16. Assume the claim is false, and let x be the lexicographically first string that violates the above claim (for some length n). Since the reduction is a computable function, and since A is a decidable promise problem, $K(x) = O(\log n)$. We have the following two cases to consider:

Case 1 — x is a YES instance: From the correctness of the reduction we have that with probability $1 - 1/n^{\omega(1)}$ the output y is a string with Kolmogorov complexity at least $|m|/2$. Since x is a violator, we have $H(C_x(r)) < k + 1 = m/2 - e(m)/2 + 1$.

First, we present some intuition. On one hand, the distribution $C_x(r)$ has large enough probability mass on the high-complexity strings (because the reduction succeeds). On the other hand, we have that since x is a low-complexity string itself, the elements of $C_x(r)$ with highest mass can be identified by short descriptions. This leads to a contradiction of simultaneously having large enough mass on the low and the high K-complexity strings.

Now, we present a more detailed argument. Let t be the entropy of the distribution $C_x(r)$. Thus, for large x, $t + O(\log m) < t + e(m)/2 - 1 < m/2$. Let $Y = \{y_1 \dots y_{2^{t+\log m}}\}$ be the heaviest elements (in terms of probability mass) of $C_x(r)$ in decreasing order. (Note that $\Pr[y_{2^{t+\log m}}] \leq \frac{1}{2^{t+\log m}}$.) Conditioned on x, the K complexity of any of these strings y_i is at most $t+O(\log m)$. Since $K(x) = O(\log n) = O(\log m)$, we have $K(y_i) \leq t+O(\log m) < m/2$. Next, we will show that there is at least mass $\frac{1}{m}$ on these strings within $C_x(r)$. This will contradict the correctness of the reduction for $x \in L$ since it cannot output strings with K complexity at most $|m|/2$ with probability $1/n^{\Omega(1)}$.

Assume not, i.e., the mass on elements of Y is at most $\frac{1}{m}$. Observe that elements of $\mathrm{Supp}(C_x(r)) - Y$ have mass no more than $2^{-(t+\log m)}$ each. Thus $t = H(C_x(r)) > \sum_{y \notin Y} \Pr[y] \log(\frac{1}{\Pr[y]}) > \sum_{y \notin Y} \Pr[y](t+\log m) > (1-1/m)(t+\log m) > t - t/m + \log m > t - \frac{1}{2} + \log m > t$, which is a contradiction.

Case 2 — x is a NO instance: From the correctness of the reduction we have that with probability at least $1 - 1/n^{\omega(1)}$ the output $f(x,r)$ is a string with K complexity at most $m/2 - e(m)$. Since x is a violator, we also have $H(C_x(r)) > k - 1 = m/2 - e(m)/2 - 1$.

We claim that the following holds:

$$\Pr_{y \sim f(x,r)}[K(y) > m/2 - e(m)] \geq 1/m.$$

Assume not. Then, since

- there are at most $2^{m/2-e(m)}$ strings y with $K(y) \leq m/2 - e(m)$, and
- entropy is maximized when probabilities are flat within a partition, and
- any element in the support has probability at least $\frac{1}{2^m}$

it follows that the entropy of $f(x,r)$ is at most $(1/m)(m) + (1-1/m)(m/2 - e(m)) \leq m/2 - e(m) + 1 < m/2 - e(m)/2 - 1$, which contradicts the lower bound on the entropy of $f(x,r)$ above.

Since the claim holds, with probability at least $1/m$ the output of the reduction is not an element of the set $N_{\widetilde{R}_K}$. Thus, the reduction fails with probability $1/n^{\Omega(1)}$. ◁

This completes the proof of Theorem 15. ◀

▶ Remark 17. The proof of the preceding theorem illustrates why we define the error threshold in our randomized reductions to be $\frac{1}{n^{\omega(1)}}$. If we assumed that A were $\leq_{\mathrm{hm}}^{\mathsf{BPP}}$-reducible to $\widetilde{R}_K$ with an inverse polynomial threshold (say $q(n)^{-1}$), then by Corollary 7 we may assume that the length of each output produced has length $Q(n) = \omega(q(n))$ (by padding with some uniformly-random bits). For strings x that are NO instances of A, when the reduction to $\widetilde{R}_K$ fails with probability $1/q(n)$, our calculation of the entropy of C_x will involve a term of $\frac{1}{q(n)}Q(n)$ (because the queries made in this case can have nearly $Q(n)$ bits of entropy). This is more than the entropy gap between the distributions corresponding to the YES and NO outputs.

▶ Remark 18. Although our focus in this paper is on $\widetilde{R}_K$, we note that one can also define an analogous problem $\widetilde{R}_{\mathsf{KT}}$ in terms of the time-bounded measure KT. The approach used in Theorem 15 also shows that every problem in NISZK is $\leq_{\mathrm{hm}}^{\mathsf{BPP}}$ reducible to $\widetilde{R}_{\mathsf{KT}}$, although we do not know how to show hardness under $\leq_{\mathrm{hm}}^{\mathsf{RP}}$ reductions. (A random sample from the low-entropy distribution is guaranteed to *always* have low K-complexity, but the tools of [14, 16] only guarantee that the output has low KT-complexity *with high probability.*)

4 More Powerful Reductions

Just as $\leq_{\mathrm{m}}^{\mathsf{RP}}$ and $\leq_{\mathrm{m}}^{\mathsf{BPP}}$ reducibilities generalize the familiar $\leq_{\mathrm{m}}^{\mathsf{P}}$ (Karp) reducibility to the setting of probabilistic computation, so also are there probabilistic generalizations of deterministic non-adaptive reductions (also known as truth-table reductions). Before presenting these probabilistic generalizations, let us review the previously-studied deterministic non-adaptive reducibilities that are relevant for this investigation. Some of them may be unfamiliar to the reader.

Ladner, Lynch, and Selman [47] considered several possible ways to define polynomial-time versions of the truth-table reducibility that had been studied in computability theory, before settling on the definition of $\leq^{\mathsf{P}}_{\mathrm{tt}}$ reducibility below. They considered only reductions between *languages*; the corresponding generalization to *promise problems* is due to [52]. In order to state this generalization formally, let us define the characteristic function χ_A of a promise problem $A = (Y, N)$ to take on the following values in three-valued logic:

- If $x \in Y$, then $\chi_A(x) = 1$.
- If $x \in N$, then $\chi_A(x) = 0$.
- If $x \notin (Y \cup N)$, then $\chi_A(x) = *$.

A Boolean circuit with n variables, when given an assignment in $\{0, 1, *\}^n$, can be evaluated using the usual rules of three-valued logic. (See, e.g., [52, Definition 4.6].)

▶ **Definition 19.** *Let $A = (Y, N)$ and $B = (Y', N')$ be promise problems. We say $A \leq^{\mathsf{P}}_{\mathrm{tt}} B$ if there is a function f computable in polynomial time, such that, for all x, $f(x)$ is of the form $(C, z_1, z_2, \ldots, z_k)$ where C is a Boolean circuit with k input variables, and $(z_1, \ldots, z_k)$ is a list of queries, with the property that*

- *If $x \in Y$, then $C(\chi_B(z_1), \ldots, \chi_B(z_k)) = 1$.*
- *If $x \in N$, then $C(\chi_B(z_1), \ldots, \chi_B(z_k)) = 0$.*

This definition ensures that the circuit C, viewed as an ordinary circuit in 2-valued logic, correctly decides membership for all $x \in (Y \cup N)$ when given any solution S for B as an oracle.

If C is a Boolean formula, instead of a circuit, then one obtains the so-called "Boolean formula reducibility" (denoted by $A \leq^{\mathsf{P}}_{\mathrm{bf}} B$), which was discussed in [47] and studied further in [46, 26]. (See also [25, 6].)

▶ **Theorem 20.** $\mathsf{SZK} = \{A : A \leq^{\mathsf{P}}_{\mathrm{bf}} \mathsf{EA}\} = \{A : A \leq^{\mathsf{P}}_{\mathrm{hbf}} \mathsf{EA}\}$.

Proof. $\mathsf{EA} \in \mathsf{NISZK} \subseteq \mathsf{SZK}$. Sahai and Vadhan [52, Corollary 4.14] showed that SZK is closed under NC^1-truth-table reductions, but the proof carries over immediately to $\leq^{\mathsf{P}}_{\mathrm{bf}}$ reductions. Thus $\{A : A \leq^{\mathsf{P}}_{\mathrm{bf}} \mathsf{EA}\} \subseteq \mathsf{SZK}$. The other inclusion was shown in [33, Proposition 5.4] (and the reduction to EA they present is honest). ◀

Notably, it is still an open question if SZK is closed under $\leq^{\mathsf{P}}_{\mathrm{tt}}$ reducibility.

Our characterization of SZK in terms of reductions to $\widetilde{R}_K$ relies on the following probabilistic generalization of $\leq^{\mathsf{P}}_{\mathrm{bf}}$:

▶ **Definition 21.** *Let $A = (Y, N)$ and $B = (Y', N')$ be promise problems. We say $A \leq^{\mathsf{BPP}}_{\mathrm{bf}} B$ with threshold $\theta > \frac{1}{2}$ if there are functions f and g computable in* **deterministic** *polynomial time, and a polynomial p, such that, for all x, $f(x)$ is a Boolean formula C (with $k = |x|^{O(1)}$ variables), with the property that*

- *If $x \in Y$, then $C(\chi_{g,B}(x, 1), \ldots, \chi_{g,B}(x, k)) = 1$,*
- *If $x \in N$, then $C(\chi_{g,B}(x, 1), \ldots, \chi_{g,B}(x, k)) = 0$,*

where

- $\chi_{g,B}(x, i) = 1$ *if* $\Pr_{r \in \{0,1\}^{p(|x|)}}[g(x, i, r) \in Y'] \geq \theta$
- $\chi_{g,B}(x, i) = 0$ *if* $\Pr_{r \in \{0,1\}^{p(|x|)}}[g(x, i, r) \in N'] \geq \theta$
- $\chi_{g,B}(x, i) = *$ *otherwise.*

Intuitively, $\leq_{\mathrm{bf}}^{\mathsf{BPP}}$ reductions generalize $\leq_{\mathrm{bf}}^{\mathsf{P}}$ reductions, in that the queries are now generated probabilistically, and the probability that any query returns a definite YES or NO answer is bounded away from $\frac{1}{2}$. Again, if all queries are of length at least n^{ϵ}, then we write $A \leq_{\mathrm{hbf}}^{\mathsf{BPP}} B$.

