Probabilistic Martingales and BPTIME Classes

Kenneth W. Regan*
State Univ. of N.Y. at Buffalo

D. Sivakumar†
University of Houston

December 31, 1998

Abstract

We define *probabilistic martingales* based on randomized approximation schemes, and show that the resulting notion of *probabilistic measure* has several desirable robustness properties. Probabilistic martingales can simulate the "betting games" of [BMR+98], and can cover the same class that a "natural proof" diagonalizes against, as implicitly already shown in [RSC95]. The notion would become a full-fledged measure on bounded-error complexity classes such as BPP and BPE *if* it could be shown to satisfy the "measure conservation" axiom of [Lut92] for these classes. We give a sufficient condition in terms of simulation by "decisive" probabilistic martingales that implies not only measure conservation, but also a much tighter bounded error probabilistic time hierarchy than is currently known. In particular it implies $\mathrm{BPTIME}[O(n)] \neq \mathrm{BPP}$, which would stand in contrast to recent claims of an oracle A giving $\mathrm{BPTIME}^A[O(n)] = \mathrm{BPP}^A$. This paper also makes new contributions to the problem of defining measure on P and other sub-exponential classes. Probabilistic martingales are demonstrably stronger than deterministic martingales in the sub-exponential case.

*Supported in part by National Science Foundation Grant CCR-9409104. Address for correspondence: Computer Science Department, 226 Bell Hall, UB North Campus, Buffalo, NY 14260-2000. Email: `regan@cs.buffalo.edu`

†Research initiated while the author was at SUNY/Buffalo, supported in part by National Science Foundation Grant CCR-9409104. Email: `siva@cs.uh.edu`

1 Introduction

Lutz's theory of resource-bounded measure [Lut92] is commonly based on *martingales* defined on strings. A martingale can be understood as a gambling strategy for betting on a sequence of events—in this case, the events are membership and non-membership of strings in a certain language. If the strategy yields unbounded profit on a language A, then it is said to *cover* A, and the class of languages covered *has measure zero* in a corresponding appropriate sense.

It is natural to ask whether gambling strategies can be improved, in the sense of covering more languages, if the gambler is allowed to randomize his bets. Randomized playing strategies are crucial in areas such as game theory, and probabilistic computation is believed or known to be more powerful than deterministic computation in various computational settings. Is this so in the setting of resource-bounded measure? We define *probabilistic martingales* precisely in order to study this, basing them on the important prior notion of a *fully polynomial randomized approximation scheme* (FPRAS) [KL83, JVV86, JS89].

Probabilistic martingales have already appeared implicitly in recent work. For every "natural proof" Π (see [RR97]) of sufficient density, there is a probabilistic martingale d of equivalent complexity that covers the class of languages that Π is useful against [RSC95]. The "betting games" of Buhrman et al. [BMR$^+$98] can be simulated by probabilistic martingales of equivalent time complexity, as essentially shown by Section 5 of that paper. Hence in particular, probabilistic martingales of $2^{O(n)}$ time complexity (which equals polynomial time in Lutz's notation with $N = 2^n$ as the input length) can cover the class of languages that are polynomial-time Turing-complete for EXP. This class is not known to be covered by any $2^{n^{O(1)}}$-time deterministic martingale—and if so, then $\mathrm{BPP} \neq \mathrm{EXP}$ [AS94, BFT95]. The ultimate point of our work here, however, is that probabilistic martingales provide a new way to study general randomized computation.

For the study of sub-exponential time bounds, we lay groundwork by offering a new extension of Lutz's measure theory to classes below E. Our extension is based on "lex-limited betting games" as defined in [BMR$^+$98]. For measure on P it is weaker than some other notions that have been proposed in [AS94, AS95, Str97, CSS97], but it satisfies all of Lutz's measure axioms (except for some "fine print" about infinite unions), and carries over desirable results and properties of Lutz's theory from E and EXP down to classes below. It also still suffices for the main result of [AS94].

We prove that a class $\mathcal{C}$ is covered by a probabilistic martingale of time complexity $T(n)$ if and only if (unrelativized) $\mathcal{C}$ has $\mathrm{DTIME}^A[T(n)]$ measure zero relative to a random oracle A. Hence in particular the EXP-complete sets have E^A-measure zero (or p-measure zero relative to A in Lutz's terms), for a random A. Our theorem is roughly analogous to the theorem that BPP equals the class of languages that belong to P^A for a random A [BG81a, Amb86]. Hence we regard our notion as the best candidate for defining a measure on bounded-error classes such as BPP, BPE, and BPEXP. The latter two classes are defined from the general definition:

Definition 1.1. A language L belongs to $\mathrm{BPTIME}[t(n)]$ if there is a probabilistic Turing machine M running in time $t(n)$ such that for all inputs x, $\Pr[M(x) \neq L(x)] < 1/3$.

If t belongs to a family of time bounds that is closed under multiplication by n, then standard "repeated trials" amplification can be used to reduce the error probability $1/3$ below $1/2^n$ on inputs of length n. Hence the definitions of BPP, BPE, and BPEXP are robust, but whether the error probability can be reduced in the resulting definition of $\mathrm{BPTIME}[O(n^k)]$ for fixed k is pertinent to open problems in this paper.

The nub is whether our candidate for probabilistic measure has the crucial "measure conservation" property, i.e. whether one can show that BPP does *not* have "BPP–measure zero," or that BPE does not have probabilistic E-measure zero, and so on. We give a sufficient condition that is natural and reasonably plausible, namely that every probabilistic martingale d can be simulated by a probabilistic martingale d' (in the sense of d' succeeding on all languages that d succeeds on and having a randomized approximation scheme of similar time complexity) such that every change in the value of d' is not negligibly close to zero. For deterministic martingales this is an easy simulation. If this holds for probabilistic martingales, then measure conservation holds, and they really define a measure.

Since we show that $\mathrm{BPTIME}[O(n)]$ is covered by a BPP-martingale, and BPP by a probabilistic quasi-polynomial time martingale, and BPE by a BPEXP martingale (and so on), this would have the following consequence:

$$\begin{aligned} \mathrm{BPTIME}[O(n)] &\subset \mathrm{BPP} \subset \mathrm{BP}[qpoly] \subset \ldots \\ &\subset \mathrm{BPE} \subset \mathrm{BPEXP}. \end{aligned}$$

None of these successive inclusions is known to be proper. Indeed, Rettinger and Verbeek [RV97] claim to have an oracle A relative to which $\mathrm{BPTIME}^A[O(n)] = \mathrm{BPP}^A = \mathrm{BPTIME}^A[qpoly]$ (hence also $\mathrm{BPE}^A = \mathrm{BPEXP}^A$ by translation), fixing a flawed result of Fortnow and Sipser [FS89, FS97]. Thus our simulation problem ties in to the important open question of whether bounded-error probabilistic time enjoys a tight time hierarchy like that of deterministic time. We propose to turn this question on its head by analyzing the problem for probabilistic martingales, which is equivalent to a more-general problem about randomized approximation schemes treated in [CLL$^+$95]. For instance, it may be useful to seek and study oracles A relative to which some probabilistic martingale has no decisive simulation, independent of whether the oracle claim of [RV97] holds up.

2 Resource-Bounded Measure

A *martingale* is explicitly defined as a function d from $\{0,1\}^*$ into the nonnegative reals that satisfies the following "exact average law": for all $w \in \{0,1\}^*$,

$$d(w) = \frac{d(w0) + d(w1)}{2}. \tag{1}$$

The interpretation in Lutz's theory is that a string $w \in \{0,1\}^*$ stands for an initial segment of a language over an arbitrary alphabet Σ as follows: Let $s_1, s_2, s_3, \ldots$ be the standard lexicographic ordering of Σ^*. Then for any language $A \subseteq \Sigma^*$, write $w \sqsubseteq A$ if for all i, $1 \leq i \leq |w|$, $s_i \in A$ iff the ith bit of w is a 1. We also regard w as a function with *domain* $\{s_1, \ldots, s_{|w|}\}$ and range $\{0,1\}$, writing $w(s_i)$ for the ith bit of w. A martingale d *succeeds on* a language A if the sequence of values $d(w)$ for $w \sqsubseteq A$ is unbounded. Let $S^\infty[d]$ stand for the (possibly empty, often uncountable) class of languages on which d succeeds.

Definition 2.1 ([Lut92]). Let Δ be a complexity class of functions. A class $\mathcal{C}$ of languages *has Δ-measure zero*, written $\mu_\Delta(\mathcal{C}) = 0$, if there is a martingale d computable in Δ such that $\mathcal{C} \subseteq S^\infty[d]$. One also says that d *covers* $\mathcal{C}$.

Lutz defined complexity bounds in terms of the length of the argument w to d, which we denote by N. However, we also work in terms of the largest length n of a string in the domain of w. For $N > 0$, n equals $\lfloor \log N \rfloor$; all we care about is that $n = \Theta(\log N)$ and $N = 2^{\Theta(n)}$. Because complexity bounds on languages we want to analyze will naturally be stated in terms of n, we generally prefer to use n for martingale complexity bounds. The following correspondence is helpful:

$$\begin{array}{lcccccc} \text{Lutz's "}p\text{"} & \sim & N^{O(1)} & = & 2^{O(n)} & \sim & \mu_{\mathrm{E}} \\ \text{Lutz's "}p_2\text{"} & \sim & 2^{(\log N)^{O(1)}} & = & 2^{n^{O(1)}} & \sim & \mu_{\mathrm{EXP}} \end{array}$$

One effect of the change is that the function class Δ corresponding to a time-bounded complexity class $\mathcal{D}$ is clear—one simply uses the same time bound for functions. We carry forward Lutz's usage of saying that a class $\mathcal{C}$ has measure zero *in* a class $\mathcal{D}$ if $\mu_{\mathcal{D}}(\mathcal{C} \cap \mathcal{D}) = 0$, and *measure one in* $\mathcal{D}$ if $\mathcal{D} \setminus \mathcal{C}$ has measure zero in $\mathcal{D}$.

