Circuit Size Relative to Pseudorandom Oracles[*]

Jack H. Lutz and William J. Schmidt
Department of Computer Science
Iowa State University
Ames, IA 50011

Abstract

Circuit-size complexity is compared with deterministic and nondeterministic time complexity in the presence of pseudorandom oracles. The following separations are shown to hold relative to *every* pspace-random oracle A, and relative to *almost every* oracle $A \in$ ESPACE.

(i) NP^A is *not* contained in $\mathrm{SIZE}^A(2^{\alpha n})$ for any real $\alpha < \frac{1}{3}$.

(ii) E^A is *not* contained in $\mathrm{SIZE}^A(\frac{2^n}{n})$.

Thus, neither NP^A nor E^A is contained in $\mathrm{P}^A/\mathrm{Poly}$.

In fact, these separations are shown to hold for almost every n. Since a randomly selected oracle is pspace-random with probability one, (i) and (ii) immediately imply the corresponding random oracle separations, thus improving a result of Bennett and Gill [9] and answering open questions of Wilson [47].

1 Introduction

The most fundamental problems of complexity theory appear to be those involving the relationships among deterministic polynomial time, nondeterministic polynomial time, and polynomial size circuits. Aside from the trivial observations that P $\subseteq$ NP, P $\subseteq$ PSIZE, and PSIZE $\not\subseteq$ NP, very little is known. It is likely that NP-complete problems are *combinatorially infeasible* in the sense that NP $\not\subseteq$ PSIZE, but even such extreme counter-assertions as P = NP, NP $\subseteq$ LINSIZE, E $\subseteq$ LINSIZE, and $\mathrm{E}_2 \subseteq$ PSIZE have yet to be disproven. (See sections 3 and 4 for definitions of complexity classes.)

The investigation of *relativized* complexity classes has arisen largely as an attempt to better understand the difficulty of these problems, and the types of techniques that will be required to solve them. Baker, Gill, and Solovay [5] exhibited oracles A and B such that $\mathrm{P}^A = \mathrm{NP}^A$ and $\mathrm{P}^B \neq \mathrm{NP}^B$. Wilson [47] exhibited oracles C, D, E, F, and G such that $\mathrm{NP}^C \subseteq \mathrm{LINSIZE}^C$, $\mathrm{E}^D \subseteq \mathrm{LINSIZE}^D$, $\mathrm{E}_2^E \subseteq \mathrm{PSIZE}^E$, $\mathrm{NP}^F \not\subseteq \mathrm{PSIZE}^F$, and $\mathrm{E}^G \not\subseteq \mathrm{PSIZE}^G$. Taken collectively, the oracles A through G testify that none of the open problems mentioned in the preceding paragraph will be solved by techniques that relativize to arbitrary oracles. This is taken as evidence that these problems may be very hard to solve. (Such evidence is to be interpreted with caution. For example, the theorems ALOG = P of Chandra, Kozen, and Stockmeyer [17] and IP = PSPACE of Shamir [42] have simple proofs but do not relativize, unless one modifies oracle access mechanisms to *force* them to relativize.)

[*] This research was supported in part by NSF Grants CCR–8809238 and CCR–9157382, and by a National Science Foundation Graduate Fellowship.

Unfortunately, oracle *existence* results of the above type provide no evidence regarding the truth or falsity of the underlying conjectures. As a remedy for this situation, Bennett and Gill [9] proposed the study of complexity classes relative to *randomly selected* oracles. In this scheme, an oracle $A \subseteq \{0,1\}^*$ is chosen probabilistically by using an independent toss of a fair coin to decide whether each string $x \in \{0,1\}^*$ is in A. Bennett and Gill [9] proved (among other things) that $\mathrm{P}^A \neq \mathrm{NP}^A$ holds with probability 1 when the oracle A is so selected. That is, the conjecture $\mathrm{P} \neq \mathrm{NP}$ holds relative to *almost every* oracle. Moreover, Bennett and Gill [9] formulated and proposed the *random oracle hypothesis*, which posits that *any* reasonably formed conjecture that holds relative to almost every oracle is in fact true. Thus, the *random oracle result,* $\mathrm{P}^A \neq \mathrm{NP}^A$ with probability 1, is regarded as *evidence* that $\mathrm{P} \neq \mathrm{NP}$.

The random oracle hypothesis was refuted by Kurtz [25], so it is not clear that random oracle separations of the above type provide evidence that the corresponding unrelativized conjectures are true. In fact, recent work of Book [11] indicates that such separations do *not* provide such evidence. Nevertheless, random oracle separations continue to be of interest. Notably, Cai [14] and Babai [4] have proven that $\mathrm{PH} \neq \mathrm{PSPACE}$ relative to almost every oracle; Kurtz, Mahaney, and Royer [26] have proven that the Berman-Hartmanis [10] isomorphism conjecture fails relative to almost every oracle; and Beigel [8] has shown that almost every oracle supports a fine hierarchy between UP and NP, based upon the number of accepting computations.

At our present state of knowledge (i.e., lack thereof), results of this type merit careful attention. There are several reasons for this. First, more often than not, random oracle results correspond to our intuitive conjectures about the unrelativized questions. A scientific analysis and explanation of this correspondence and its limitations is likely to be instructive.

Second, oracle properties that hold with probability 1 have proven to be useful for characterizing complexity classes. Bennett and Gill [9] and Ambos-Spies [2] have shown that a language L is in BPP if and only if $L \in \mathrm{P}^A$ for almost every oracle A. Nisan and Wigderson [40] have given a similar characterization of the Arthur-Merlin class AM of Babai [3], showing that a language L is in AM if and only if $L \in \mathrm{NP}^A$ for almost every oracle A. Other complexity classes have been given similar characterizations by Ambos-Spies [2], Tang and Watanabe [45], and Book and Tang [12]. Results of this type indicate that a systematic study of random oracle properties may be a fruitful enterprise.

Random oracle results, though interesting, are uninformative in a crucial respect. For example, consider the random oracle separation of P from NP. This results tells us that *almost every* oracle A achieves the separation $\mathrm{P}^A \neq \mathrm{NP}^A$, but gives no information as to *which* oracles A achieve this separation.

To deal with this matter, this paper introduces *pseudorandom relativization*, a new, more sophisticated successor to the random oracle technique. Roughly speaking, a pseudorandom oracle separation result for a relativized separation condition SEP^A (e.g., the condition $\mathrm{P}^A \neq \mathrm{NP}^A$) identifies a *level of (pseudo)randomness* Δ for which the following two conditions hold.

(i) *Every* oracle A that is Δ-random satisfies the condition SEP^A.

(ii) A randomly selected oracle A is Δ-random with probability 1.

Taken together, of course, (i) and (ii) give the corresponding random oracle separation, namely that a randomly selected oracle A satisfies the condition SEP^A with probability 1.

However, (i) gives more information than this by identifying the Δ-randomness of any *individual* oracle A as a *sufficient* condition for SEP^A to hold.

The notion of Δ-randomness used here was developed and investigated by Lutz [35, 38] and is discussed in some detail in section 3 below. It is the level $\Delta = \mathrm{pspace}$ that is of interest in this paper. Briefly, a language A (equivalently, the characteristic sequence of A) is pspace-random if and only if it has no "pspace-specifiable special properties", i.e., if it is in no pspace-measure 0 set of languages. (See section 3 for details.) This definition resembles the Martin-Löf [39] definition of *random* sequences; indeed every random sequence is pspace-random. Since Martin-Löf [39] proved that a randomly selected oracle A is random with probability 1, it immediately follows that property (ii) above holds when $\Delta = \mathrm{pspace}$. However, much more is true. The definition of Δ-randomness is based on the resource-bounded measure theory developed by Lutz [35, 37]. This underlying measure theory articulates the internal measure-theoretic structure of various complexity classes and, as it turns out, ensures that most *decidable* languages are Δ-random. For example, almost every language decidable in $2^{\mathrm{polynomial}}$ space is pspace-random [38]. Since no decidable (or even recursively enumerable) language is random [39], then, pspace-random languages are *pseudo*random, with pspace specifying the "level of randomness".

It is shown in Corollary 5.2 below that, relative to *every* pspace-random oracle A, $\mathrm{P}^A \neq \mathrm{NP}^A$. Thus (i) and (ii) above hold for this separation property when $\Delta = \mathrm{pspace}$. This refines the random oracle separation of Bennett and Gill [9]. (Such refinements are not automatic. For example, the separation $\mathrm{P}^A \not\subseteq \mathrm{REC}$ holds for a randomly selected oracle with probability one, but fails for every decidable pspace-random oracle.)

This improvement, from randomly selected relativization to pseudorandom relativization, is only one dimension of the progress reported in this paper. Equally significant is the fact that the results reported here give quantitative comparisons of circuit-size complexity with deterministic and nondeterministic time complexity.

The main result of this paper, Theorem 5.1, compares circuit size to nondeterministic time, relative to pseudorandom oracles. After constructing the above-mentioned oracles C and F, Wilson [47] asked what occurs with high probability relative to a randomly selected oracle. Theorem 5.1 below implies immediately that oracle F represents the typical case, i.e., that $\mathrm{NP}^A \not\subseteq \mathrm{PSIZE}^A$ holds with probability 1. However, Theorem 5.1 is the much stronger fact that, for every real $\alpha < \frac{1}{3}$, the condition $\mathrm{NP}^A \not\subseteq \mathrm{SIZE}^A_{\mathrm{i.o.}}(2^{\alpha n})$ is a pspace-*test* (defined in section 3). This is stronger than the answer to Wilson's question in the following three respects.

(a) The fact that the separation condition is a pspace-test implies that the separation holds for *every* pspace-random oracle A and for *almost every* oracle A that is decidable in 2^{linear} space.

(b) The separation holds even when the size bound on the right is $2^{\alpha n}$ ($\alpha < \frac{1}{3}$). That is, the separation condition forces NP^A to contain problems with exponential circuit-size complexity relative to A.

(c) The separation holds for *almost every* n in the sense that it holds even if the circuits on the right are only required to be small for infinitely many n.

Thus Theorem 5.1 is a very strong pseudorandom oracle separation of nondeterministic polynomial time from submaximal exponential circuit size.

Theorem 6.2 gives an even stronger separation for deterministic exponential time. In this case, the result states that the condition $\mathrm{E}^A \not\subseteq \mathrm{SIZE}^A_{\mathrm{i.o.}}(\frac{2^n}{n})$ is a pspace-test. This answers another open question of Wilson [47], since it implies that, of the above-mentioned oracles D, E, and G, oracle G represents the probability-one case. Moreover, Theorem 6.2 is stronger than the answer to Wilson's question in respects (a), (b), and (c) above. In fact, (b) is even stronger in this case because the $\frac{2^n}{n}$ circuit-size lower bound is essentially maximal. Theorem 6.2 is a very strong pseudorandom oracle separation of deterministic exponential time from slightly-submaximal circuit size.