The following proposition is immediate from the definitions.

▶ **Proposition 22.** *If $A \leq_{\mathrm{hbf}}^{\mathsf{P}} B$ and $B \leq_{\mathrm{hm}}^{\mathsf{BPP}} C$ with threshold θ, then $A \leq_{\mathrm{hbf}}^{\mathsf{BPP}} C$ with threshold θ.*

▶ **Corollary 23.** $\mathsf{SZK} \subseteq \{A : A \leq_{\mathrm{hbf}}^{\mathsf{BPP}} \widetilde{R}_K\}$ *with threshold* $1 - \frac{1}{n^{\omega(1)}}$.

Proof. Immediate from Theorem 20 and Theorem 15. ◀

There are (at least) three other variants of probabilistic nonadaptive reducibility that we should mention. The first of these is the notion that goes by the name "nonadaptive BPP reducibility" or "randomized nonadaptive reductions" in work such as [53, 14, 23] and elsewhere.

▶ **Definition 24.** *Let $A = (Y, N)$ and $B = (Y', N')$ be promise problems. We say $A \leq_{\mathrm{tt}}^{\mathsf{BPP}} B$ if there are a function f computable in polynomial time and a polynomial p such that, for all x and all r of length $p(|x|)$, $f(x, r)$ is of the form $(C, z_1, z_2, \ldots, z_k)$ where C is a Boolean circuit with k input variables, and $(z_1, \ldots, z_k)$ is a list of queries, with the property that*

- *If $x \in Y$, then $\Pr_r[C(\chi_B(z_1), \ldots, \chi_B(z_k)) = 1] \geq \frac{2}{3}$.*
- *If $x \in N$, then $\Pr_r[C(\chi_B(z_1), \ldots, \chi_B(z_k)) = 0 \geq \frac{2}{3}$.*

(The threshold $\frac{2}{3}$ can be replaced by any threshold between n^{-k} and 2^{-n^k}, by the usual method of taking the majority vote of several independent trials.)

Saks and Santhanam showed that if $A \leq_{\mathrm{htt}}^{\mathsf{BPP}} \widetilde{R}_K$, then $A \in \mathsf{AM} \cap \mathsf{coAM}$ [53]. The most important ways in which $\leq_{\mathrm{bf}}^{\mathsf{BPP}}$ and $\leq_{\mathrm{tt}}^{\mathsf{BPP}}$ reducibility differ from each other, are (1) in $\leq_{\mathrm{bf}}^{\mathsf{BPP}}$ reducibility, the query evaluation is performed by a Boolean formula, instead of a circuit, and (2) in $\leq_{\mathrm{tt}}^{\mathsf{BPP}}$ reducibility, the circuit that is chosen to do the evaluation depends on the choice of random bits, whereas in $\leq_{\mathrm{bf}}^{\mathsf{BPP}}$ reducibility, the formula is chosen deterministically. Making different choices in these two dimensions gives rise to two other notions:

▶ **Definition 25.** *Let $A = (Y, N)$ and $B = (Y', N')$ be promise problems. We say $A \leq_{\mathrm{rbf}}^{\mathsf{BPP}} B$ if there are a function f computable in polynomial time and a polynomial p such that, for all x and all r of length $p(|x|)$, $f(x, r)$ is of the form $(C, z_1, z_2, \ldots, z_k)$ where C is a Boolean formula with k input variables, and $(z_1, \ldots, z_k)$ is a list of queries, with the property that*

- *If $x \in Y$, then $\Pr_r[C(\chi_B(z_1), \ldots, \chi_B(z_k)) = 1] \geq \frac{2}{3}$.*
- *If $x \in N$, then $\Pr_r[C(\chi_B(z_1), \ldots, \chi_B(z_k)) = 0] \geq \frac{2}{3}$.*

(The threshold $\frac{2}{3}$ can be replaced by any threshold between n^{-k} and 2^{-n^k}, simply by incorporating a Boolean formula that takes the majority vote of several independent trials.).

The notation $\leq_{\mathrm{rbf}}^{\mathsf{BPP}}$ is intended to suggest "random Boolean formula", since the Boolean formula is chosen randomly.

▶ **Definition 26.** *Let $A = (Y, N)$ and $B = (Y', N')$ be promise problems. We say $A \leq_{\mathrm{circ}}^{\mathsf{BPP}} B$ with threshold $\theta > \frac{1}{2}$ if there are functions f and g computable in* **deterministic** *polynomial time, and a polynomial p, such that, for all x, $f(x)$ is a Boolean circuit (with $k = |x|^{O(1)}$ variables), with the property that*

- *If $x \in Y$, then $C(\chi_{g,B}(x, 1), \ldots, \chi_{g,B}(x, k)) = 1$,*
- *If $x \in N$, then $C(\chi_{g,B}(x, 1), \ldots, \chi_{g,B}(x, k)) = 0$,*

where

- $\chi_{g,B}(x,i) = 1$ *if* $\Pr_{r\in\{0,1\}^{p(|x|)}}[g(x,i,r) \in Y'] \geq \theta$
- $\chi_{g,B}(x,i) = 0$ *if* $\Pr_{r\in\{0,1\}^{p(|x|)}}[g(x,i,r) \in N'] \geq \theta$
- $\chi_{g,B}(x,i) = *$ *otherwise.*

If the reduction is honest, we write $A \leq^{\mathsf{BPP}}_{\mathrm{hcirc}} B$.

We show in this paper that SZK is the class of problems $\leq^{\mathsf{BPP}}_{\mathrm{hbf}}$ reducible to $\widetilde{R}_K$. We are not able to show that the class of problems (honestly) $\leq^{\mathsf{BPP}}_{\mathrm{rbf}}$ reducible to $\widetilde{R}_K$ is contained in SZK, although we do observe that SZK is closed under this type of reducibility.

▶ **Theorem 27.** $\mathsf{SZK} = \{A : A \leq^{\mathsf{BPP}}_{\mathrm{rbf}} \mathsf{EA}\}$.

Proof. The inclusion of SZK in $\{A : A \leq^{\mathsf{BPP}}_{\mathrm{rbf}} \mathsf{EA}\}$ is immediate from Theorem 20. For the other direction, let $A \leq^{\mathsf{BPP}}_{\mathrm{rbf}} \mathsf{EA}$. Thus there are a function f computable in polynomial time, and a polynomial p such that, for all x and all r of length $p(|x|)$, $f(x,r)$ is of the form $(C, z_1, z_2, \ldots, z_k)$, where evaluating the Boolean formula $C(\chi_B(z_1), \ldots, \chi_B(z_k))$ gives a correct answer for all $x \in Y \cup N$ with error at most 2^{-n^2}. Here is a zero-knowledge interactive protocol for A. The verifier sends a random string r to the prover. The prover and the verifier can each compute $f(x,r) = (C, z_1, z_2, \ldots, z_k)$, and then (as in [52, Corollary 4.14]) compute an instance (D,E) of SD such that (D,E) is a YES instance of SD if $C(\chi_B(z_1), \ldots, \chi_B(z_k)) = 1$, and (D,E) is a NO instance of SD if $C(\chi_B(z_1), \ldots, \chi_B(z_k)) = 0$. The prover and the verifier can then run the SZK protocol for the SD instance (D,E). The verifier clearly accepts each YES instance with high probability, and cannot be convinced to accept any NO instance with more than negligible probability. The simulator, given input x, will generate the string r uniformly at random, and then compute $f(x,r)$ and compute the instance (D,E) as above, and then produce the transcript that is produced by the SD simulator on input (D,E). It is straightforward to observe that, if $x \in Y$, then this distribution is very close to the distribution induced by the honest prover and verifier. ◀

It is straightforward to observe that $\leq^{\mathsf{BPP}}_{\mathrm{tt}}$ and $\leq^{\mathsf{BPP}}_{\mathrm{rbf}}$ are transitive relations. It is not clear that $\leq^{\mathsf{BPP}}_{\mathrm{bf}}$ and $\leq^{\mathsf{BPP}}_{\mathrm{circ}}$ are transitive. But for promise problems that reduce to $\widetilde{R}_K$, a similar property holds.

▶ **Theorem 28.** *If* $A \leq^{\mathsf{BPP}}_{\mathrm{bf}} B$ *and* $B \leq^{\mathsf{BPP}}_{\mathrm{hbf}} \widetilde{R}_K$, *then* $A \leq^{\mathsf{BPP}}_{\mathrm{hbf}} \widetilde{R}_K$.

Proof. If $B \leq^{\mathsf{BPP}}_{\mathrm{bf}} \widetilde{R}_K$, then $B \in \mathsf{SZK}$ by Theorem 29. Since $A \leq^{\mathsf{BPP}}_{\mathrm{bf}} B \in \mathsf{SZK}$, it follows that $A \leq^{\mathsf{BPP}}_{\mathrm{rbf}} B \leq^{\mathsf{BPP}}_{\mathrm{rbf}} \mathsf{EA}$ and hence (by Theorem 27) $A \in \mathsf{SZK}$. Thus (by Theorem29) $A \leq^{\mathsf{BPP}}_{\mathrm{hbf}} \widetilde{R}_K$. ◀

5 A New Characterization of SZK

▶ **Theorem 29.** *The following are equivalent, for any decidable promise problem A:*

1. $A \in \mathsf{SZK}$.
2. $A \leq^{\mathsf{BPP}}_{\mathrm{hbf}} \widetilde{R}_K$ *with threshold* $1 - \frac{1}{n^{\omega(1)}}$.