The desire to extend Lutz's theory to define measures on classes $\mathcal{D}$ below E runs into well-known technical problems. We will later propose a new definition that works well across the board for all families of sub-exponential time bounds that are closed under squaring, and is equivalent to Lutz's for bounds at E and above. However, we prefer to introduce probabilistic martingales in the specific case of E-measure and defer the more-general results to later sections.

3 Measure on E via Probabilistic Computations

Our first definition of measure via probabilistic computations has the following basic idea. Lutz's adaptation of classical measure theory to complexity classes is based on *real-valued martingale functions*. To turn this into a meaningful complexity-based notion, Lutz appeals to Turing's notion of computing a real number by arbitrarily close dyadic rational approximations (see [Ko83]). In a similar vein, we would like to define probabilistic measure via martingales that can be *efficiently computed probabilistically to arbitrary accuracy*. Among many possible notions of probabilistic computation of numerical functions, the following natural definition due to Karp and Luby [KL83] has risen to prominence.

Definition 3.1. A function $f : \Sigma^* \to \mathbf{Q}^{\geq 0}$ has a *fully polynomial-time randomized approximation scheme* (FPRAS) if there are a probabilistic Turing machine M and a polynomial p such that for all $x \in \Sigma^*$ and $\epsilon, \delta \in \mathbf{Q}^{>0}$,

$$\Pr[(1-\epsilon)f(x) \leq M(x,\epsilon,\delta) \leq (1+\epsilon)f(x)] \geq 1-\delta, \tag{2}$$

and $M(x,\epsilon,\delta)$ halts within $p(|x| + (1/\epsilon) + \log(1/\delta))$ steps.

(It is equivalent to remove δ and write "$1-\epsilon$" on the right-hand side of (2), but the above form is most helpful to us.)

Definition 3.2. A *probabilistic* E*-martingale* is a martingale that has an FPRAS.

This definition bounds the time to approximate martingale values $d(w)$ by a polynomial in $N = |w|$, which is the same as $poly(2^n) = 2^{O(n)}$. Our later generalization of an FPRAS will work nicely for all time bounds, stated in terms of n.

Now to obtain a notion of probabilistic E-measure, we simply carry over Definition 2.1.

Definition 3.3.
A class $\mathcal{C}$ of languages *has probabilistic* E*-measure zero*, written $\mu_{\mathrm{BPE}}(\mathcal{C}) = 0$, if there is a probabilistic E-martingale d such that $\mathcal{C} \subseteq S^\infty[d]$.

A viewpoint orthogonal to randomized approximation arises naturally from classical measure theory.

Definition 3.4. A "randomized martingale machine" M has access to an infinite sequence $\rho \in \{0,1\}^\omega$, which it uses as a source of random bits. Every fixed ρ defines a martingale d_M^ρ, and for any input w, M first computes $d_M^\rho(v)$ on the successive prefixes $v \sqsubset w$ before computing $d_M^\rho(w)$, drawing successive bits from ρ without repetition. M runs in E-time if this computation takes time $2^{O(n)} = poly(|w|)$.

Now we want to say that a class $\mathcal{C}$ has probabilistic measure zero if a "random" ρ makes d_M^ρ cover $\mathcal{C}$. To do this, we simply use the long-established definition of "random" via *classical Lebesgue measure* on the space $\{0,1\}^\omega$ of sequences ρ, which is much the same as the familiar Lebesgue measure on the real numbers in $[0,1]$.

Definition 3.5. A class $\mathcal{C}$ is "E-random-null" if there is an E-time randomized martingale machine M such that for all $A \in \mathcal{C}$, the set

$$\{\rho : A \in S^\infty[d_M^\rho]\}$$

has Lebesgue measure one.

For countable classes $\mathcal{C}$, it is equivalent to stipulate that the set of ρ such that $\mathcal{C} \subseteq S^\infty[d_M^\rho]$ has Lebesgue measure one. In the case of uncountable classes such as $\mathrm{P}/poly$, however, we do not know such an equivalence. Nevertheless, Definition 3.5 is equivalent to Definition 3.3, and to a third condition stated familiarly in terms of "random oracles," which differ from random sequences ρ in that bits can be used with repetitions. Note that $\mathcal{C}$ is unrelativized in the third case.

Theorem 3.1 *For any class $\mathcal{C}$ of languages, the following statements are equivalent.*

(a) There is a probabilistic E-martingale that covers $\mathcal{C}$.

(b) $\mathcal{C}$ is E-random-null.

(c) For a random oracle A, $\mathcal{C}$ has E^A-measure zero.

The neat thing about the equivalence is that Definition 3.3 allows us to treat a "probabilistic martingale" as a single entity rather than the ensemble of Definition 3.5, while Definition 3.5 does not involve an approximation parameter ϵ, and makes no reference to bounded-error computation at all! With this understood, we can point out that probabilistic E-martingales have already implicitly been used in the literature. The construction in Theorem 18 of [RSC95] shows that for every "natural proof/property" Π of sufficient density, there is a probabilistic martingale of equivalent complexity that covers the class of languages that Π is useful against. The constructions in Section 5 of [BMR$^+$98] show that every "E-betting game" can be simulated by a probabilistic E-martingale. These two constructions hold for EXP in place of E as well.

The proof of Theorem 3.1 is deferred until Section 6, where the generalization to other time bounds is stated and proved. The proof is more informative in the general setting. Its general workings are similar to the proofs by Bennett and Gill [BG81b] and Ambos-Spies [Amb86] that BPP equals the class of languages that belong to P^A for a random oracle A. These proofs extend to show that BPE equals the class of languages that belong to E^A for a random oracle A. The next section explores how well our unified definitions serve as a notion of measure "on" BPE.

4 Measure Properties and BP Time Hierarchies

The important axioms for a measure μ on a complexity class $\mathcal{D}$, as formulated by Lutz [Lut92] and summarized in [AS94], are:

M1 *Easy unions of null sets are null.* We skirt the difficult formal definition of an "easy" infinite union in [Lut92] and concentrate on the "finite unions" case: if $\mathcal{C}_1$ and $\mathcal{C}_2$ are subclasses of $\mathcal{D}$ with $\mu(\mathcal{C}_1) = 0$ and $\mu(\mathcal{C}_2) = 0$, then $\mu(\mathcal{C}_1 \cup \mathcal{C}_2) = 0$.

M2 *Singleton sets of easy languages are null.* For all languages $L \in \mathcal{D}$, $\mu(\{ L \}) = 0$.

M3 *The whole space is not null.* In other words, it is not the case that $\mu(\mathcal{D}) = 0$. This is called "measure conservation" in [Lut92]. Under the stipulation that for $\mathcal{C} \subseteq \mathcal{D}$, $\mu(\mathcal{C}) = 1 \iff \mu(\mathcal{D} \setminus \mathcal{C}) = 0$, this can be rewritten as $\mu(\mathcal{D}) = 1$.

All of the recent attempts to strengthen Lutz's measure framework to make more classes null have missed out on one of these axioms, most notably finite unions [AS95, BMR$^+$98], and this paper is no exception.

Why is it interesting to meet these axioms? A "pure" reason is that they abstract out the distinguishing characteristics of Lebesgue measure, and meeting them assures the integrity of a measure notion. They can also have direct connection to open problems in complexity theory, however. A recent example is that if the "betting-game measure" of [BMR$^+$98] has the finite-unions property **M1**, then the nonrelativizable consequence $\mathrm{BPP} \neq \mathrm{EXP}$ follows. A similar interest applies here: We show that probabilistic martingales satisfy **M1** and **M2** with $\mathcal{D} = \mathrm{BPE}$. If they also satisfy **M3**, then *tight* BPTIME *hierarchies follow*—in particular, $\mathrm{BPTIME}[O(n)] \subset \mathrm{BPP} \subset \mathrm{BPTIME}[qpoly(n)]$.

This consequence is interesting at this time because Rettinger and Verbeek [RV97] have recently claimed to construct an oracle A relative to which $\mathrm{BPTIME}^A[O(n)] = \mathrm{BPP}^A = \mathrm{BPTIME}^A[qpoly(n)]$, which would fix the main result of Fortnow and Sipser ([FS89], withdrawn in [FS97]). We have not yet been able to verify this claim, but the point is that bounded-error probabilistic time is not known to have a tight hierarchy like that for deterministic or nondeterministic or even unbounded-error probabilistic time. Karpinski and Verbeek [KV87] proved that BPP, and also $\mathrm{BPTIME}[qpoly(n)]$ and $\mathrm{BPTIME}[t(n)]$ for some bounds $t(n)$ slightly above quasi-polynomial, are properly contained in $\cap_{\epsilon>0}\mathrm{BPTIME}[2^{n^\epsilon}]$. This result and its translates are basically the best ones known. Thus interest in our work partly depends on how well the following notion provides a new angle on the problem of diagonalizing out of smaller BPTIME classes into larger ones.

Intuitively, a martingale d is *decisive* if it never makes a bet so small that its winning is insubstantial for the goal of succeeding on any language. In the present case of exponential time, suppose d on strings w of length N makes bets of magnitude less than $1/N^2$. Since $\Pi_{N\geq 1}(1 + 1/N^2) < \infty$, even if *all* such bets win along a language A, d still does not succeed on A. Hence we would consider any individual bet of this size to be insubstantial. Our formal definition actually relaxes the threshold from $1/N^2$ to $1/N^k$ for any fixed k.

Definition 4.1. A (probabilistic) E-martingale d is *decisive* if there exists $k \geq 1$ such that for all w, either $d(w1) - d(w) = 0$, or $|d(w1) - d(w)| \geq 1/|w|^k$.

Proposition 4.1 *(a) Probabilistic* E*-martingales satisfy* **M1** *and* **M2***.*

(b) Decisive probabilistic E*-martingales satisfy* **M2** *and* **M3***.*

Proof. (a) Given d_1 and d_2 covering $\mathcal{C}_1$ and $\mathcal{C}_2$, the function $d_3 = (d_1 + d_2)/2$ covers $\mathcal{C}_1 \cup \mathcal{C}_2$, and has a randomized approximation scheme of the same order of running time and precision as those for d_1 and d_2. (Note that d_3 is the same as flipping a coin to decide whether to bet according to d_1 or d_2 on a given string.) For infinite unions, we defer the proof until Section 7.

