Relative to a Random Oracle, NP Is Not Small

Steven M. Kautz *
Department of Mathematics
Randolph-Macon Woman's College
2500 Rivermont Avenue
Lynchburg, VA 24503

Peter Bro Miltersen †
Department of Computer Science
University of Toronto
King's College Road
Toronto Ontario M5S 1A4
CANADA

March 13, 2008

Abstract

Resource-bounded measure as originated by Lutz is an extension of classical measure theory which provides a probabilistic means of describing the relative sizes of complexity classes. Lutz has proposed the hypothesis that NP does not have *p-measure zero*, meaning loosely that NP contains a non-negligible subset of exponential time. This hypothesis implies a strong separation of P from NP and is supported by a growing body of plausible consequences which are not known to follow from the weaker assertion P $\neq$ NP.

It is shown in this paper that relative to a random oracle, NP does not have p-measure zero. The proof exploits the following *independence* property of algorithmically random sequences: if A is an algorithmically random sequence and a subsequence A_0 is chosen by means of a *bounded Kolmogorov-Loveland*

*Much of this author's research was performed while visiting Iowa State University, supported by National Science Foundation Grant CCR-9157382, with matching funds from Rockwell International and Microware Systems Corporation.

†Supported by a grant of the Danish Science Research Council and partially supported by the ESPRIT II Basic Research Actions Program of the European Community under contract No. 7141 (project ALCOM II).

place selection, then the sequence A_1 of unselected bits is random relative to A_0, i.e., A_0 and A_1 are independent. A bounded Kolmogorov-Loveland place selection is a very general type of recursive selection rule which may be interpreted as the sequence of oracle queries of a time-bounded Turing machine, so the methods used may be applicable to other questions involving random oracles.

1 Introduction

The conjecture P $\neq$ NP is a reasonable working hypothesis because of the plausibility of its consequences and the body of empirical evidence supporting it. Lutz has proposed a stronger hypothesis, that NP does not have *p-measure zero*. This hypothesis has greater explanatory power than P $\neq$ NP and is supported by a number of credible consequences, some of which are summarized later in this section. The main result of this paper is that a strong form of Lutz's hypothesis holds relative to a random oracle, improving the previous result of Bennett and Gill [2] that P $\neq$ NP relative to a random oracle.

The meaning of "measure zero" is in terms of *resource-bounded measure*, an extension of classical measure or probability theory on $\{0,1\}^\infty$, due to Lutz [28]. The aspects of resource-bounded measure needed for the present results are introduced in Section 4. Intuitively, "NP does not have p-measure zero", written $\mu_p(\text{NP}) \neq 0$, indicates that NP contains a non-negligible subset of exponential time, or very loosely, a random exponential time language has nonzero probability of being in NP. Resource-bounded measure provides a meaningful answer to the question of what it means for an exponential time language to be "random". Since it is known that the class P does have p-measure zero, the hypothesis $\mu_p(\text{NP}) \neq 0$ implies that P $\neq$ NP.

Lutz and Mayordomo cite evidence in [26] for the plausibility of the hypothesis $\mu_p(\text{NP}) \neq 0$. In particular, its negation would imply the the existence of a betting algorithm for efficiently predicting membership in NP languages, a consequence which turns out to be intuitively quite unlikely. We discuss this in greater detail in Section 4 after giving the relevant definitions.

The hypothesis $\mu_p(\text{NP}) \neq 0$ also has a number of plausible consequences which are not known to follow from the weaker assertion P $\neq$ NP. In particular in [26] Lutz and Mayordomo prove that if $\mu_p(\text{NP}) \neq 0$, then the "Cook versus Karp-Levin" (CvKL) conjecture holds for NP, that is, there is a language which is $\leq_T^P$-complete but not $\leq_m^P$-complete for NP. Evidence for the plausibility of the CvKL conjecture as cited in [26] includes the following facts: The CvKL conjecture holds for E = DTIME(2^{linear}) (Ko and Moore, [20]) and for NE (Watanabe [45], Buhrman, Homer, and Torenvliet [4]). Under certain additional hypotheses it holds for PSPACE (Watanabe and Tang [46]). If E $\neq$ NE the CvKL conjecture holds for NP $\cup$ co-NP and if E $\neq$ NE $\cap$ co-NE it holds for NP (Selman [36]). Longpré and Young [25] also show that Cook reducibility is faster than Karp-Levin reducibility for certain classes of NP-complete sets. At this

time the CvKL conjecture is not known to be a consequence of the assertion P $\neq$ NP. This fact and the plausibility of the CvKL conjecture itself suggest that a stronger class separation such as $\mu_p(\text{NP}) \neq 0$ is likely to be true.

Other consequences of the hypothesis $\mu_p(\text{NP}) \neq 0$ cited in [26] include the following (here EE denotes the doubly-exponential class $\bigcup_{c=0}^{\infty} \text{DTIME}(2^{2^{n+c}})$ and NEE denotes the corresponding nondeterministic class):

1. E $\neq$ NE and EE $\neq$ NEE (Mayordomo [30], Lutz and Mayordomo [26]).

2. There exist NP search problems which are not reducible to the corresponding decision problems (this follows from item 1 above and a result of Bellare and Goldwasser [1]).

3. Every $\leq_m^P$-complete language for NP contains a dense exponential complexity core (Juedes and Lutz [11]).

4. For every real number $\alpha < 1$, every $\leq_{n^\alpha - tt}^P$-hard language for NP is dense (Lutz and Mayordomo [27]).

Our main result is that relative to a random oracle, NP does not have measure zero in E, meaning that NP$\cap$E does not have p-measure zero. This result implies that Lutz's hypothesis and its many consequences hold relative to a random oracle, and implies additionally that NP does not have measure zero in $\text{E}_2 = \text{DTIME}(2^{\text{polynomial}})$, i.e., NP does not have "p_2-measure zero" relative to a random oracle. (Here p and p_2 refer to the appropriate resource bounds for measure in E and in E_2, respectively; see Section 4 for definitions.) A further consequence is that, by virtue of a recent result by Regan, Sivakumar, and Cai [33], NP is *not measurable* in E or E_2 relative to a random oracle (see Section 6). Our arguments actually apply not just to NP but to some smaller classes such as FewP $-$ coNP; see Section 6.

It is difficult at this point to assess the exact meaning of a random oracle separation such as Bennett and Gill's [2] or the stronger result proved here. In [2], Bennett and Gill proposed the *random oracle hypothesis*, i.e., if a property holds relative to almost every oracle, then it must hold in unrelativized form. The random oracle hypothesis was first shown to be false in general by Kurtz [21], and more recently it has been shown that IP = PSPACE ([37]) but $\text{IP}^A \neq \text{PSPACE}^A$ for a random oracle A ([8], [15]). We do not suggest that the present result should be interpreted as "evidence" for the hypothesis $\mu_p(\text{NP}) \neq 0$, only that it is an interesting, related result. The view of the first author is that random oracle results are useful largely for the insight they give into the properties of algorithmically random sequences and into the nature of probabilistic computation.

In any event the present paper is a nontrivial improvement of [2] and introduces techniques which may be useful in other contexts. In particular we exploit the *independence properties* of sequences which are algorithmically random (in the sense of Martin-Löf, Levin, or Chaitin; see Section 3). Roughly, two sequences A_0 and A_1 are

independent if each is algorithmically random relative to the other. For example, it is shown in [42] and in [17] that if A is algorithmically random and A_0, A_1 denote the even and odd bits of A, respectively, then A_0 and A_1 are independent in this sense. Independence results for a number of other kinds of subsequences are given in [17]. Recently one of the authors [19] has shown that if the subsequence A_0 is chosen from A according to a *bounded Kolmogorov-Loveland place selection*, and A_1 denotes the nonselected bits, then A_0 and A_1 are independent. A Kolmogorov-Loveland place selection (see Definition 3.5) is a very general kind of recursive selection rule in which the nth bit selected may depend *adaptively* upon the $n-1$ previous bits in the subsequence as well as upon previously examined bits which are not necessarily included in the subsequence. The sequence of oracle queries of a Turing machine is an example of a subseqence chosen according to this type of rule; the result applies to a restriction of Kolmogorov-Loveland place selections which is nonetheless useful for dealing with time-bounded computations.

In the next section we review some of the terminology and notation to be used; Section 3 introduces algorithmic randomness, and in Section 4 we cover the pertinent notions of resource-bounded measure. Section 5 contains the proof of our main theorem, and in Section 6 we indicate some extensions and consequences of the main result.

2 Preliminaries

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$ denote the natural numbers. A *string* is an element of $\{0,1\}^*$ or $\{0, 1, \perp\}^*$, where the symbol $\perp$ is called an *undefined* bit. The concatenation of strings x and y is denoted xy. For any string x, $|x|$ denotes the length of x, and λ is the unique string of length 0. If $x \in \{0, 1, \perp\}^*$ and $j, k \in \mathbb{N}$ with $0 \leq j \leq k < |x|$, $x[k]$ is the kth bit (symbol) of x and $x[j..k]$ is the string consisting of the jth through kth bits of x (note that the "first" bit of x is the 0th). For an infinite binary sequence $A \in \{0,1\}^\infty$, the notations $A[k]$ and $A[j..k]$ are defined analogously. For any $x, y \in \{0, 1, \perp\}^*$, $x \sqsubseteq y$ means that if $x[k]$ is defined, then $y[k]$ is also defined and $x[k] = y[k]$; we say that x is an *initial segment*, or *predecessor*, of y or that y is an *extension* of x. Likewise for $A \in \{0,1\}^\infty$, $x \sqsubseteq A$ means $x[k] = A[k]$ whenever bit $x[k]$ is defined. Strings x and y are said to be *incompatible*, or *disjoint*, if there is no string z which is an extension of both x and y; when $x, y \in \{0,1\}^*$, this simply means that $x \not\sqsubseteq y$ and $y \not\sqsubseteq x$.

Fix a standard enumeration of $\{0,1\}^*$, $s_0 = \lambda, s_1 = 0, s_2 = 1, s_3 = 00, s_4 = 01, \ldots$. A *language* is a subset of $\{0,1\}^*$; a language A will be identified with its characteristic sequence $\chi_A \in \{0,1\}^\infty$, defined by $s_y \in A \iff \chi_A[y] = 1$ for $y \in \mathbb{N}$. We will consistently write A for χ_A. $\overline{A}$ denotes the bitwise complement of A, i.e., the set-theoretic complement of the language A in $\{0,1\}^*$. For $X \subseteq \{0,1\}^\infty$, X^c denotes the complement of X in $\{0,1\}^\infty$.