Proof. Corollary 23 states that all problems in SZK $\leq^{\mathsf{BPP}}_{\mathrm{hbf}}$-reduce to $\widetilde{R}_K$. Thus we need only show the converse containment. Let $A \leq^{\mathsf{BPP}}_{\mathrm{hbf}} \widetilde{R}_K$. As in the proof of Theorem 15, we will build circuits $C_{x,i}(r)$ that model the computation that produces the i^{th} query that is asked on input x, when using random bits r. As in the proof of Theorem 15, we claim that if a $1 - \frac{1}{n^{\omega(1)}}$ fraction of the strings of the form $C_{x,i}(r)$ are in $Y_{\widetilde{R}_K}$, then $C_{x,i}$ represents a distribution with entropy at least $m/2 - e(m)/2 + 1$, and if a $1 - \frac{1}{n^{\omega(1)}}$ fraction of the strings of the form $C_{x,i}(r)$ are in $N_{\widetilde{R}_K}$, then $C_{x,i}$ represents a distribution with entropy at most $m/2 - e(m)/2 - 1$. Indeed, the proof is essentially identical. Assume that there are infinitely many x that are not don't care instances, where replacing the $\widetilde{R}_K$ oracle with the EA oracle does not yield the correct answer. Given n, we can find the lexicographically-least string x of length n for which the reduction fails. Since the reduction fails, there must be some i such that the i^{th} query in the formula yields the wrong answer. Thus, given (n, i), we can find x and build the circuit $C_{x,i}$ of Kolmogorov complexity $O(\log n)$ that yields a correct answer when given $\widetilde{R}_K$ as an oracle, but fails when queries are made to EA instead. The analysis is identical to the argument in the proof of Theorem 15. ◀

We have nothing to say, regarding the problems that are reducible to $\widetilde{R}_K$ via $\leq_{\text{tt}}^{\mathsf{BPP}}$ or $\leq_{\text{rbf}}^{\mathsf{BPP}}$ reductions, other than to refer to the $\mathsf{AM} \cap \mathsf{coAM}$ upper bound provided by Saks and Santhanam [53]. We do have a somewhat better bound to report, regarding $\leq_{\text{circ}}^{\mathsf{BPP}}$ reducibility.

▶ **Theorem 30.** *The following are equivalent, for any decidable promise problem A:*

1. $A \leq_{\text{hcirc}}^{\mathsf{BPP}} \widetilde{R}_K$ *with threshold* $1 - \frac{1}{n^{\omega(1)}}$.
2. $A \leq_{\text{htt}}^{\mathsf{P}} \mathsf{EA}$.
3. $A \leq_{\text{tt}}^{\mathsf{P}} B$ *for some* $B \in \mathsf{SZK}$.

Proof. Item 2 obviously implies item 3. To see that item 3 implies item 1, observe that if $A \leq_{\text{tt}}^{\mathsf{P}} B$ for some $B \in \mathsf{SZK}$, then we know that $A \leq_{\text{htt}}^{\mathsf{P}} B \times 0^* \in \mathsf{SZK}$, and hence $A \leq_{\text{htt}}^{\mathsf{P}} \mathsf{EA} \leq_{\text{hm}}^{\mathsf{BPP}} \widetilde{R}_K$. The composition of a $\leq_{\text{htt}}^{\mathsf{P}}$ reduction with a $\leq_{\text{hm}}^{\mathsf{BPP}}$ reduction is clearly a $\leq_{\text{hcirc}}^{\mathsf{BPP}}$ reduction (as in Proposition 22). Finally, the proof of the remaining implication (item 1 implies item 2) follows along the same lines as the proof of Theorem 29. We still build circuits $C_{x,i}$ that produce the i^{th} query, and use the oracle for EA to determine if those circuits represent distributions of high or low entropy. Since we are assuming only that $A \leq_{\text{hcirc}}^{\mathsf{BPP}} \widetilde{R}_K$ (instead of $A \leq_{\text{hbf}}^{\mathsf{BPP}} \widetilde{R}_K$) we end by concluding only $A \leq_{\text{htt}}^{\mathsf{BPP}} \widetilde{R}_K$. ◀

6 Less Powerful Reductions

The standard complete problems EA and SD remain complete for NISZK and SZK, respectively, even under more restrictive reductions such as $\leq_{\text{m}}^{\mathsf{L}}, \leq_{\text{m}}^{\mathsf{AC}^0}, \leq_{\text{m}}^{\mathsf{NC}^0}$ and $\leq_{\text{m}}^{\mathsf{proj}}$. In this section, we show that it is worthwhile considering probabilistic versions of $\leq_{\text{m}}^{\mathsf{L}}, \leq_{\text{m}}^{\mathsf{AC}^0}$ and $\leq_{\text{m}}^{\mathsf{NC}^0}$ reducibility to $\widetilde{R}_K$.

▶ **Definition 31.** *For a class $\mathcal{C}$, a promise problem $A = (Y, N)$ is $\leq_{\text{m}}^{\mathsf{R}\mathcal{C}}$-reducible to $B = (Y', N')$ with threshold θ if there are a function $f \in \mathcal{C}$ and a polynomial p such that*

- $x \in Y$ *implies* $\Pr_{r \in \{0,1\}^{p(|x|)}}[f(x, r) \in Y'] \geq \theta$.
- $x \in N$ *implies* $\Pr_{r \in \{0,1\}^{p(|x|)}}[f(x, r) \in N'] = 1$.

A is $\leq_{\text{m}}^{\mathsf{BP}\mathcal{C}}$-reducible to B with threshold θ if there are a function $f \in \mathcal{C}$ and a polynomial p such that

- $x \in Y$ *implies* $\Pr_{r \in \{0,1\}^{p(|x|)}}[f(x, r) \in Y'] \geq \theta$.
- $x \in N$ *implies* $\Pr_{r \in \{0,1\}^{p(|x|)}}[f(x, r) \in N'] \geq \theta$.

We are particularly interested in the cases $\mathcal{C} = \mathsf{L}, \mathcal{C} = \mathsf{AC}^0$, and $\mathcal{C} = \mathsf{NC}^0$. Note especially that, in the definitions of $\leq_{\text{m}}^{\mathsf{RL}}$ and $\leq_{\text{m}}^{\mathsf{BPL}}$, the logspace computation has full (two-way) access to the random bits r. This is consistent with the way that probabilistic logspace computation is used in the context of the "verifier" and "simulator" in the complexity classes $\mathsf{SZK}_{\mathsf{L}}$ and $\mathsf{NISZK}_{\mathsf{L}}$ [30, 14].

$\mathsf{SZK}_{\mathsf{L}}$, the "logspace version" of SZK, was introduced in [30], primarily as a tool to discuss the complexity of problems involving distributions realized by extremely limited circuits (such as NC^0 circuits). It is shown in [30] that $\mathsf{SZK}_{\mathsf{L}}$ contains many of the problems of cryptographic significance that lie in SZK. $\mathsf{NISZK_L}$ was introduced in [14] as the "non-interactive" counterpart to $\mathsf{SZK_L}$, by analogy with NISZK, primarily as a tool to investigate the complexity of computing time-bounded Kolmogorov complexity. It was subsequently studied in [15], where it was shown to be robust to several changes to the definition. It is shown in [30, 14] that complete problems for $\mathsf{SZK_L}$ and $\mathsf{NISZK_L}$ arise by considering restrictions of the standard complete problems for SZK and NISZK where the distributions under consideration are represented either by branching programs (in $\mathsf{EA_{BP}}$), or by NC^0 circuits where each output bit depends on at most 4 input bits (in $\mathsf{SD_{NC^0}}$ and $\mathsf{EA_{NC^0}}$).

Following the pattern we established in Section 2, we now define $\mathsf{SZK_L}$ and $\mathsf{NISZK_L}$ in terms of their complete problems, rather than presenting the definitions in terms of interactive proofs:

▶ **Definition 32.** $\mathsf{SZK_L} = \{A : A \leq_{\mathrm{m}}^{\mathsf{proj}} \mathsf{SD_{NC^0}}\} = \{A : A \leq_{\mathrm{m}}^{\mathsf{L}} \mathsf{SD_{BP}}\}$
$\mathsf{NISZK_L} = \{A : A \leq_{\mathrm{m}}^{\mathsf{proj}} \mathsf{EA_{NC^0}}\} = \{A : A \leq_{\mathrm{m}}^{\mathsf{L}} \mathsf{EA_{BP}}\}$.

▶ **Theorem 33.** *The following are equivalent, for any decidable promise problem A:*
- $A \in \mathsf{NISZK_L}$
- $A \leq_{\mathrm{hm}}^{\mathsf{RNC}^0} \widetilde{R}_K$
- $A \leq_{\mathrm{hm}}^{\mathsf{BPNC}^0} \widetilde{R}_K$
- $A \leq_{\mathrm{hm}}^{\mathsf{RAC}^0} \widetilde{R}_K$
- $A \leq_{\mathrm{hm}}^{\mathsf{BPAC}^0} \widetilde{R}_K$
- $A \leq_{\mathrm{hm}}^{\mathsf{RL}} \widetilde{R}_K$
- $A \leq_{\mathrm{hm}}^{\mathsf{BPL}} \widetilde{R}_K$

Proof. The proof that $A \in \mathsf{NISZK_L}$ implies $A \leq_{\mathrm{hm}}^{\mathsf{RNC}^0} \widetilde{R}_K$ proceeds as in the proof of Theorem 15. Whereas the proof of Theorem 15 takes as its starting point the problem EA', we make use of the analogous problem $\mathsf{EA'_{NC^0}}$, defined exactly as EA' except that the input is an NC^0 circuit where each output bit depends on at most four input bits. It is shown in [15, Theorem 13] that a promise problem denoted $\mathsf{SDU'_{NC^0}}$ is complete for $\mathsf{NISZK_L}$ under uniform projections. The problem $\mathsf{SDU'_{NC^0}}$ has YES instances consisting of distributions with statistical distance at most 2^{-n^ϵ} from the uniform distribution, and NO instances consisting of distributions with suppport of size at most 2^{n-n^ϵ} for some fixed $\epsilon > 0$. Thus, precisely as in the proof of Lemma 14, we obtain that $\mathsf{EA'_{NC^0}}$ is complete for $\mathsf{NISZK_L}$ under uniform projections.

We continue to follow the outline of the proof of Theorem 15. The second paragraph of that proof makes use of Corollary 18 of [14], and instead we appeal to the analogous result [14, Corollary 43] (presenting a nonuniform $\leq_{\mathrm{m}}^{\mathsf{proj}}$ reduction from $\mathsf{EA_{NC^0}}$ to $\widetilde{R}_K$).