2 Preliminaries

A *binary string* is a finite sequence $x \in \{0,1\}^*$. A *binary sequence* is an infinite sequence $x \in \{0,1\}^\infty$. We write $\{0,1\}^n$ for the set of strings of length n, and $\{0,1\}^{\leq n}$ for the set of strings of length at most n. We use variables x, y, z, etc., to denote strings or sequences. We write $|x|$ for the length of x. Thus $|x| \in \mathbf{N} \cup \{\infty\}$, where $\mathbf{N}$ is the set of nonnegative integers. The unique string of length 0 is λ, the empty string. We write $x[i]$ for the i^{th} bit of x. Thus $x = x[0]x[1]\cdots x[|x|-1]$ if x is a string.

If x is a string and y is a string or sequence, then xy is the concatenation of x and y. If x is already a sequence, then $xy = x$. If x is a string and $k \in \mathbf{N}$, then x^k is the k-fold concatenation of x with itself. Thus $x^0 = \lambda$ and $x^{k+1} = xx^k$.

Complexity classes are usually defined as sets of languages. A *language* here is a set $L \subseteq \{0,1\}^*$, i.e., a set of binary strings. We fix the lexicographic enumeration $s_0 = \lambda$, $s_1 = 0, s_2 = 1, s_3 = 00, \ldots$ of $\{0,1\}^*$ and identify each language L with its *characteristic sequence* $\chi_L \in \{0,1\}^\infty$ defined by

$$\chi_L[k] \quad = \quad \begin{cases} 1 & \text{if } s_k \in L \\ 0 & \text{if } s_k \notin L. \end{cases}$$

This identifies the set $\mathcal{P}(\{0,1\}^*)$ of all languages with the set $\{0,1\}^\infty$ of all binary sequences. We use X, Y, Z, etc., to denote sets of languages (equivalently, to denote sets of binary sequences). The *complement* of a set X is $X^c = \mathcal{P}(\{0,1\}^*) \backslash X = \{0,1\}^\infty \backslash X$. We sometimes write $L_{\leq n}$ for $L \cap \{0,1\}^{\leq n}$.

We fix once and for all a one-to-one pairing function $\langle , \rangle$ from $\{0,1\}^* \times \{0,1\}^*$ onto $\{0,1\}^*$ such that the pairing function and its associated projections, $\langle x, y\rangle \mapsto x$ and $\langle x, y\rangle \mapsto y$ are computable in polynomial time. We insist further that $\langle x, y\rangle \in \{0\}^*$ if and only if $x, y \in \{0\}^*$. This condition canonically induces a pairing function $\langle , \rangle$ from $\mathbf{N} \times \mathbf{N}$ onto $\mathbf{N}$.

We say that a condition $\varphi(n)$ holds *almost everywhere* (*a.e.*) if it holds for all but finitely many $n \in \mathbf{N}$. We say that $\varphi(n)$ holds *infinitely often* (*i.o.*) if it holds for infinitely many $n \in \mathbf{N}$.

We use the discrete logarithm

$$\log n = \min\{k \in \mathbf{N} \mid 2^k \geq n\}.$$

Note that $\log 0 = 0$.

We will use the following combinatorial bound in section 5.

Proposition 2.1 (Chernoff [18]). For $0 < b < a < 1$,

$$\sum_{i=0}^{bt} \binom{t}{i} a^i (1-a)^{t-i} \leq 2^{-ct},$$

where $c = b(\log b - \log a) + (1-b)[\log(1-b) - \log(1-a)] > 0$. □

3 Resource-Bounded Measure and Pseudorandomness

In this section we review those aspects of resource-bounded measure and pseudorandomness that are essential to this paper. The interested reader is referred to [35, 37, 38] for a more complete treatment.

We work in two alphabets, the binary alphabet $\{0,1\}$ and the extended binary alphabet $\Sigma = \{0,1,\perp\}$. The symbol $\perp$ ("bottom") denotes an "undefined bit." We fix the partial ordering $\sqsubseteq$ of Σ in which $\perp \sqsubseteq 0$, $\perp \sqsubseteq 1$, and 0 and 1 are incomparable. Given a string or sequence $x \in \Sigma^* \cup \Sigma^\infty$, we write $x[i]$ for the i^{th} bit of x and $x[i..j]$ for the string consisting of the i^{th} through j^{th} bits of x. We also fix the standard enumeration $s_0 = \lambda, s_1 = 0, s_2 = 1, s_3 = 00, \ldots$ of $\{0,1\}^*$, and write $x[w] = x[i]$ whenever $w = s_i$ and $0 \leq i < |x|$. We extend $\sqsubseteq$ bitwise to strings and sequences, i.e., $x \sqsubseteq y$ iff $(\forall i \in \mathbf{N}) x'[i] \sqsubseteq y'[i]$, where $x' = x$ if $|x| = \infty$, $x' = x\perp^\infty$ if $|x| < \infty$, and y' is defined similarly. The *cylinder specified by* a string $x \in \Sigma^*$ is $C_x = \{A \subseteq \{0,1\}^* \mid x \sqsubseteq \chi_A\}$, where $\chi_A \in \{0,1\}^\infty$ is the *characteristic sequence* of A, i.e., each $\chi_A[i]$ is 1 if $s_i \in A$ and 0 otherwise. We use the symbol $\top$ ("top") to specify the empty set, i.e., $C_\top = \emptyset$. For $x, y \in \Sigma^*$, we let $x \wedge y$ be the shortest string such that $C_{x\wedge y} = C_x \cap C_y$. Note that $x \wedge y = \top$ if x and y are *incompatible*, i.e., if $C_x \cap C_y = \emptyset$. The *measure* $\mu(x)$ of a cylinder C_x is the probability that $A \in C_x$ when $A \subseteq \{0,1\}^*$ is chosen according to the random experiment in which an independent toss of a fair coin is used to decide whether each string $w \in \{0,1\}^*$ is in A. Thus if we let $\#(b,x)$ denote the number of occurrences of the symbol b in the string x and define

$$\|x\| = \begin{cases} \#(0,x) + \#(1,x) & \text{if } x \in \Sigma^* \\ \infty & \text{if } x = \top, \end{cases}$$

then $\mu(x) = 2^{-\|x\|}$ for all $x \in \Sigma^* \cup \{\top\}$.

We fix once and for all a one-to-one pairing function $\langle , \rangle$ from $\{0,1\}^* \times \{0,1\}^*$ onto $\{0,1\}^*$ such that the pairing function and its associated projections, $\langle x,y\rangle \mapsto x$ and $\langle x,y\rangle \mapsto y$ are computable in polynomial time. We insist further that this pairing function satisfy the following condition for all $x, y \in \{0,1\}^*$: $\langle x,y\rangle \in \{0\}^*$ if and only if $x, y \in \{0\}^*$. This condition canonically induces a pairing function $\langle , \rangle$ from $\mathbf{N} \times \mathbf{N}$ onto $\mathbf{N}$. We write $\langle x,y,z\rangle$ for $\langle x, \langle y,z\rangle\rangle$, etc., so that tuples of any fixed length are coded by the pairing function.

We let $\mathbf{D} = \{m2^{-n} \mid m,n \in \mathbf{N}\}$ be the set of nonnegative *dyadic rationals.* Many functions in this paper take their values in $\mathbf{D}$ or in $[0,\infty)$, the set of nonnegative real numbers. In fact, with the exception of some functions that map into $[0,\infty)$, our functions are of the form $f : X \to Y$, where each of the sets X, Y is $\mathbf{N}$, $\{0,1\}^*$, $\mathbf{D}$, or some cartesian product of these sets. Formally, in order to have uniform criteria for their computational complexities, we regard all such functions as mapping $\{0,1\}^*$ into $\{0,1\}^*$. For example, a function $f : \mathbf{N}^2 \times \{0,1\}^* \to \mathbf{N} \times \mathbf{D}$ is formally interpreted as a function $\tilde{f} : \{0,1\}^* \to \{0,1\}^*$. Under this interpretation, $f(i,j,w) = (k,q)$ means that $\tilde{f}(\langle 0^i, \langle 0^j, w\rangle\rangle) = \langle 0^k, \langle u, v\rangle\rangle$, where u and v are the binary representations of the integer and fractional parts of q, respectively. Moreover, we only care about the values of $\tilde{f}$ for arguments of the form $\langle 0^i, \langle 0^j, w\rangle\rangle$, and we insist that these values have the form $\langle 0^k, \langle u, v\rangle\rangle$ for such arguments.

For a function $f : \mathbf{N} \times X \to Y$ and $k \in \mathbf{N}$, we define the function $f_k : X \to Y$ by $f_k(x) = f(\langle 0^k, x\rangle)$. We then regard f as a "uniform enumeration" of the functions $f_0, f_1, f_2, \ldots$. For a function $f : \mathbf{N}^n \times X \to Y$ ($n \geq 2$), we write $f_{k,l} = (f_k)_l$, etc. For a function $f : \{0,1\}^* \to \{0,1\}^*$, we write f^n for the n-fold composition of f with itself.

We work with the resource bound

$$\text{pspace} = \{f : \{0,1\}^* \to \{0,1\}^* \mid f \text{ is computable in polynomial space}\}.$$

(The length $|f(x)|$ of the output *is* included as part of the space used in computing f.)

Resource-bounded measure and pseudorandomness were originally developed in terms of "modulated covering by cylinders" [32, 33, 34]. Though the main results of these papers are true, the underlying development was technically flawed. This situation was remedied in [35], where resource-bounded measure was reformulated in terms of density functions. We review relevant aspects of the latter formulation here.

A *density function* is a function $d : \{0,1\}^* \to [0,\infty)$ satisfying

$$d(x) \geq \frac{d(x0) + d(x1)}{2}$$

for all $x \in \{0,1\}^*$. The *global value* of a density function d is $d(\lambda)$. An n-dimensional *density system (n-DS)* is a function $d : \mathbf{N}^n \times \{0,1\}^* \to [0,\infty)$ such that $d_{\vec{k}}$ is a density function for every $\vec{k} \in \mathbf{N}^n$. It is sometimes convenient to regard a density function as a 0-DS.

A *computation* of an n-DS d is a function $\hat{d} : \mathbf{N}^{n+1} \times \{0,1\}^* \to \mathbf{D}$ such that

$$\left|\hat{d}_{\vec{k},r}(x) - d_{\vec{k}}(x)\right| \leq 2^{-r} \tag{3.1}$$

for all $\vec{k} \in \mathbf{N}^n$, $r \in \mathbf{N}$, and $x \in \{0,1\}^*$. A pspace-*computation* of an n-DS d is a computation $\hat{d}$ such that $\hat{d} \in$ pspace. An n-DS is pspace-*computable* if there exists a pspace-computation $\hat{d}$ of d. (Note that (3.1) implies that

$$d_{\vec{k}}(x) = \lim_{r\to\infty} \hat{d}_{\vec{k},r}(x)$$

for all $\vec{k} \in \mathbf{N}^n$ and $x \in \{0,1\}^*$.)