For **M2** in (a) and (b), given a fixed language $A \in \mathrm{BPE}$, use amplification to find a probabilistic $2^{O(n)}$-time TM M_A such that for all x of any length n, $\Pr[M_A(x) = A(x)] > 1 - 1/2^{n^2}$. Now let d be the trivial martingale that doubles its capital at every step along A and gets wiped out to zero everywhere else. Then d is decisive. To compute an FPRAS for $d(w)$, use M_A to test for all i whether $w_i = 1 \iff s_i \in A$. With high probability all the tests are correct, and so the value—either zero or $2^{|w|}$—is correct with the same probability.

(b) For **M3**, given a decisive probabilistic E-martingale d, define a sequence $\lambda = w_0 \sqsubset w_1 \sqsubset w_2 \ldots$ inductively by

$$\begin{aligned} w_{i+1} &= w_i 1 \ \text{ if } d(w_i 1) < d(w_i), \\ &= w_i 0 \ \text{ otherwise.} \end{aligned}$$

This infinite sequence defines a language A. Given k from Definition 4.1, we can use amplification to obtain an E-computable FPRAS M for d such that for all w, $\Pr[|M(w) - d(w)| < 1/N^{k+1}] > 1 - 1/2^{n^2}$. (Recall that $N = 2^n = |w|$.) Now define M_A to be a machine that on any input x first runs itself recursively on all inputs $y < x$. The recursive calls build up a string w whose domain is all the strings up to but not including x. Again with high probability, $w = w_{N-1}$. Finally, M_A computes $M(w1)$ and compares it to its already-computed value $M(w)$. If $M(w1) < M(w)$, M_A accepts x; else M_A rejects x.

The point is that owing to decisiveness, whenever $x \in A$, $|d(w1) - d(w)|$ is large enough that the approximating values $M(w)$ and $M(w1)$ will show $M(w1) < M(w)$ with high probability, so M_A will accept. Similarly whenever $x \notin A$, M_A will most likely not get $M(w1) < M(w)$, and with high probability will correctly reject x. Since $M_A(x)$ does little more than run the FPRAS M 2^n times, M_A runs in $2^{O(n)}$ time, and so $A \in \mathrm{BPE}$. $\square$

The construction in (a) fails to establish **M1** for decisive E-martingales. The problem is that d_1 and d_2 can be decisive, but a value $d_1(w)$ can be positive and $d_2(w)$ negative such that $d_1(w) - d_2(w)$ is close to but different from zero. We do not know of any other strategy that makes **M1** hold for decisive martingales, so that they would yield a fully-qualified notion of measure. Now do we know whether **M1** holding for them would have any larger-scale complexity consequences.

The "bigger game," of course, is whether every probabilistic E-martingale d can be simulated by a decisive one d', in the sense that $S^\infty[d] \subseteq S^\infty[d']$, from which **M3** would follow. For deterministic E-martingales this is a simple simulation: take $d' = (d+e)/2$, where e bets an amount $(1/2) \cdot (1/N^2)$ in the direction that takes the combined bet away from zero. The particular probabilistic martingales d that we care about are those that arise in the proof of the next theorem.

Theorem 4.2 *For all fixed* $c > 0$, $\mathrm{BPTIME}[2^{cn}]$ *can be covered by a probabilistic* E-*martingale.*

The tricky part of this, compared to the simple proof that DTIME$[2^{cn}]$ has measure zero in E, is that it may not be possible to obtain a recursive enumeration of "BPTIME[...] machines." However, probabilistic martingales (though maybe not decisive ones) can take up the slack of starting with a larger enumeration of *unbounded-error* probabilistic TMs, and arrange to succeed on those TMs that happen to have bounded error.

Proof. Take $P_1, P_2, \ldots$ to be a standard recursive enumeration of probabilistic Turing machines that run in time 2^{cn}. We define a "randomized martingale machine" M as follows. M divides its initial capital $C_0 = 1$ into infinitely many "shares" $s_k = 1/2k^2$ for $k \geq 0$ (with an unused portion $1 - \pi^2/12$ of C_0 left over). Each share s_k is assigned to the corresponding machine P_k, maintains its own capital, and (for a fixed random input ρ) computes a martingale that bets a nonzero amount only on strings x of the form $y10^k$. The martingale computed by M is well-defined by the sum of the shares.

To play share s_k on a string x, M uses its random bits to simulate P_k 2^{cn}-many times, treating acceptance as $+1$ and rejection as -1, and lets ν be the sample mean of the results. (Or M can apply the construction in [GZ97], which is defined for any probabilistic TM even though it only amplifies when P_k has bounded error.) M then bets a portion $\nu/2$ of the current capital of share s_k with the same sign as ν. Then M runs in time $O(2^{2cn})$.

For any P_k that has bounded error probability, a measure-one set of random sequences give:

- for all but finitely many $0^n \in L(P_k)$, $\nu > 1/2$, and
- for all but finitely many $0^n \notin L(P_k)$, $\nu < -1/2$.

For any sequence in this set, share s_k survives a possible finite sequence of losses and eventually grows to $+\infty$. Hence M succeeds on $L(P_k)$. □

By our equivalence theorem, Theorem 3.1, M defines a probabilistic martingale d that has an FPRAS. We may fail to obtain such a d that is decisive, however, for two main reasons. First and foremost, when P_k has unbounded error, the sample mean ν may be very close to zero. However, this does not prevent *an individual share* s_k from playing a decisive betting strategy s'_k: if $|\nu| < 1/3$ then bet zero. Other thresholds besides $1/3$ can be used, and can be varied for different x, or scaled toward zero as $1/|x|^2$, and so on. Since the shares play on different strings, the combination of the revised shares s'_k yields a function d' that is decisive (i.e., this is not the problem in **M1** for decisive martingales). The second rub, however, is that d' may no longer be fully randomly approximable. This is because a tiny difference in a reported ν may cross the threshold and cause a displacement in the value of d' that is larger than the scheme allows. Put another way, the random variable ν for a particular x may happen to be centered on the currently-used threshold, so that two widely-displaced values are output with roughly equal probability.

Seen in isolation, the problem of simulating a time-$t(n)$ probabilistic martingale d by a decisive d' is tantalizing. There seems to be slack for fiddling with thresholds to define d', or trying to take advantage of the fact that a martingale must make infinitely many large bets along any language that it succeeds on. Or one could try to make one of the parts of the proof of Theorem 6.2 below produce a decisive probabilistic martingale from the given one. However, this problem is tied to (and basically a re-casting of) the longer-studied problem of diagonalizing against bounded-error probabilistic machines:

Theorem 4.3 *If all probabilistic* E*-martingales can be simulated by decisive ones, then for all* $k > 0$*,* $\mathrm{BPTIME}[n^k] \neq \mathrm{BPP}$.

Proof. By Theorem 4.2, it immediately follows that for all $c > 0$, $\mathrm{BPTIME}[2^{cn}] \neq \mathrm{BPE}$. The conclusion then follows by familiar "translation" or "padding" techniques. □

Rather than rely on translation/padding results as in the proof of Theorem 4.3, however, we find it more informative to do the measure and diagonalization directly on BPP. The next section makes this possible, and independently contributes to the growing debate about the "proper" way to extend Lutz's theory to measure on sub-exponential time classes.

5 A New Take on Sub-Exponential Measure

The key idea is to focus on the "betting strategy" that a martingale represents. The strategy plays on an unseen language A, and tries to win money by "predicting" the membership or non-membership of successive strings x in A. Standardly, a martingale d corresponds to the strategy that starts by betting the amount $B_1 = d(1) - d(\lambda)$ "on" the assertion $\lambda \in A$, and given a string w that codes the membership of all strings $y < x$ in A, bets $B_x = d(w1) - d(w)$ on x. Here a negative B_x means that the bet wins if $x \notin A$. For measure at E and above, one can freely switch between the two views because the (upper bound on the) time to compute all of $d(\lambda), d(1), \ldots, d(w)$ has the same order as the time to compute $d(w)$ alone.

For sub-exponential time bounds, however, one has to choose one's view. Previous proposals for measures on classes below E [May94, AS94, AS95, Str97, CSS97] have worked directly with the martingales. We apply time bounds directly to the betting strategies, relaxing the condition that they must bet on *successive* strings, but maintaining that bets be in lexicographic order. The following is equivalent to the way a "lex-limited betting game" is defined in [BMR$^+$98]. (The general notion of betting games is obtained by replacing "lex-legal" by the simpler requirement that G can never bet twice on the same string.)

Definition 5.1. For any time bound $t(n)$, a *time-$t(n)$ martingale* is one computed by a machine G that executes one infinite computation as follows. G maintains a "capital tape" and a "bet tape," in addition to its other worktapes, and works in *stages* $i = 1, 2, 3\ldots$ Beginning each stage i, the capital tape holds a nonnegative rational number C_{i-1}. Initially $C_0 = 1$. G computes a query string x_i to bet on and a *bet amount* B_i, $-C_{i-1} \leq B_i \leq C_{i-1}$, where again a bet with negative B_i wins if x_i is not in the language being played on. If the bet wins, then the new capital C_i equals $C_{i-1} + |B_i|$, else it is $C_{i-1} - |B_i|$.

G is allowed to choose the next string x_{i+1} to bet on depending on the results of previous bets. The computation is *lex-legal* so long as the sequence of bet strings x_i is in ascending lexicographical order. G *runs in time* $t(n)$ if for all n, every bet on a string of length n is made within the first $t(n)$ steps.

The martingale computed by G is defined for all w by $d_G(w) =$ the capital after the finite sequence of bets that are resolved by w. (The "lex" limitation here makes this a martingale, unlike the corresponding situation in [BMR$^+$98].)

Now we simply carry over Definition 2.1 to the new definition of running time.

Definition 5.2. A class $\mathcal{C}$ has *time-$t(n)$ measure zero* if there is a time-$t(n)$ martingale d such that $\mathcal{C} \subseteq S^\infty[d]$.

For time bounds closed under multiplication by 2^n, this is equivalent to Lutz's definition. Our point is that for smaller time bounds $t(n)$ that meet the following definition, time-$t(n)$ martingales define a notion of measure that meets all the measure axioms.