In more detail: as in the proof of Theorem 15, given x, our reduction constructs a sequence of independent copies of insteances of $\mathsf{EA'_{NC^0}}$. The proof of Corollary 43 in [14] shows that these NC^0 circuits can be constructed via uniform *projections*. Let $f(x, r)$ denote the function that takes input x (an instance of the promise problem A) and random sequence r as input, and first constructs (via a projection) the sequence $C_1, C_2, ..., C_{|x|^{O(1)}}$ of NC^0 circuits, and then produces as output the result of partitioning the bits of r into inputs r_i for each C_i, computing $C_i(r_i)$, and concatenating the results. Thus each output bit of $f(x, r)$ is computed by a gadget that is connected to $O(1)$ random bits (i.e., the bits that are fed into the circuit computing the distribution), along with at most one bit from the input x (determining the circuitry internal to the gadget). The rest of the analysis (showing that, if the $\mathsf{EA'_{NC^0}}$ instance has high entropy, then $f(x, r)$ has high Kolmogorov complexity with high probability, and if the $\mathsf{EA'_{NC^0}}$ instance has small support, then $f(x, r)$ has low Kolmogorov complexity) is similar to that in the proof of Theorem 15.

All of the other implications clearly follow, if we show that if A is decidable and $A \leq_{\mathrm{hm}}^{\mathsf{BPL}} \widetilde{R}_K$, then $A \in \mathsf{NISZK_L}$.

If A is decidable and $A \leq_{\mathrm{hm}}^{\mathsf{BPL}} \widetilde{R}_K$, then, as in the proof of Theorem 15, we build a device $C_x(r)$ that simulates the computation that produces queries to $\widetilde{R}_K$ on input x. However, now C_x is a branching program, and thus we replace queries to $\widetilde{R}_K$ by queries to $\mathsf{EA_{BP}}$. Since $\mathsf{EA_{BP}} \in \mathsf{NISZK_L}$, this shows that A is also in $\mathsf{NISZK_L}$. Again, the analysis is similar to that in the proof of Theorem 15. ◀

We end this section, with an analogous characterization of $\mathsf{SZK_L}$.

▶ **Definition 34.** *Let $A = (Y, N)$ and $B = (Y', N')$ be promise problems. We say $A \leq_{\mathrm{bf}}^{\mathsf{L}} B$ if there is a function f computable in logspace such that, for all x, $f(x)$ is of the form $(C, z_1, z_2, \ldots, z_k)$ where C is a Boolean formula with k input variables, and $(z_1, \ldots, z_k)$ is a list of queries, with the property that*
- *If $x \in Y$, then $C(\chi_B(z_1), \ldots, \chi_B(z_k)) = 1$.*
- *If $x \in N$, then $C(\chi_B(z_1), \ldots, \chi_B(z_k)) = 0$.*

Earlier work that studied $\leq_{\mathrm{bf}}^{\mathsf{L}}$ reducibility can be found in [25, 6].

We say $A \leq_{\mathrm{bf}}^{\mathsf{BPL}} B$ with threshold $\theta > \frac{1}{2}$ if there are functions f and g computable in **deterministic** *logspace, and a polynomial p, such that, for all x, $f(x)$ is a Boolean formula (with $k = |x|^{O(1)}$ variables), with the property that*
- *If $x \in Y$, then $C(\chi_{g,B}(x, 1), \ldots, \chi_{g,B}(x, k)) = 1$,*
- *If $x \in N$, then $C(\chi_{g,B}(x, 1), \ldots, \chi_{g,B}(x, k)) = 0$,*

where
- *$\chi_{g,B}(x, i) = 1$ if $\Pr_{r \in \{0,1\}^{p(|x|)}}[g(x, i, r) \in Y'] \geq \theta$*
- *$\chi_{g,B}(x, i) = 0$ if $\Pr_{r \in \{0,1\}^{p(|x|)}}[g(x, i, r) \in N'] \geq \theta$*
- *$\chi_{g,B}(x, i) = *$ otherwise.*

If the reduction is honest, then we write $A \leq_{\mathrm{hbf}}^{\mathsf{BPL}} B$

(Similarly, one can define AC^0 versions of $\leq_{\mathrm{bf}}^{\mathsf{L}}$, although, since an AC^0 circuit cannot evaluate a Boolean formula, we do not pursue that direction here.)

▶ **Theorem 35.** *The following are equivalent, for any decidable promise problem A:*
- *$A \in \mathsf{SZK_L}$.*
- *$A \leq_{\mathrm{bf}}^{\mathsf{L}} \mathsf{EA_{NC^0}}$.*
- *$A \leq_{\mathrm{hbf}}^{\mathsf{BPL}} \widetilde{R}_K$ with threshold $1 - \frac{1}{n^{\omega(1)}}$.*

Proof. The first two items are equivalent, because (a) $\mathsf{SZK_L}$ is closed under $\leq_{\mathrm{bf}}^{\mathsf{L}}$ reducibility [15], and (b) the argument in [33], showing that SZK $\leq_{\mathrm{bf}}^{\mathsf{L}}$-reduces to NISZK carries over directly to $\mathsf{SZK_L}$ and $\mathsf{NISZK_L}$. Furthermore, the reduction to $\mathsf{EA_{NC^0}}$ is length-increasing, and hence honest.

Since $\mathsf{EA_{NC^0}}$ is complete for $\mathsf{NISZK_L}$, Theorem 33 implies that every $A \in \mathsf{NISZK_L}$ is $\leq_{\mathrm{hbf}}^{\mathsf{BPL}}$-reducible to $\widetilde{R}_K$. The argument that every decidable A that $\leq_{\mathrm{hbf}}^{\mathsf{BPL}}$-reduces to $\widetilde{R}_K$ lies in $\mathsf{SZK_L}$ is similar to the argument in Theorem 29. ◀

7 How important is the "Honesty" Condition?

Our main results (Theorems 15 and 33) rely on restricting randomized reductions to $\widetilde{R}_K$ to be honest. In this section, we consider what happens when this "honesty" condition is dropped, for related notions of reducibility. First, we consider a seemingly much more powerful notion of reducibility, and show that we still obtain a complexity-theoretic upper bound.

▶ **Theorem 36.** *Let A be a decidable promise problem. Let R_{K_U} be the set $\{x : K_U(x) \geq |x|\}$. If $A \leq_{\mathrm{m}}^{\mathsf{NP}} R_{K_U}$ for every universal Turing machine U, then A has a solution in $\mathsf{PP}^{\mathsf{NP}}$.*

Note that, in contrast to Theorem 15, we no longer assume any approximation error, we no longer assume that the reduction is honest, and we are assuming a $\leq_{\mathrm{m}}^{\mathsf{NP}}$ reduction, instead of a $\leq_{\mathrm{m}}^{\mathsf{RP}}$ reduction. This means that there is a deterministic Turing machine M running in polynomial time $p(n)$ such that $x \in A_Y$ implies there exists a string r of length at most $p(|x|)$ such that $M(x, r) \in R_{K_U}$, and $x \in A_N$ implies that no such string r exists.

Proof. It will suffice to show that, for any decidable promise problem A that has no solution in $\mathsf{PP}^{\mathsf{NP}}$, there is a universal Turing machine U such that $A \not\leq_{\mathrm{m}}^{\mathsf{NP}} R_{K_U}$. We will follow the approach of [8, Theorem 14].

Let U_{st} be some "standard" universal Turing machine that is used to define $K(x)$. Now define a new Turing machine U such that $U(00d) = U_{st}(d)$ for every string d. Note that, for every string $x, K_U(x) \leq K(x) + 2$, and thus U is a Universal Turing machine. Next, we describe a stage construction that will define the behavior of U on inputs not in $00\{0,1\}^*$. We accomplish this by presenting an enumeration of pairs (d, y); that is, $U(d) = y$ if the pair (d, y) appears in the enumeration. In stage i, we will guarantee that the i^{th} nondeterministic Turing machine N_i (with a run-time of n^i) does not define a $\leq_{\mathrm{m}}^{\mathsf{NP}}$ reduction of A to R_{K_U}.

At the start of stage i, there is a length ℓ_i with the property that at no later stage will any string d of length less than ℓ_i or any string y of length less than $2\ell_i$ be enumerated into our list of pairs (d, y). (At stage 1, let $\ell_1 = 1$.)

For any string x, denote by $Q_i(x)$ the set of outputs produced along some branch of $N_i(x)$, and let $Q'_i(x)$ be the set of strings in $Q_i(x)$ having length less than ℓ_i.

In Stage i, the construction starts searching through all strings of length $2\ell_i$ or greater, until two strings x_0 and x_1 are found, such that

- $x_0 \in A_N$,
- $x_1 \in A_Y$,
- $Q'(x_0) = Q'(x_1)$, and
- One of the following holds:
  - $Q_i(x_1)$ contains no more than $2^{\lfloor m/2 \rfloor - 2}$ elements from $\{0,1\}^m$ for each length $m \geq 2\ell_i$, or
  - $Q_i(x_0)$ contains more than $2^{\lfloor m/2 \rfloor - 2}$ elements from $\{0,1\}^m$ for some length $m \geq 2\ell_i$. .

We argue below that strings x_0 and x_1 will be found after a finite number of steps.

If $Q_i(x_1)$ contains no more than $2^{\lfloor m/2 \rfloor - 2}$ elements from $\{0,1\}^m$ for each length $m \geq \ell_i$, then for each string y of length $m \geq \ell_i$ in $Q_i(x_1)$, pick a different d of length $\lfloor m/2 \rfloor - 2$ and add the pair $(1d, y)$ to the enumeration. This guarantees that $Q_i(x_1)$ contains no element of R_{K_U} of length $\geq 2\ell_i$. Thus if N_i is to be a $\leq_{\mathrm{m}}^{\mathsf{NP}}$ reduction of A to R_{K_U}, it must be the case that $Q'_i(x_1)$ contains an element of R_{K_U}. However, since $Q'_i(x1) = Q'_i(x_0)$ and $x_0 \notin A$, we see that N_i is not a $\leq_{\mathrm{m}}^{\mathsf{NP}}$ reduction of A to R_{K_U}

If $Q_i(x_0)$ contains more than $2^{\lfloor m/2 \rfloor - 2}$ elements from $\{0,1\}^m$ for some length $m \geq 2\ell_i$, then note that at least one of these strings is not produced as output by $U(00d)$ for any string d of length $\leq \frac{m}{2} - 2$. We will guarantee that U does not produce any of these strings on any description $d \notin 00\{0,1\}^*$, and thus one of these strings must be in R_{K_U}, and hence N_i is not a $\leq_{\mathrm{m}}^{\mathsf{NP}}$ reduction of A to R_{K_U}.