The *set covered by* a density function d is

$$S[d] = \bigcup_{x \in \{0,1\}^* \wedge d(x) \geq 1} C_x.$$

A density function d *covers* a set X of languages if $X \subseteq S[d]$. A *null cover* of a set X of languages is a 1-DS d such that, for all $k \in \mathbf{N}$, d_k covers X with global value $d_k(\lambda) \leq 2^{-k}$.

It is easy to show [37] that a set X of languages has classical Lebesgue measure 0 (i.e., probability 0 in the coin-tossing random experiment) if and only if there exists a null cover of X. In this paper we are interested in the situation where the null cover d is pspace-computable.

A pspace-*null cover* of a set X of languages is a null cover of X that is computable. A set X has pspace-*measure 0*, and we write $\mu_{\text{pspace}}(X) = 0$, if there exists a pspace-null cover of X. A set X has pspace-*measure 1*, and we write $\mu_{\text{pspace}}(X) = 1$, if $\mu_{\text{pspace}}(X^c) = 0$. Thus a set X has pspace-measure 0 if pspace provides sufficient computational resources to uniformly enumerate pspace-covers of X with rapidly vanishing total measure.

We say that a set X has *measure 0 in* ESPACE $=$ DSPACE(2^{linear}), and write $\mu(X \mid \text{ESPACE}) = 0$, if $\mu_{\text{pspace}}(X \cap \text{ESPACE}) = 0$. A set X has *measure 1 in* ESPACE, and we write $\mu(X \mid \text{ESPACE}) = 1$, if $\mu(X^c \mid \text{ESPACE}) = 0$. In this case we say that *almost every* sequence in ESPACE is in X. The following routine result of [35] relates pspace-measure to measure in ESPACE and to classical Lebesgue measure.

Lemma 3.1. Let X be a set of languages.

(a) If $\mu_{\text{pspace}}(X) = 0$, then $\mu(X \mid \text{ESPACE}) = 0$.

(b) If $\mu_{\text{pspace}}(X) = 0$, then $\mu(X) = 0$, where $\mu(X)$ is the classical Lebesgue measure of the set X. □

It is shown in [35] that the above definitions endow ESPACE with internal measure-theoretic structure. Specifically, if $\mathcal{I}$ is either the collection $\mathcal{I}_{\text{pspace}}$ of all pspace-measure 0 sets or the collection $\mathcal{I}_{\text{ESPACE}}$ of all sets of measure 0 in ESPACE, then $\mathcal{I}$ is a "pspace-ideal," i.e., is closed under subsets, finite unions, and "pspace-unions" (countable unions that can be generated in polynomial space). More importantly, it is shown that the ideal $\mathcal{I}_{\text{ESPACE}}$ is a *proper* ideal, i.e., that ESPACE does *not* have measure 0 in ESPACE.

We need a polynomial notion of convergence for infinite series. All our series here consist of nonnegative terms. A *modulus* for a series $\sum_{n=0}^{\infty} a_n$ is a function $m : \mathbf{N} \to \mathbf{N}$ such that

$$\sum_{n=m(j)}^{\infty} a_n \leq 2^{-j}$$

for all $j \in \mathbf{N}$. A series is p-*convergent* if it has a modulus that is a polynomial. We will use the following sufficient condition for p-convergence. (This well-known lemma is easily verified by routine calculus.)

Lemma 3.2. Let $a_t \in [0, \infty)$ for all $t \in \mathbf{N}$. If there exists a real $\varepsilon > 0$ such that $a_t \leq 2^{-t^{\varepsilon}}$ for all sufficiently large $t \in \mathbf{N}$, then the series $\sum_{t=0}^{\infty} a_t$ is p-convergent. □

In sections 5 and 6 we will use two theorems that provide sufficient conditions for sets to have pspace-measure 0. The first is a special case (for pspace) of a resource-bounded version of the classical first Borel-Cantelli lemma.

Theorem 3.3 (Borel [13], Cantelli [15], Lutz [35]). If d is a pspace-computable 1-DS such that the series $\sum\limits_{t=0}^{\infty} d_t(\lambda)$ is p-convergent, then

$$\mu_{\mathrm{pspace}}\left(\bigcap_{k=0}^{\infty}\bigcup_{t=k}^{\infty} S[d_t]\right) = \mu_{\mathrm{pspace}}\left(\{A \mid A \in S[d_t] \text{ i.o.}\}\right) = 0. \qquad \square$$

Our second sufficient condition for a set to have pspace-measure 0 involves space-bounded Kolmogorov complexity. Kolmogorov complexity (also called program-size complexity) was introduced independently by Solomonoff [44], Kolmogorov [23], and Chaitin [16]. Time-bounded, space-bounded, and conditional Kolmogorov complexities have since been studied by Hartmanis [19], Sipser [43], Levin [27], Huynh [20], Ko [22], Longpré [30], Lutz [32, 35], and many others. For an overview of work in this area, see Kolmogorov and Uspenskii [24] or Li and Vitanyi [28]. We begin with some terminology.

Given a deterministic machine M, a space bound $t : \mathbf{N} \to \mathbf{N}$, a language $L \subseteq \{0,1\}^*$, and a natural number n, the *t-space-bounded Kolmogorov complexity* of $L_{\leq n}$ relative to M is

$$\mathrm{KS}_M^t(L_{\leq n}) = \min\{|\pi| \mid M(\pi, n) = \chi_{L_{\leq n}} \text{ in } \leq t(2^n) \text{ space}\},$$

i.e., the length of the shortest program π such that M, on input (π, n), outputs the characteristic string of $L_{\leq n}$ and halts without using more than $t(2^n)$ workspace.

Well-known simulation techniques show that there exists a machine U that is *optimal* in the sense that for each machine M there is a constant c such that for all t, L, and n we have

$$\mathrm{KS}_U^{ct+c}(L_{\leq n}) \leq \mathrm{KS}_M^t(L_{\leq n}) + c.$$

As usual, we fix an optimal machine U and omit it from the notation.

Theorem 3.4 (Lutz [35]). Let q be any polynomial, let $\varepsilon > 0$ be real, and let X be the set of all languages L such that $\mathrm{KS}^q(L_{\leq n}) < 2^{n+1} - 2^{\varepsilon n}$ i.o. Then $\mu_{\mathrm{pspace}}(X) = 0$. □

We end this section with a discussion of pseudorandom languages.

A *pspace-test* is a set X such that $\mu_{\mathrm{pspace}}(X) = 1$. A language $A \subseteq \{0,1\}^*$ *passes* a pspace-test X if $A \in X$. A language $A \subseteq \{0,1\}^*$ is *pspace-random* if A passes all pspace-tests. It is easily shown that every language A that is *random* (i.e., whose characteristic sequence $\chi_A \in \{0,1\}^\infty$ is random) in the sense of Martin-Löf [39] is also pspace-random. As discussed in the introduction, this implies that a randomly selected language is pspace-random with probability one. Thus, separations that hold relative to every pspace-random oracle also hold relative to a randomly selected oracle with probability one. It has also been shown in [38] that results about pspace-random sequences give information about reasonably low complexity classes. Specifically, *almost every* language in $\mathrm{E_2SPACE} = \mathrm{DSPACE}(2^{\mathrm{polynomial}})$ is pspace-random, but *no* language in ESPACE is pspace-random.

There are several additional properties of pspace-random languages that support characterizing them as pseudorandom. For example, every pspace-random language L has nearly maximal circuit-size complexity and nearly maximal space-bounded Kolmogorov complexity almost everywhere [35]. Also, every pspace-random sequence $x \in \{0,1\}^\infty$ is a structurally adequate source for every bounded-error probabilistic machine [36].

4 Relativized Complexity

We use the oracle Turing machine and the oracle circuit as our models of relativized uniform and nonuniform complexity, respectively. For a formal definition of the oracle Turing machine, see for example Balcázar, Díaz, and Gabarró [7]. Recall that we write DTIME($T(n)$) [resp., NTIME($T(n)$)] for the set of languages accepted by deterministic [resp., nondeterministic] Turing machines in $O(T(n))$ time. Analogously, we write DTIME$^A(T(n))$ [resp., NTIME$^A(T(n))$] for the set of languages accepted by deterministic [resp., nondeterministic] oracle Turing machines in $O(T(n))$ time using oracle set A. We will use the following relativized and unrelativized uniform complexity classes.

$$\begin{array}{lll@{\qquad}lll}
\mathrm{P} & = & \bigcup_{k\geq 0}\mathrm{DTIME}(n^k) & \mathrm{P}^A & = & \bigcup_{k\geq 0}\mathrm{DTIME}^A(n^k)\\
\mathrm{NP} & = & \bigcup_{k\geq 0}\mathrm{NTIME}(n^k) & \mathrm{NP}^A & = & \bigcup_{k\geq 0}\mathrm{NTIME}^A(n^k)\\
\mathrm{E} & = & \bigcup_{c\geq 0}\mathrm{DTIME}(2^{cn}) & \mathrm{E}^A & = & \bigcup_{c\geq 0}\mathrm{DTIME}^A(2^{cn})
\end{array}$$

A (*deterministic*) *oracle circuit* is a directed acyclic graph $\gamma = (V, E)$ with vertex set $V = I \cup G_s \cup G_o$, where $I = \{w_1, \ldots, w_n\}$ is the set of *inputs*, G_s is the set of *standard gates*, and G_o is the set of *oracle gates*. Each input has indegree 0; each standard gate has indegree 0, 1, or 2; and each oracle gate may have any indegree $k \in \mathbf{N}$. Each standard gate of indegree 0 is labeled either by the constant 0 or by the constant 1. Each standard gate of indegree 1 is labeled either by the identity function ID: $\{0,1\} \to \{0,1\}$ or by the negation function NOT: $\{0,1\} \to \{0,1\}$. Each standard gate of indegree 2 is labeled either by the conjunction AND: $\{0,1\}^2 \to \{0,1\}$ or by the disjunction OR: $\{0,1\}^2 \to \{0,1\}$. The function computed by an oracle gate is dependent on the *oracle set* $A \subseteq \{0,1\}^*$ that is "attached" to the circuit. A k-input oracle gate computes $A \cap \{0,1\}^k$; thus all k-input oracle gates in γ compute the same function. Intuitively, when an oracle gate g is presented with a string of inputs $x \in \{0,1\}^k$, g "queries" A about x, producing 1 if $x \in A$ and 0 otherwise. Without loss of generality, we insist that each oracle circuit contains at most one 0-input oracle gate.

An n-input oracle circuit γ with attached oracle set A computes a Boolean function $\gamma^A : \{0,1\}^n \to \{0,1\}$ in the usual way. For $w \in \{0,1\}^n$, $g^A(w)$ is the value computed at gate g of γ, and $\gamma^A(w)$ is the value computed at the unique *output gate* of γ, when the inputs are assigned the bits $w_1, \ldots, w_n$ of w. The *set computed by* an n-input oracle circuit γ *relative to* an oracle A is then the set of all $w \in \{0,1\}^n$ such that $\gamma^A(w) = 1$. Two n-input oracle circuits γ_1 and γ_2 are *functionally distinct* if there exists an oracle A relative to which γ_1 and γ_2 compute different sets.