Typically strings in $\{0, 1, \perp\}^*$ will be used to represent *partially defined languages*, and will generally be represented by lower-case greek letters. For $\sigma \in \{0, 1, \perp\}^*$, when

no confusion is likely to result we will regard σ, $\sigma \perp^k$, and $\sigma \perp^\infty$ as essentially the same object, since all specify the same language fragment. We avoid using the notation $|\sigma|$ unless $\sigma \in \{0,1\}^*$, however following [28] we let $||\sigma||$ denote the number of defined bits in σ. When $\alpha \in \{0,1,\perp\}^*$ and $\tau \in \{0,1\}^*$, the notation $\alpha \downarrow \tau$ ("τ inserted into α") is defined by

$$(\alpha \downarrow \tau)[x] = \begin{cases} \alpha[x] & \text{if } \alpha[x] \text{ is defined,} \\ \tau[j] & \text{if } x \text{ is the } j\text{th undefined} \\ & \text{position in } \alpha \text{ and } j < |\tau|, \\ \perp & \text{otherwise.} \end{cases}$$

For $A, B \in \{0,1\}^\infty$, A/B is the subsequence of A *selected* by B, i.e., if $y_0, y_1, \ldots$ are the positions of the 1-bits of B in increasing order, then $(A/B)[x] = A[y_x]$. Note that A/B is a finite string if B contains only finitely many 1's. For $\sigma, \tau \in \{0,1\}^*$, σ/τ may be defined analogously. For $A, B \in \{0,1\}^\infty$, the sequence $A \oplus B$ is defined by

$$A \oplus B[x] = \begin{cases} A[\frac{x}{2}] & \text{if } x \text{ is even,} \\ B[\frac{x+1}{2}] & \text{if } x \text{ is odd.} \end{cases}$$

$\mathrm{E} = \mathrm{E}_1$ denotes the class $\mathrm{DTIME}(2^{\mathrm{linear}})$ and E_2 denotes $\mathrm{DTIME}(2^{\mathrm{polynomial}})$. Given a function $f : \{0,1\}^* \to \{0,1\}^*$, we say f is in the class $p = p_1$ if $f(x)$ is computable in time polynomial in $|x|$, and f is in the class p_2 if $f(x)$ is computable in time $|x|^{(log|x|)^{O(1)}}$. We assume there is a fixed pairing function on strings so that, for example, a function on $\mathbb{N} \times \{0,1\}^*$ can be interpreted as a function on $\{0,1\}^*$ (where the numeric input is represented as a unary string).

A string $\sigma \in \{0,1,\perp\}^*$ defines the subset $\mathrm{Ext}(\sigma) = \{A \in \{0,1\}^\infty : \sigma \sqsubseteq A\}$ of $\{0,1\}^\infty$, called a *cylinder*. $\mathrm{Ext}(\sigma)$ is referred to as an *interval* if $\sigma \in \{0,1\}^*$; evidently any cylinder is a union of intervals. Likewise if S is a subset of $\{0,1,\perp\}^*$, $\mathrm{Ext}(S)$ denotes $\bigcup_{\sigma \in S} \mathrm{Ext}(\sigma)$. By a *measure* we simply mean a probability distribution on $\{0,1\}^\infty$, and for our present purposes it is sufficient to consider the uniform distribution, i.e., each bit is equally likely to be a zero or a one, also called Lebesgue measure. The measure of a subset $\mathcal{E}$ of $\{0,1\}^\infty$, denoted $\mathbf{Pr}(\mathcal{E})$, can be intuitively interpreted as the probability that a sequence produced by tossing a fair coin is in the set $\mathcal{E}$; in particular the measure of an interval $\mathrm{Ext}(\sigma)$, abbreviated $\mathbf{Pr}(\sigma)$, is just $(\frac{1}{2})^{|\sigma|}$ (or $(\frac{1}{2})^{||\sigma||}$ if $\sigma \in \{0,1,\perp\}^*$). For S a set of strings, we abbreviate $\mathbf{Pr}(\mathrm{Ext}(S))$ by $\mathbf{Pr}(S)$; if S is *disjoint*, i.e., all strings in S are pairwise incompatible, then

$$\mathbf{Pr}(S) = \sum_{\sigma \in S} \mathbf{Pr}(\sigma).$$

Standard results of measure theory (see [13]) show that $\mathcal{E}$ is *measurable* (meaning that $\mathbf{Pr}(\mathcal{E})$ is defined) as long as $\mathcal{E}$ is a *Borel* set, i.e., built up from intervals by some finite iteration of countable union and complementation operations; however, for the most part we will be concerned with the measure of intervals and cylinders, which depend on only a finite number of bits, so our probability calculations are typically just finite combinatorial arguments.

3 Algorithmic randomness

"Random" sequences are not defined in classical measure theory; a statement of the form "Property $P(X)$ holds for a random sequence X" is a euphemism for

$$\mathbf{Pr}\{X : P(X)\} = 1. \tag{1}$$

An alternative approach is to explicitly define a class of sequences $\mathcal{R}$, show that $\mathbf{Pr}(\mathcal{R}) = 1$ and that sequences in $\mathcal{R}$ have the intuitive and mathematical properties normally associated with "randomness", and then replace (1) with the statement

$$\text{Property } P(X) \text{ holds for every } X \text{ in } \mathcal{R}. \tag{2}$$

This approach has some potential advantages: First, (2) is more informative than (1); (1) asserts that sequences with property P are plentiful, while (2) offers more or less explicit instructions (depending on the definition of $\mathcal{R}$) on how to find them. Second, the explicitness of the definition of $\mathcal{R}$ may actually make (2) easier to prove than (1). Attempts to employ notions of computability to define a suitable class $\mathcal{R}$ can be traced to Church's attempt [9] to interpret von Mises' intuitive definition of random sequences [44]. See [23], [40], and [42] for background; in particular [42] includes a critical examination of the role of computation in characterizing randomness.

Here we will simply take as $\mathcal{R}$ the class of *algorithmically random* sequences defined below. The definition we present is due to Martin-Löf [29]. The class defined is extremely robust, and various equivalent definitions have been given by Levin [22], Schnorr [34], Chaitin [5, 6, 7], and Solovay [39].

Recall that associated with any recursively enumerable (r.e.) set of strings is an index, e.g., the code of a program for enumerating the set (possibly relative to an oracle). A sequence $\{S_i\}$ of r.e. sets is *uniform* if there is a recursive function g such that for each i, $g(i)$ is an index of S_i.

Definition 3.1 A *constructive null cover*, or *Martin-Löf test*, is a uniform sequence of r.e. sets $\{S_i\}$ such that $\mathbf{Pr}(S_i) \leq 2^{-i}$. A sequence $A \in \{0,1\}^\infty$ is *algorithmically random*, or 1-*random*, if it is not contained in any constructive null cover, that is, for every constructive null cover $\{S_i\}$, $A \notin \bigcap_i \mathrm{Ext}(S_i)$.

Since $\mathbf{Pr}(\bigcap_i \mathrm{Ext}(S_i)) = 0$ for any constructive null cover $\{S_i\}$ and since there are only countably many of them, it is evident from the definition that the class $\mathcal{R}$ of algorithmically random sequences has measure 1. Martin-Löf [29] offers a rationale for believing that the definition captures every property of randomness that mathematicians are ever likely to require (see [42] for an alternative viewpoint).

The following characterization of algorithmic randomness, due to Solovay, provides an effective form of the Borel-Cantelli lemma (see [12, p.188]), i.e., if $\{E_i\}$ is a sequence of events (subsets of $\{0,1\}^\infty$) whose probabilities are summable, then with probability one only finitely many of the E_i occur. A proof can be found in [17], [38], or [39].

Theorem 3.2 *A sequence $A \in \{0,1\}^\infty$ is algorithmically random if and only if for every uniform sequence of r.e. sets $\{S_i\}$ such that $\sum_i \mathbf{Pr}(S_i) < \infty$, A is in only finitely many of the classes* $\mathrm{Ext}(S_i)$.

The proof of our main result rests heavily on the notion of *relative randomness*. We say that A is random relative to B, or A is *independent* of B, to mean in a strong sense that B "has no information about A" [41]. That is, not only is it the case that A cannot be computed from B, but access to B provides no help in guessing or approximating initial segments of A. In [41] van Lambalgen proposes an axiomatization of independence relations appropriate for characterizing randomness, and there it is shown that relative randomness as defined below satisfies the independence axioms.

Definition 3.3 Let $B \in \{0,1\}^\infty$. A *constructive null cover relative to B* is a uniform sequence $\{S_i\}$ of sets of strings, where each S_i is r.e. relative to B and $\mathbf{Pr}(S_i) \leq 2^{-i}$. A sequence $A \in \{0,1\}^\infty$ is *algorithmically random relative to B* if for every constructive null cover $\{S_i\}$ relative to B, $A \notin \bigcap_i \mathrm{Ext}(S_i)$. If A is algorithmically random relative to B and B is algorithmically random relative to A, then we say that A and B are *independent*.

A fundamental result about independence is the following; see [17] or [41].

Theorem 3.4 *(i) If $A \in \{0,1\}^\infty$ is algorithmically random and $A = A_0 \oplus A_1$, then A_0 and A_1 are independent. (ii) Conversely, if A_1 is algorithmically random and A_0 is algorithmically random relative to A_1, then $A_0 \oplus A_1$ is algorithmically random.*

Note that when the hypothesis of (ii) holds, it follows from (i) that A_1 is algorithmically random relative to A_0, i.e., independence is symmetric.

There is nothing special, of course, about splitting A into the even bits and odd bits as in Theorem 3.4; the proof of Theorem 3.4 is easily modified to show that for any recursive B, the subsequences A/B and $A/\overline{B}$ are independent. We are interested, however, in subsequences which depend on the sequence A itself. The definition below describes a very general selection process which encompasses several special cases of interest. The process may be pictured as follows, as suggested in [40]: Suppose the sequence A is represented as a row of cards laid face down; on the face of the ith card is either a zero or a one, corresponding to $A[i]$. We have two recursive functions, F and G, which are used to select some of the cards to create a second sequence, which we continue to call a "subsequence" even though the order of the cards may be changed. Both F and G look at the history of the selection process, that is, the sequence of cards turned over so far. The value of F is a natural number indicating the position of the next card to be turned over. The value of G is either 0 or 1; if the value is 0, the card is merely turned over and observed, while if the value is 1 the card is also *selected*, i.e., added onto the end of the subsequence.

Definition 3.5 A *Kolmogorov-Loveland place selection* [40] is a pair of partial recursive functions $F : \{0,1\}^* \to \mathbb{N}$ and $G : \{0,1\}^* \to \{0,1\}$. Let $A \in \{0,1\}^\infty$; F and G select a subsequence Q^* from A as follows. First define sequences of strings $\xi_0 \sqsubseteq \xi_1 \sqsubseteq \cdots$ and $\rho_0 \sqsubseteq \rho_1 \sqsubseteq \cdots$ such that $\xi_0 = \rho_0 = \lambda$, $\xi_{j+1} = \xi_j A[F(\xi_j)]$, and $\rho_{j+1} = \rho_j G(\xi_j)$ (with the proviso that ξ_{j+1} is undefined if $F(\xi_j) = F(\xi_i)$ for some $i < j$ or if either F or G fails to converge). If ξ_j and ρ_j are defined for all j let $Q = \lim_j \xi_j$ and $R = \lim_j \rho_j$. Thus Q represents the sequence of all bits of A examined by F, in the order examined. A given bit $Q[j] = A[F(\xi_j)]$ is included in the subsequence Q^* just if $G(\xi_j) = 1$, i.e. F determines which bits of A to examine, and G determines which ones to include in the sequence Q^*. Formally we define $Q^* = Q/R$. A Kolmogorov-Loveland place selection will be called *bounded* if the function G is determined by a partial recursive function $H : \{0,1\}^* \to \mathbb{N}$ with the following properties:

(i) H is nondecreasing, i.e., if $\xi \sqsubseteq \xi'$ then $H(\xi) \leq H(\xi')$,

(ii) H is unbounded, i.e., if ξ_j and ρ_j are defined for all j then $\lim_j H(\xi_j) = \infty$, and

(iii) G is determined by H according to the rule

$$\begin{aligned} F(\xi) < H(\xi) &\Rightarrow G(\xi) = 0 \\ F(\xi) \geq H(\xi) &\Rightarrow G(\xi) = 1. \end{aligned}$$

It is also useful to define a sequence B by $B[z] = 1$ if and only if for some j, $F(\xi_j) = z$ and $G(\xi_j) = 1$, so that $N = A/\overline{B}$ consists of the "nonselected" bits of A, in their natural order.