Let ℓ_{i+1} be the maximum of the lengths of x_0, x_1 and the lengths of the strings in $Q_i(x_0)$ and $Q_i(x_1)$.

It remains only to show that strings x_0 and x_1 will be found after a finite number of steps. Assume otherwise. It follows that $A_Y \cup A_N$ can be partitioned into a finite number of equivalence classes, where y and z are equivalent if both y and z have length less than $2\ell_i$, or if they have length $\geq 2\ell_i$ and $Q'_i(y) = Q'_i(z)$. Furthermore, for the equivalence classes containing long strings, if the class contains both strings in A and in $\overline{A}$, then the strings in A are exactly the strings on which N_i queries more than $2^{\lfloor m/2 \rfloor - 2}$ elements of $\{0,1\}^m$ for some length $m \geq 2\ell_i$. This can be decided by making a truth-table reduction to the set $\{(x, m) : N_i(x)$ queries at least $2^{\lfloor m/2 \rfloor - 2}$ strings of length $m\}$, which is in $\mathsf{PP}^{\mathsf{NP}}$. Since PP^B is closed under polynomial-time truth-table reductions for every oracle B [32], it follows that A has a solution in $\mathsf{PP}^{\mathsf{NP}}$, in contradiction to our choice of A. ◀

Theorem 36 highlights a weakness of $\leq_{\mathrm{m}}^{\mathsf{NP}}$ reducibility, in comparison to $\leq_{\mathrm{T}}^{\mathsf{P}}$ reducibility. By [36], every problem in $\mathsf{EXP}^{\mathsf{NP}}$ is $\leq_{\mathrm{T}}^{\mathsf{P}}$-reducible to R_{K_U} for every universal machine U, whereas Theorem 36 shows that any set $\leq_{\mathrm{m}}^{\mathsf{NP}}$ reducible to R_{K_U} for every U lies in $\mathsf{PP}^{\mathsf{NP}}$, which seems to be a much smaller class.

Theorem 36 gives an *upper* bound on the complexity of problems $\leq_{\mathrm{m}}^{\mathsf{NP}}$ reducible to R_{K_U}; what can we say about lower bounds? It is clear that every set in NP is $\leq_{\mathrm{m}}^{\mathsf{NP}}$ reducible to any set other than the empty set and Σ^*, and Theorem 15 implies that every problem in NISZK is also reducible to R_{K_U} in this way. (Note that NISZK is not known to be contained in NP.) But if we impose an "honesty" restriction on $\leq_{\mathrm{m}}^{\mathsf{NP}}$ reductions, then it is not at all clear that all problems in NP reduce to R_{K_U}, although Theorem 15 implies that problems in NISZK reduce not only to R_{K_U}, but to the more restrictive problem $\widetilde{R}_K$, using the even more restrictive $\leq_{\mathrm{hm}}^{\mathsf{RP}}$ reductions.

Now we turn to the $\leq_{\mathrm{m}}^{\mathsf{RP}}$ reductions that yield one of our characterizations of NISZK, but dropping the "honesty" condition.

▶ **Theorem 37.** *Let A be a decidable promise problem. If $A \leq_{\mathrm{m}}^{\mathsf{RP}} \widetilde{R}_K$, then A has a solution in $\mathsf{AM} \cap \mathsf{coAM}$.*

Proof. If $A \leq_{\mathrm{m}}^{\mathsf{RP}} \widetilde{R}_K$, then there is a single reduction R such that, for each universal Turing machine U, R reduces A to R_{K_U} for all large inputs. We make use of this (weaker) assumption, without relying on the $\omega(\log n)$ "approximation" term in the definition of $\widetilde{R}_K$. Thus Theorem 37 is incomparable with the main result of [53], where the same upper bound of $\mathsf{AM} \cap \mathsf{coAM}$ is presented for more general nonadaptive reductions, but with an "honesty" restriction, and requring a superlogarithmic approximation term for the Kolmogorov complexity promise problem.

We follow a similar strategy to the proof of Theorem 36, while also incorporating some of the techniques of [39, Theorem 2]. Let A be any decidable promise problem with no solution in AM. We will show that, for every machine R computing a (possible) $\leq_{\mathrm{m}}^{\mathsf{RP}}$ reduction, there is a universal Turing machine U such that there are infinitely many inputs on which R fails to reduce A to R_{K_U}.

Let R be any probabilistic polynomial-time Turing machine that (possibly) computes a $\leq_{\mathrm{m}}^{\mathsf{RP}}$ reduction to R_{K_U} for every U (for all large inputs), and let $p(n)$ be the running time of R. Define $\delta(n) = 1/p(n)^{11}$, and let $\delta'(n) = 3p(n)\delta(n)$.

On input x, the reduction R may query strings of various lengths j. Let $R_j(x)$ be the set of all random sequences r such that $R(x, r)$ outputs a string of length j. For a given U, define $P_j(x)$ to be $\Pr[R(r, x) \in R_{K_U} | r \in R_j(x)]$. (The machine U under consideration will always be clear from context.)

▷ Claim 38. If R is computing a $\leq_{\mathrm{m}}^{\mathsf{RP}}$ reduction to R_{K_U} on input x, then

- If the reduction accepts on input x, then there is some j such that $\Pr[r \in R_j(x)] \geq 2\delta(n)$ and $P_j(x) \geq 1 - \delta'(n)$.

- If the reduction rejects on input x, then for all j such that $\Pr[r \in R_j(x)] > 0, P_j(x) = 0$.

Proof. The first item is proved along the lines of [39, Claim 14]: By definition, the probability that the reduction accepts on input x is

$$\Pr_r\left[K_U(R(x,r)) \geq \frac{|R(x,r)|}{2}\right] = \sum_j \Pr[r \in R_j(x)] \cdot P_j(x).$$

If R is a $\leq_{\mathrm{m}}^{\mathsf{RP}}$ reduction to R_{K_U} then this probability is $1 - \frac{1}{n^{\omega(1)}} \geq 1 - \delta(n)^2$. Assume by way of contradiction that $P_j(x) < 1 - \delta'(n)$ for every j such that $\Pr[r \in R_j(x) \geq 2\delta(n)$. Then

$$\begin{aligned}
1 - \delta(n)^2 &\leq \sum_j \Pr[r \in R_j(x)] \cdot P_j(x) \\
&= \sum_{\{j : P_j(x) \geq 2\delta(n)\}} \Pr[r \in R_j(x)] \cdot P_j(x) \ + \sum_{\{j : P_j(x) < 2\delta(n)\}} \Pr[r \in R_j(x)] \cdot P_j(x) \\
&\leq (1 - \delta'(n)) + p(n)2\delta(n) = 1 - 3p(n)\delta(n) + p(n)2\delta(n) = 1 - p(n)\delta(n)
\end{aligned}$$

and thus $p(n) \leq \delta(n) < 1$, which is a contradiction.

The second item follows immediately from the definition of a $\leq_{\mathrm{m}}^{\mathsf{RP}}$ reduction. If the reduction rejects on input x, then every query must be non-random. ◀

Let us say that j is *popular for x* if $\Pr[r \in R_j(x)] \geq 2\delta(n)$. Since the running time of R is $p(n)$, and since R outputs a string of some length (at most $p(n)$) along every path, there is always some j such that $\Pr[r \in R_j(x)] \geq \frac{1}{p(n)} \geq 2\delta(n)$, and thus there is always at least one j that is popular for x.

Let U_{st} be some "standard" universal Turing machine that is used to define $K(x)$. Now define a new Turing machine U such that $U(00d) = U_{st}(d)$ for every string d. Note that, for every string $x, K_U(x) \leq K(x) + 2$, and thus U is a Universal Turing machine. Next, we describe a stage construction that will define the behavior of U on inputs not in $00\{0,1\}^*$. We accomplish this by presenting an enumeration of pairs (d, y); that is, $U(d) = y$ if the pair (d, y) appears in the enumeration. In stage i, we will guarantee that there are at least i inputs on which R fails to reduce A to R_{K_U}.

At the start of stage i, there is a length ℓ_i with the property that at no later stage will any string d of length less than ℓ_i or any string y of length less than $2\ell_i$ be enumerated into our list of pairs (d, y). (At stage 1, let $\ell_1 = 1$.)

Let us say that a query q of length j is *β-heavy* on input x if $\Pr_{r \in R_j}[R(x,r) = q] \geq \beta$.

In Stage i, the construction starts searching through all strings of length $2\ell_i$ or greater, until two strings x_0 and x_1 are found, such that

- $x_0 \in A_N$,
- $x_1 \in A_Y$, and
- For each $y \in \{x_0, x_1\}$, there is a $j \geq \ell_i$ such that j is popular for y.
- One of the following holds:
  - For some $j \geq \ell_i$ that is popular for x_1, letting $m = \lfloor j/2 \rfloor$, and setting $\beta = \frac{1}{2^{m+13}}$, $\Pr_{r \in R_j(x_1)}[R(x,r)$ is β heavy$] \geq \frac{1}{4}$.
  - For every $j \geq \ell_i$ that is popular for x_0, as above letting $m = \lfloor j/2 \rfloor$, and setting $\beta = \frac{1}{2^{m+13}}$, $\Pr_{r \in R_j(x_0)}[R(x,r)$ is $2^{11}\beta$ heavy$] \leq \frac{3}{4}$.

We claim that some such pair (x_0, x_1) will be found after a finite number of steps, and that R fails to reduce A to R_{K_U} on either x_0 or x_1. Thus, at the end of stage i we will have found at least i strings on which R fails to reduce A to R_{K_U}. Then we set ℓ_i to be larger than the length of any query that is made by R on either x_0 and x_1, and move on to the next stage.