This model was first introduced by Wilson [46, 47]. As defined in these references, the size of a circuit $\gamma = (V, E)$ is equal to $|E|$, i.e., the number of "wires" in γ, or the sum of the indegrees of γ's component gates. We will find it convenient to use the following "almost equivalent" definition. The *size* of an oracle circuit γ is given by

$$\mathrm{size}(\gamma) = 2|G_s| + \sum_{g \in G_o} k_g,$$

where k_g is the indegree of oracle gate g. Thus every standard gate is considered to contribute a count of 2 to the size of the circuit, rather than its actual indegree. This will facilitate some counting arguments below.

The *circuit-size complexity* of a language $L \subseteq \{0,1\}^*$ *with respect to* an oracle set A is the function $\mathrm{CS}_L^A : \mathbf{N} \to \mathbf{N}$ defined by

$$\mathrm{CS}_L^A(n) = \min\{\mathrm{size}(\gamma) \mid \gamma^A \text{ computes } L \cap \{0,1\}^n\}.$$

We define the relativized circuit-size complexity classes

$$\begin{array}{lcl} \mathrm{SIZE}^A(f(n)) & = & \{L \mid \mathrm{CS}_L^A(n) \leq f(n) \text{ a.e.}\} \\ \mathrm{SIZE}_{\text{i.o.}}^A(f(n)) & = & \{L \mid \mathrm{CS}_L^A(n) \leq f(n) \text{ i.o.}\} \\ \mathrm{LINSIZE}^A & = & \bigcup_{k\geq 0} \mathrm{SIZE}^A(kn) \\ \mathrm{PSIZE}^A & = & \bigcup_{k\geq 0} \mathrm{SIZE}^A(n^k) \\ \mathrm{PSIZE}_{\text{i.o.}}^A & = & \bigcup_{k\geq 0} \mathrm{SIZE}_{\text{i.o.}}^A(n^k) \end{array}$$

The oracle circuit model is an extension of the unrelativized circuit model in which oracle gates do not appear, and in which the size of a circuit is simply the number of its constituent gates. In this model, the *circuit-size complexity* of a language $L \subseteq \{0,1\}^*$ is the function $\mathrm{CS}_L(n) : \mathbf{N} \to \mathbf{N}$ defined by

$$\mathrm{CS}_L(n) = \min\{\mathrm{size}(\gamma) \mid \gamma \text{ computes } L \cap \{0,1\}^n\}.$$

Using this function, the unrelativized circuit-size classes $\mathrm{SIZE}(f(n))$, $\mathrm{SIZE}_{\text{i.o.}}(f(n))$, LINSIZE, PSIZE, and $\mathrm{PSIZE}_{\text{i.o.}}$ are defined analogously to their relativized counterparts.

Fix a standard enumeration of all oracle circuits in which no circuit precedes a circuit of lesser size. Call an n-input oracle circuit γ *novel* for n if γ is functionally distinct from every n-input oracle circuit that precedes it. Observe that testing whether a given circuit γ is novel for n can clearly be done using workspace that is polynomial in 2^n.

Let $\mathcal{C}$ be a class of languages, and let $\mathcal{F}$ be a class of *advice functions* from $\mathbf{N}$ into $\{0,1\}^*$. As in Karp and Lipton [21], we define $\mathcal{C}/\mathcal{F}$ to be the class of languages B for which there exists a set $C \in \mathcal{C}$ and a function $f \in \mathcal{F}$ such that $B = \{x \mid \langle x, f(|x|)\rangle \in C\}$. The standard proof (see, for instance, Schöning [41]) that PSIZE = P/Poly may easily be modified to show that $\mathrm{PSIZE}^A = \mathrm{P}^A/\mathrm{Poly}$, and that $\mathrm{PSIZE}_{\text{i.o.}}^A = \mathrm{P}^A/\mathrm{Poly}^{\text{i.o.}}$ for every oracle A.

Recall that a *partition* of an integer s is a nonincreasing sequence of positive integers $(s_1, s_2, \ldots, s_k)$ such that $\sum_{i=1}^k s_i = s$. Define a *gate partition* of an integer s to be a partition $(t_1, t_2, \ldots, t_k)$ of s with the special property that each t_i with a value of 2 is also assigned a label from the set $\{\text{oracle}, \text{standard}\}$, with no such oracle label preceding a standard one. Intuitively, a gate partition represents a particular set of gate types that may be used to construct a circuit of size s. We will occasionally abuse notation and use the term "gate partition" to refer to this set of gate types. We say that a gate partition $(t_1, t_2, \ldots, t_k)$ of s is *equivalent* to a partition $(s_1, s_2, \ldots, s_k)$ of the same integer s if $s_i = t_i$ for each $1 \leq i \leq k$, regardless of the special labels.

Lemma 4.1. For each positive integer s, the number G of gate partitions of s is less than $(2e)^s$.

Proof. We can put a weak upper bound on the number P of partitions of s by counting the number of ways of putting numbers between 1 and s into no more than s slots, under the condition that the numbers are selected in nonincreasing order such that the sum of the numbers selected never exceeds s. There are no more than s ways to select the first number, no more than $\left\lfloor \frac{s}{2} \right\rfloor$ ways to select the second number, and so on. This gives

$$P \leq \left\lfloor \tfrac{s}{1} \right\rfloor \cdot \left\lfloor \tfrac{s}{2} \right\rfloor \cdot \cdots \cdot \left\lfloor \tfrac{s}{s} \right\rfloor \leq \frac{s^s}{s!} < \frac{s^s}{\left(\frac{s}{e}\right)^s} = e^s.$$

Since each partition is equivalent to no more than $s+1$ gate partitions, we have $G \leq (s+1)P < (2e)^s$. □

Lemma 4.2. Given any $n, s \in \mathbf{N}$ with $s > n$, the number $H(s)$ of functionally distinct n-input oracle circuits γ such that $\operatorname{size}(\gamma) = s$ is less than $2685(4es)^s$.

Proof. For each gate partition π of s, let $F_\pi(s)$ be the number of functionally distinct such circuits having gates sized according to π. By Lemma 4.1, we have

$$H(s) < (2e)^s \max_\pi F_\pi(s). \tag{4.1}$$

In fact, every oracle circuit is functionally identical to one containing at most one oracle gate of size 0, so (4.1) holds even if the maximum is only taken over those gate partitions π that allow at most one oracle gate of size 0. It thus suffices to show that

$$F_\pi(s) \leq 2685(2s)^s \tag{4.2}$$

holds for every such π.

Let π be such a gate partition allowing g standard gates and k oracle gates. The total size of the oracle gates is thus $m = s - 2g$. Since π allows at most one oracle gate of size 0, we have $k + 2g \leq s + 1$. Since $s > n$, it follows that

$$n + k + 2g \leq 2s. \tag{4.3}$$

We now prove (4.2). There are two cases. First, suppose that $g = 0$. Then $F_\pi(s)$ is simply the number of ways to select the source for each of the s inputs to the oracle gates. Each gate input may be taken from one of the n circuit inputs or from one of the $k-1$ other gate inputs. Thus $F_\pi(s) < (n+k)^s$. It follows by (4.3) that $F_\pi(s) < (2s)^s$, so (4.2) is affirmed in this case.

Next, suppose that $g > 0$. Observe that the number of potential sources of input for any gate is less than $n + k + g$. There are m oracle gate inputs, each of which may come from one of the n circuit inputs or from one of the $k + g - 1$ other gate outputs. Thus there are fewer than $(n+k+g)^m$ ways to configure the oracle gates. For each of the standard gates, there are 6 choices of gate type and fewer than $n + k + g$ choices of source for each of its at most 2 inputs. Thus there are fewer than $6^g(n+k+g)^{2g}$ ways to configure the standard gates. The total number of circuits is thus less than $6^g(n+k+g)^{m+2g} = 6^g(n+k+g)^s$. By (4.3), this is less than or equal to $6^g(2s)^s$. Note, however, that these circuits are not all functionally distinct. Each of these circuits is equivalent to at least $(g-1)!$ circuits obtained by permuting its non-output standard gates. Since all these circuits are sized according to π, it follows that

$$F_\pi(s) < \frac{6^g(2s)^s}{(g-1)!} = \frac{g \cdot 6^g(2s)^s}{g!}.$$

Using the weak Stirling approximation $g! > (\frac{g}{e})^g$, this gives $F_\pi(s) < g^{1-g}(6e)^g(2s)^s$. Routine calculus shows that $g^{1-g}(6e)^g$ takes its absolute maximum at the solution of $\ln\frac{g}{6} = \frac{1}{g}$. This solution satisfies $6.93 < g < 6.94$ and gives a maximum value less than 2685. Thus we have $F_\pi(s) < 2685(2s)^s$, again affirming (4.2). □

It should be noted that care must be taken in comparing standard and relativized circuit-size results. Define a *degenerate oracle circuit* to be a circuit whose size is defined according to the oracle circuit model described above, but which does not contain any oracle gates. Then it is clear that any degenerate oracle circuit of size s is equivalent to a standard circuit of size exactly $s/2$. As a result of the fact that any language accepted by a family of standard circuits can be accepted by a family of oracle circuits, a proof of Lupanov [31] gives us the following useful fact.

Proposition 4.3. For every language L and oracle A,

$$\mathrm{CS}_L^A(n) \le \frac{2^{n+1}}{n}\left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right).$$

□

If we write pspace^A for the set of all functions computable in polynomial workspace relative to oracle A, and ESPACE^A for DSPACE$^A(2^{\mathrm{linear}})$, it is straightforward to prove a relativized version of Lemma 3.1(a). Then using methods of Lutz [35], together with the counting argument of Lemma 4.2 above, the following can also be shown.

Proposition 4.4. For every fixed oracle A and every $\alpha < 1$, the set of all L such that $\mathrm{CS}_L^A(n) > \frac{2^n}{n}\left(1 + \frac{\alpha \log n}{n}\right)$ a.e. has pspace^A-measure 1, hence measure 1 in ESPACE^A. □

This strengthens a result of Wilson [47] and, together with Proposition 4.3, shows that ESPACE^A exhibits a weak Shannon effect: For any fixed oracle A, almost every language in ESPACE^A has circuit-size complexity that is within a factor of 2 of maximal. The linear separation from maximal size in Proposition 4.4 will resurface in the main result of section 6, below.

We will need the following facts in sections 5 and 6.

Lemma 4.5. For every $0 < \alpha < \alpha' < 1$ and all sufficiently large n, the number of functionally distinct n-input oracle circuits having size $\le 2^{\alpha n}$ is less than $2^{2^{\alpha' n}}$.

Proof. Fix $\alpha < \alpha' < 1$ and let $s = 2^{\alpha n}$ in Lemma 4.2. (Note that every oracle circuit of size $< s$ is functionally equivalent to some circuit of size s.) This gives

$$H(2^{\alpha n}) < 2685[4e(2^{\alpha n})]^{2^{\alpha n}} < 2^{2^{\alpha' n}}$$

for all sufficiently large n. □

Lemma 4.6. For all sufficiently large n, the number of functionally distinct n-input oracle circuits having size $\le \frac{2^n}{n}$ is less than $2^{2^n(\frac{n-1}{n})}$.