One special case of such a selection process, called a *Mises-Wald-Church place selection* (see [9], [23], [40], and [43]) is essentially a gambling strategy for a game in which the bits of A are revealed sequentially, such as by successive coin tosses, and a gambler may examine the past history of outcomes and decide via some algorithm whether to place a bet of a fixed amount on (i.e., to "select") the next toss. This process defines a subsequence, namely, the sequence of bits representing the outcomes on which the gambler placed a bet.

The real usefulness of Definition 3.5 for our purposes, however, is that the sequence of queries of a Turing machine using a sequence A as an oracle may be construed as a sequence selected according to a rule of this form. We might not see any immediate use for the idea that a bit may be observed without being selected, but as we will discover in Section 5, the extra generality is precisely what is needed to define a version of the oracle query sequence that excludes "irrelevant" information.

We are interested in whether the analog of Theorem 3.4 holds for more general subsequences. It is not difficult to show that if A is algorithmically random and Q^* is obtained by means of a Kolmogorov-Loveland place selection, then Q^* is algorithmically random also. (In effect this is a version of a classical result on the

impossibility of successful gambling strategies; see [13] or [10].) It is not known at this time whether, in general, Q^* and N are independent or even whether N is algorithmically random. The theorem below does answer the corresponding questions for *bounded* Kolmogorov-Loveland place selections, and will play a crucial role in the proof of our main result. The proof of Theorem 3.6 is based on techniques developed in [19] and in [17], where a number of independence properties are established for subsequences of random sequences.

Theorem 3.6 *Let F, G, and H be partial recursive functions determining a bounded Kolmogorov-Loveland place selection, let $A \in \{0,1\}^\infty$, and let N and Q^* be as in Definition 3.5. If A is algorithmically random and N is infinite, then N is algorithmically random relative to Q^*; thus by Theorem 3.4, N and Q^* are independent.*

4 Resource-bounded measure

Resource-bounded measure theory, as formulated by Lutz [28], is a form of effective measure theory which provides a means of describing the measure or probability of sets of languages within complexity classes and of defining the random languages (i.e., the pseudorandom sequences) within a complexity class. The formulation of Lutz is extremely general—for example, classical Lebesgue measure on $\{0,1\}^\infty$ is a special case—and is presented in [28] in terms of the powerful notion of *n-dimensional density systems*, but its origins may be traced back to the work of Schnorr on computable martingales [35]. Our presentation here is highly abbreviated, covering just those aspects needed for the proofs at hand, that is, to describe measure in the classes E and E_2, and we use the simpler language of martingales rather than density systems. The reader is encouraged to consult [31] or [28] for a more complete development of resource-bounded measure; see [13] or [47] for general background on martingales. There is one instance in the proof of Theorem 5.7 where we will need the slight extra generality of a *density function* as defined in [28], also called a *supermartingale*.

Definition 4.1 A *density function* is a function $d : \{0,1\}^* \to [0,\infty)$ such that for all $\sigma \in \{0,1\}^*$,

$$d(\sigma) \geq \frac{1}{2}d(\sigma 0) + \frac{1}{2}d(\sigma 1). \tag{3}$$

If (3) holds with equality then d is a *martingale*.

A martingale may be intuitively understood as a strategy for betting on the values of successive bits of a binary sequence. We picture the space $\{0,1\}^\infty$ of all possible sequences as a tree, the value $d(\lambda)$ at the root as the gambler's initial *capital*, and the value $d(\sigma)$ at node σ as the amount of capital she would possess after the initial sequence of outcomes σ. (Thus the "bet" at node σ on $i = 0$ or 1 corresponds to the

amount B, $0 \leq B \leq d(\sigma)$, for which $d(\sigma i) = d(\sigma) + B$ and $d(\sigma(1-i)) = d(\sigma) - B$.) The coefficient $\frac{1}{2}$ on $d(\sigma i)$ represents the conditional probability that the next bit is i, given the initial sequence of outcomes σ. Condition (3) with equality asserts that the game is *fair*, i.e., the gambler's expected gain at each node is zero. The sequences of particular interest are those for which the capital becomes unbounded as the game progresses.

Definition 4.2 A martingale (or density function) d *succeeds* on a sequence $A \in \{0,1\}^\infty$ if

$$\limsup_{n\to\infty} d(A[0..n]) = \infty.$$

Thus for a martingale d to succeed on a sequence A, it must be able to make a good prediction of the $(n+1)$st bit of A from the first n bits, and must do so often enough to win an infinite amount of money. Intuitively it is not surprising that this hardly ever happens; given a martingale or density function d, if a sequence A is generated by repeatedly tossing a fair coin, then d succeeds on A with probability zero. This is a consequence of the following lemma, which is a special case of a standard result known as *Kolmogorov's inequality for martingales* (see [13, p. 242]).

Lemma 4.3 *Let d be a density function and $a > 0$; then*

$$\mathbf{Pr}\{\sigma \in \{0,1\}^* : d(\sigma) > a\} < \frac{d(\lambda)}{a}.$$

Proof. Let $S = \{\sigma \in \{0,1\}^* : d(\sigma) > a\}$, and let $S_n = \{\sigma \in S : |\sigma| \leq n\}$. Since each string in S_n has a shortest predecessor in S_n, we can assume that S_n is disjoint, so by repeated application of (3) we have

$$d(\lambda) \geq \sum_{\sigma\in S_n} \frac{1}{2^{|\sigma|}} d(\sigma) > \sum_{\sigma\in S_n} \frac{a}{2^{|\sigma|}} = a \cdot \mathbf{Pr}(S_n).$$

The lemma then follows from the fact that $\mathbf{Pr}(S) = \lim_n \mathbf{Pr}(S_n)$. □

In particular if d (or some close approximation to d) is a recursive function, the sequence $\{S_i\}$ defined by $S_i = \{\sigma \in \{0,1\}^* : d(\sigma) > 2^i\}$ is a constructive null cover.

Definition 4.4 A *computation* of a martingale d is a function $\hat{d}$ from $\mathbb{N} \times \{0,1\}^*$ into the rational numbers such that $\left|\hat{d}_t(\sigma) - d(\sigma)\right| \leq 2^{-t}$ for all $\sigma \in \{0,1\}^*$ and $t \in \mathbb{N}$. For $i = 1$ or 2, $\hat{d}$ is a *p_i-computation* of d if $\hat{d} \in p_i$; then we also refer to d as a *p_i-martingale*. If the function d itself is in p_i, then d is called an *exact* p_i-martingale.

In this paper we are only concerned with showing that a given p_i-martingale is unable to succeed on a particular language. The following lemma, due independently to Juedes and Lutz [16] and to Mayordomo [31], ensures that for our purposes it suffices to consider only *exact* p_i-martingales. (See [16] for a somewhat stronger version.)

Lemma 4.5 *Let d be a p_i-martingale. Then there is an exact p_i-martingale $\tilde{d}$ such that if d succeeds on $A \in \{0,1\}^\infty$, then $\tilde{d}$ succeeds on A also.*

The measure structure of the class E_i is defined in terms of p_i-martingales. The key definition is the following.

Definition 4.6 Let $i = 1$ or 2. A class $X \subseteq \{0,1\}^\infty$ has *p_i-measure zero*, written $\mu_{p_i}(X) = 0$, if there is a p_i-martingale which succeeds on every sequence $A \in X$. Likewise $X \subseteq \{0,1\}^\infty$ has *p_i-measure one*, written $\mu_{p_i}(X) = 1$, if the complement X^c of X has p_i-measure zero. The class $X \subseteq \{0,1\}^\infty$ has *measure zero in* E_i, written $\mu(X \mid \mathrm{E}_i) = 0$, if $\mu_{p_i}(X \cap \mathrm{E}_i) = 0$; X has *measure one in* E_i if $\mu_{p_i}(X^c \cap \mathrm{E}_i) = 1$.

We also write $\mu_{p_i}(X) \neq 0$ to indicate that X does not have p_i-measure zero; note that this does not imply that $\mu_{p_i}(X)$ has some nonzero value, since it may be undefined, i.e., X may not be p_i-measurable. In particular if X is *closed under finite variation*—that is, any $A \in \{0,1\}^\infty$ which differs from some $B \in X$ on only a finite number of bits is also in X—then there are only three possibilities for the p_i-measure of X: $\mu_{p_i}(X) = 0$, $\mu_{p_i}(X) = 1$, or X is not p_i-measurable. (This is the resource-bounded form of the Kolmogorov zero-one law; see [31, p. 37] for a proof.) Note that Definition 4.4 may easily be relativized to an oracle $A \in \{0,1\}^\infty$, in which case we may write $\mu_{p_i}^A$ instead of μ_{p_i} in Definition 4.6.

If $\mu(X \mid \mathrm{E}_i) = 0$, we say that X is a *negligibly small* part of E_i; if $\mu(X \mid \mathrm{E}_i) = 1$, then *almost every* language $A \in \mathrm{E}_i$ is in X. Lutz [28] has proved a number of results justifying the use of this terminology, e.g., the measure zero sets in E_i behave set-theoretically like "small" sets and the measure one sets behave set-theoretically like "large" sets. It is also not difficult to show that $\mu(\mathrm{E}_i \mid \mathrm{E}_i) \neq 0$ and that $\mu(\mathrm{P} \mid \mathrm{E}_i) = 0$. We are interested in the size of NP in the classes E and E_2. Figure 1, adapted from [26], summarizes the known relationships among "non-smallness" conditions on NP. All the implications are easy consequences of the definitions except for the one marked with an asterisk, which follows from the lemma below due to Juedes and Lutz [16]. (The notation "$\mathrm{P}_m(X)$" denotes the downward closure of X under polynomial-time many-one reductions.)