Definition 5.3. A collection T of time bounds is "well-behaved" if it is closed under squaring, i.e. if $t \in T \implies t^2 \in T$, and if every $t \in T$ is fully time-constructible and at least linear.

Examples are polynomial time, *quasi*-polynomial time (i.e., $T = \{$functions $2^{c(\log n)^d}$ for $c, d \in \mathbf{Q}^+\}$), linear exponential time (functions 2^{cn} for $c \in \mathbf{Q}+$), and poly-exp. time (functions 2^{cn^d} for $c, d \in \mathbf{Q}^+$), when the corresponding DTIME$[T]$ classes are P, DTIME$[qpoly]$, E, and EXP. The corresponding bounded 2-sided error probabilistic complexity classes are here called BPP, $\mathrm{BPTIME}[qpoly]$, BPE, and BPEXP.

Proposition 5.1 *For any well-behaved collection T of time bounds, time-$t(n)$ martingales for $t \in T$ define a measure on* DTIME$[T]$ *that meets measure axioms* **M1**–**M3**.

Proof. For the finite-union version of **M1**, suppose we are given G_1 and G_2 from Definition 5.1. Define G_3 to divide its initial capital into two equal "shares" s_1 and s_2, which follow the respective betting strategies used by G_1 and G_2. In contrast to the situation for general betting games in [BMR+98], where closure under finite unions implies $\mathrm{BPP} \neq \mathrm{EXP}$, the point is that owing to the lex-order stipulation, G_3 can play the two shares side-by-side with no conflicts. Whichever share's choice of next string to bet on is the lesser gets to play next; if they both bet on a string x, then G_3's bet is the algebraic sum of the two bets.

We postpone the definition and treatment of the infinite-unions case until the end of this section.

For **M2**, we actually show that for any time-$t(n)$ martingale d, there is a *time-$t(n)$ printable* language A that is not covered by d. Let M start simulating the G for d, and define $x \in A$ iff G makes a negative bet on x. Then M can print out all the strings in A of length up to n within its first $t(n)$ steps.

For **M3**, we show the stronger result that for any $t \in T$, DTIME$[t(n)]$ has time-T measure zero. Take $P_1, P_2, P_3, \ldots$ to be a recursive presentation of time-$t(n)$ Turing machines so that the language $\{\,(i, x) : x \in L(P_i)\,\}$ can be recognized by a machine M that runs in time, say, $t(n)^2$. Let G divide its initial capital into infinitely many "shares" s_i, where s_i has initial capital $1/2i^2$. To assure meeting the time bound, G bets only on tally strings $x = 0^j$ as follows: Let i be maximum such that 2^i divides j. Then bet all of the current value of share s_i positively if $x \in L(P_i)$, negatively otherwise. For all $A \in$ DTIME$[t(n)]$, the share s_i corresponding to any P_i that accepts A doubles its capital infinitely often, making G succeed on A regardless of how the other shares do. Then G runs in time $O(nt(n)^2)$, which belongs to T. $\square$

Definition 5.1 defines a single infinite process that begins with empty input. We want an equivalent definition in terms of the more-familiar kind of machine that has finite computations on given inputs. We use the same model of "random input access" Turing machines M used by others to define measures on classes below E [May94, AS94, AS95], but describe it a little differently: Let M have an "input-query" tape on which it can write a string s_i and receive the bit w_i of w that indexes s_i. Initially we place on M's actual input tape the lexically last string x that is indexed by w (if $|w| = N$, then $x = s_N$), and write $M(w : x)$ or $M(w : N)$ to represent this initial configuration.

It is now believed that the class of martingales computed by machines of this kind running in $poly(n)$ time or space is too big to define a good notion of measure on P or PSPACE, respectively. Up to now, the main solution has been to impose some "dependency set" restriction on the queries made by M. Given M, define a directed graph Γ_M by making (s_j, x) an edge if there exists some w of length n such that $M(w : x)$ queries bit j of w. Necessarily $s_j < x$. The condition used by Allender and Strauss to define "conservative P-measure" [AS95] (and used in [AS94]) is that for every x, the set of y such that there is a path from y to x in Γ_M has $poly(|x|)$ size and can be output in polynomial time. We define a condition that is incomparable with theirs, and seems to yield an incomparable notion of measure.

Definition 5.4. An input-query machine M "runs lex-nicely in time $t(n)$" if for all w, x, and $m \leq |x|$, the computation $M(w : x)$ makes its input queries in lex order, and queries any s_j of length m within the first $t(m)$ steps of the computation.

The intent of the definition is also clear if we add some $O(n)$-sized "extras" to the input tape, such as an index $i < n$ or some error parameters ϵ, δ.

An example where this condition is more liberal is an M that first queries the strings 0^i for $0 \leq i < n$, reads the results as a binary string x of length n, queries x, and then queries 0^{n+1}. Then 0^{n+1} has exponentially-many predecessors in Γ_M, but M still runs lex-nicely in quadratic time. A poly-time machine of the Allender-Strauss kind can, however, query all strings of length $O(\log n)$, which our machines cannot do.

The technical nub now is whether the extra information that writing x, or equivalently N, on the input tape of M imparts is enough to compute more martingales than in Definition 5.1. The nice robustness property is that the answer is *no*. Our proof is similar in spirit to robustness proofs in [AS95], and we would like to know whether it can be simplified or cast as a simple patch to one of their proofs.

Lemma 5.2 *Let T be a well-behaved collection of time bounds. Then d is a time-$t(n)$ martingale for some $t \in T$ if and only if d is computed by an input-query machine M that runs lex-nicely in some time $t'(n)$ with $t' \in T$.*

Proof. The forward direction is immediate: given G, $M(w : x)$ just simulates G up through all stages that bet on strings indexed by w and outputs the final capital. Since $t(|x|)$ is time-constructible, M knows to shut itself off if G dithers for $t(n)$ steps without betting on any more strings of length n. The point of the converse is that the extra information given to $M(w : x)$ in the form of x (compared to how betting games receive empty input) does *not* help M compute any more martingales. The effect is similar to the robustness results for the "conservative" measure-on-P notions of [AS95]. The main point is the following claim:

Claim 5.3 *Suppose v is a proper prefix of w such that $d(v1) \neq d(v)$. Then $M(w : x)$ must query the string y_v indexed by the '1' in $v1$.*

To prove this, suppose not. Take $W = \{ w' : |w'| = |w| \text{ and } v \sqsubseteq w' \}$, $W_1 = \{ w' \in W : v1 \sqsubseteq w' \}$, and $W_0 = W \setminus W_1$. Thanks to the lex-order restriction, *none* of the $w' \in W$ cause $M(w' : x)$ to query y_v—they all make the same queries lexically less than y_v, and then either all halt with no further queries, or all make the same query higher than y_v and can never query y_v from that point on. Now for all $w_1 \in W_1$, there is a corresponding $w_0 \in W_0$ that differs only in the bit indexing y_v. It follows that $M(w_1 : x)$ and $M(w_0 : x)$ have the same computation. But then the average of $M(w_1 : x)$ over $w_1 \in W_1$ must equal the average of $M(w_0 : x)$ over $w_0 \in W_0$, which contradicts $d(v0) \neq d(v1)$ since d is a martingale. This proves the claim.

It follows that if v is a shortest initial segment such that $d(v1) \neq d(v)$, then for *every* w with $|w| > |v|$, the computation $M(w : |w|)$ queries y_v, after perhaps querying some strings lexically before y_v. Then for a shortest v_0 extending $v0$ with $d(v_0 1) \neq d(v_0)$, and all w with $|w| > |v_0|$ and $v0 \sqsubseteq w$, the computation $M(w : |w|)$ must query y_{v_0}, and so on…and similarly for v_1 extending $v1$…

Now the lex-limited betting game G simulating M takes shape: For all n, the phase of the computation in which G bets on some (or no) strings of length n begins by simulating $M(w : 1^{t(n)})$. Whenever $M(w : 1^{t(n)})$ queries a string x of length n, call x a "top-level query." G then simulates $M(w : x)$ in order to learn what to bet on x. G maintains a table of all queries made by M and their results. If a string $y < x$ queried by $M(w : x)$ is a top-level query, by the lex-order restriction applied to $M(w : 1^{t(n)})$, y will already be in the table. If y is not in the table, then G does *not* query y—indeed, it might violate the runtime-proviso to query y if y is short. Instead, G proceeds as though the answer to y is "no." Finally, if $M(w : x)$ queries x, G simulates both the "yes" and "no" branches and takes half the signed difference as its bet. (If $M(w : x)$ does not query x, then G bets zero on x.) By Claim 5.3, the average of the two branches is the same as the value of $M(w : x')$ for the top-level query x' lexically preceding x (or the average is $d(\lambda) = 1$ in case x is the first query). By induction (see next paragraph), this equals the capital G has at this stage, so G has sufficient capital to make the bet. Then G queries x. When $t(n)$ steps of the simulation of $M(w : 1^{t(n)})$ have gone by, or when $M(w : 1^{t(n)})$ wishes to query a string of length $> n$, G abruptly skips to the next stage of simulating $M(w : 1^{t(n+1)})$ and handling any queries of length $n + 1$ that it makes.

For correctness, it suffices to argue that the values G computes while simulating $M(w : x)$ always equal $M(w : x)$, despite the possibly-different answers to non-top-level queries y. Suppose v and v' be two initial segments up to x that agree on all top-level queries, such that $d(v) \neq d(v')$. Let $W = \{ w : v \sqsubseteq w \ \wedge \ x_w = 1^{t(n)} \}$ and $W' = \{ w' : v' \sqsubseteq w' \ \wedge \ x_{w'} = 1^{t(n)} \}$. Then for every $w \in W$ there is a $w' \in W'$, obtained by altering the bits indexing non-top-level queries to make v' a prefix, on which $M(w' : 1^{t(n)})$ has the same computation as $M(w : 1^{t(n)})$. Hence the average of $M(w : 1^{t(n)})$ over $w \in W$ equals that of $M(w' : 1^{t(n)})$ over $w' \in W'$, but the former equals $d(v)$ and the latter equals $d(v')$ since d is a martingale, a contradiction. So G always gets the correct values of $d(w)$ as it bets.