Proof. Letting $s = \frac{2^n}{n}$ in Lemma 4.2, we have

$$\begin{aligned} H\left(\frac{2^n}{n}\right) &< 2685\left[4e\left(\frac{2^n}{n}\right)\right]^{\frac{2^n}{n}} \\ &\leq (2^{n-1})^{\frac{2^n}{n}} \\ &= 2^{2^n(\frac{n-1}{n})} \end{aligned}$$

for all sufficiently large n. □

5 Nondeterministic Time Versus Circuit Size

In this, the main section of the paper, we prove that *every* pspace-random oracle A supports the separation $\mathrm{NP}^A \not\subseteq \mathrm{SIZE}^A_{\mathrm{i.o.}}(2^{\alpha n})$ for every real $\alpha < \frac{1}{3}$. We begin by showing that the desired separation is a pspace-test.

Theorem 5.1. For every $0 < \alpha < \frac{1}{3}$,

$$\mu_{\mathrm{pspace}}(\{A \mid \mathrm{NP}^A \not\subseteq \mathrm{SIZE}^A_{\mathrm{i.o.}}(2^{\alpha n})\}) = 1.$$

Proof. Fix $0 < \alpha < \frac{1}{3}$. For each $y \in \{0,1\}^*$, let $\frac{1}{3}y$ denote the string consisting of the first $\frac{|y|}{3}$ bits of y. (For clarity, we omit the floors and ceilings required for strict accuracy in this proof.) For each oracle A, define the function θ_A by

$$\theta_A(y) = (\tfrac{1}{3}y)[\![y0 \in A]\!][\![y0^2 \in A]\!]\cdots[\![y0^{\frac{2|y|}{3}} \in A]\!]$$

and let

$$L_A = \mathrm{range}(\theta_A) = \{x \mid (\exists y)\, \theta_A(y) = x\}.$$

It is clear that $L_A \in \mathrm{NP}^A$ for every oracle A, so it suffices to show that $\mu_{\mathrm{pspace}}(X) = 0$, where

$$X = \{A \mid L_A \in \mathrm{SIZE}^A_{\mathrm{i.o.}}(2^{\alpha n})\}.$$

For each $n \in \mathbf{N}$, partition $\{0,1\}^n$ into *blocks*

$$B_x = \{y \in \{0,1\}^n \mid \tfrac{1}{3}y = \tfrac{1}{3}x\}$$

for $x \in \{0,1\}^n$, and define the set

$$\mathrm{BR}_n = \{u0^{\frac{2n}{3}} \mid |u| = \tfrac{n}{3}\}$$

of *block representatives.* Note that $\theta_A(B_x) \subseteq B_x$ for all x. Our proof focuses on the difficulty of determining that block representatives are not in range(θ_A).

Throughout this proof, to simplify notation, we write $s = 2^{\alpha n}$ for the circuit-size bound used in the definition of X. For each $n \in \mathbf{N}$, let $\mathrm{CIRC}(s) = \mathrm{CIRC}(2^{\alpha n})$ be the set of all novel n-input oracle circuits γ with size$(\gamma) \leq s$, and define the set

$$X_n = \bigcup_{\gamma \in \mathrm{CIRC}(s)} X_{n,\gamma},$$

where

$$X_{n,\gamma} = \{A \mid L(\gamma^A) \cap \mathrm{BR}_n = L_A \cap \mathrm{BR}_n\}$$

for each $\gamma \in \mathrm{CIRC}(s)$. Note that

$$X \subseteq \{A \mid A \in X_n \text{ i.o.}\}. \tag{5.1}$$

For each $n \in \mathbf{N}$, let

$$Y_n = \{A \mid |\mathrm{BR}_n \setminus L_A| \geq \tfrac{1}{4}|\mathrm{BR}_n|\}.$$

That is, Y_n is the set of all oracles A for which at least 25% of the block representatives $u0^{\frac{2n}{3}} \in \mathrm{BR}_n$ escape being in $\mathrm{range}(\theta_A) = L_A$. Finally, let

$$\begin{aligned} Y &= \{A \mid A \in Y_n \text{ a.e.}\} \\ Z &= \{A \mid A \in X_n \cap Y_n \text{ i.o.}\} \end{aligned}$$

By (5.1), $X \subseteq Y^c \cup Z$, so it suffices to prove that

$$\mu_{\mathrm{pspace}}(Y^c) = 0 \tag{5.2}$$

and

$$\mu_{\mathrm{pspace}}(Z) = 0. \tag{5.3}$$

Our proof is thus in two parts, establishing (5.2) and (5.3) separately. Note that the definition of Y does not involve circuits, so the verification of (5.3) is the main part of this proof.

To establish (5.2), it suffices by Theorem 3.4 to show that there is a polynomial q such that the implication

$$A \in Y_n^c \Longrightarrow \mathrm{KS}^q(A_{\leq 2n}) < 2^{2n+1} - 2^{\frac{n}{4}} \tag{5.4}$$

holds for all sufficiently large n. For each $n \in \mathbf{N}$, let $S_1, \ldots, S_{I(n)}$ be the lexicographic enumeration of all sets $S \subseteq \{0,1\}^{\leq 2n}$ such that $S \in Y_n^c$. It is routine to design a deterministic machine M that takes inputs $i, t \in \mathbf{N}$ in binary and has the following property. If $1 \leq i \leq I(n)$, then $M(i, 2n)$ is the $(2^{2n+1} - 1)$-bit characteristic string of S_i, and this computation is carried out using workspace that is polynomial in 2^n. For all n, it is clear that $Y_n = \{A \mid A_{\leq 2n} \in Y_n\}$, since $\theta_A(\{0,1\}^n)$ is entirely determined by $A_{\leq 2n}$. Since we have fixed an optimal machine in defining KS, it follows that there exist a polynomial q and a constant a such that the implication

$$A \in Y_n^c \Longrightarrow \mathrm{KS}^q(A_{\leq 2n}) \leq \log I(n) + a \tag{5.5}$$

holds for all sufficiently large n. We thus estimate $\log I(n)$.

Intuitively, $I(n)$ is small because for most sets S, approximately $\frac{1}{e}|\mathrm{BR}_n| \gg \frac{1}{4}|\mathrm{BR}_n|$ of the elements of BR_n escape being in $\mathrm{range}(\theta_S)$. To formalize this intuition, consider the random experiment in which a set $S \subseteq \{0,1\}^{\leq 2n}$ is chosen probabilistically according to the uniform distribution on all such sets. It is clear that

$$\log I(n) < 2^{2n+1} + \log \Pr[S \in Y_n^c]. \tag{5.6}$$

For $x \in \mathrm{BR}_n$, let $Y_{n,x}$ be the event that $x \notin \mathrm{range}(\theta_S)$. For each n, the events $Y_{n,x}$ are independent for distinct $x \in \mathrm{BR}_n$ and the probability $p_n = \Pr[S \in Y_{n,x}]$ does not depend on x. In fact,

$$p_n = \left(1 - 2^{-\frac{2n}{3}}\right)^{2^{\frac{2n}{3}}} > \tfrac{1}{3} \text{ a.e.,}$$

since $p_n \to \frac{1}{e}$ as $n \to \infty$. By Proposition 2.1,

$$\begin{aligned} \Pr[S \in Y_n^c] &< \sum_{i=0}^{\frac{1}{4}2^{\frac{n}{3}}} \binom{2^{\frac{n}{3}}}{i} \left(\frac{1}{3}\right)^i \left(\frac{2}{3}\right)^{2^{\frac{n}{3}}-i} \\ &\le 2^{-c2^{\frac{n}{3}}} \qquad\qquad (c > 0) \end{aligned}$$

so it follows by (5.5) and (5.6) that

$$\begin{aligned} \mathrm{KS}^q(A_{\le 2n}) &< 2^{2n+1} - c2^{\frac{n}{3}} + a \\ &< 2^{2n+1} - 2^{\frac{n}{4}} \end{aligned}$$

for all $A \in Y_n^c$, for all sufficiently large n, confirming (5.4) and hence (5.2). This completes the first part of the proof.

The second, and main, part of the proof is to establish (5.3). For this, by Theorem 3.3, it suffices to exhibit a pspace-computable 1-DS d such that

$$\sum_{t=0}^{\infty} d_t(\lambda) \text{ is p-convergent} \tag{5.7}$$

and

$$Z \subseteq \bigcap_{k=0}^{\infty} \bigcup_{t=k}^{\infty} S[d_t]. \tag{5.8}$$

Define $d : \mathbf{N} \times \{0,1\}^* \to [0,\infty)$ by

$$d_t(w) = \begin{cases} \sum\limits_{\gamma \in \mathrm{CIRC}(s)} \Pr(X_{n,\gamma} \cap Y_n \mid C_w) & \text{if } t = 2^n \\ 0 & \text{if } t \text{ is not a power of 2,} \end{cases} \tag{5.9}$$

where the conditional probabilities $\Pr(X_{n,\gamma} \cap Y_n \mid C_w) = \Pr[A \in X_{n,\gamma} \cap Y_n \mid A \in C_w]$ are computed according to the random experiment in which the language $A \subseteq \{0,1\}^*$ is chosen probabilistically, using an independent toss of a fair coin to decide membership of each string in A.

For each $n \in \mathbf{N}$ and $\gamma \in \mathrm{CIRC}(s)$, it is immediate from the definition of conditional probability that

$$\Pr(X_{n,\gamma} \cap Y_n \mid C_w) = \frac{\Pr(X_{n,\gamma} \cap Y_n \mid C_{w0}) + \Pr(X_{n,\gamma} \cap Y_n \mid C_{w1})}{2}.$$

It follows from this and (5.9) that d is a 1-DS.

Now let Ω_n be the set of all subsets of $\{0,1\}^{\le \max\{s,2n\}}$ and let

$$\Psi_{n,\gamma} = \Omega_n \cap X_{n,\gamma} \cap Y_n$$

for each $\gamma \in \mathrm{CIRC}(s)$. (Note that $s = 2^{\alpha n} \geq 2n$ for all sufficiently large n.) Intuitively, $\Psi_{n,\gamma}$ is the set of all $T \subseteq \{0,1\}^{\leq \max\{s,2n\}}$ such that (i) γ^T correctly decides L_T when restricted to inputs from BR_n; and (ii) at least 25% of the elements of BR_n escape being in $\mathrm{range}(\theta_T) = L_T$. Since $L_T \cap \{0,1\}^n$ depends only upon $T_{\leq 2n}$, and since circuits of size $\leq s$ only query strings of length $\leq s$, we have

$$X_{n,\gamma} \cap Y_n = \{A \mid A_{\leq \max\{s,2n\}} \in \Psi_{n,\gamma}\}$$

for all $n \in \mathbf{N}$ and $\gamma \in \mathrm{CIRC}(s)$. It follows immediately from this that, for all $n \in \mathbf{N}$, $\gamma \in \mathrm{CIRC}(s)$, and $w \in \{0,1\}^*$,

$$\Pr(X_{n,\gamma} \cap Y_n \mid C_w) = \frac{|\Psi_{n,\gamma} \cap C_{w'}|}{|\Omega_n \cap C_{w'}|}, \tag{5.10}$$

where $w' = w[0..m-1]$, $m = \min\{|w|, 2^{\max\{s,2n\}+1} - 1\}$.