Lemma 4.7 *Let $X \subseteq \{0,1\}^\infty$. If $\mu(X \mid \mathrm{E}) \neq 0$, then $\mu(\mathrm{P}_m(X) \mid \mathrm{E}_2) \neq 0$.*

Lutz has suggested that the conditions in Figure 1 be investigated as scientific hypotheses, i.e., evaluated in terms of explanatory power and intrinsic plausibility. In Section 1 we discussed some of the consequences of the hypothesis $\mu_p(\mathrm{NP}) \neq 0$; we conclude this section with a brief intuitive argument, originally given in [26], for the intrinsic plausibility of the hypothesis. The condition $\mu_p(\mathrm{NP}) = 0$ would imply that there exists a single p-martingale d which succeeds on every language $A \in \mathrm{NP}$. This means that there is a *fixed* polynomial x^c such that for every NP language A, given the first x bits of A, d has time $x^c \approx 2^{cn}$ to compute its bet on whether $s_x \in A$, where

$$\begin{array}{ccc} \mu(\mathrm{NP} \mid \mathrm{E}_2) \neq 0 & \overset{*}{\Longleftarrow} & \mu(\mathrm{NP} \mid \mathrm{E}) \neq 0 \\ \Updownarrow & & \Downarrow \\ \mu_{p_2}(\mathrm{NP}) \neq 0 & \Longrightarrow & \mu_p(\mathrm{NP}) \neq 0 \\ & & \Downarrow \\ & & \mathrm{P} \neq \mathrm{NP} \end{array}$$

Figure 1: Non-smallness conditions.

$n = |s_x| \approx \log x$. However, for arbitrarily large k, there are NP languages A for which determining whether $s_x \in A$ apparently requires checking 2^{kn} potential witnesses (possible nondeterministic computation paths). Thus an individual in possession of the algorithm d could successfully bet on *all* NP languages while only examining the fraction $2^{cn}/2^{kn}$ of the search space of potential witnesses. Since c is fixed and k is arbitrarily large, the fraction $2^{cn}/2^{kn} = 2^{c-k}$ is arbitrarily small; thus it seems extremely unlikely that such an algorithm could exist.

5 Main result

Our main result is that the strongest of the conditions in Figure 1 holds relative to a random oracle.

Theorem 5.1 *If $A \in \{0,1\}^\infty$ is algorithmically random, then*

(i) $\mu_p^A(\mathrm{NP}^A \cap \mathrm{E}^A) \neq 0$, *and*

(ii) $\mu_{p_2}^A(\mathrm{NP}^A \cap \mathrm{E}_2^A) \neq 0$.

Note that as stated, (ii) is a consequence of (i) by Lemma 4.7; however, we will see in Section 6 that the proof actually applies to some slightly smaller classes than NP, such as $\mathrm{NP} - \mathrm{coNP}$, which are not closed downward and hence not subject to Lemma 4.7. In any case the proofs of (i) and (ii) are essentially identical, so we will present both together and remark on the minor differences where appropriate. Theorem 5.1 is equivalent to Corollary 5.8 below, which is in turn a consequence of Theorems 3.6 and 5.7. Most of our work will be devoted to the proof of Theorem 5.7. Before we can formally state it we need to develop a number of definitions and preliminary results.

Remark 5.2 We make some simplifying assumptions about martingales. For $i = 1$ or 2, let d be an exact p_i-martingale (see lemma 4.5). We assume that $d(\lambda)$ is always 1. We assume that the values $d(\sigma 0)$ and $d(\sigma 1)$ are produced simultaneously, e.g., as a pair, and we use the notation $d(\sigma\Box)$ to denote this pair of values. The computation of $d(\sigma\Box)$ always begins by precisely duplicating the computations of $d(\sigma[0..i])$ for

$i = 0, 1, \ldots, |\sigma| - 1$, in order. Associated with d is a function $f \in p_i$ such that on an input sequence $\sigma \in \{0,1\}^*$ of length m, $d(\sigma\Box)$ runs for exactly $f(m)$ steps. We assume that each step includes exactly one query of an oracle A and that no bit of the oracle is queried more than once during the computation of $d(\sigma\Box)$. It is important to note that these assumptions do not restrict the sets of languages having p_i^A-measure zero. We suppress the superscript A throughout the sequel.

Throughout the discussion below, let $i = 1$ or $i = 2$ be fixed, let d be an exact p_i-martingale, and let f denote the time bound function for d as in Remark 5.2 above. We first define the construction of a language $L_A \in \mathrm{NP}^A \cap \mathrm{E}_i^A$ depending in a uniform way on $A \in \{0,1\}^\infty$; Theorem 5.1 will be established once we show that when A is algorithmically random, d does not succeed on L_A, where the computation of d is relative to oracle A.

If $y = |\sigma|$, then we think of $d(\sigma\Box)$ as first producing the value of the capital at node σ, and then determining how to bet on the next bit $L_A[y]$, i.e., on whether $s_y \in L_A$. To make its decision, d has time $f(y)$; note that since the yth bit of L_A represents a string s_y of length $n = \lfloor \log_2(y+1) \rfloor$, when expressed in terms of n the time bound on d is dominated by a function of the form $2^{t(n)}$, where $t(n) = cn$ if $f \in p_1$ and $t(n) = n^c$ if $f \in p_2$, for some constant c. Let

$$u(n) = t(n) + 2n + 1.$$

Let $\tilde{v}(n)$ be the real-valued function defined by

$$\left(1 - \frac{1}{2^{u(n)}}\right)^{\tilde{v}(n)} = \frac{1}{2},$$

and define

$$v(n) = \lfloor \tilde{v}(n) \rfloor.$$

Now given $A \in \{0,1\}^\infty$, we partition A into independent blocks of contiguous bits and let each block determine a single bit of L_A. For each y, the block corresponding to s_y will consist of $u(n) \cdot v(n)$ bits of A, where $n = |s_y|$. Specifically let

$$b_0 = 0, \qquad b_y = \sum_{x<y} u(|s_x|)v(|s_x|).$$

We will refer to $A[b_y..b_{y+1} - 1]$ as the yth *block* of A. The yth block of A determines the bit $L_A[y]$ according to the mapping Φ defined below.

Definition 5.3 Let u, v, and b_y be as described above, let $y \in \mathbb{N}$, $n = |s_y|$, and $A \in \{0,1\}^\infty$. Define

$$\Phi_y(A) = \begin{cases} 1 & \text{if } (\exists x < v(n))\,[A[b_y + x \cdot u(n) + j] = 0 \text{ for } 0 \le j < u(n)], \\ 0 & \text{otherwise.} \end{cases}$$

Then L_A is the language defined by $s_y \in L_A \iff \Phi_y(A) = 1$, i.e., $L_A[y] = \Phi_y(A)$. We may similarly define $\Phi_y(\sigma)$ for any string σ which is defined on bits $b_y, \ldots, b_{y+1} - 1$.

That is, $L_A[y] = 1$ just if for some x, the xth group of $u(n)$ bits within the yth block of A is all zeros. Such an x will be called a *witness* for $L_A[y] = 1$, and we say that d *finds a witness* for $L_A[y] = 1$ if for some x, d queries all $u(n)$ bits in the xth group and determines that all are zeros. We verify in Lemma 5.4 that the function v has been defined so that approximately half of the possible configurations of $A[b_y..b_{y+1} - 1]$ correspond to $L_A[y] = 0$. We also verify that $L_A \in \mathrm{NP}^A \cap \mathrm{E}_i^A$. The proof is a somewhat technical argument which can be skipped on a first reading without loss of continuity.

Lemma 5.4 *(i) Let $y \in \mathbb{N}$ and $n = |s_y|$; then*

$$\frac{1}{2} \leq \mathbf{Pr}(L_A[y] = 0) < \frac{1}{2} + \frac{1}{2^{u(n)}}.$$

(ii) $L_A \in \mathrm{NP}^A \cap \mathrm{E}_i^A$.

Proof. (i) For any fixed $x, 0 \leq x \leq v(n)$,

$$\begin{aligned} \mathbf{Pr}(x \text{ is a witness for } L_A[y] = 1) &= \mathbf{Pr}((\forall j < u(n))A[b_y + x \cdot u(n) + j] = 0) \\ &= \frac{1}{2^{u(n)}} \end{aligned}$$

and so

$$\mathbf{Pr}(x \text{ is not a witness for } L_A[y] = 1) = 1 - \frac{1}{2^{u(n)}}. \tag{4}$$

It follows that

$$\mathbf{Pr}(L_A[y] = 0) = \left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)}.$$

By the definition of $\tilde{v}(n)$ and the fact that $v(n) \leq \tilde{v}(n) < v(n) + 1$, we have

$$\left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)+1} < \frac{1}{2} \leq \left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)},$$

establishing the first inequality, and then

$$\begin{aligned} \left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)} - \frac{1}{2} &< \left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)} - \left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)+1} \\ &= \left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)} \left[1 - \left(1 - \frac{1}{2^{u(n)}}\right)\right] \\ &= \left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)} \left(\frac{1}{2^{u(n)}}\right) \\ &< \frac{1}{2^{u(n)}}, \end{aligned} \tag{5}$$

which establishes the second.

(ii) Since

$$\left(1 - \frac{1}{2^{u(n)}}\right)^{2^{u(n)}} \longrightarrow \frac{1}{e}$$

from below,

$$\left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)} \geq \frac{1}{2} > \left(1 - \frac{1}{2^{u(n)}}\right)^{2^{u(n)}}.$$

Hence we know that

$$v(n) < 2^{u(n)} \tag{6}$$

for all $n \geq 1$. Given $y \in \mathbb{N}$, let $n = |s_y|$; to determine whether $s_y \in L_A$ requires examining the yth block of A, which consists of $u(n) \cdot v(n)$ bits. For $i = 1$, $u(n) = (c+2)n + 1$, so by (6), $u(n) \cdot v(n)$ is of the form 2^{linear}; likewise for $i = 2$, $u(n) = n^c + 2n + 1$, so $u(n) \cdot v(n)$ is of the form $2^{\text{polynomial}}$. This establishes that $L_A \in \mathrm{E}_i^A$. A nondeterministic computation of $L_A[y]$ need only check the polynomially many bits in the yth block constituting the witness. Moreover, for any bit z in the yth block we have $b_y \leq z < b_{y+1}$, and it follows from (6) and the definition of b_y that $|s_z|$ is bounded by a polynomial; hence $L_A \in \mathrm{NP}^A$. □

Our object in defining L_A is to show that when A is algorithmically random, d cannot succeed on L_A. However, since L_A is obtained deterministically from A, d can get information about L_A by querying the oracle A. For any σ, $y = |\sigma|$ and $n = |s_y|$, the bit $L_A[y]$ depends on $u(n) \cdot v(n)$ bits of A, so there are $v(n)$ potential witnesses for $L_A[y] = 1$. During the computation of $d(\sigma\Box)$, i.e., in deciding how to bet on $L_A[y]$, d may make $2^{t(n)}$ queries. In particular this means that d can examine less than $2^{t(n)}$ potential witnesses. It turns out that $v(n)$ is of roughly the same order as $2^{u(n)} = 2^{t(n)+2n+1}$, so $2^{t(n)}$ is actually a very small fraction, about 2^{-2n}, of the potential witnesses. The lemma below confirms that d can gain only a very slight advantage by querying A; again, the proof could be skipped on a first reading.

Lemma 5.5 *(i) There exists a sequence $\{\delta_y\}$ with $\sum_y \delta_y < \infty$ such that for all $y \in \mathbb{N}$,*

$$\mathbf{Pr}(d \text{ finds a witness for } L_A[y] = 1) < \delta_y.$$

It follows that if A is algorithmically random, d finds a witness for $L_A[y] = 1$ for only finitely many y.