Finally, the running time of this process up through queries of length n is bounded by $\sum_{m=1}^{n} t(m) \leq nt(n)$. One technicality needs to be mentioned, however: G needs to maintain a dynamic table to record and look up queries. On a fixed-wordsize RAM model a string x can be added or looked up in time $O(|x|)$, but it is not known how to do this on a Turing machine. However, we can appeal to the fact that a Turing machine can simulate t steps of a fixed-wordsize RAM in time $O(t^2)$. Hence the final runtime is at most $(nt(n))^2$, which by the closure under squaring in "well-behaved" is a bound in T. $\square$

Now we can *define* the infinite-unions case in a way that carries over Lutz's intent. Say that a sequence $d_1, d_2, d_3, \ldots$ of martingales is *time-$t(n)$-presented* if there is an input-query machine M that given $N\#i$ on its input tape computes $d_i(w)$ (for all w with $|w| = N$) lex-nicely in time $t(n)$. A time-$t(n)$ infinite union of measure-zero classes is then defined by a sequence $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3 \ldots$ of classes for which there is a time-$t(n)$ sequence of martingales $d_1, d_2, d_3, \ldots$ with $\mathcal{C}_i \subseteq S^\infty[d_i]$ for each i. The niggling extra condition we seem to need restricts attention to complexity classes $\mathcal{C}$ that are closed under finite variations, meaning that whenever $A \in \mathcal{C}$ and $A \triangle B$ is finite, also $B \in \mathcal{C}$.

Proposition 5.4 *Let T be a well-behaved family of time bounds. If $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3 \ldots$ is a time-$t(n)$ infinite union of measure-zero classes, and each $\mathcal{C}_i$ is closed under finite variations, then $\cup_i \mathcal{C}_i$ has time-$t(n)$ measure zero.*

Proof. We build an M' that divides its initial capital into infinitely many shares s_i, each with initial capital $1/2i^2$. (The portion $1-\pi^2/12$ of the capital left over is ignored.) Take M to be the time-$t(n)$ input-query machine computing the time-$t(n)$ sequence of martingales $d_1, d_2, d_3, \ldots$ from the above definitions. Then share s_i will *try to* simulate computations $M(w : x\#i)$.

Given w and x, where $|x| = n$, $M'(w : x)$ loops over m from 1 to n. At each stage m it allots $t(m)$ steps to each of the computations $M(w : x\#1), \ldots, M(w : x\#m)$. *The rub* is that the last of these, namely $M(w : x\#m)$, was not included in the previous iteration $m-1$ and before, and conceivably may want to query strings of length $< m$ that M' has already passed over. The fix is that we may begin the simulation of $M(w : x\#m)$ by answering "no" for each such query, without submitting the query.

For queries made by these m computations on strings of length m itself, M' works in the same parallel fashion as in the proof for finite unions in Proposition 5.1. Namely, whichever of the computations wants to query the lexicographically least string is the one that receives attention. This polling takes $O(m^2)$ extra time per step on strings of length m. If two or more wish to bet on the same string, then M' submits the algebraic sum of the bets. The running time of iteration m of the for-loop is thus $O(m^3 t(m))$, and summing this tells us that M' runs lex-nicely in time $n^4 t(n)$, which bound belongs to T.

For any language $A \in \cup_i \mathcal{C}_i$, there exists an i such that not only $A \in \mathcal{C}_i$, but also all finite variations of A belong to $\mathcal{C}_i$, and hence are covered by d_i. In particular, the finite variation A' that deletes all strings of length less than i is covered by d_i. Then share s_i imitates the simulation by M of d_i playing on A', and hence sees its capital grow to infinity. $\square$

Whether we can claim that our measure satisfies Lutz's infinite-unions axiom, and hence **all** the measure axioms, is left in a strangely indeterminate state. All complexity classes of interest are closed under finite variations, and we've shown that the infinite-unions axiom holds for them. If we could show that $\mathcal{C}$ null $\Longrightarrow$ the closure $\mathcal{C}^f$ of $\mathcal{C}$ under finite variations is null, then the construction would probably be uniform enough for time bounds in T to plug in to the above proof and remove the condition on the $\mathcal{C}_i$. But as it stands, this last is an open problem, and a niggling chink in what is otherwise a healthily robust notion of measure.

For measure on P in particular, our notion lives at the weak end insofar as the P-printable sets, and hence the sparse sets, do not have P-measure zero. However, it is strong enough for the construction in the main result of [AS94]. For any fixed $\epsilon > 0$, let E_ϵ stand for the collection of time bounds 2^{n^δ} for $\delta < \epsilon$. This is closed under multiplication and hence well-behaved.

Theorem 5.5 (after [AS94]) *For every $\epsilon > 0$, the class of languages $A \in \mathrm{E}_\epsilon$ such that $\mathrm{BPP} \not\subseteq \mathrm{P}^A$ has E_ϵ-measure zero (in our terms).*

Proof Sketch. The key detail in the proof in [AS94] is that the dependency sets for computations $M(w : x)$ have the form $\{ 0^{2^{b|y|}} y : |y| \leq (\log n)/b \}$, where b is a constant that depends on ϵ and $n = |x|$. These sets are computable in polynomial time, and more important, are sparse enough that every string of length m in the set can be queried in $poly(m)$ time, for all $m < n$. Hence we can construct an E_ϵ-martingale to simulate the martingale in that proof. $\square$

6 Probabilistic Sub-Exponential Measure

We first wish to generalize the notion of an FPRAS to general time bounds $t(n)$. The indicated way to do this might seem to be simply replacing "p" by "t" in Definition 3.1. However, we argue that the following is the *correct* conceptual generalization.

Definition 6.1. Let T denote an arbitrary collection of time bounds. A function $f : \Sigma^* \to \mathbf{Q}^{\geq 0}$ has a *fully poly-T randomized approximation scheme* (T-FPRAS) if there are a probabilistic Turing machine M, a bound $t \in T$, and a polynomial p such that for all $x \in \Sigma^*$ and $\epsilon, \delta \in \mathbf{Q}^{>0}$,

$$\Pr[(1-\epsilon)f(x) \leq M(x, \epsilon, \delta) \leq (1+\epsilon)f(x)] \geq 1 - \delta,$$

and $M(x, \epsilon, \delta)$ halts within $p(t(|x|) + (1/\epsilon) + \log(1/\delta))$ steps.

That is, we have replaced "$|x|$" in Definition 3.1 by "$t(|x|)$." If T is closed under squaring (i.e., well-behaved), then $p(t(|x|))$ is a time bound t' in T, and the time bound in Definition 6.1 could essentially be rewritten as $t'(|x|) \cdot p((1/\epsilon) + \log(1/\delta))$. The point is that the time to achieve a given target accuracy ϵ or error δ remains polynomial (for any T) and is not coupled with the running time $t(n)$, which figuratively represents the time for an individual sample. The application in this section is entirely general and typical of the way an FPRAS is constructed and used. Hence we assert that the result supports our choice of generalization.

Now we would like to say simply that a probabilistic time-T martingale is a martingale that has a fully poly-T randomized approximation scheme. However, recall the discussion in the last section before Definition 5.4 that the unrestricted definition of a deterministic time-$t(n)$ martingale is considered too broad. We need to work the "lex-nicely" condition into the manner of computing the approximation, and so define:

Definition 6.2. A *probabilistic time-T martingale* is a martingale that has a fully poly-T randomized approximation scheme computed by a machine M that runs lex-nicely in some time $t \in T$.

Expanded, the lex-nicely proviso in this case says that in the randomized computation of $M(w : N, \epsilon, \delta)$, accesses to w must be in ascending (i.e., left-to-right) order, and any request s_j for bit j of w must be made within the first $t(|s_j|)$ steps. Here we could weaken $t(|s_j|)$ to $p(t(|s_j|), \epsilon, \log(1/\delta))$ without affecting any of our results. This proviso may seem artificial, but it makes M play by the same rules used to define sub-2^n time martingales in the first place. Anyway, for time bounds T at E *and above*, this technicality can be ignored, and $M(w, \epsilon, \delta)$ can be any T-FPRAS for $d(w)$.

The other main definition in Section 3 carries over without a hitch. We simply give lex-limited betting games G access to a source ρ of random bits, calling the resulting machine G_ρ.

Definition 6.3. A class $\mathcal{C}$ is "time-T-random-null" if there is a time-T randomized martingale machine G such that for all $A \in \mathcal{C}$, the set

$$\{ \rho : A \in S^\infty[d_{G_\rho}] \}$$

has Lebesgue measure one.

Before we prove that this definition yields the same "null" classes as the previous one, we give a noteworthy motivation. A sequence $[\epsilon_n]$ is "polynomially non-negligible" if there exists $c > 0$ such that for all but finitely many n, $\epsilon_n > 1/n^c$.

Theorem 6.1 *For any polynomially non-negligible sequence $[\epsilon_n]$, the class of languages of density at most $1/2 - \epsilon_n$ is polynomial-time random null, but it—nor the subclass of languages of density at most ϵ_n—does not have measure zero in any sub-exponential time bound.*

Proof Sketch. Let L be any language of density at most $1/2 - 1/n^c$. Then (for all large enough n) the probability that a random string of length n belongs to L is at most $1/2 - 1/n^c$. By sampling $O(n^{2c})$ strings, we create a process in which with probability $> 1 - 1/n^c$ there is an excess of at least n^c strings that are not in L. A martingale that bets conservatively on strings not being in L can more than double its value whwnever that event occurs. Since the product of $1 - 1/n^c$ converges for $c > 1$, and this analysis holds for any such L, the class of such L is polynomial-time random null.

However, every deterministic time-$t(n)$ martingale fails to cover some time-$O(t(n))$ printable language, so when $t(n) = o(2^n)$, not all languages of that density can be covered. When t belongs to a well-behaved family T that does not include 2^n, it follows that for all $c > 0$, some language of density $1/n^c$ is not covered. $\square$

Theorem 6.2 *Let $\mathcal{C}$ be a class of languages, and let T denote a well-behaved collection of time bounds. Then the following are equivalent:*

(a) There is a probabilistic time-T martingale that covers $\mathcal{C}$.