The denominator of (5.10) is triply exponential in n, hence too large to store in polynomial space. Nevertheless, $\Pr(X_{n,\gamma} \cap Y_n \mid C_w)$ can be computed in space polynomial in $t + |w|$, where $t = 2^n$. To see this, let $n \in \mathbf{N}$, $\gamma \in \mathrm{CIRC}(s)$, and $w \in \{0,1\}^*$. We first compute the number m and the string w' as in (5.10). Now note that, for $T \in \Omega_n$, membership of T in $\Psi_{n,\gamma}$ depends upon at most $|\{0,1\}^n| \cdot \frac{2n}{3} + |\mathrm{BR}_n| \cdot s$ bits of the characteristic string of T. (The first term counts all bits affecting $\mathrm{range}(\theta_T)_{=n}$, while the second term bounds the total number of oracle queries on inputs $x \in \mathrm{BR}_n$.) For sufficiently large n, these terms are less than t^2 and t, respectively. For each $y \in \{0,1\}^{t^2}$ and $z \in \{0,1\}^t$, construct a partial oracle specification $\mathrm{oracle}(y,z) \in \{0,1,\perp\}^*$ as follows. The length of $\mathrm{oracle}(y,z)$ is $2^{\max\{s,2n\}+1} - 1$, i.e., $\mathrm{oracle}(y,z)$ decides (some) strings of length $\leq \max\{s,2n\}$. Initially, $\mathrm{oracle}(y,z)$ is of the form $w'\perp^l$, where w' is as in (5.10). (This ensures that $C_{\mathrm{oracle}(y,z)} \subseteq C_{w'}$.) The "bit sources" y and z are then used to further specify $\mathrm{oracle}(y,z)$ in the following two phases.

Phase I. For each $x \in \{0,1\}^n$ and each $1 \leq i \leq \frac{2n}{3}$ (in some canonical order), if the bit of $\mathrm{oracle}(y,z)$ corresponding to $x0^i$ is $\perp$, then the first bit of y is deleted from y and used to replace this $\perp$ in $\mathrm{oracle}(y,z)$. At the end of Phase I, $\mathrm{oracle}(y,z)$ completely determines $\mathrm{range}(\theta_T)_{=n}$ for all $T \in C_{\mathrm{oracle}(y,z)}$. We let y' be the prefix of (the original string) y consisting of those bits actually used in this phase.

Phase II. For each $x \in \mathrm{BR}_n$, simulate the oracle circuit γ on input x. During the course of this simulation, oracle queries are handled as follows. If the bit of $\mathrm{oracle}(y,z)$ corresponding to the queried string is $\perp$, then the first bit of z is deleted from z and used to replace this $\perp$ in $\mathrm{oracle}(y,z)$. Then, in any case, the bit of $\mathrm{oracle}(y,z)$ corresponding to the queried string is used as the answer to the query. At the end of Phase II, $\mathrm{oracle}(y,z)$ completely determines membership (or non-membership) of T in $\Psi_{n,\gamma}$ for all $T \in C_{\mathrm{oracle}(y,z)}$. We let z_y be the prefix of (the original string) z consisting of those bits actually used in this phase. Since z_y depends only upon the prefix y' of y, and since $\mathrm{oracle}(y,z)$ depends only upon y' and z_y, we write $\mathrm{oracle}(y', z_{y'})$ for $\mathrm{oracle}(y,z)$. We also let $T(y', z_{y'})$ be the smallest language in $C_{\mathrm{oracle}(y',z_{y'})}$.

We are primarily concerned with the strings y' and $z_{y'}$, which are the bit sources actually used in Phases I and II above. Accordingly, we define the set

$$\mathrm{SOURCES}(n,\gamma) = \{(y', z_{y'}) \mid y \in \{0,1\}^{t^2}, z \in \{0,1\}^t\}.$$

It is important to note that

$$\{C_{\mathrm{oracle}(y',z_{y'})} \mid (y', z_{y'}) \in \mathrm{SOURCES}(n,\gamma)\}$$

is a partition of $C_{w'}$, whence (5.10) tells us that

$$\begin{aligned}\Pr(X_{n,\gamma} \cap Y_n \mid C_w) &= \sum_{(y',z_{y'})\in\mathrm{SOURCES}(n,\gamma)} \frac{\left|\Psi_{n,\gamma} \cap C_{\mathrm{oracle}(y',z_{y'})}\right|}{|\Omega_n \cap C_{w'}|} \\ &= \sum_{(y',z_{y'})\in\mathrm{SOURCES}(n,\gamma)} \frac{\left|\Omega_n \cap C_{\mathrm{oracle}(y',z_{y'})}\right| \cdot [\![T(y',z_{y'}) \in \Psi_{n,\gamma}]\!]}{|\Omega_n \cap C_{w'}|} \\ &= \sum_{(y',z_{y'})\in\mathrm{SOURCES}(n,\gamma)} 2^{-(|y'|+|z_{y'}|)} \cdot [\![T(y',z_{y'}) \in \Psi_{n,\gamma}]\!].\end{aligned} \tag{5.11}$$

We use (5.11) as the basis for our computation of $\Pr(X_{n,\gamma} \cap Y_n \mid C_w)$. Having computed m and w' as in (5.10), we can, for any $y \in \{0,1\}^{t^2}$ and $z \in \{0,1\}^t$, compute the partial specification oracle(y,z). This can be done in space polynomial in $t + |w| = 2^n + w$ because at most $|w| + t^2 + t$ bits of oracle(y,z) are not $\perp$. (We thus represent oracle(y,z) in a compressed form by a list of positions i at which $w[i] = 0$ and a list of positions j at which $w[j] = 1$.) Once we have oracle(y,z), we can test the condition $T(y', z_{y'}) \in \Psi_{n,\gamma}$ in space polynomial in $t + |w|$. Thus we can use (5.11) to compute each $\Pr(X_{n,\gamma} \cap Y_n \mid C_w)$ in space polynomial in $t + |w|$. It follows by (5.9) that $d : \mathbf{N} \times \{0,1\}^* \to \mathbf{D}$ and $d \in$ pspace. Thus d is a pspace-computable 1-DS.

To see that (5.8) holds, let $A \in Z$. Then the set

$$K = \{2^n \mid A \in X_n \cap Y_n\}$$

is infinite. Moreover, for each $t = 2^n \in K$, if we let $w = \chi_A[0..2^{\max\{s,2n\}+1} - 1]$, then

$$\begin{aligned} d_t(w) &= \sum_{\gamma\in\mathrm{CIRC}(s)} \Pr(X_{n,\gamma} \cap Y_n \mid C_w) \\ &\geq \Pr\Big(\bigcup_{\gamma\in\mathrm{CIRC}(s)} X_{n,\gamma} \cap Y_n \;\Big|\; C_w\Big) \\ &= \Pr(X_n \cap Y_n \mid C_w) \\ &= 1, \end{aligned}$$

so $A \in C_w \subseteq S[d_t]$. Thus $A \in S[d_t]$ i.o., confirming (5.8).

All that remains is to verify (5.7). For this it suffices by Lemma 3.2 to prove that

$$d_t(\lambda) \leq 2^{-t^{\alpha}} \tag{5.12}$$

for all sufficiently large t. This is trivial if t is not a power of 2, so for the rest of the proof we assume that $t = 2^n$; we will show that (5.12) holds for all sufficiently large n.

By (5.9) and (5.10), we have

$$d_t(\lambda) = \sum_{\gamma\in\mathrm{CIRC}(s)} \Pr(X_{n,\gamma} \cap Y_n) = \sum_{\gamma\in\mathrm{CIRC}(s)} \frac{|\Psi_{n,\gamma}|}{|\Omega_n|} \tag{5.13}$$

for all $n \in \mathbf{N}$. We thus seek an upper bound for $|\Psi_{n,\gamma}|$. We use a refinement of the measure-preserving transformation argument of [9]. However, we give our argument in purely combinatorial terms. For convenience, write $N = \frac{1}{8}2^{\frac{n}{3}}$ and let

$$\Delta_n = \{z \in \{0,1\}^{2N} \mid \#(0,z) = \#(1,z) = N\}.$$

Intuitively, we will show that $|\Psi_{n,\gamma}|$ cannot be much larger than $|\Omega_n|/|\Delta_n|$. Roughly, the idea is that each $T \in \Psi_{n,\gamma}$ must have $\gamma^T(x) = 0$ for at least $\frac{1}{4}2^{\frac{n}{3}} = 2N$ of the strings $x \in \mathrm{BR}_n$. We can thus use each $z \in \Delta_n$ as a selection of N of these $2N$ strings at which to "introduce an error," creating from T a new set $U \in \Omega_n$ such that (i) $\gamma^U(x) = \gamma^T(x)$ for every $x \in \mathrm{BR}_n$, but (ii) the N strings selected by z are in range(θ_U). If our construction made the function $(T, z) \mapsto U$ one-to-one from $\Psi_{n,\gamma} \times \Delta_n$ into Ω_n, we could conclude that $|\Psi_{n,\gamma}| \leq |\Omega_n|/|\Delta_n|$. However, matters are not so simple. To make our function one-to-one, we must carry extra information, namely the "old" values of θ_T that were changed to put N new block representatives into the range. Also, to ensure the condition $\gamma^U(x) = \gamma^T(x)$, we avoid changing the answers to queries of $\gamma^T(x)$, thereby slightly restricting our freedom to choose preimages for the N new elements of the range. Thus our construction is a little more elaborate and does not quite achieve $|\Psi_{n,\gamma}| \leq |\Omega_n|/|\Delta_n|$.