To see that a pair (x_0, x_1) will always be found, observe that otherwise, a string x of length greater than $2\ell_i$ in $A_N \cup A_Y$ is a YES instance if for every $j \geq \ell_i$ that is popular for x, $\Pr_{r\in R_j(x)}[R(x,r)$ is β heavy$] < \frac{1}{4}$, and x is a NO instance if there is some $j \geq \ell_i$ that is popular for x, where $\Pr_{r\in R_j(x)}[R(x,r)$ is $2^{11}\beta$ heavy$] > \frac{3}{4}$.[8] But these conditions can both be checked in $\mathsf{AM} \cap \mathsf{coAM}$, which places A in $\mathsf{AM} \cap \mathsf{coAM}$, contrary to our choice of A. To see this, note that the distribution given by $R(x,r)$ for uniformly sampled $r \in R_j(x)$ is very close to a polynomial-time samplable distribution if j is popular. (Simply choose r uniformly at random for a large polynomial number of tries, until some r is found such that $R(x,r)$ has length j, and output this $R(x,r)$. By sampling r for a large enough polynomial number of times, the resulting distribution D has the property that $|\Pr_{r\sim D}[R(x,r)$ is β heavy$] - \Pr_{r\in R_j(x)}[R(x,r)$ is β heavy$]| < \frac{1}{8}$), and similarly the probabilities of sampling a $2^{11}\beta$-heavy string in the two distributions are very close.) Thus we can appeal to the heavy samples protocol of Bogdanov and Trevisan [23], as presented in [39, Lemma 13]:

▶ **Lemma 39.** *Let $q(n)$ be a polynomial. There is an $\mathsf{AM} \cap \mathsf{coAM}$ protocol that solves the following promise problem: Given a circuit of size $q(n)$ producing output of length n representing a distribution D, and given a threshold $\beta = \frac{a}{b} \in (0,1)$ where a and b are represented in binary notation, accept if* $\Pr_{y\sim D}[y$ is $2^{11}\beta-$heavy$] \geq \frac{7}{8}$*, and reject if* $\Pr_{y\sim D}[y$ is $\beta-$heavy$] \leq \frac{1}{8}$.[9]

This gives the desired $\mathsf{AM} \cap \mathsf{coAM}$ protocol. (More precisely, Arthur can use BPP computation to determine which j are popular, and then construct the circuits that approximate the distributions required, to run the heavy samples protocol in parallel for each popular $j \geq \ell_i$.)

If the pair (x_0, x_1) that is found in stage i satisfies the second condition (namely: for every $j \geq \ell_i$ that is popular for x_0, $\Pr_{r\in R_j(x_0)}[R(x,r)$is $2^{11}\beta$ heavy$] \leq \frac{3}{4}$) we can conclude that R does not define a $\leq^{\mathsf{RP}}_{\mathrm{m}}$ reduction of A to R_{K_U} on x_0, since (a) there must be some $j \geq \ell_i$ that is popular for x_0, and (b) there must be more than $2^{\lfloor j/2 \rfloor}$ strings of length j that are queried by R on input x_0, and thus at least one of them must be random. To see this, order the 2^j possible queries of length j in decreasing order of weight, $q_1, q_2, \ldots, q_{2^m}, \ldots q_{2^{m+2}}, \ldots, q_{2^j}$, where $m = \lfloor j/2 \rfloor$ and $2^{11}\beta = \frac{1}{2^{m+2}}$. Let $w(q_i)$ denote the weight of q_i; thus $w(q_i) \geq w(q_{i+1})$ and $w(q_i) \leq \frac{1}{i}$. It suffices to show that, if no more than 2^m strings of length j are queried,

[8] There is actually one other possibility: that all j that are popular for x are less than ℓ_i. However, in this case the probability given to longer queries is no more than $p(n)\delta(n) = \frac{1}{p(n)^{10}}$ and thus the short queries determine the outcome of the reduction. Thus in BPP we can determine which $j \leq \ell_i$ are popular and simulate the reduction on those short queries, using a finite table to answer all of the short queries.

[9] This is not precisely the way that the heavy samples lemma is stated in [39], but the proof that is presented there establishes this version of the lemma.

then $\Pr_{r\in R_j(x_0)}[R(x,r) \text{ is } 2^{11}\beta \text{ heavy}] > \frac{3}{4}$.

$$\begin{aligned}\Pr_{r\in R_j(x_0)}[R(x,r) \text{ is } 2^{11}\beta \text{ heavy}] &= \sum_{\{i:w(q_i)\geq 2^{-m-2}\}} w(q_i)\\ &= 1 - \sum_{\{i:w(q_i)<2^{-m-2}\}} w(q_i)\\ &> 1 - \sum_{\{i:w(q_i)<2^{-m-2}\}} 2^{-m-2}\\ &\geq 1-(2^m\cdot 2^{-m-2}) = \frac{3}{4}.\end{aligned}$$

On the other hand, if the pair that is found in stage i satisfies the first condition (namely: for some $j \geq \ell_i$ that is popular for x_1, $\Pr_{r\in R_j(x_1)}[R(x,r) \text{ is } \frac{1}{2^{m+13}} \text{ heavy}] \geq \frac{1}{4}$), then – as above – order the 2^j possible queries of length j in decreasing order of weight, $q_1, q_2, \ldots, q_{2^{m-2}}, \ldots q_{2^m}, \ldots, q_{2^j}$. For each $q \in S = \{q_h : h \leq 2^{m-2}\}$ choose a distinct description d of length $m-2$ and enumerate $(1d, q)$ into the description of U, thereby assuring that the heaviest queries made by R on input x_1 are all non-random. The probability mass of the heaviest queries is minimized if as much mass as possible is shifted to the lighter queries. Let i be the largest number such that $w(q_i) \geq \beta$. In this case, $\Pr_{r\in R_j(x_1)}[R(x,r) \text{ is } \frac{1}{2^{m+13}} \text{ heavy}] = i\beta \geq \frac{1}{4}$, and hence $i \geq 2^{m+13}$. In particular, we can conclude that the probability that $R(x_1)$ outputs one of the 2^{m-2} strings in S (conditioned on R producing a string of length j with weight at least β) is minimized if all strings of weight at least β have equal probability, and in particular $w(q_{2^{m-2}}) = \beta$. Thus $\Pr[R(x_1,r) \in S | R(x_1,r) \text{ has weight } \geq \beta \text{ and has length } j] \geq \frac{2^{\mathrm{m}-2}}{2^{\mathrm{m}+13}} = \frac{1}{2^{15}}$. Thus

$$\begin{aligned}&\Pr_{r\in R_j(x_1)}[R(x,r)\in S]\\ &= \Pr_{r\in R_j(x_1)}[R(x,r)\in S | R(x,r) \text{ is } \frac{1}{2^{m+13}} \text{ heavy}]\cdot \Pr_{r\in R_j(x_1)}[R(x,r) \text{ is } \frac{1}{2^{m+13}} \text{ heavy}]\\ &\geq \frac{1}{2^{15}}\cdot\frac{1}{4}.\end{aligned}$$

Thus, since j is popular for x_1, $R(x_1,r)$ is producing as output a non-random string with probability at least $2\delta(n)/2^{17}$, which means that R is failing to compute a $\leq_{\mathrm{m}}^{\mathsf{RP}}$ reduction of A to R_{K_U} (since this would require that $R(x_1)$ output a random string with probability $1 - \frac{1}{n^{\omega(1)}}$).

◀

▶ Remark 40. The proof of Theorem 37 carries over, with only minor changes, to nonadaptive probabilistic reductions that make at most one query along any path.

8 Discussion

There are not many examples of natural computational problems that are known or conjectured to lie outside of P, such that the class of problems reducible to them via $\leq_{\mathrm{m}}^{\mathsf{P}}$ and $\leq_{\mathrm{m}}^{\mathsf{L}}$ (or $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$) reductions differ (or are conjectured to differ). Is it the case that the problems reducible to $\widetilde{R}_K$ via $\leq_{\mathrm{hm}}^{\mathsf{RP}}$ and $\leq_{\mathrm{hm}}^{\mathsf{RL}}$ (or $\leq_{\mathrm{hm}}^{\mathsf{RAC}^0}$) reductions differ? Or should this be taken as evidence that NISZK and $\mathsf{NISZK}_{\mathsf{L}}$ coincide?

Similarly, there are not many examples of natural computational problems such that the classes of problems reducible to them via $\leq_{\mathrm{tt}}^{\mathsf{P}}$ and $\leq_{\mathrm{bf}}^{\mathsf{P}}$ reductions differ (or are conjectured to differ). For example, these reducibilities coincide for SAT [26]. Is it the case that $\leq_{\mathrm{bf}}^{\mathsf{BPP}}$ and $\leq_{\mathrm{circ}}^{\mathsf{BPP}}$ reducibilities differ for $\widetilde{R}_K$? Or should this be taken as evidence that SZK is closed under $\leq_{\mathrm{tt}}^{\mathsf{P}}$ reducibility?

Perhaps our new characterizations of statistical zero knowledge classes will be useful in answering these questions.

It is known that every promise problem in $\mathsf{NISZK_L}$ reduces to $\widetilde{R}_K$ via *nonuniform projections* [14, 4]. The following quote from [4] is worth paraphrasing here:

> ...no complexity class larger than $\mathsf{NISZK_L}$ is known to be (non-uniformly) $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reducible to the Kolmogorov-random strings [14]. It seems unlikely that this is optimal.

The discussion in [4] was referring to reductions to an oracle for the *exact* Kolmogorov-complexity function. Our results show that, for reductions to an *approximation* to the Kolmogorov-complexity function, $\mathsf{NISZK_L}$ *is* essentially "optimal".

9 An Application

Finally, let us observe that our new characterizations of $\mathsf{NISZK_L}$ may open new avenues of attack on questions such as whether $\mathsf{NP} = \mathsf{NL}$. MKTP, the problem of computing KT complexity, lies in NP and is hard for co-$\mathsf{NISZK_L}$ under nonuniform projections [14]. If $\mathsf{MKTP} \in \mathsf{NISZK_L}$, then there must be a nonuniform projection f that takes strings of low KT-complexity (and hence low K-complexity) to strings of high K complexity, and simultaneously maps strings of high KT complexity to strings of low K-complexity.[10] It is plausible that one could show unconditionally that no such projection can exist. Among other things, this would show that $\mathsf{NP} \neq \mathsf{DET}$ (where DET is the complexity class, containing NL, of problems that reduce to the determinant) since $\mathsf{DET} \subseteq \mathsf{NISZK_L}$ [14].[11]

It may be useful to observe that, if $\mathsf{MKTP} \in \mathsf{NISZK_L}$, then the projection discussed in the preceding paragraph can be assumed without loss of generality to have a very specific form.