(ii) There exists a sequence $\{\epsilon_y\}$ with $\sum_y \epsilon_y < \infty$ such that for all $y \in \mathbb{N}$ and $n = |s_y|$, if $x_1, x_2, \ldots, x_r$ is any sequence of r natural numbers with $x_j < v(n)$ and $r \leq 2^{t(n)}$, then

$$\frac{1}{2} \leq \mathbf{Pr}(L_A[y] = 0 \mid x_1, \ldots, x_r \text{ are not witnesses for } L_A[y] = 1) < \frac{1}{2} + \epsilon_y.$$

Proof. We begin with a couple of technical inequalities. First note that since

$$\left(1-\frac{1}{2^x}\right)^{2^x} \longrightarrow \frac{1}{e}$$

monotonically from below and the left-hand-side is equal to $\frac{1}{4}$ when $x=1$,

$$\frac{1}{4} \leq \left(1-\frac{1}{2^x}\right)^{2^x} < \frac{1}{2}$$

for all $x>0$. It follows that

$$1-\frac{1}{2^x} \;<\; \left(\frac{1}{2}\right)^{2^{-x}}. \tag{7}$$

Similarly since

$$\left(1+\frac{1}{2^x}\right)^{2^x} \longrightarrow e > 2,$$

and again the limit is monotonic from below,

$$2^{2^{-x}} \;<\; 1+\frac{1}{2^x} \tag{8}$$

for all $x>0$. Then we can write

$$\begin{aligned}
\left(1-\frac{1}{2^{u(n)}}\right)^{2^{t(n)}} &= \left[\left(1-\frac{1}{2^{u(n)}}\right)^{2^{u(n)}}\right]^{2^{t(n)-u(n)}} \\
&\geq \left(\frac{1}{4}\right)^{2^{t(n)-u(n)}} \\
&= \left(\frac{1}{2}\right)^{2^{t(n)-u(n)+1}} \\
&= \left(\frac{1}{2}\right)^{2^{-2n}} \quad (\text{since } u(n)=t(n)+2n+1) \\
&> 1-\frac{1}{2^{2n}} \;\text{ by (7).}
\end{aligned} \tag{9}$$

It also follows that

$$\left(1-\frac{1}{2^{u(n)}}\right)^{-2^{t(n)}} \leq 2^{2^{-2n}} < 1+\frac{1}{2^{2n}} \;\text{ by (8).} \tag{10}$$

Now to prove (i), let us say that during the computation of $d(\sigma\Box)$, $|\sigma|=y$, d *examines* a potential witness $x<v(n)$ when d first queries some bit within the xth group of $u(n)$ bits in the yth block of A, i.e., within $A[b_y+x\cdot u(n)..b_y+(x+1)u(n)-1]$. There are $2^{u(n)}$ possible configurations of these $u(n)$ bits, and for only one of them

is x a witness; however, the next and subsequent bits queried by d, and in particular the next potential witness examined, may depend on the exact configuration of the xth group. In addition, d may query bits of A outside the yth block altogether. Each $A \in \{0,1\}^\infty$ determines a sequence $x_1, x_2, \ldots, x_r$, $r \leq 2^{t(n)}$, of witnesses examined by d. Since d is time-bounded, there is an $M \in \mathbb{N}$ such that while computing $d(\sigma\Box)$, d can only query bits $y \leq M$. Let $\alpha \sqsubseteq A$ denote the string $A[0..M]$ with each bit in the yth block replaced by the $\perp$ symbol, i.e.,

$$\alpha = A[0..b_y - 1](\perp)^{u(n)\cdot v(n)} A[b_{y+1}..M].$$

Let x_j denote the jth potential witness examined by d and let τ_j denote the x_jth group of $u(n)$ bits in the yth block. Clearly x_{j+1} depends on $\alpha, \tau_1, \ldots, \tau_j$, but it is always the case that the conditional probability that x_{j+1} is not a witness, *given* $\alpha, \tau_1, \ldots, \tau_j$, is $1 - 2^{-u(n)}$, by (4). It follows that

$$\begin{aligned} \mathbf{Pr}(\ x_1, \ldots, x_j \text{ are all not witnesses} \mid \alpha) &= \left(1 - \frac{1}{2^{u(n)}}\right)^r \\ &\geq \left(1 - \frac{1}{2^{u(n)}}\right)^{2^{t(n)}}, \end{aligned}$$

and since the strings α form a disjoint cover of $\{0,1\}^\infty$, we can sum over all α to conclude that

$$\mathbf{Pr}(d \text{ finds no witnesses}) \geq \left(1 - \frac{1}{2^{u(n)}}\right)^{2^{t(n)}}$$

and hence

$$\begin{aligned} \mathbf{Pr}(d \text{ finds at least one witness}) &\leq 1 - \left(1 - \frac{1}{2^{u(n)}}\right)^{2^{t(n)}} \\ &< \frac{1}{2^{2n}} \text{ by (9).} \end{aligned}$$

Let

$$\delta_y = \frac{1}{2^{2n}},$$

where $n = |s_y|$. Note that

$$\sum_y \delta_y = \sum_y \sum_{n=|s_y|} \frac{1}{2^{2n}} = \sum_n 2^n \cdot \frac{1}{2^{2n}} < \infty. \tag{11}$$

We noted above that since d makes only finitely many queries in computing $d(\sigma\Box)$, only a finite initial segment $A[0..M]$ is required to determine whether d finds a witness for $L_A[y] = 1$. Hence for each y we can define a set S_y to consist of all strings of the form $A[0..M]$ such that d finds a witness for $L_A[y] = 1$; evidently the sets S_y are uniformly r.e. and $\mathbf{Pr}(S_i) < \delta_y$. It follows from the Borel-Cantelli lemma (Theorem

3.2) that if $A \in \{0,1\}^\infty$ is algorithmically random, d finds a witness for $L_A[y] = 1$ for only finitely many y.

Then for (ii), using (4) we see that

$$\mathbf{Pr}(L_A[y] = 0 \mid x_1, \ldots, x_r \text{ are not witnesses for } L_A[y] = 1) \;=\; \left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)-r},$$

and since $v(n) - r < v(n) \leq \tilde{v}(n)$,

$$\left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)-r} \geq \left(1 - \frac{1}{2^{u(n)}}\right)^{\tilde{v}(n)} = \frac{1}{2},$$

which provides the first inequality. For the second inequality,

$$\begin{aligned}
\left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)-r} &\leq \left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)-2^{t(n)}} \\
&= \left(1 - \frac{1}{2^{u(n)}}\right)^{v(n)} \left(1 - \frac{1}{2^{u(n)}}\right)^{-2^{t(n)}} \\
&\leq \left(\frac{1}{2} + \frac{1}{2^{u(n)}}\right) \left(1 - \frac{1}{2^{u(n)}}\right)^{-2^{t(n)}} \text{ by (5)} \\
&< \left(\frac{1}{2} + \frac{1}{2^{u(n)}}\right) \left(1 + \frac{1}{2^{2n}}\right) \text{ by (10)} \\
&= \frac{1}{2} + \epsilon_y,
\end{aligned}$$

where

$$\epsilon_y = \frac{1}{2^{u(n)}} + \frac{1}{2^{2n+1}} + \frac{1}{2^{u(n)+2n}}$$

and $n = |s_y|$. Clearly $\{\epsilon_y\}$ is summable as in (11). □

Part (i) shows that if A is algorithmically random, d can actually find a witness for $L_A[y] = 1$ only finitely often. Part (ii) then asserts that the information d gains by *not* finding a witness gives it only a very slight advantage which is bounded by a rapidly decreasing sequence ϵ_y. The difficulty ahead of us is to show that the slight advantage d gains by querying A is not enough to enable it to succeed on L_A.

We will first define Q_A^*, the *bounded query sequence for* L_A, to consist of just those bits in the yth block of A which d was able to query before having to decide the value of $d(\sigma\Box)$, where $y = 0, 1, 2, \ldots$, and $\sigma = L_A[0..y-1]$. Informally, for any $z \in \mathbb{N}$ let y be the integer for which $b_y \leq z < b_{y+1}$, and define $B[z] = 1$ just if $A[z]$ is queried during the computation of $d(\sigma\Box)$, where $\sigma = L_A[0..y-1]$. Then Q_A^* will consist of the bits of A/B, not in their "natural" order in A, but in the order queried by d. We will also define $N_A = A/\overline{B}$, the *nonselected* bits of A. For the purpose of proving Theorem 5.7 below, we will use the following formal definition.

Definition 5.6 Fix an oracle $A \in \{0,1\}^\infty$. For a string σ with $|\sigma| = y$, define a function $F_\sigma : \{0,1\}^* \longrightarrow \mathbb{N}$ as follows: if ξ is the sequence of responses to the first $|\xi| < f(y)$ oracle queries in the computation of $d(\sigma\Box)$ relative to A, then $F_\sigma(\xi)$ is the position of the next bit to be queried. For an infinite sequence $C \in \{0,1\}^\infty$ let $F_C(\xi) = F_\sigma(\xi)$, where $\sigma = C[0..y-1]$ for the least y such that $|\xi| < f(y)$. Let

$$G_C(\xi) = \begin{cases} 0 & \text{if } F_C(\xi) < b_y \\ 1 & \text{if } F_C(\xi) \geq b_y. \end{cases}$$

Define sequences of strings $\xi_0 \sqsubseteq \xi_1 \sqsubseteq \cdots$ and $\rho_0 \sqsubseteq \rho_1 \sqsubseteq \cdots$ such that $\xi_0 = \rho_0 = \lambda$, $\xi_{j+1} = \xi_j A[F_C(\xi_j)]$, and $\rho_{j+1} = \rho_j G_C(\xi_j)$. Let $Q_C = \lim_j \xi_j$ and $R_C = \lim_j \rho_j$. Then the *bounded query sequence for* C is the sequence $Q_C^* = Q_C / R_C$, and in particular to avoid double subscripts we let Q_A^* denote $Q_{L_A}^*$, the bounded query sequence for $C = L_A$. We also define $B \in \{0,1\}^\infty$ by $B[z] = 1$ if and only if for some j, $F_{L_A}(\xi_j) = z$ and $G_{L_A}(\xi_j) = 1$; then $N_A = A/\overline{B}$ is the sequence of *nonselected* bits of A.

Q_A^* includes a relatively small part of the yth block of A, and moreover by Lemma 5.5(ii), if we look at the bits of N_A within the yth block (i.e., everything remaining in the yth block after the selection of Q_A^*), approximately half the possible configurations correspond to $L_A[y] = 0$ and half to $L_A[y] = 1$. Thus if d is successful in predicting whether $L_A[y] = 1$ based on the partial information represented in Q_A^*, then d in effect has a great deal of information about the nonqueried bits in the yth block of A, which should be impossible unless A itself has some kind of internal regularity, i.e., is nonrandom. What we prove is that if d succeeds on L_A, there is a martingale h (actually a density function) recursive in Q_A^* which succeeds on N_A. We then invoke Kolmogorov's inequality (Lemma 4.3) to conclude that N_A is contained in a constructive null cover relative to Q_A^*.