Formally, for sufficiently large n, and for each $\gamma \in \mathrm{CIRC}(s)$, we will exhibit a function

$$f_{n,\gamma} : \Psi_{n,\gamma} \times \Delta_n \times \{0,1\}^{\frac{2nN}{3} - N} \overset{\text{one-to-one}}{\longrightarrow} \Omega_n \times \{0,1\}^{\frac{2nN}{3}}. \tag{5.14}$$

The existence of such a function implies that each $|\Psi_{n,\gamma}| \leq 2^N|\Omega_n|/|\Delta_n|$, whence each

$$\frac{|\Psi_{n,\gamma}|}{|\Omega_n|} \leq \frac{2^N}{|\Delta_n|} = \frac{2^N}{\binom{2N}{N}}.$$

Using the estimate $e(\frac{n}{e})^n < n! < en(\frac{n}{e})^n$ gives

$$\binom{2N}{N} > \frac{2^{2N}}{4N^2},$$

whence we have

$$\frac{|\Psi_{n,\gamma}|}{|\Omega_n|} < 2^{2-N}N^2 \tag{5.15}$$

for all $\gamma \in \mathrm{CIRC}(s)$, for all sufficiently large n. Now fix $\beta > 1$ such that $\alpha\beta < \frac{1}{3}$. By (5.13), (5.15), and Lemma 4.5, we have, for all sufficiently large n,

$$\begin{aligned} d_t(\lambda) &\leq 2^{2-N}N^2|\mathrm{CIRC}(s)| \\ &< 2^{2-N}N^2 2^{s^\beta} \\ &= 2^{2-N+t^{\alpha\beta}}N^2. \end{aligned}$$

Recalling that $N = \frac{1}{8}2^{\frac{n}{3}} = \frac{1}{8}t^{\frac{1}{3}}$ and $\alpha < \alpha\beta < \frac{1}{3}$, it follows that (5.12) holds for all sufficiently large n. Thus, to complete the proof, it suffices to define the functions $f_{n,\gamma}$ as in (5.14).

Fix $n_0 \in \mathbf{N}$ such that $2^{(\frac{1}{3}+\alpha)n} < 2^{\frac{2n}{3}-1}$ for all $n \geq n_0$. Given $n \geq n_0$, $\gamma \in \mathrm{CIRC}(s)$, $T \in \Psi_{n,\gamma}$, $z \in \Delta_n$, and $v \in \{0,1\}^{\frac{2nN}{3}-N}$, we now describe the value

$$f_{n,\gamma}(T, z, v) = (U, w) \in \Omega_n \times \{0,1\}^{\frac{2nN}{3}}. \tag{5.16}$$

We will write $v = v_1 \cdots v_N$ and $w = w_1 \cdots w_N$, where each $|v_i| = \frac{2n}{3} - 1$ and each $|w_i| = \frac{2n}{3}$. (Intuitively, $v_1, \ldots, v_N$ specify a choice of preimages, while $w_1, \ldots, w_N$ specify the "old" values of θ_T at these preimages.) Since $T \in \Psi_{n,\gamma}$, we have $\gamma^T(x) = 0$ for at least $\frac{1}{4} 2^{\frac{n}{3}} = 2N$ of the strings $x \in \mathrm{BR}_n$. Call these strings $x'_1, x'_2, \ldots$ (in lexicographical order) and let $D(T, z)$ be the set consisting of all x'_i such that $1 \le i \le 2N$ and the i^{th} bit of z is 1. Note, then, that $|D(T, z)| = N$; write $D(T, z) = \{x_1, \ldots, x_N\}$ in lexicographic order.

Let $Q(T)$ be the set of all strings $y \in \{0,1\}^n$ such that, for some $x \in \mathrm{BR}_n$ and $1 \le j \le \frac{2n}{3}$, $\gamma^T(x)$ queries $y0^j$. Note that $|Q(T)| \le |\mathrm{BR}_n| \cdot \mathrm{size}(\gamma) \le 2^{(\frac{1}{3}+\alpha)n} < 2^{\frac{2n}{3}-1}$. For each $x \in \mathrm{BR}_n$ and each $v' \in \{0,1\}^{\frac{2n}{3}-1}$, let $x \star v'$ denote the j^{th} string in $B_x \setminus Q(T)$, where v' is the j^{th} string in $\{0,1\}^{\frac{2n}{3}-1}$. (Note that $x \star v'$ exists because $|B_x \setminus Q(T)| \ge |B_x| - |Q(T)| > 2^{\frac{2n}{3}} - 2^{\frac{2n}{3}-1} = 2^{\frac{2n}{3}-1}$.)

The pair (U, w) of (5.16) is now defined by

$$U = T \setminus \{(x_i \star v_i)0^j \mid 1 \le i \le N,\ 1 \le j \le \tfrac{2n}{3}\}$$

and

$$w_i = [\![(x_i \star v_i)0 \in T]\!] \cdots [\![(x_i \star v_i)0^{\frac{2n}{3}} \in T]\!]$$

for $1 \le i \le N$. Intuitively, U is obtained from T by making just those changes required to establish the conditions

$$\theta_U(x_i \star v_i) = x_i$$

for $1 \le i \le N$. The string w satisfies

$$\theta_T(x_i \star v_i) = (\tfrac{1}{3}x_i)w_i$$

for $1 \le i \le N$. Note that γ^T does not query any string in $T \bigtriangleup U = T \setminus U$, so $\gamma^U(x) = \gamma^T(x)$ for every $x \in \mathrm{BR}_n$.

To see that the resulting function $f_{n,\gamma}$ is one-to-one, it suffices to show that T, z, and v can be recovered from U and w. First note that $D(T, z)$ is precisely the set of all $x \in \mathrm{BR}_n$ such that $\gamma^U(x) = 0$ but $x \in \mathrm{range}(\theta_U)$. Thus $D(T, z)$ and z are determined by U. Now each $x_i \in D(T, z)$ has a unique preimage under θ_U. This preimage is $x_i \star v_i$, so v is also determined by U. Finally,

$$T = U \cup \{(x_i \star v_i)0^j \mid \text{the } j^{\text{th}} \text{ bit of } w_i \text{ is } 1\},$$

so T is determined by U and w. This establishes (5.14) and completes the proof of Theorem 5.1. □

Corollary 5.2. For *every* pspace-random oracle A and every real $\alpha < \frac{1}{3}$, $\mathrm{NP}^A \not\subseteq \mathrm{SIZE}^A_{\text{i.o.}}(2^{\alpha n})$.

Proof. By Theorem 5.1, this condition is a pspace-test. □

Note that Corollary 5.2 gives an explicit, sufficient condition for an oracle A to support the indicated separation.

Corollary 5.3. For every real $\alpha < \frac{1}{3}$, for almost every oracle $A \in \mathrm{ESPACE}$, $\mathrm{NP}^A \not\subseteq \mathrm{SIZE}^A_{\text{i.o.}}(2^{\alpha n})$.

Proof. Immediate from Theorem 5.1 and Lemma 3.1. □

Corollary 5.4. For every real $\alpha < \frac{1}{3}$, for a randomly selected oracle A,

$$\Pr\left[\mathrm{NP}^A \not\subseteq \mathrm{SIZE}^A_{\text{i.o.}}(2^{\alpha n})\right] = 1.$$

Proof. Immediate from Theorem 5.1 and Lemma 3.1. □

Wilson [47] constructed oracles A and B such that $\mathrm{NP}^A \subseteq \mathrm{LINSIZE}^A$ and $\mathrm{NP}^B \not\subseteq \mathrm{P}^B/\mathrm{Poly}$, and asked which of these holds with probability one. We can now answer this question.

Corollary 5.5. For a randomly selected oracle A,

$$\Pr[\mathrm{NP}^A \not\subseteq \mathrm{P}^A/\mathrm{Poly}] = 1.$$

Proof. Immediate from Corollary 5.4. □

Of course the original random oracle result is an immediate consequence of Corollary 5.4.

Corollary 5.6 (Bennett and Gill [9]). For a randomly selected oracle A,

$$\Pr\left[\mathrm{P}^A \neq \mathrm{NP}^A\right] = 1.$$

□

Often one is only interested in the case of polynomial advice.

Corollary 5.7. For every pspace-random oracle A, $\mathrm{NP}^A \not\subseteq \mathrm{P}^A/\mathrm{Poly}$.

Proof. Immediate from Corollary 5.2. □

6 Deterministic Time Versus Circuit Size

It is interesting to observe that the test language L_A of the previous section is computable in 2^{linear} time relative to A for each oracle A. That is, we have the following.

Corollary 6.1. $\mu_{\text{pspace}}(\{A \mid \mathrm{E}^A \not\subseteq \mathrm{SIZE}^A_{\text{i.o.}}(2^{\alpha n})\}) = 1$ for every $0 < \alpha < \frac{1}{3}$. □

In this section, we show that this separation condition remains a pspace-test even when the size bound becomes virtually maximal.

Theorem 6.2. $\mu_{\text{pspace}}\left(\left\{A \mid \mathrm{E}^A \not\subseteq \mathrm{SIZE}^A_{\text{i.o.}}\left(\frac{2^n}{n}\right)\right\}\right) = 1.$

Proof. For each oracle A, let

$$L_A = \{x \mid x0^{2^{|x|}} \in A\}.$$

Clearly, $L_A \in \mathrm{E}^A$ for each A. Thus it suffices to prove that $\mu_{\text{pspace}}(Y) = 0$, where

$$Y = \left\{A \mid L_A \in \mathrm{SIZE}^A_{\text{i.o.}}\left(\frac{2^n}{n}\right)\right\}.$$

For this, by Theorem 3.3, it suffices to exhibit a pspace-computable 1-DS d such that

$$\sum_{t=0}^{\infty} d_t(\lambda) \text{ is p-convergent} \tag{6.1}$$

and

$$Y \subseteq \bigcap_{k=0}^{\infty} \bigcup_{t=k}^{\infty} S[d_t]. \tag{6.2}$$

Throughout this proof, to simplify notation, we write $s = \frac{2^n}{n}$ for the circuit-size bound used in the definition of Y. For each $n \in \mathbf{N}$, then, let $\mathrm{CIRC}(s)$ be the set of all novel n-input oracle circuits that have size $\leq s$, and define the set

$$Y_n = \bigcup_{\gamma \in \mathrm{CIRC}(s)} Y_{n,\gamma},$$

where each

$$Y_{n,\gamma} = \{A \mid L(\gamma^A) = (L_A)_{=n}\}.$$

Define $d : \mathbf{N} \times \{0,1\}^* \to [0,\infty)$ by

$$d_t(w) = \begin{cases} \sum\limits_{\gamma \in \mathrm{CIRC}(s)} \Pr(Y_{n,\gamma} \mid C_w) & \text{if } t = 2^n \\ 0 & \text{if } t \text{ is not a power of 2,} \end{cases} \tag{6.3}$$

where the conditional probabilities $\Pr(Y_{n,\gamma} \mid C_w) = \Pr[A \in Y_{n,\gamma} \mid A \in C_w]$ are computed according to the random experiment in which $A \subseteq \{0,1\}^*$ is chosen probabilistically, using an independent toss of a fair coin to decide membership of each string in A.

As in the proof of Theorem 5.1, it is easily checked that d is a 1-DS. By a bit source argument analogous to (but simpler than) the one in the proof of Theorem 5.1, d is pspace-computable. All that remains, then, is to verify conditions (6.1) and (6.2).