(b) $\mathcal{C}$ is "time-T-random-null."

(c) For a random oracle A, $\mathcal{C}$ has DTIME$[T]^A$*-measure zero.*

Proof. (a) $\Longrightarrow$ (b): Let d be a probabilistic time-T martingale that covers $\mathcal{C}$, and let M compute a T-FPRAS for d. Then there are $t \in T$ and a polynomial p such that M runs lex-nicely in time $p(t(\ldots), \ldots)$. Here we may suppose that $t(n) \geq 2n$ and that $t(n)$ is fully time constructible. We describe a probabilistic lex-limited betting game G with an auxiliary sequence $\rho \in \{0,1\}^\omega$ that works as follows. G_ρ carries out the simulation of "M" in Lemma 5.3, using ρ to supply random bits requested by M. Whenever M makes a (top-level) query x of length n, G takes $\epsilon_n = 1/Kt(n)^3$ and $\delta_n = 1/4t(n)^3$, where the quantity K is described below, and simulates $M(w : x, \epsilon_n, \delta_n)$. Note that G has time to write ϵ_n and δ_n down before betting on a string of length n. (Also note that $p(t(n), \epsilon_n, \log(1/\delta_n))$ is likewise a polynomial in $t(n)$, which is why the change remarked after Definition 6.2 does not affect this result. Moreover, we can choose δ_n as low as $2^{-poly(t(n))}$ rather than essentially taking $\delta_n = \epsilon_n$; this slack is not surprising given the remark on slack in defining an FPRAS after Definition 3.1.)

Now suppose that G_ρ has current capital C just before querying x. Let w index the strings up to but not including x. Let C_1 be the result of simulating $M(w1 : x, \epsilon_n, \delta_n)$, and C_0 the result of simulating $M(w0 : x, \epsilon_n, \delta_n)$. If we had complete confidence in the estimates C_0 and C_1 for the values $d(w0)$, and $d(w1)$, respectively, then we would bet $B_x = C\frac{C_1 - C_0}{C_1 + C_0}$ on the event x is "in." However, since the estimate may err by a factor of $(1 \pm \epsilon_n)$ even when M's approximation is successful, we make G_ρ play a little conservatively. Specifically, we will make G suppose that C_1 underestimates $d(w1)$, and also that C_0 *underestimates* $d(w0)$. Imitating the equations in the proof of Theorem 18 in [RSC95], we define:

$$B_x = C\frac{C_1 - C_0}{(C_1 + \epsilon_n C) + (C_0 + \epsilon_n C)}. \tag{3}$$

Making B_x smaller in absolute value in this way also has the effect of preventing the current capital from going to zero even after a finite sequence of bad bets resulting from incorrect estimates by M. This scaling-down works even if C itself falls below ϵ_n.

Claim 6.3 *Let A be a language on which d succeeds. If ρ is such that, starting with the current bet x, all estimates by M are within the prescribed FPRAS bounds, then G_ρ playing on A grows its capital C to infinity, regardless of how small C is at the current stage.*

We could appeal to the proof of Theorem 18 in [RSC95], but here we sketch a different way to do the argument. We first show that the *proportion* of C that is bet (i.e., the fractional part of B_x) is close to the ideal proportion given by the values $d(w)$, $d(w1)$, and $d(w0)$. Without loss of generality suppose $C_1 \geq C_0$ so that $B_x \geq 0$—the case $C_0 \leq C_1$ is handled symmetrically. Suppose first that B_x is a *losing* bet, i.e. $x \notin A$. We want to prove that $(1 - \frac{B_x}{C}) \geq \frac{d(w0)}{d(w)}(1 - K\epsilon)$ (writing ϵ for ϵ_n). Now

$$(1 - \frac{B_x}{C}) = 1 - \frac{C_1 - C_0}{C_1 + C_0 + 2C\epsilon} = \frac{2C\epsilon + 2C_0}{2C\epsilon + C_1 + C_0}.$$

This is least when $C_0 = d(w0)(1-\epsilon)$ and $C_1 = d(w1)(1+\epsilon)$ within the bounds allowed by the FPRAS. Then we obtain

$$\begin{array}{rcl} C_1 & = & (2d(w) - d(w0))(1+\epsilon) \\ C_1 + C_0 & = & 2d(w) + 2\epsilon d(w) - 2\epsilon d(w0), \text{ and} \\ C_1 \geq C_0 & \Longrightarrow & d(w) + \epsilon d(w) \geq d(w0). \end{array}$$

Hence

$$\begin{array}{rcl} (1 - \dfrac{B_x}{C}) & = & \dfrac{d(w0) + \epsilon(C - d(w0))}{d(w)(1+\epsilon) + \epsilon(C - d(w0))} \\ & = & \dfrac{d(w0)(1-\epsilon) + \epsilon C}{d(w)(1+\epsilon) - \epsilon d(w0) + \epsilon C} \\ & \geq & \dfrac{d(w0)(1-\epsilon) + \epsilon C}{d(w)(1+\epsilon) - \epsilon(d(w) + \epsilon d(w)) + \epsilon C} \\ & = & \dfrac{d(w0)(1-\epsilon) + \epsilon C}{d(w)(1-\epsilon^2) + \epsilon C} \\ & \geq & \dfrac{d(w0)}{d(w)(1+\epsilon)} \geq \dfrac{d(w)}{d(w0)}(1-\epsilon). \end{array}$$

The last line follows from the identity $b, x \geq 0 \wedge b \geq a \Longrightarrow \frac{a+x}{b+x} \geq \frac{a}{b}$, and $d(w0)(1-\epsilon) = a \geq b = d(w)(1-\epsilon^2$ because $d(w)(1+\epsilon) \geq d(w0)$. That finishes this case, with $K = 1$.

The case of a winning bet is not quite symmetrical. We want to show $(1 + \frac{B_x}{C}) \geq \frac{d(w1)}{d(w)}(1 - K\epsilon)$. We have

$$(1 + \frac{B_x}{C}) = 1 + \frac{C_1 - C_0}{C_1 + C_0 + 2C\epsilon} = \frac{2C\epsilon + 2C_1}{2C\epsilon + C_1 + C_0}.$$

This is least when $C_1 = d(w1)(1-\epsilon)$ and $C_0 = d(w0)(1+\epsilon)$ within the bounds allowed by the FPRAS. Then we obtain

$$\begin{aligned} C_0 &= (2d(w) - d(w1))(1+\epsilon) \\ C_1 + C_0 &= 2d(w) + 2\epsilon d(w) - 2\epsilon d(w1), \text{ and} \\ C_1 \geq C_0 &\Longrightarrow d(w1) \geq d(w)(1+\epsilon). \end{aligned}$$

The last line is the part that isn't symmetrical. Now we get:

$$(1 + \frac{B_x}{C}) = \frac{d(w1) + \epsilon(C - d(w1))}{d(w)(1+\epsilon) + \epsilon(C - d(w1))}$$

Now we want to use the identity

$$(x < 0 \ \wedge \ (b+x) > 0 \ \wedge \ a \geq b) \Longrightarrow \frac{a+x}{b+x} \geq \frac{a}{b}.$$

If $C < d(w1)$, this is satisfied with $x = \epsilon(C - d(w1))$, $a = d(w1)$, and $b = d(w)(1+\epsilon)$. This is because $C_1 \geq C_0 \Longrightarrow a \geq b$, and $b + x > d(w) + \epsilon d(w) - \epsilon d(w1) \geq d(w)(1-\epsilon)$ since $C > 0$ and $d(w1) \leq 2d(w)$. Thus we get

$$(1 + \frac{B_x}{C}) \geq \frac{d(w1)}{d(w)(1+\epsilon)} \geq \frac{d(w1)}{d(w)}(1-\epsilon)$$

and we're home, again with $K = 1$. Now what if $C \geq d(w1)$? In this analysis, that implies $C \geq d(w)(1+\epsilon)$. We can wave this case away by reasoning that if the current capital is already doing that much better than the "true value" $d(w)$ then there is nothing to prove. Or we can make G always keep some of its capital in reserve, so that C stays less than $d(w)$ unless the FPRAS estimates are violated on the high side. Finally, we could also change the "$+2\epsilon C$" in the denominator of (3) to something else, at the cost of making K higher.

(*Remark.* One interesting thing about this argument is that the inequalities resulting from the worst-case choices of C_1 and C_0, namely $d(w1) \geq d(w)(1+\epsilon)$ for a winning bet and $d(w0) \leq d(w)(1+\epsilon)$ for a losing bet, hold automatically if d is a decisive probabilistic martingale, as defined below in Definition 7.1. Here the latter inequality is trivial, but in the symmetrical case $C_0 \leq C_1$ the losing-bet case has the nontrivial inequality.)

The basic point in any event is that for any fixed K, we can make $\prod_n (1 - t(n)K\epsilon_n)$ stay bounded below by a constant, by choosing $\epsilon_n = 1/Kt(n)^3$ (note also $t(n) \geq 2n$). The leading $t(n)$ comes in because M can make up to $t(n)$ queries of length n, and $(1 - K\epsilon_n)^{t(n)} \geq (1 - t(n)K\epsilon_n)$. Hence as $d(w) \rightarrow \infty$ for $w \sqsubseteq A$, $C \rightarrow \infty$ in a constant proportion to $d(w)$. This proves the claim.

Next we observe that with the choice of δ_n, the probability that *all* estimates by M are within the FPRAS bound, which is lower-bounded by $\prod_{n\geq 1}(1-2t(n)\delta_n)$, is bounded away from zero by a fixed constant. Thus the set of ρ such that G_ρ succeeds on A has nonzero measure.

Now we finish the argument by claiming that G^ρ succeeding on A is a *tail event*, i.e., independent of any changes to a finite initial segment of ρ. Because of the way (3) is defined as a proportion of the current capital C of G, and because the "conservative" adjustment preserves a chunk $\epsilon_n C$ of the current capital C even in cases where $d(w1)$ actually equals zero and $x \notin A$, the strategy can recover from a finite initial sequence of inaccurate bets. The recovery can happen within the prescribed time bound because even if all $t(n)$ bets at length n get wiped out down to the $\epsilon_n C$ chunk, the resulting capital $C \cdot \epsilon_n^{t(n)} = (C/K) \cdot (1/t(n)^{3t(n)})$ can still be written down in $3t(n)\log t(n)$ time. All told, G runs in time at most $p((t(n))^3)$, which by well-behavedness is still a time bound in T.