Theorem 5.7 *Let $A \in \{0,1\}^\infty$, and let L_A, Q_A^*, and N_A be as in Definition 5.6. Suppose that d succeeds on L_A using oracle A, and that d finds a witness for $L_A[y] = 1$ for only finitely many $y \in \mathbb{N}$. Then N_A is contained in a constructive null cover relative to Q_A^*.*

It is not difficult to see that the bounded query sequence of Definition 5.6 is an instance of a sequence selected via a bounded Kolmogorov-Loveland place selection (Definition 3.5), and the similarity in notation is deliberate. While the selection function F_{L_A} of Definition 5.6 apparently depends on L_A, this is really just an artifact of the time bound on d. Note that for a given string ξ, if y is the number for which $f(y-1) \leq |\xi| < f(y)$, then $F_{L_A}(\xi)$ is defined as $F_\sigma(\xi)$, where $\sigma = L_A[0..y-1]$, i.e., $F_{L_A}(\xi)$ depends only on an initial segment of L_A of length y. Moreover, $L_A[0..y-1]$ is determined completely by $A[0..b_y - 1]$, and as soon as $f(y)$ queries have been made no bit of A to the left of b_{y+1} can thereafter be added to the bounded query sequence, so in the absence of the time bound there is no reason that d could not then systematically examine all bits of A to the left of b_{y+1} and hence determine

$L_A[0..y]$. We can define p.r. functions F and H which select the subsequence Q_A^* from A according to Definition 3.5 as follows: Initially $H(\lambda) = 0$ and F simulates F_{L_A} on inputs ξ of length $|\xi| < f(0)$; since $F_{L_A}(\xi) = F_\lambda(\xi)$, no knowledge of L_A is required. For $y > 0$, having accumulated knowledge of $L_A[0..y-1]$, F may simulate $F_{L_A}(\xi)$ for $|\xi| < f(y)$. Upon reaching the point in the simulation that $|\xi| = f(y)$, the value of H is set to b_{y+1} and F then queries all previously unexamined bits in the yth block of A and thus determines $L_A[0..y]$. The subsequence Q^* selected by F and H is precisely the bounded query sequence of Definition 5.6. Note also that N_A is infinite since F_{L_A} can select at most $f(y)$ bits from $A[0..b_{y+1}-1]$.

It therefore follows from Theorem 3.6 that if A is algorithmically random, N_A is algorithmically random relative to Q_A^*; moreover by Lemma 5.5(i), d finds a witness for $L_A[y] = 1$ only finitely often. Then the desired conclusion is immediate from Theorem 5.7:

Corollary 5.8 *Let $A \in \{0,1\}^\infty$ and let L_A be the test language of Definition 5.3. Let d be an exact p_i-martingale as in Remark 5.2. If A is algorithmically random, d does not succeed on L_A relative to A.*

The proof of our main result, Theorem 5.1, will be complete once we prove Theorem 5.7.

Proof of Theorem 5.7. The plan of the proof is as follows. We first construct a function d^* recursive in Q_A^*, and then define the martingale h. The function d^* may be regarded more or less as a partially defined martingale which is attempting to succeed on N_A; thus we think of the inputs ρ to d^* as possible initial segments of N_A. The construction of d^* proceeds in stages; at stage $y+1$, d^* attempts to simulate the computation of $d(\sigma\Box)$ for strings σ of length y. Consider steps $f(y-1)$ through $f(y)-1$ in the computation of $d(\sigma\Box)$; if d queries bit z of the oracle, where $z \geq b_y$, the value of $A[z]$ is available from Q_A^*. The information in Q_A^* can then be used by d^* to construct an approximation of A, that is, to fill in some of the bits of a string $\alpha_\rho \in \{0,1,\perp\}^*$ which d^* "believes" to be an initial segment of a sequence A associated with a given input ρ. However, during this part of the computation of $d(\sigma\Box)$, values for $d(\sigma[0..j])$, $j < y$, have already been produced, so if a bit $z < b_y$ is queried by d, the value $A[z]$ is not part of the bounded query sequence Q_A^*, i.e, it resides in N_A. What d^* does is to use the input string ρ (which it hopes is an initial segment of N_A) to fill in the values of bits $z < b_y$ of α_ρ which are not provided by Q_A^*. Most input strings are not initial segments of N_A, of course, so most of the time the attempted simulation of d is incorrect, but there must be one sequence of inputs $\rho_0 \sqsubseteq \rho_1 \sqsubseteq \cdots$ which are true initial segments of N_A, and on these inputs d^* will correctly simulate d. We will then arrange to define $d^*(\rho)$ to be equal to an associated value $d(\sigma_\rho)$, where σ_ρ is a true initial segment of L_A, so that if $\limsup_j d(L_A[0..j]) = \infty$, then $\limsup_j d^*(N_A[0..j]) = \infty$ also.

The function d^*, unfortunately, is not a martingale, but in some sense it is a "biased" martingale, and the extent to which it is biased—the extent to which the probability associated with each bit differs from $\frac{1}{2}$—is bounded by the sequence ϵ_y of Lemma 5.5. Thus instead of a strategy d which has an advantage ϵ_y in betting on L_A, we have in a sense the "dual" problem of a strategy d^* betting on a sequence where the odds may be biased by ϵ_y. This is actually a fortuitous state of affairs, since the effect of the bias ϵ_y can now be more readily analyzed. The idea is to define the (unbiased) density function h by making careful adjustments in the values of d^* so that condition (3) can be satisfied, and to do so in such a way that the lim sup is preserved. The key to being able to do this is the fact that the sequence ϵ_y is rapidly decreasing.

We next give a formal description of the construction of d^*. The construction takes place in stages. If $d^*(\rho)$ is defined during stage y, we will say that the node ρ is *active* at stage $y+1$. At each stage, for each active node ρ, all the extensions of ρ having a certain fixed length will become defined. Associated with each active node ρ at stage $y+1$ is the following cast of characters:

α_ρ : belief about A at node ρ. α_ρ is a string over $\{0, 1, \perp\}^*$, but all bits $z < b_y$ are defined.

σ_ρ : belief about L_A corresponding to α_ρ, i.e., $\sigma_\rho[j] = \Phi_j(\alpha_\rho)$ for $j = 0, \ldots, y-1$. σ_ρ is a string of length y. (Φ is as in Definition 5.3.)

Γ_ρ : an integer representing the next bit of Q_A^* to be queried by d^*.

ξ_ρ : sequence of all query responses produced by the simulation of $d(\sigma_\rho)$. ξ_ρ is a string of length $f(y-1)$ (where we define $f(-1) = 0$).

Construction at stage 0: Let $d^*(\lambda) = d(\lambda) = 1$, and

$$\begin{aligned} \alpha_\lambda &= \perp^\infty, \\ \sigma_\lambda &= \lambda, \\ \Gamma_\lambda &= 0, \text{ and} \\ \xi_\lambda &= \lambda. \end{aligned}$$

Construction at stage y + 1: For each active node ρ, let $\alpha = \alpha_\rho$, $\sigma = \sigma_\rho$, $\Gamma = \Gamma_\rho$, and $\xi = \xi_\rho$. Note that $y = |\sigma|$ by construction. We simulate the computation of $d(\sigma\Box)$ on steps $f(y-1)$ through $f(y)-1$, extending α and ξ as follows:

for $j := f(y-1)$ **to** $f(y)-1$ **do**
  $z := F_\sigma(\xi)$
  if $z \geq b_y$ **then**
    $\xi[j] := Q_A^*[\Gamma]$
    $\alpha[z] := Q_A^*[\Gamma]$

$\Gamma := \Gamma + 1$

else

$\xi[j] := \alpha[z]$

Simulate the jth step in
the computation of $d(\sigma\Box)$,
using $\xi[j]$ for the oracle response.

Then let

$$\begin{aligned} a &= d(\sigma 0), \\ b &= d(\sigma 1), \\ \text{and } k &= \text{number of undefined bits} \\ &\quad \text{in } \alpha[b_y..b_{y+1}-1]. \end{aligned} \tag{12}$$

For each string $\tau \in \{0,1\}^k$ define d^* on node $\rho\tau$ by:

$$d^*(\rho\tau) = \begin{cases} a & \text{if } \Phi_y(\alpha \downarrow \tau) = 0 \\ b & \text{if } \Phi_y(\alpha \downarrow \tau) = 1. \end{cases}$$

Finally let

$$\sigma_{\rho\tau} = \begin{cases} \sigma 0 & \text{if } \Phi_y(\alpha \downarrow \tau) = 0 \\ \sigma 1 & \text{if } \Phi_y(\alpha \downarrow \tau) = 1, \end{cases}$$

$$\begin{aligned} \alpha_{\rho\tau} &= \alpha \downarrow \tau, \\ \Gamma_{\rho\tau} &= \Gamma, \text{and} \\ \xi_{\rho\tau} &= \xi. \end{aligned}$$

This completes stage $y+1$.

Claim 5.9 *Let $\lambda = \rho_0 \sqsubseteq \rho_1 \sqsubseteq \rho_2 \sqsubseteq \cdots$ be the unique sequence of strings such that for each y, $\rho_y \sqsubseteq N_A$ and ρ_y is active at stage $y+1$. If d succeeds on L_A, then $\limsup_y d^*(\rho_y) = \infty$.*

Proof of Claim 5.9. It can be shown by induction that at stage $y+1$,

$$\begin{aligned} \alpha_{\rho_y} &\sqsubseteq A, \\ \sigma_{\rho_y} &= L_A[0..y-1], \text{ and} \\ d^*(\rho_y) &= d(\sigma_{\rho_y}). \end{aligned}$$

Hence $\limsup_y d^*(\rho_y) = \limsup_y d(L_A[0..y-1]) = \infty$. $\Box$

We next define the density function h. During the following discussion let ρ be an active node at any stage $y+1$, and let $\lambda = \rho_0 \sqsubseteq \rho_1 \sqsubseteq \cdots \sqsubseteq \rho_y = \rho$ be the predecessors of ρ such that ρ_j is active at stage $j+1$. Let $\alpha = \alpha_\rho$ and $\sigma = \sigma_\rho$ be the strings associated with ρ in the construction at stage $y+1$, let a, b, and k be as in (12), and let $c = d^*(\rho) = d(\sigma)$. The plan is to define h by adjusting the values of d^* at each node so the condition

$$h(\rho) \;\geq\; \frac{1}{2^k} \sum_{|\tau|=k} h(\rho\tau) \tag{13}$$

can be satisfied (note that (13) is just the obvious extension of (3)). First we need the following:

Definition 5.10 The node ρ is *bad* if $\Phi_y(\alpha \downarrow \tau) = 1$ for every $\tau \in \{0,1\}^k$. If ρ is not bad it is *good*. If ρ is bad and $b = 0$, then each of the nodes $\rho\tau$ will be called *dead*.

Note that a node is bad if the bits of the oracle queried by $d(\sigma\Box)$ actually include a group of zeros witnessing that $L_A[y] = 1$. It follows that $d^*(\rho\tau) = d(\sigma 1) = b$ for every $\tau \in \{0,1\}^k$. Note also that by construction, if σ_ρ is the string associated with the active node ρ so that $d^*(\rho) = d(\sigma_\rho)$, then for every active node ρ' extending ρ, the associated string $\sigma_{\rho'}$ extends σ_ρ. It follows that if a node ρ is dead, then $d^*(\rho') = d(\sigma_\rho) = 0$ for every ρ' extending ρ. We should also note that the goodness or badness of a node ρ is evident at the stage in the construction where ρ is active.