▶ **Theorem 41.** *Let ϵ be any number greater than zero, and let $e(m)$ be any function computable in AC^0, where $\omega(\log m) < e(m) < m^{o(1)}$. If $\mathsf{MKTP} \in \mathsf{NISZK_L}$, then there is a (non-uniform, polynomial-size) projection f mapping strings of length n to strings of length m, such that*
- $\mathsf{KT}(x) \leq \frac{n}{3}$ *implies* $K(f(x)) > \frac{m}{2}$, *and*
- $\mathsf{KT}(x) > \frac{n}{3}$ *implies* $K(f(x)) < \frac{m}{2} - e(m)$

and furthermore, $f(x)$ has the following form: Given input $x = x_1 x_2 \dots x_n$,

$$f(x) = y_n g_1(x_1) g_2(x_2) \dots g_n(x_n),$$

where y_n has length $\geq m - m^{\epsilon}$ and depends only on n, and each each g_i depends on only a single bit of x, and all of the strings $g_1(0), g_1(1), g_2(0), g_2(1), \dots, g_n(0), g_n(1)$ have the same length.

Proof. (Sketch) If $\mathsf{MKTP} \in \mathsf{NISZK_L}$, then the language A consisting of all strings x such that $\mathsf{KT}(x) < \frac{|x|}{3}$ is also in $\mathsf{NISZK_L}$, and hence, by Theorem 33 $A \leq_{\mathrm{hm}}^{\mathsf{RNC}^0} \widetilde{R}_K$, via a function $f_0(x, r)$

[10] Similarly, under the same assumption, there is a nonuniform projection that takes strings of low KT complexity to strings of high KT complexity, and simultaneously maps trings of high KT complexity to strings of low KT complexity.

[11] More precisely, as observed in [17], the Rigid Graph (non-) Isomorphism problem is hard for DET [55], and the Rigid Graph Non-Isomorphism problem is in $\mathsf{NISZK_L}$ [14, Corollary 23].

computable in *uniform* NC^0. Furthermore, as in Propositions 3 and 5, we may assume that many of the output bits in $f_0(x, r)$ do not depend on x at all, but are simply "padding". In fact, as in [14, Theorem 39], the error probability for the reduction is exponentially small, and a deterministic (but *nonuniform*) reduction can be obtained by hardwiring in a fixed choice of r. As described in the proof of [14, Corollary 41], this yields a function $f_1(x)$ that is a *projection*; briefly, this is because each output bit of $f_0(x, r)$ depends on at most one bit of x (and depends on $O(1)$ bits of r).

Many of the output bits of $f_1(x)$ are fixed by the choice of r, and do not depend on x at all. In fact, since $f_0(x, r)$ is in *uniform* NC^0, and since many of the output bits are the result of padding, there are at least $m - m^\epsilon$ bit positions that we can easily find that do not depend on x. Let y_n be the string that results from concatenating those bit positions consecutively. All of the bit positions of $f_0(x, r)$ that do not correspond to a bit in y_n are all connected to exactly one bit position of x. Let k_i be the number of output bits connected to x_i, and let k be the maximum of all of the k_i; note that k can easily be computed, given n.

Let $g_i(b)$ be the string of length k consisting of the concatenation of the bits of $f_0(x, r)$ that depend on x_i, when $x_i = b$ (padded out with zeros, if necessary, to obtain a string of length k).

Let $f_2(x) = y_n g_1(x_1) \ldots g_n(x_n)$. It is easy to see that $K(f_1(x)) = K(f_2(x)) \pm O(1)$. (Given a short description of $f_1(x)$ or $f_2(x)$, the other string can be obtained by simply rearranging the bits, using the uniform description of f_0 to indicate which bits should be moved where. This function f_2 is the projection f in the statement of the theorem. The proof is completed, by noticing that the proof of Theorem 33 carries over for any promise problem defined as $\widetilde{R}_K$, but with the YES instances consisting of strings z with $K(z) > \frac{|z|}{2} + c$ for any constant c. ◀

We do not know if a version of Theorem 41 holds, where K-complexity is replaced by KT-complexity.

We have not been able to prove that there is no projection reducing MKTP to $\widetilde{R}_K$. In fact, we do not even know whether there is a projection reducing the halting problem to $\widetilde{R}_K$. The structure of the computably-enumerable degrees of languages under non-uniform projections does not seem to have been studied in any depth. Indeed, it is easy to observe that non-uniform projections do not behave similarly to the more-commonly studied m-reductions:

▶ **Theorem 42.** *The halting problem $\leq_{\mathrm{m}}^{\mathsf{proj}}$-reduces to its complement.*

Proof. Let $H = \{(M, x) : M \text{ halts on input } x\}$. Let $n_H = H \cap \{y : |y| \leq n\}$. Note that the set $A = \{(y, i) : \text{there are at least } i \text{ strings } x \neq y \text{ in } H \text{ having length at most } n\}$ is computably-enumerable, and thus there is a projection f reducing A to H. Let y have length n. Note that $y \notin H$ if and only if $f(y, n_H) \in H$. ◀

Although we do not know how to prove that there is no projection reducing MKTP to $\widetilde{R}_K$, we note there there is provably no projection reducing MKTP to a related problem $\widetilde{R'}_K$, where the "gap" between the YES and NO instances is larger than in $\widetilde{R}_K$. Define $\widetilde{R'}_K$ to have YES instances $\{x : K(x) \geq \frac{4|x|}{5}\}$ and NO instances $\{x : K(x) \leq \frac{|x|}{5}\}$.

▶ **Theorem 43.** *There is no projection reducing MKTP to $\widetilde{R'}_K$.*

Proof. Since PARITY is in $\mathsf{co\text{-}NISZK_L}$, we know that PARITY $\leq_{\mathrm{m}}^{\mathsf{proj}}$ MKTP. Thus if $\mathsf{MKTP} \leq_{\mathrm{m}}^{\mathsf{proj}} \widetilde{R'}_K$ it follows that PARITY $\leq_{\mathrm{m}}^{\mathsf{proj}} \widetilde{R'}_K$. We apply the techniques of [20, Lemma 6] to show that no such projection can exist. More precisely, we show that if A is any language that projection reduces to $\widetilde{R'}_K$, then the 1-block sensitivity of A is at most 2. (Since the 1-block sensitivity of PARITY is n, this suffices to prove the theorem.)

Let $x \in A$ be such that the block sensitivity at x is at least 3. Thus there are three disjoint blocks of input bits B_1, B_2, B_3, such that flipping the bits in any block B_i produces a string $x_i \notin A$. If f is a projection reducing A to $\widetilde{R'}_K$, then $K(f(x)) \geq \frac{4m}{5}$, where $m = |f(x)|$, whereas $K(f(x_i)) \leq \frac{m}{5}$. Let d_i be a short description of x_i; thus $U(d_i) = x_i$, where U is the universal Turing machine from the definition of Kolmogorov complexity. Any bit of the output of f depends on at most 1 input bit. Thus, for any i, the i^{th} bit of $f(x)$ agrees with the i^{th} bit of at least 2 of $\{f(x_1), f(x_2), f(x_3)\}$ (since the blocks B_1, B_2, and B_3 are disjoint). Thus we can simply take the majority vote of $\{U(d_1), U(d_2), U(d_3)\}$ to obtain any bit of $f(x)$. It follows that $K(f(x)) \leq |d_1| + |d_2| + |d_3| + O(\log m) < \frac{4m}{5}$. This is a contradiction. ◀

In this vein, let us also remark that Kolmogorov complexity has already proved useful in developing nonrelativizing proof techniques [37], and also that the machinery of perfect randomized encodings (which were developed in [21] and which are essential to the results of [14]) also does not seem to relativize in any obvious way.

Acknowledgments

We thank Sam Buss, Johannes Köbler, and Uwe Schöning for discussions concerning Boolean formula reducibility. This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing.

References

1 Leonard M. Adleman and Kenneth L. Manders. Reducibility, randomness, and intractability (abstract). In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–163. ACM, 1977. doi:10.1145/800105.803405.

2 Eric Allender. Curiouser and curiouser: The link between incompressibility and complexity. In *Proc. Computability in Europe (CiE)*, volume 7318 of *Lecture Notes in Computer Science*, pages 11–16. Springer, 2012. doi:10.1007/978-3-642-30870-3_2.

3 Eric Allender. The complexity of complexity. In *Computability and Complexity: Essays Dedicated to Rodney G. Downey on the Occasion of his 60th Birthday*, volume 10010 of *Lecture Notes in Computer Science*, pages 79–94. Springer, 2017. doi:10.1007/978-3-319-50062-1_6.

4 Eric Allender. Vaughan Jones, Kolmogorov complexity, and the new complexity landscape around circuit minimization. *New Zealand journal of mathematics*, 52, 2021. doi:10.53733/148.

5 Eric Allender, José L. Balcázar, and Neil Immerman. A first-order isomorphism theorem. *SIAM J. Comput.*, 26(2):557–567, 1997. doi:10.1137/S0097539794270236.

6 Eric Allender, David A. Mix Barrington, Tanmoy Chakraborty, Samir Datta, and Sambuddha Roy. Planar and grid graph reachability problems. *Theory of Computing Systems*, 45(4):675–723, 2009. doi:10.1007/s00224-009-9172-z.

7 Eric Allender, Harry Buhrman, Luke Friedman, and Bruno Loff. Reductions to the set of random strings: The resource-bounded case. *Logical Methods in Computer Science*, 10(3), 2014. doi:10.2168/LMCS-10(3:5)2014.

8 Eric Allender, Harry Buhrman, and Michal Koucký. What can be efficiently reduced to the Kolmogorov-random strings? *Annals of Pure and Applied Logic*, 138:2–19, 2006.