(b) $\iff$ (c): The infinite sequence ρ of independent random bits given to G can be simulated by an oracle TM G' that never repeats a query. For the converse, there is the problem that G'^R may succeed on A for a random oracle R but by dint of repeating oracle queries. However, a betting game G with random coin-flips has time to maintain a dynamic table of all the oracle queries (note—these are separate from queries to w) made by G', and in case of a repeat, answer from the table rather than flip a fresh coin. Then the behavior of G over ρ is identically distributed to that of G' over oracle languages.

(c) $\Longrightarrow$ (a): Let G be an betting machine that takes an auxiliary sequence ρ and runs in time $O(t(n))$, such that for every $L \in \mathcal{C}$, the set of sequences ρ such that G^ρ succeeds on L has Lebesgue measure 1. By the "slow-but-sure winnings" lemma of [BMR+98] (or similar lemmas in [May94, BL96]), we may assume that the capital G has before any string of length $n+1$ is queried is at most $O(t(n))$, irrespective of ρ or the language A that G is betting on. (This is done by restricting G to never bet more than one unit of its capital.)

For every n, the computation of G^ρ along A can use at most $t(n)$ bits of ρ before all queried strings of length n are queried. Thus a finite initial segment $\sigma \sqsubseteq \rho$ of length $t(n)$ suffices to determine the capital that G^ρ has at any point prior to querying a string of length $n+1$. Now define, for any w of length $N \approx 2^n$, $d(w)$ to be the average, over all sequences σ of length $t(n)$, of the capital that G^σ has after it has queried all strings (that it queries) whose membership is specified by w. It is immediate that d is a martingale.

Claim 6.4 $\mathcal{C} \subseteq S^\infty[d]$.

Proof. (of Claim 6.4). Suppose there is a language $L \in \mathcal{C}$ on which d does not succeed. Then there is a natural number m such that for all $w \sqsubseteq L$, $d(w) \leq m$. Now for each N

define

$$\mathcal{O}_N := \{ \rho : (\exists w \sqsubseteq L, |w| \leq N) G^{\rho}(w) \geq 2m \}, \tag{4}$$

and finally define $\mathcal{O} = \cup_{w \sqsubseteq L} \mathcal{O}_w$. Then each $\mathcal{O}_N$ is topologically open in the space $\{ 0, 1 \}^{\omega}$, because $\rho \in \mathcal{O}_N$ is witnessed by a finite initial segment $\sigma \sqsubseteq \rho$ of length $t(n)$, which is long enough to fix the computation $G^{\sigma}(w)$. Also clearly $\mathcal{O}_1 \subseteq \mathcal{O}_2 \subseteq \mathcal{O}_3 \ldots$, so that $\mathcal{O}$ is an increasing union of the $\mathcal{O}_N$.

Finally and crucially, each $\mathcal{O}_N$ has measure at most $1/2$. This is because the measure of $\mathcal{O}_N$ is just the proportion of σ of length $t(n)$ such that $G^{\sigma}(w) \geq 2m$. If this proportion were $> 1/2$, then the overall average $d(w)$ would be $> m$, contradicting the choice of L and m.

Thus we have an increasing countable union of open sets, each of measure at most $1/2$. It follows that their union $\mathcal{O}$ has measure at most $1/2$. The easiest way to see this is to write $\mathcal{O} = \mathcal{O}_1 \cup (\mathcal{O}_2 \setminus \mathcal{O}_1) \cup (\mathcal{O}_3 \setminus \mathcal{O}_2) \cup \ldots$. This makes $\mathcal{O}$ a disjoint union of pieces $\mathcal{O}_N \setminus \mathcal{O}_{N-1}$, and bounds $\mu(\mathcal{O})$ by an infinite sum whose summands are non-negative and whose partial sums are all $\leq 1/2$. Hence the sum converges to a value $\leq 1/2$.

Hence the complement $\mathcal{A}$ of $\mathcal{O}$ has measure at least $1/2$. For every $\rho \in \mathcal{A}$, and every N, $\rho \notin \mathcal{O}_N$. This implies that for all $w \sqsubseteq L$ (of any length N), $G^{\rho}(w) \leq 2m$. It follows that G^{ρ} does not succeed on L. Thus the set $\{ \rho : L \notin S^{\infty}[G^{\rho}] \}$ has measure at least $1/2$, contradicting the hypothesis that $\{ \rho : L \in S^{\infty}[G^{\rho}] \}$ has measure one.

Finally, we note that d has a full time-T randomized approximation scheme. This is simply because the averages can be estimated to within a desired $poly(t(n))$ mean error by making $poly(t(n))$ random samples, in time bounded by a polynomial in $t(n)$. More details follow.

Let A be a language along which G succeeds, fix a prefix w of A of length 2^n, and for an auxiliary sequence σ of length $t(n)$, let $G^{\sigma,w}$ denote the capital that G^{σ} has prior to querying any string of length $n+1$. As mentioned above, we will assume that the "slow-but-sure" construction of [BMR+98] has been applied to G. This ensures that the capital G has before querying any string of length $n+1$ is at most $O(t(n))$ (irrespective of its auxiliary sequence and the language that it is betting on). Given ϵ and δ, we will divide the $O(t(n))$-sized range into $O(t(n))$ tiny intervals of constant size. Pick $O(\frac{(t(n))^2}{\epsilon} \log \frac{t(n)}{\delta})$ random auxiliary sequences σ, and for each sequence σ, compute $G^{\sigma,w}$ and find out which interval this capital falls within. This computation takes time $t(n) O(\frac{(t(n))^2}{\epsilon} \log \frac{t(n)}{\delta}) = (t(n))^{O(1)} O(\frac{1}{\epsilon} \log \frac{1}{\delta})$. By standard Chernoff bounds, for any interval I, with probability $1 - \delta/\Omega(t(n))$, the probability that $G^{\sigma,w}$ falls within I is accurate to within $\epsilon/\Omega((t(n))^2)$. Thus with probability at least $1 - O(t(n))(\delta/\Omega(t(n)))$, the probabilities are accurate to within $\epsilon/\Omega((t(n))^2)$ for every interval. Therefore, the estimate of $d(w)$ made this way is accurate within $O(t(n)) \times O(t(n)) \times \epsilon/\Omega((t(n))^2) = \epsilon$ with probability at least $1 - \delta$. $\square$

In the above proof of (c) $\Longrightarrow$ (a), by replacing "$2m$" by "Km" for larger K in (4), we can show that the measure of $\mathcal{O}$ arbitrarily close to zero. Hence we have: given G and L, if $\{ \rho : G^\rho \text{ covers } L \}$ does not have measure 1, then it has measure 0. This seems curious, because "G^ρ covers L" is not in general a "tail event." Nevertheless, nothing in this part of the argument requires this to be a tail event—nor even that G be a betting game! Thus our proof actually shows more generally that "FPRAS" is a robust notion of computing real-valued functions on Σ^*, namely that it is equivalent to the "measure one" type definitions that are possible.

7 Measuring Sub-Exponential BP Time Classes Directly

Now we can carry over the definitions and results of Section 4 to sub-exponential time bounds $t(n)$, starting right away with the notion of decisiveness.

Definition 7.1. A martingale d is *$t(n)$-decisive* if there exists $k > 0$ such that for all w, taking $N = |w|$ and $n = \lceil \log_2 \rceil N$, either $d(w1) - d(w) = 0$ or $|d(w1) - d(w)| \geq 1/t(n)^k$.

Recall that a family T of time bounds is well-behaved if for all $t \in T$ and $k > 0$, the function $t(n)^k$ also belongs to T. The threshold $1/t(n)^k$ is fine enough to make time-$t(n)$ martingales that bet below it fail to succeed, and coarse enough to enable a time-$T(n)$ randomized approximation scheme's estimates to be finer than it, with high probability.

Although giving proof details here is somewhat redundant with Section 4, we want to make it fully clear that the results really do carry over to sub-exponential time bounds.

Proposition 7.1 *Decisive probabilistic martingales for well-behaved time bounds T satisfy measure axioms* **M2** *and* **M3***. That is:*

(a) For any language $A \in \mathrm{BPTIME}[T(n)]$, there is a decisive $\mathrm{BPTIME}[T(n)]$ martingale d such that $A \in S^{\infty[d]}$.

(b) For every decisive $\mathrm{BPTIME}[T(n)]$ martingale d, there is a language $A \in \mathrm{BPTIME}[T(n)]$ such that $A \notin S^\infty[d]$.

Proof. (a) Take $t \in T$ such that $A \in \mathrm{BPTIME}[t(n)]$. We can find a probabilistic TM M_A running in time $O(t(n))$ such that for all x of any length n, $\Pr[M_A(x) = A(x)] > 1 - 1/2^{2t(n)}$, which is bounded below by $1 - 1/2^{2n}$ since $t(n) \geq n$. Now let d be the martingale induced by the lex-limited betting game that plays only on strings 0^n and bets all of its capital on $A(0^n)$. Then d is trivially decisive, and is approximable using M_A in time $O(t(n))$, so it is a time-$T(n)$ probabilistic martingale that covers A.