Now if ρ is good, we define the quantity

$$q_\rho = \frac{\Big|\{\tau \in \{0,1\}^k : d^*(\rho\tau) = a\}\Big|}{2^k},$$

if $a \neq b$, and $q_\rho = \frac{1}{2}$ if $a = b$. That is, q_ρ is the proportion of the extensions $\rho\tau$ of ρ, $|\tau| = k$, for which $d^*(\rho\tau) = a$, or equivalently, for which $\Phi_y(\alpha \downarrow \tau) = 0$. We will shortly need the following crucial fact about q_ρ:

Claim 5.11 *Let $\{\epsilon_y\}$ be the sequence defined in Lemma 5.5(ii), and let ρ be a node active at stage $y+1$ as above. Then*

$$\frac{1}{2} \leq q_\rho < \frac{1}{2} + \epsilon_y. \tag{14}$$

Proof of Claim 5.11 Let $x_1, \ldots, x_r$ denote those potential witnesses $x_j < v(n)$ such that the x_jth group of $u(n)$ bits in the yth block of α_ρ includes a defined bit; note $r \leq k \leq 2^{t(n)}$. Then

$$\begin{aligned}
\frac{1}{2} \;&\leq\; \mathbf{Pr}(L_A[y] = 0) \\
&=\; \mathbf{Pr}\{\tau \in \{0,1\}^{u(n)v(n)} : \Phi_y(\tau) = 0\} \\
&\leq\; \mathbf{Pr}\{\tau \in \{0,1\}^k : \Phi_y(\alpha \downarrow \tau) = 0\} \\
&\leq\; \mathbf{Pr}(L_A[y] = 0 \mid x_1, \ldots, x_r \text{ are not witnesses}) \\
&<\; \frac{1}{2} + \epsilon_y.
\end{aligned}$$

The first inequality is Lemma 5.4(i); the last is Lemma 5.5(ii). □

Let $m_\lambda = 1$; if ρ is good, for each string $\tau \in \{0,1\}^k$ let

$$m_{\rho\tau} = \begin{cases} \dfrac{1}{2q_\rho} & \text{if } d^*(\rho\tau) = a \\ \\ \dfrac{1}{2(1-q_\rho)} & \text{if } d^*(\rho\tau) = b, \end{cases}$$

and if ρ is bad let

$$m_{\rho\tau} = \begin{cases} \dfrac{c}{b} & \text{if } b \neq 0 \\ \\ 0 & \text{if } b = 0 \end{cases}$$

We will also need the observation that since $(a+b)/2 = c$ (by (3), since d is a martingale), it is always the case that if $b \neq 0$, then

$$\frac{c}{b} = \frac{1}{2} + \frac{a}{2b} \geq \frac{1}{2}. \tag{15}$$

Now to define $h(\rho)$, let $h(\lambda) = d^*(\lambda) = 1$, let

$$M_\rho = \prod_{j=0}^{y} m_{\rho_j},$$

and let

$$h(\rho) = M_\rho \cdot d^*(\rho).$$

Claim 5.12 *If the strings $\rho\tau$ are not dead, then*

$$h(\rho) = \frac{1}{2^k} \sum_{|\tau|=k} h(\rho\tau).$$

Proof of Claim 5.12. The numbers m_ρ have been defined so that the following conditions hold. If ρ is good,

$$\begin{aligned} d^*(\rho) \;=\; c \;&=\; \frac{a+b}{2} \\ &=\; \frac{1}{2^k}\left[2^k q_\rho \left(\frac{1}{2q_\rho}\right) a + 2^k(1-q_\rho)\left(\frac{1}{2(1-q_\rho)}\right) b\right] \\ &=\; \frac{1}{2^k} \sum_{|\tau|=k} m_{\rho\tau} d^*(\rho\tau) \end{aligned}$$

since $2^k q_\rho$ is the number of extensions $\rho\tau$ for which $d^*(\rho\tau) = a$, and $2^k(1-q_\rho)$ is the number for which $d^*(\rho\tau) = b$. If ρ is bad, then $d^*(\rho\tau) = b$ for all $\tau \in \{0,1\}^k$, so as long as $b \neq 0$ (by hypothesis the nodes $\rho\tau$ are not dead), we have

$$\begin{aligned} d^*(\rho) \;=\; c \;&=\; \frac{c}{b} \cdot b \\ &=\; m_{\rho\tau} d^*(\rho\tau) \\ &=\; \frac{1}{2^k} \sum_{|\tau|=k} m_{\rho\tau} d^*(\rho\tau). \end{aligned}$$

Thus in either case

$$\begin{aligned} M_\rho \cdot c \;&=\; \frac{1}{2^k} \sum_{|\tau|=k} M_\rho \cdot m_{\rho\tau} d^*(\rho\tau), \text{ i.e.,} \\ h(\rho) \;&=\; \frac{1}{2^k} \sum_{|\tau|=k} h(\rho\tau). \end{aligned}$$

□

We can extend the domain of h to all strings γ as follows: if $\gamma 0, \gamma 1$ are dead strings of the form $\rho\tau$, then let $h(\gamma) = c = d^*(\rho)$; otherwise simply require

$$h(\gamma) = \frac{1}{2} h(\gamma 0) + \frac{1}{2} h(\gamma 1).$$

Claim 5.12 ensures that this can be done. (We have not defined d^* on strings other than the *active* nodes, but it is not necessary to do so.) Since $h(\rho\tau) = 0$ when $\rho\tau$ is dead and thus (13) is trivially satisfied, we have shown:

Claim 5.13 *h is a density function.*

The next step is to establish a relationship between h and d^*. We have seen that to obtain each value $h(\rho)$ we multiplied the corresponding value of $d^*(\rho)$ by a number M_ρ. The idea in the arguments that follow is to show that the numbers M_ρ cannot get too small, so that $h(\rho)$ cannot be too much smaller than $d^*(\rho)$. Let $\{\epsilon_y\}$ be the sequence defined in Lemma 5.5. We first need a minor technical fact.

Claim 5.14 *There exists a constant J such that*

$$\prod_y \frac{1}{1+2\epsilon_y} > 2^{-J}.$$

Proof of Claim 5.14. Since

$$(1+2\epsilon_y)^{\frac{1}{2\epsilon_y}} \longrightarrow e < 4,$$

it follows that

$$(1+2\epsilon_y) < 4^{2\epsilon_y} = 2^{4\epsilon_y}$$

and hence

$$\log(1+2\epsilon_y) < 4\epsilon_y;$$

since $\{\epsilon_y\}$ is summable we may let J be any number for which

$$\sum_y \log(1+2\epsilon_y) < J < \infty$$

Thus

$$\sum_y \log\left(\frac{1}{1+2\epsilon_y}\right) > -J$$

and

$$\prod_y \frac{1}{1+2\epsilon_y} > 2^{-J}.$$

□

Claim 5.15 *Let $\rho_0 \sqsubseteq \rho_1 \sqsubseteq \rho_2 \sqsubseteq \cdots$ be a finite or infinite sequence of nodes such that ρ_j is active at stage $j+1$, none of the ρ_j are dead, and only finitely many of the ρ_j are bad. Let K denote the number of bad nodes in the sequence and J the constant of Claim 5.14. Then for each node ρ_j in the sequence,*

$$M_{\rho_j} > 2^{-J}2^{-K}.$$

Proof of Claim 5.15. If ρ_j is a good node, then $\frac{1}{2} \leq q_{\rho_j} < \frac{1}{2} + \epsilon_j$ by (14), so $1 \leq 2q_{\rho_j} < 1 + 2\epsilon_j$ and

$$\frac{1}{1+2\epsilon_j} < \frac{1}{2q_{\rho_j}} \leq 1.$$

For good nodes ρ we always have

$$\frac{1}{2(1-q_\rho)} \geq m_{\rho\tau} \geq \frac{1}{2q_\rho},$$

and for the bad nodes ρ it is always the case by (15) that

$$m_{\rho\tau} \geq \frac{1}{2}.$$

Thus for any ρ_y in the sequence,

$$\begin{aligned} M_{\rho_y} &= \prod_{\substack{j<y \\ \rho_j \text{ good}}} m_{\rho_{j+1}} \cdot \prod_{\substack{j<y \\ \rho_j \text{ bad}}} m_{\rho_{j+1}} \\ &\geq \prod_{j=0}^{y} m_{\rho_j} \cdot \left(\frac{1}{2}\right)^K \end{aligned}$$

$$\begin{aligned} &\geq \prod_{j=0}^{y} \frac{1}{2q_{\rho_j}} \cdot 2^{-K} \\ &\geq \prod_{j} \frac{1}{1+2\epsilon_j} \cdot 2^{-K} \\ &\geq 2^{-J}2^{-K}. \end{aligned}$$

□

Let $\rho_0 \sqsubseteq \rho_1 \sqsubseteq \rho_2 \sqsubseteq \cdots$ be the unique sequence of active nodes with $\rho_y \sqsubseteq N_A$. By hypothesis d finds a witness for $L_A[y] = 1$ only finitely often, so only finitely many of the ρ_y are bad; let K_0 denote the number of bad nodes in the sequence. For each $t \in \mathbb{N}$, we may define a set S_t as the enumeration of those strings ρ satisfying the following conditions:

(i) ρ is active at some stage in the construction,

(ii) at most K_0 of the active nodes $\rho' \sqsubseteq \rho$ are bad,

(iii) $d^*(\rho) > 2^{J+K_0} \cdot 2^t$.