To see that (6.1) holds, fix $n \in \mathbf{N}$ and let $t = 2^n$. By (6.3),

$$d_t(\lambda) = \sum_{\gamma \in \mathrm{CIRC}(s)} \Pr(Y_{n,\gamma}).$$

For all $\gamma \in \mathrm{CIRC}(s)$, all $x \in \{0,1\}^n$, and all oracles A, the string $x0^{2^{|x|}}$ is not queried in the computation of $\gamma^A(x)$. Thus, for all $\gamma \in \mathrm{CIRC}(s)$, $\Pr(Y_{n,\gamma}) = 2^{-|\{0,1\}^n|} = 2^{-t}$. By Lemma 4.6, it follows that

$$d_t(\lambda) = |\mathrm{CIRC}(s)| \cdot 2^{-t} < 2^{-\frac{t}{\log t}} < 2^{-t^{\frac{1}{2}}}$$

if n is sufficiently large. By (6.3), then, $d_t(\lambda) \leq 2^{-t^{\frac{1}{2}}}$ for almost all t, whence (6.1) follows from Lemma 3.2.

Finally, to verify (6.2), let $n \in \mathbf{N}$ and $A \in Y_n$. Fix $\gamma \in \mathrm{CIRC}(s)$ such that $A \in Y_{n,\gamma}$ and let w be the characteristic string of $A_{\leq n+t}$, where $t = 2^n$. Then $A \in C_w$ and

$$d_t(w) \geq \Pr(Y_{n,\gamma} \mid C_w) = 1,$$

so $A \in S[d_t]$. Thus $Y_n \subseteq S[d_t]$ for all $n \in \mathbf{N}$, where $t = 2^n$. It follows that

$$\begin{aligned} A \in Y \quad &\Longleftrightarrow \quad A \in Y_n \text{ i.o.} \\ &\Longrightarrow \quad A \in S[d_t] \text{ i.o.}, \end{aligned}$$

whence (6.2) holds. This completes the proof of Theorem 6.2. □

Corollary 6.3. For *every* pspace-random oracle A, $\mathrm{E}^A \not\subseteq \mathrm{SIZE}^A_{\mathrm{i.o.}}(\frac{2^n}{n})$.

Proof. By Theorem 6.2, this condition is a pspace-test. □

Corollary 6.4. For almost every oracle $A \in$ ESPACE, $\mathrm{E}^A \not\subseteq \mathrm{SIZE}^A_{\mathrm{i.o.}}(\frac{2^n}{n})$.

Proof. Immediate from Theorem 6.2 and Lemma 3.1. □

Corollary 6.5. For a randomly selected oracle A,

$$\Pr\left[\mathrm{E}^A \not\subseteq \mathrm{SIZE}^A_{\mathrm{i.o.}}\left(\frac{2^n}{n}\right)\right] = 1.$$

Proof. Immediate from Theorem 6.2 and Lemma 3.1. □

Corollary 6.6. For a randomly selected oracle A,

$$\Pr[\mathrm{E}^A \not\subseteq \mathrm{P}^A/\mathrm{Poly}] = 1.$$

Proof. Immediate from Corollary 6.5. □

Although Corollary 6.6 is considerably weaker than Corollary 6.5 (which, in turn, is much weaker than Theorem 6.2), Corollary 6.6 gives an explicit answer to an open question of Wilson [47]. Specifically, after exhibiting oracles A, B, and C such that $\mathrm{E}^A \subseteq \mathrm{LINSIZE}^A$, $\mathrm{E}_2^B \subseteq \mathrm{P}^B/\mathrm{Poly}$, and $\mathrm{E}^C \not\subseteq \mathrm{P}^C/\mathrm{Poly}$, Wilson asked what relation holds for randomly selected oracles. Corollary 6.6 tells us that oracle C gives the typical situation, while oracles A and B are exceptional.

7 Conclusion

We have established pspace-randomness as a sufficient condition for an oracle to achieve certain separations. Intuitively, for example, we now know that $\mathrm{NP}^A \not\subseteq \mathrm{P}^A/\mathrm{Poly}$ for *every* oracle A whose information content is high enough that A is pspace-random. In contrast, work of Hartmanis [19], Long and Selman [29], Balcázar and Book [6], and Allender and Rubinstein [1], can be used to show the following. If there *exists* an oracle A, whose information content is sufficiently low (e.g., $A \in \mathbf{K[log,poly]}$), such that $\mathrm{NP}^A \not\subseteq \mathrm{P}^A/\mathrm{Poly}$, then the unrelativized separation $\mathrm{NP} \not\subseteq \mathrm{P}/\mathrm{Poly}$ follows. It will be interesting to see these high and low information criteria pushed closer together.

Acknowledgments. We thank Ron Book and David Juedes for helpful discussions. We also thank two anonymous referees for useful observations.

References

[1] E. Allender and R. Rubinstein, P-printable sets, *SIAM Journal on Computing* **17** (1988), pp. 1193–1202.

[2] K. Ambos-Spies, Randomness, relativizations, and polynomial reducibilities, *Proceedings of the First Structure in Complexity Theory Conference*, 1986, pp. 23–34.

[3] L. Babai, Trading group theory for randomness, *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, 1985, pp. 421–429.

[4] L. Babai, Random oracles separate PSPACE from the polynomial-time hierarchy, *Information Processing Letters* **26** (1987), pp. 51–53.

[5] T. Baker, J. Gill, and R. Solovay, Relativizations of the P =?NP question, *SIAM Journal on Computing* **4** (1975), pp. 431–442.

[6] J. L. Balcázar and R. V. Book, Sets with small generalized Kolmogorov complexity, *Acta Informatica* **23** (1986), pp. 679–688.

[7] J. L. Balcázar, J. Díaz, and J. Gabarró, *Structural Complexity I*, Springer-Verlag, 1988.

[8] R. Beigel, On the relativized power of additional accepting paths, *Proceedings of the Fourth Annual Structure in Complexity Theory Conference*, 1989, pp. 216–224.

[9] C. H. Bennett and J. Gill, Relative to a random oracle A, $\mathrm{P}^A \neq \mathrm{NP}^A \neq \text{co-NP}^A$ with probability 1, *SIAM Journal on Computing* **10** (1981), pp. 96–113.

[10] L. Berman and J. Hartmanis, On isomorphism and density of NP and other complete sets, *SIAM Journal on Computing* **6** (1977), pp. 305–322.

[11] R. V. Book, Some observations on separating complexity classes, *SIAM Journal on Computing* **20** (1991), pp. 246–258.

[12] R. V. Book and S. Tang, Characterizing polynomial complexity classes by reducibilities, *Mathematical Systems Theory* **23** (1990), pp. 165–174.

[13] É. Borel, Sur les probabilités dénombrables et leurs applications arithmétiques, *Rend. Circ. Mat. Palermo* **27** (1909), pp. 247–271.

[14] J. Cai, With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy, *Journal of Computer and System Sciences* **38** (1989), pp. 68–85.

[15] F. P. Cantelli, La tendenza ad un limite nel senzo del calcolo delle probabilita, *Rend. Circ. Mat. Palermo* **41** (1916), pp. 191–201.

[16] G. J. Chaitin, On the length of programs for computing finite binary sequences, *Journal of the Association for Computing Machinery* **13** (1966), pp. 547–569.

[17] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer, Alternation, *Journal of the ACM* **28** (1981), pp. 114–133.

[18] H. Chernoff, A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations, *Annals of Mathematical Statistics* **23** (1952), pp. 493–509.

[19] J. Hartmanis, Generalized Kolmogorov complexity and the structure of feasible computations, *Proceedings of the 24th IEEE Symposium on the Foundations of Computer Science*, 1983, pp. 439–445.

[20] D. T. Huynh, Resource-bounded Kolmogorov complexity of hard languages, *Structure in Complexity Theory*, pp. 184–195, Springer-Verlag, 1986.

[21] R. M. Karp and R. J. Lipton, Some connections between nonuniform and uniform complexity classes, *Proceedings of the 12th ACM Symposium on Theory of Computing*, 1980, pp. 302–309.

[22] K. Ko, On the notion of infinite pseudorandom sequences, *Theoretical Computer Science* **48** (1986), pp. 9–33.

[23] A. N. Kolmogorov, Three approaches to the quantitative definition of 'information', *Problems of Information Transmission* **1** (1965), pp. 1–7.

[24] A. N. Kolmogorov and V. A. Uspenskii, Algorithms and randomness, translated in *Theory of Probability and its Applications* **32** (1987), pp. 389–412.

[25] S. Kurtz, On the random oracle hypothesis, *Information and Control* **57** (1983), pp. 40–47.

[26] S. Kurtz, S. Mahaney, and J. Royer, The isomorphism conjecture fails relative to a random oracle, *Proceedings of the 21st ACM Symposium on Theory of Computing*, 1989, pp. 157–166.

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

[28] M. Li and P. M. B. Vitanyi, Kolmogorov complexity and its applications, in J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A*, pp. 187–254. Elsevier, 1990.

[29] T. J. Long and A. L. Selman, Relativizing complexity classes with sparse oracles, *Journal of the ACM* **33** (1986), pp. 618–627.

[30] L. Longpré, *Resource Bounded Kolmogorov Complexity, a Link Between Computational Complexity and Information Theory*, PhD thesis, Cornell University, 1986, Technical Report TR-86-776.

[31] O. B. Lupanov, On the synthesis of contact networks, *Dokl. Akad. Nauk SSSR* **119** (1958), pp. 23–26.

[32] J. H. Lutz, Category and measure in complexity classes, *SIAM Journal on Computing* **19** (1990), pp. 1100–1131.

[33] J. H. Lutz, Pseudorandom sources for BPP, *Journal of Computer and System Sciences* **41** (1990), pp. 307–320.

[34] J. H. Lutz, Almost everywhere high nonuniform complexity, *Proceedings of the Fourth Structure in Complexity Theory Conference*, 1989, pp. 37–53.

[35] J. H. Lutz, Almost everywhere high nonuniform complexity, *Journal of Computer and System Sciences* (1991), to appear.

[36] J. H. Lutz, A pseudorandom oracle characterization of BPP, *SIAM Journal on Computing*, to appear.

[37] J. H. Lutz, Resource-bounded measure, in preparation.

[38] J. H. Lutz, Intrinsically pseudorandom sequences, in preparation.

[39] P. Martin-Löf, On the definition of random sequences, *Information and Control* **9** (1966), pp. 602–619.

[40] N. Nisan and A. Wigderson, Hardness vs. randomness, *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science*, 1988, pp. 2–11.

[41] U. Schöning, *Complexity and Structure*, Springer-Verlag, 1986.

[42] A. Shamir, IP = PSPACE, *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, 1990, pp. 11–15.

[43] M. Sipser, A complexity-theoretic approach to randomness, *Proceedings of the 15th ACM Symposium on Theory of Computing*, 1983, pp. 330–335.

[44] R. J. Solomonoff, A formal theory of inductive inference, *Information and Control* **7** (1964), pp. 1–22, 224–254.

[45] S. Tang and O. Watanabe, On tally relativizations of BP-complexity classes, *SIAM Journal on Computing* **18** (1989), pp. 449–462.

[46] C. B. Wilson, Relativization, reducibilities, and the exponential hierarchy, Technical Report TR-140, University of Toronto, 1980.

[47] C. B. Wilson, Relativized circuit complexity, *Journal of Computer and System Sciences* **31** (1985), pp. 169–181.