(b) Define A to be the diagonal language of the martingale d, viz. $A = \{x : d \text{ loses money on } x\}$. We need to show that $A \in \mathrm{BPTIME}[T(n)]$. Take M be a time-$T(n)$ randomized approximation scheme that with high probability approximates (the betting strategy used by) d to within $1/2t(n)^k$, and let M_A accept x iff M says that the bet on x is negative. Owing to decisiveness, whenever $x \in A$, M will say "negative" with high probability, and similarly for $x \notin A$. Hence $A \in \mathrm{BPTIME}[T(n)]$. $\square$

Proposition 7.2 *Probabilistic time-T martingales satisfy* **M1***–finite unions and* **M2***, and satisfy* **M1***–infinite unions for classes closed under finite variations.*

Proof. The proof for **M2** is immediate by the last proof. The proof for the finite case of **M1** is the same as that of (a) in Proposition 4.1. For infinite unions, we combine the construction in Proposition 5.4 with the idea for finite unions. By running approximations for values $d_1(w), \ldots, d_n(w)$ so that each comes within a factor of $(1 \pm \epsilon_n/n)$ with probability at least $(1 - \delta_n/n)$, we can approximate the desired weighted sum of $d_1(w), \ldots, d_n(w)$ to within a factor of $(1 \pm \epsilon_n)$, with probability at least $(1 - \delta_n)$. Setting $\epsilon_n = 1/t(n)^3$ and δ_n similarly does the trick. $\square$

Theorem 7.3 *For all individual time bounds $t \in T(n)$, $\mathrm{BPTIME}[t(n)]$ can be covered by a time-$T(n)$ probabilistic martingale.*

Proof. Take $P_1, P_2, \ldots$ to be a standard recursive enumeration of probabilistic Turing machines that run in time $t(n)$. Now play the randomized lex-limited betting game M defined as follows. M divides its initial capital $C_0 = 1$ into infinitely many "shares" $s_k = 1/2k^2$ for $k \geq 1$ (with an unused portion $1 - \pi^2/12$ of C_0 left over). Each share s_k is assigned to P_k, maintains its own capital, and plays on infinitely many strings x of the form $x = 0^n$, where n is divisible by 2^k but not by 2^{k+1}. Then no two shares play on the same string.

To play share s_k on a string $x = 0^n$, M uses its random bits to simulate P_k $t(n)$-many times, treating acceptance as $+1$ and rejection as -1, and lets ν be the sample mean of the results. M then bets a portion $\nu/2$ of the current capital of share s_k with the same sign as ν. Then M runs in time $O(t(n)^2)$.

As in the proof of Theorem 4.2, For any P_k that has bounded error probability, a measure-one set of random sequences make $\nu > 1/2$ for all but finitely many $0^n \in L(P_k)$ and $\nu < -1/2$ for all but finitely many $0^n \in\sim L(P_k)$. Hence for any such sequence, share s_k survives a possible finite sequence of losses and eventually grows to $+\infty$. Hence M succeeds on $L(P_k)$. $\square$

The corollary now follows directly from the measure, rather than relying on padding results.

Corollary 7.4 *If probabilistic time-$T(n)$ martingales can be simulated by decisive ones, then for all $t(n) \in T$,* $\mathrm{BPTIME}[t(n)] \neq \mathrm{BPTIME}[T(n)]$.

If the decisive simulation is uniform enough to apply to any well-behaved T, then

$$\begin{aligned} \mathrm{BPTIME}[O(n)] &\subset \mathrm{BPP} \subset \mathrm{BPTIME}[qpoly], \text{ and} \\ \mathrm{BPE} &\subset \mathrm{BPEXP}, \end{aligned}$$

all of which are unknown and possibly contradicted by oracle evidence.

Cai et al. [CLL$^+$95] define the notion of a *feasible generator* to be a prob. polynomial time machine that, on input 1^n, generates a string of length n according to some probability distribution $D = \{D_n\}_{n=0}^{\infty}$. In that paper we raised the following question of whether every feasible generator has a *monic refinement*: for every feasible generator M, is there a machine M' such that for all n, there exists $y \in \{0,1\}^n$ s.t. $\Pr[M'(1^n) = y] > 3/4$? We also studied the analogue of this question for arbitrary generators (i.e., non-feasible generators), and showed that, in general, this is impossible.

Further, in [CLL$^+$95] we observed a connection between this problem and the notion of fully polynomial time randomized approximation scheme (FPRAS), and established the following two facts:

(1) For every function f with an FPRAS, there is a machine M_f such that for every x and ϵ there are two values y_1 and y_2 such that $(1-\epsilon)f(x) \leq y_1 \leq y_2 \leq (1+\epsilon)f(x)$ and such that for any δ, $\Pr[M_f(x, \epsilon, \delta) \in \{y_1, y_2\}] > 3/4$.

(2) If there is a machine M_f that achieves an effect similar to (1) with just one value y instead of two values y_1 and y_2, then every feasible generator has a monic refinement.

It follows from our results that if every FPRAS has a symmetry breaking algorithm as in (2), then one would obtain a tight BPTIME hierarchy.

8 Conclusions

We have defined a natural and interesting candidate notion of probabilistic measure. The notion has already been applied to "natural proofs" and a simulation of "betting games," from which it follows that it measures classes not known to be measured deterministically. We have proved it to be fairly robust and appropriate for BPTIME classes. We have tied questions about its suitability to the longstanding open problem of whether there is a tight BPTIME hierarchy. One footnote on the latter deserves measure. It follows from our work that one of the following is true:

- $\#\mathrm{P} \neq \mathrm{P}$

- BPP has a nontrivial time hierarchy, viz. for some k, and all c, $\mathrm{BPTIME}[n^c] \neq \mathrm{BPTIME}[n^{kc}]$.

While this can be argued directly by letting k be something like the time for computing the permanent assuming $\#\mathrm{P} = \mathrm{P}$, the connection through measure is interesting.

References

[Amb86] K. Ambos-Spies. Relativizations, randomness, and polynomial reducibilities. In *Proceedings, First Annual Conference on Structure in Complexity Theory*, volume 223 of *Lect. Notes in Comp. Sci.*, pages 23–34. Springer Verlag, 1986.

[AS94] E. Allender and M. Strauss. Measure on small complexity classes, with applications for BPP. In *Proc. 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 807–818, 1994.

[AS95] E. Allender and M. Strauss. Measure on P: Robustness of the notion. In *Proc. 20th International Symposium on Mathematical Foundations of Computer Science*, volume 969 of *Lect. Notes in Comp. Sci.*, pages 129–138. Springer Verlag, 1995.

[BFT95] H. Buhrman, L. Fortnow, and L. Torenvliet. Using autoreducibility to separate complexity classes. In *36th Annual Symposium on Foundations of Computer Science*, pages 520–527, Milwaukee, Wisconsin, 23–25 October 1995. IEEE.

[BG81a] C. Bennett and J. Gill. Relative to a random oracle A, $\mathrm{P}^A \neq \mathrm{NP}^A \neq \mathrm{coNP}^A$ with probability 1. *SIAM J. Comput.*, 10:96–113, 1981.

[BG81b] C. Bennett and J. Gill. Relative to a random oracle A, $\mathrm{P}^A \neq \mathrm{NP}^A \neq \mathrm{coNP}^A$ with probability 1. *SIAM J. Comput.*, 10:96–113, 1981.

[BL96] H. Buhrman and L. Longpré. Compressibility and resource bounded measure. In *13th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1046 of *lncs*, pages 13–24, Grenoble, France, 22–24 February 1996. Springer.

[BMR+98] H. Buhrman, D. van Melkebeek, K. Regan, D. Sivakumar, and M. Strauss. A generalization of resource-bounded measure, with an application. In *Proc. 15th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1373 of *Lect. Notes in Comp. Sci.*,, pages 161–171. Springer Verlag, 1998.

[CLL$^+$95] J.-Y. Cai, R. Lipton, L. Longpré, M. Ogihara, K. Regan, and D. Sivakumar. Communication complexity of key agreement on limited ranges. In *Proc. 12th Annual Symposium on Theoretical Aspects of Computer Science*, volume 900 of *Lect. Notes in Comp. Sci.*, pages 38–49. Springer Verlag, 1995.

[CSS97] J.-Y. Cai, D. Sivakumar, and M. Strauss. Constant depth circuits and the Lutz hypothesis. In *Proc. 38th Annual IEEE Symposium on Foundations of Computer Science*, pages 595–604, 1997.

[FS89] L. Fortnow and M. Sipser. Probabilistic computation and linear time. In *Proc. 21st Annual ACM Symposium on the Theory of Computing*, pages 148–156, 1989.

[FS97] L. Fortnow and M. Sipser. Retraction of "Probabilistic computation and linear time". In *Proc. 29th Annual ACM Symposium on the Theory of Computing*, page 750, 1997.

[GZ97] O. Goldreich and D. Zuckerman. Another proof that BPP $\subseteq$ PH (and more). Technical Report TR97–045, Electronic Colloquium on Computational Complexity (ECCC), September 1997.

[HILL91] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. Construction of a pseudorandom generator from any one-way function. Technical Report 91–68, International Computer Science Institute, Berkeley, 1991.

[JS89] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18:1149–1178, 1989.

[JVV86] M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comp. Sci.*, 43:169–188, 1986.

[KL83] R. Karp and M. Luby. Monte-Carlo algorithms for enumeration and reliability problems. In *Proc. 24th Annual IEEE Symposium on Foundations of Computer Science*, pages 56–64, 1983.

[Ko83] K. Ko. On the definitions of some complexity classes of real numbers. *Math. Sys. Thy.*, 16:95–109, 1983.

[KV87] M. Karpinski and R. Verbeek. Randomness, provability, and the separation of Monte Carlo time and space. In *Computation Theory and Logic*, volume 270 of *Lect. Notes in Comp. Sci.*, pages 189–207. Springer Verlag, 1987.

[Lut92] J. Lutz. Almost everywhere high nonuniform complexity. *J. Comp. Sys. Sci.*, 44:220–258, 1992.

[May94] E. Mayordomo. *Contributions to the Study of Resource-Bounded Measure*. PhD thesis, Universidad Polytécnica de Catalunya, Barcelona, April 1994.

[NW88] N. Nisan and A. Wigderson. Hardness vs. randomness. In *Proc. 29th Annual IEEE Symposium on Foundations of Computer Science*, pages 2–11, 1988.

[RR97] A. Razborov and S. Rudich. Natural proofs. *J. Comp. Sys. Sci.*, 55:24–35, 1997.

[RSC95] K. Regan, D. Sivakumar, and J.-Y. Cai. Pseudorandom generators, measure theory, and natural proofs. In *Proc. 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 26–35, 1995.

[RV97] R. Rettinger and R. Verbeek. BPP equals BPLINTIME under an oracle (extended abstract), November 1997.

[Str97] M. Strauss. Measure on P—strength of the notion. *Inform. and Comp.*, 136:1–23, 1997.