Since d^* is recursive in Q_A^*, the sets S_t are uniformly r.e. relative to Q_A^*. By Claim 5.9, S_t contains some initial segment of N_A, so

$$N_A \in \bigcap_t \mathrm{Ext}(S_t).$$

Claim 5.16 $\mathbf{Pr}(S_t) \leq 2^{-t}$, *i.e.,* $\{S_t\}$ *is a constructive null cover.*

Proof of Claim 5.16. First define

$$S_t' = \{\rho : h(\rho) > 2^t\}.$$

By Kolmogorov's inequality (Lemma 4.3), $\mathbf{Pr}(S_t') < 2^{-t}$, so it will suffice to show that $S_t \subseteq S_t'$. Suppose $\rho \in S_t$; let $y+1$ be the stage at which ρ is active, and let $\rho_0 \sqsubseteq \cdots \sqsubseteq \rho_y = \rho$ be the active predecessors of ρ. Note that none of the ρ_j are dead (see the remarks following Definition 5.10). It then follows from Claim 5.15 that

$$h(\rho) = M_\rho d^*(\rho) > 2^{-J-K_0} \cdot d^*(\rho) > 2^t,$$

so $\rho \in S_t'$. □

This completes the proof of Theorem 5.7. □

6 Remarks

It is not difficult to see that the test language L_A of Theorem 5.1 is not in coNP^A, and hence that the theorem actually applies to NP − coNP: Consider the class $\mathcal{C} = \{A : L_A \in \mathrm{coNP}^A\}$. If $\mathcal{C}$ has positive measure, there is a fixed nondeterministic machine M with polynomial time bound g such that the class

$$\{A : (\forall y)[L_A[y] = 0 \iff M^A(y) \text{ has an accepting path}]\} \tag{16}$$

has positive measure, and hence (using standard techniques) there is an interval α such that the density of (16) in $\mathrm{Ext}(\alpha)$ is at least $\frac{3}{4}$, i.e.,

$$\mathbf{Pr}\{(\forall y)[L_A[y] = 0 \iff M^A(y) \text{ has an accepting path}] \mid \alpha \sqsubseteq A\} > \frac{3}{4}. \tag{17}$$

Suppose $y \in \mathbb{N}$ is sufficiently large that (using the notation of Lemma 5.5) $b_y > |\alpha|$, $\epsilon_y < \frac{1}{4}$, and $g(n) < 2^{t(n)}$, where $n = |s_y|$. $M^A(y)$ examines fewer than $g(n)$ potential witnesses $x_1, \ldots, x_r$ in the yth block of A, and by Lemma 5.5(ii),

$$\mathbf{Pr}(L_A[y] = 1 \mid \ x_1, \ldots, x_r \text{ are not witnesses }) \geq \frac{1}{2} - \epsilon_y > \frac{1}{4}$$

where the probability is independent of α since $b_y > |\alpha|$. Certainly if $M^A(y)$ has an accepting path, so does $M^{\tilde{A}}(y)$ for any $\tilde{A}$ having the same values as A on the $g(n)$-bit query sequence for the accepting path. It follows that there exists a y such that at least one-fourth of the measure of the set $\{A : M^A(y) \text{ accepts}\}$ consists of sequences $\tilde{A}$ for which $L_{\tilde{A}}[y] = 1$, and hence

$$\mathbf{Pr}\{(\exists y)[L_A[y] = 1 \text{ and } M^A(y) \text{ has an accepting path}] \mid \alpha \sqsubseteq A\} > \frac{1}{4},$$

contradicting (17); thus $\mathbf{Pr}(\mathcal{C}) = 0$, i.e., $L_A \notin \mathrm{coNP}^A$ with probability one. To see that $L_A \notin \mathrm{coNP}^A$ for every *algorithmically* random A, it is enough to note that $\mathcal{C}$ is a union of recursively closed sets (a Σ^0_2-class) and hence contains no algorithmically random sequences by the "effective zero-one law" of [18] or by Theorem 2 of [3].

It was observed by Longpré [24] that the proof of Theorem 5.1 shows that $\mu(\mathrm{FewP} \mid \mathrm{E}_i) \neq 0$, since with probability one, the number of witnesses for $L_A[y] = 1$ is less than $n = |s_y|$ for all but finitely y. To see this, note that the inequality (6) can be extended to show that for all sufficiently large n,

$$2^{u(n)-1} < v(n) < 2^{u(n)},$$

so that

$$\frac{1}{2} < \frac{v(n)}{2^{u(n)}} < 1. \tag{18}$$

For a given y, the number X of witnesses for $L_A[y] = 1$ is a binomial random variable corresponding to $v(n)$ independent trials with probability $2^{-u(n)}$, where $n = |s_y|$; the mean of X is $v(n)/2^{u(n)}$. Using (18) and Chernoff bounds (see [32, p. 71] or [14]),

$$\mathbf{Pr}(X > n) < \frac{1}{\sqrt{e}} \left(\frac{e}{n}\right)^{\frac{n}{2}}.$$

Since

$$\sum_y \frac{1}{\sqrt{e}} \left(\frac{e}{n}\right)^{\frac{n}{2}} = \sum_n \frac{2^n}{\sqrt{e}} \left(\frac{e}{n}\right)^{\frac{n}{2}} < \infty,$$

it follows from the Borel-Cantelli lemma (Theorem 3.2) than when A is algorithmically random there are only finitely many y with more than $n = |s_y|$ witnesses.

Regan, Sivakumar and Cai [33] have recently shown that if $\mathcal{C}$ is any class closed under finite unions and intersections such that $\mathrm{E}_i - \mathcal{C}$ is nonempty, then $\mathcal{C}$ cannot have measure one in E_i. Since it can be shown that for an algorithmically random oracle A, $\mathrm{E}_i^A - \mathrm{NP}^A$ is nonempty, it follows that NP^A is neither p- nor p_2-measurable. The same reasoning can be applied to any class $\mathcal{C}$ containing FewP such that relative to a random oracle A, $\mathcal{C}^A$ does not contain all of E_i^A. For example, we have that relative to a random oracle the classes $\oplus$P, PP, Σ_k^p, Π_k^p, PH and PSPACE are all not measurable in E_2.

Acknowledgments

The authors would like to acknowledge the contributions of Jack Lutz and Elvira Mayordomo, each of whom has discussed the problem at length with one or both of the authors. We thank the referee for several suggestions that simplified the presentation. The first author would also like to thank Jack Lutz for providing the moral and financial support for the author's visit to Iowa State University in the summer of 1993, where much of this work took place.

References

[1] M. Bellare and S. Goldwasser. The complexity of decision versus search. To appear in *SIAM Journal on Computing*.

[2] C. Bennett and J. Gill. Relative to a random oracle, $P^A \neq NP^A \neq$ co-NP^A. *SIAM Journal on Computing*, 10:96–113, 1981.

[3] R. Book, J. Lutz, and K. Wagner. An observation on probability versus randomness with applications to complexity classes. *Math. Systems Theory*, 27:201–209, 1994.

[4] H. Buhrman, S. Homer, and L. Torenvliet. Completeness for nondeterministic complexity classes. *Mathematical Systems Theory*, 24:179–200, 1991.

[5] G.J. Chaitin. *Algorithmic Information Theory*. Cambridge University Press, 1987.

[6] G.J. Chaitin. Incompleteness theorems for random reals. *Advances in Applied Mathematics*, 8:119–146, 1987.

[7] G.J. Chaitin. A theory of program size formally identical to information theory. *J. Assoc. Comput. Mach.*, 22:329–340, 1975.

[8] B. Chor, O. Goldreich, and J. Hastad. *The Random Oracle Hypothesis is False*. Technical Report 631, Department of Computer Science, Technion, 1990.

[9] A. Church. On the concept of a random sequence. *Bulletin of the AMS*, 46:130–135, 1940.

[10] J.L. Doob. Note on probability. *Annals of Mathematics*, 37(2):363–367, 1936.

[11] D. W. Juedes and J. H. Lutz. The complexity and distribution of hard problems. *SIAM Journal on Computing*, 24(2):279–295, 1995.

[12] W. Feller. *An Introduction to Probability Theory and its Applications*. Volume 1, John Wiley and Sons, Inc., 1957.

[13] W. Feller. *An Introduction to Probability Theory and its Applications*. Volume 2, John Wiley and Sons, Inc., 1971.

[14] T. Hagerup and C Rüb. A guided tour of Chernoff bounds. *Information Processing Letters*, 33:305–308, 1989/90.

[15] J. Hartmanis, R. Chang, D. Ranjan, and P. Rohatgi. Structural complexity: recent surprises. In *Proceedings of the Second Scandinavian Workshop on Algorithm Theory*, pages 1–12, Springer-Verlag, 1990.

[16] D. Juedes and J. Lutz. Weak completeness in E and E_2. *Theoretical Computer Science*, 143:149–158, 1995.

[17] S. M. Kautz. *Degrees of Random Sets*. PhD thesis, Cornell University, 1991.

[18] S. M. Kautz. An improved zero-one law for algorithmically random sequences. 1994. Manuscript.

[19] S. M. Kautz. Independence properties of algorithmically random sequences. 1994. Manuscript.

[20] K. Ko and D. Moore. Completeness, approximation, and density. *SIAM Journal on Computing*, 10:787–796, 1981.

[21] S.A. Kurtz. On the random oracle hypothesis. *Information and Control*, 57:40–47, 1983.

[22] L.A. Levin. On the notion of a random sequence. *Soviet Math. Dokl.*, 14:1413–1416, 1973.

[23] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, 1993.

[24] L. Longpré. 1993. Personal communication.

[25] L. Longpré and P. Young. Cook reducibility is faster than Karp reducibility in NP. *Journal of Computer and System Sciences*, 41:389–401, 1990.

[26] J. H. Lutz and E. Mayordomo. Cook versus Karp-Levin: separating completeness notions if NP is not small. In *Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science*, Springer-Verlag, 1994. To appear in *Theoretical Computer Science*.

[27] J. H. Lutz and E. Mayordomo. Measure, stochasticity, and the density of hard languages. *SIAM Journal on Computing*, 23:762–779, 1994.

[28] J.H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44:220–258, 1992.

[29] P. Martin-Löf. The definition of random sequences. *Information and Control*, 9:602–619, 1966.

[30] E. Mayordomo. Almost every set in exponential time is P-bi-immune. *Theoretical Computer Science*, 136:487–506, 1994. Also in *Seventeenth International Symposium on Mathematical Foundations of Computer Science*, Springer-Verlag, 1992.

[31] E. Mayordomo. *Contributions to the Study of Resource-Bounded Measure*. PhD thesis, Universitat Politècnica de Catalunya, 1994.

[32] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge, 1995.

[33] K. Regan, D. Sivakumar, and J. Cai. *Pseudorandom Generators, Measure Theory, and Natural Proofs*. UBCS–TR 95-2, Computer Science Dept., University at Buffalo, 1995. To appear in FOCS 1995.

[34] C. P. Schnorr. Process complexity and effective random tests. *Journal of Computer and System Sciences*, 7:376–378, 1973.

[35] C. P. Schnorr. *Zufälligkeit und Wahrscheinlichkeit*. Volume 218 of *Springer Lecture Notes in Mathematics*, Springer-Verlag, 1971.

[36] A.L. Selman. P-selective sets, tally languages, and the behavior of polynomial-time reducibilities on NP. *Mathematical Systems Theory*, 13:55–65, 1979.

[37] A. Shamir. IP = PSPACE. *JACM*, 39:869–877, 1992.

[38] A. Kh. Shen´. On relations between different algorithmic definitions of randomness. *Soviet Math. Dokl.*, 38(2):316–319, 1989.

[39] R.M. Solovay. 1975. Reported in [6].

[40] V.A. Uspenskii, A.L. Semenov, and A. Kh. Shen´. Can an individual sequence of zeros and ones be random? *Russian Math Surveys*, 45:121–189, 1990.

[41] M. van Lambalgen. The axiomatization of randomness. *Journal of Symbolic Logic*, 55:1143–1167, 1990.

[42] M. van Lambalgen. *Random Sequences*. PhD thesis, University of Amsterdam, 1987.

[43] M. van Lambalgen. Von Mises' definition of random sequences reconsidered. *Journal of Symbolic Logic*, 52(3):725–755, 1987.

[44] R. von Mises. *Probability, Statistics and Truth*. George Allen & Unwin Ltd., 1957. Reprinted by Dover Publications, New York, NY in 1981. Originally published in German by J. Springer in 1928.

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

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

[47] D. Williams. *Probability With Martingales*. Cambridge University Press, 1991.