9 Eric Allender, Harry Buhrman, Michal Kouckỳ, Dieter Van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35(6):1467–1493, 2006. doi:10.1137/050628994.

10 Eric Allender, Mahdi Cheraghchi, Dimitrios Myrisiotis, Harsha Tirumala, and Ilya Volkovich. One-way functions and a conditional variant of MKTP. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 213 of *LIPIcs*, pages 7:1–7:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.FSTTCS.2021.7.

11 Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. *Information and Computation*, 256:2–8, 2017. Special issue for MFCS '14. doi:10.1016/j.ic.2017.04.004.

12 Eric Allender, George Davie, Luke Friedman, Samuel B. Hopkins, and Iddo Tzameret. Kolmogorov complexity, circuits, and the strength of formal theories of arithmetic. *Chicago Journal of Theoretical Computer Science*, 2013(5), April 2013. doi:10.4086/cjtcs.2013.005.

13 Eric Allender, Luke Friedman, and William Gasarch. Limits on the computational power of random strings. *Information and Computation*, 222:80–92, 2013. ICALP 2011 Special Issue. doi:10.1016/j.ic.2011.09.008.

14 Eric Allender, John Gouwar, Shuichi Hirahara, and Caleb Robelle. Cryptographic hardness under projections for time-bounded Kolmogorov complexity. *Theoretical Computer Science*, 940(B):206–224, 2023. doi:10.1016/j.tcs.2022.10.040.

15 Eric Allender, Jacob Gray, Saachi Mutreja, Harsha Tirumala, and Pengxiang Wang. Robustness for space-bounded statistical zero knowledge. In Nicole Megow and Adam Smith, editors, *Proc. International Workshop on Randomization and Computation (RANDOM 2023)*, volume 275 of *LIPIcs*, pages 56:1–56:21, Dagstuhl, Germany, 2023. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik. See also ECCC Report TR22-138. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.56.

16 Eric Allender, Joshua A Grochow, Dieter Van Melkebeek, Cristopher Moore, and Andrew Morgan. Minimum circuit size, graph isomorphism, and related problems. *SIAM Journal on Computing*, 47(4):1339–1372, 2018. doi:10.1137/17M1157970.

17 Eric Allender and Shuichi Hirahara. New insights on the (non-) hardness of circuit minimization and related problems. *ACM Transactions on Computation Theory*, 11(4):1–27, 2019. doi:10.1145/3349616.

18 Eric Allender, Shuichi Hirahara, and Harsha Tirumala. Kolmogorov complexity characterizes statistical zero knowledge. Technical Report TR22-127, Electronic Colloquium on Computational Complexity (ECCC), 2022.

19 Eric Allender, Shuichi Hirahara, and Harsha Tirumala. Kolmogorov complexity characterizes statistical zero knowledge. In *14th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 251 of *LIPIcs*, pages 3:1–3:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.3.

20 Eric Allender, Rahul Ilango, and Neekon Vafa. The non-hardness of approximating circuit size. *Theory of Computing Systems*, 65(3):559–578, 2021. doi:10.1007/s00224-020-10004-x.

21 Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. *SIAM Journal on Computing*, 36(4):845–888, 2006. doi:10.1137/S0097539705446950.

22 David A. Mix Barrington, Neil Immerman, and Howard Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41(3):274–306, 1990. doi:10.1016/0022-0000(90)90022-D.

23 Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for NP problems. *SIAM J. Comput.*, 36(4):1119–1159, 2006. doi:10.1137/S0097539705446974.

24 Harry Buhrman, Lance Fortnow, Michal Koucký, and Bruno Loff. Derandomizing from random strings. In *25th IEEE Conference on Computational Complexity (CCC)*, pages 58–63. IEEE, 2010. doi:10.1109/CCC.2010.15.

25 Harry Buhrman, Edith Spaan, and Leen Torenvliet. The relative power of logspace and polynomial time reductions. *Computational Complexity*, 3:231–244, 1993. doi:10.1007/BF01271369.

26 Samuel R. Buss and Louise Hay. On truth-table reducibility to SAT. *Information and Computation*, 91(1):86–102, 1991. doi:10.1016/0890-5401(91)90075-D.

27 Mingzhong Cai, Rodney Downey, Rachel Epstein, Steffen Lempp, and Joseph Miller. Random strings and tt-degrees of Turing complete c.e. sets. *Logical Methods in Computer Science*, 10(3):1–24, 2014. doi:10.2168/LMCS-10(3:15)2014.

28 Richard Chang, Jim Kadin, and Pankaj Rohatgi. On unique satisfiability and the threshold behavior of randomized reductions. *Journal of Computer and System Sciences*, 50(3):359–373, 1995. doi:10.1006/jcss.1995.1028.

29 R. Downey and D. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer, 2010.

30 Zeev Dvir, Dan Gutfreund, Guy N Rothblum, and Salil P Vadhan. On approximating the entropy of polynomial mappings. In *Second Symposium on Innovations in Computer Science*, 2011.

31 Friederike Anna Dziemba. Uniform diagonalization theorem for complexity classes of promise problems including randomized and quantum classes. *CoRR*, abs/1712.07276, 2017.

32 Lance Fortnow and Nick Reingold. PP is closed under truth-table reductions. *Information and Computation*, 124(1):1–6, 1996. doi:10.1006/inco.1996.0001.

33 Oded Goldreich, Amit Sahai, and Salil Vadhan. Can statistical zero knowledge be made non-interactive? or On the relationship of SZK and NISZK. In *Annual International Cryptology Conference*, pages 467–484. Springer, 1999. doi:10.1007/3-540-48405-1_30.

34 Joachim Grollmann and Alan L. Selman. Complexity measures for public-key cryptosystems. *SIAM J. Comput.*, 17(2):309–335, 1988. doi:10.1137/0217018.

35 Shuichi Hirahara. Unexpected hardness results for Kolmogorov complexity under uniform reductions. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1038–1051. ACM, 2020. doi:10.1145/3357713.3384251.

36 Shuichi Hirahara. Unexpected power of random strings. In *11th Innovations in Theoretical Computer Science Conference, ITCS*, volume 151 of *LIPIcs*, pages 41:1–41:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.41.

37 Shuichi Hirahara. NP-hardness of learning programs and partial MCSP. In *63rd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 968–979. IEEE, 2022. doi:10.1109/FOCS54457.2022.00095.

38 Shuichi Hirahara and Akitoshi Kawamura. On characterizations of randomized computation using plain Kolmogorov complexity. *Computability*, 7(1):45–56, 2018. doi:10.3233/COM-170075.

39 Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In *31st Conference on Computational Complexity (CCC)*, volume 50 of *LIPIcs*, pages 18:1–18:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.18.

40 Shuichi Hirahara and Osamu Watanabe. On nonadaptive reductions to the set of random strings and its dense subsets. In Ding-Zhu Du and Jie Wang, editors, *Complexity and Approximation - In Memory of Ker-I Ko*, volume 12000 of *Lecture Notes in Computer Science*, pages 67–79. Springer, 2020. doi:10.1007/978-3-030-41672-0_6.

41 Rahul Ilango. Approaching MCSP from above and below: Hardness for a conditional variant and $AC^0[p]$. In *11th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 151 of *LIPIcs*, pages 34:1–34:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.34.

42 Rahul Ilango. Constant depth formula and partial function versions of MCSP are hard. In *61st IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 424–433. IEEE, 2020. doi:10.1109/FOCS46700.2020.00047.

43 Rahul Ilango, Bruno Loff, and Igor Carboni Oliveira. NP-hardness of circuit minimization for multi-output functions. In *35th Computational Complexity Conference (CCC)*, volume 169 of *LIPIcs*, pages 22:1–22:36. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.22.

44 Rahul Ilango, Hanlin Ren, and Rahul Santhanam. Robustness of average-case meta-complexity via pseudorandomness. In *54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1575–1583. ACM, 2022. doi:10.1145/3519935.3520051.

45 Neil Immerman. *Descriptive complexity*. Graduate texts in computer science. Springer, 1999. doi:10.1007/978-1-4612-0539-5.

46 Johannes Köbler, Uwe Schöning, and Klaus W. Wagner. The difference and truth-table hierarchies for NP. *RAIRO Theor. Informatics Appl.*, 21(4):419–435, 1987. doi:10.1051/ita/1987210404191.

47 Richard E. Ladner, Nancy A. Lynch, and Alan L. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–123, 1975. doi:10.1016/0304-3975(75)90016-X.

48 Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications, 4th Edition*. Texts in Computer Science. Springer, 2019. doi:10.1007/978-3-030-11298-1.

49 Yanyi Liu and Rafael Pass. On one-way functions from NP-complete problems. In *37th Computational Complexity Conference (CCC)*, volume 234 of *LIPIcs*, pages 36:1–36:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CCC.2022.36.

50 Kenneth W. Regan. A uniform reduction theorem - extending a result of J. Grollmann and A. Selman. In *Proc. International Conference on Automata, Languages, and Programming (ICALP)*, volume 226 of *Lecture Notes in Computer Science*, pages 324–333. Springer, 1986. doi:10.1007/3-540-16761-7_82.

51 Hanlin Ren and Rahul Santhanam. Hardness of KT characterizes parallel cryptography. In *36th Computational Complexity Conference (CCC)*, volume 200 of *LIPIcs*, pages 35:1–35:58. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CCC.2021.35.

52 Amit Sahai and Salil P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, 2003. doi:10.1145/636865.636868.

53 Michael Saks and Rahul Santhanam. On randomized reductions to the random strings. In *37th Computational Complexity Conference (CCC)*, volume 234 of *LIPIcs*, pages 29:1–29:30. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CCC.2022.29.

54 Rahul Santhanam. Personal communication, 2022.

55 Jacobo Torán. On the hardness of graph isomorphism. *SIAM Journal on Computing*, 33(5):1093–1108, 2004. doi:10.1137/S009753970241096X.

56 Salil Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. Springer, 2023. To appear.

57 Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(3):85–93, 1986. doi:10.1016/0304-3975(86)90135-0.

58 Heribert Vollmer. *Introduction to circuit complexity: a uniform approach*. Springer Science & Business Media, 1999. doi:10.1007/978-3-662-03927-4.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il