Electronic Colloquium on Computational Complexity, Report No. 58 (2002)

A Generalization of Lutz's Measure to Probabilistic Classes

Philippe Moser*

Abstract

We extend Lutz's measure to probabilistic classes, and obtain notions of measure on probabilistic complexity classes C such as BPP, BPE and BPEXP. Unlike former attempts, all our measure notions satisfy all three Lutz's measure axioms, that is every singleton $\{L\}$ has measure zero in C, the whole space C has measure one in C, and "easy infinite unions" of measure zero sets have measure zero. Finally we prove a conditional time hierarchy Theorem for probabilistic classes, and show that under the same assumption, both the class of $\leq_T^p$-autoreducible sets and the class of $\leq_T^p$-complete sets for EXP have measure zero in BPE.

1 Introduction

Resource-bounded measure was introduced by Lutz in [Lut90] and [Lut92] for both complexity classes EXP and E. It provides a means of investigating the sizes of various subsets of E and EXP. Given a subset C of EXP such as P, NP or BPP one tries to determine whether C is a small subset of EXP i.e. has measure zero, or is a large subset, i.e. has measure one. Resource-bounded measure has been used with many successes to understand the structure of the exponential time classes E and EXP.

The first goal of Lutz's approach was to extend existence results, such as "there is a language in C satisfying property P, to abundance results such as "most languages in C satisfy property P", which is more informative since an abundance result reflects the typical behavior of languages in a class, whereas an existence result could as well correspond to an exception in the class. For instance the set of $\leq_m^p$-complete languages for E has measure zero in E [JL93].

Another application of resource-bounded measure is in relation with the probabilistic method. Suppose we want to prove the existence of a set L in E satisfying property P. It is often easier to prove that the subset of E not satisfying property P is small i.e. has measure zero, than explicitly constructing a set L with property P. In [LM94] Lutz and Mayordomo used this technique to prove results about the density of hard languages.

Plausible but unproven hypothesis such as P $\neq$ NP and "the polynomial time hierarchy does not collapse" are useful to provide information concerning complexity theoretical propositions. Resource-bounded measure can also be used to formulate new plausible working hypothesis such as "NP is not a small subset of E". For instance Impagliazzo and Moser showed in [IM02] that under the hypothesis "NP has p-measure non zero" full derandomization of AM was possible, i.e. NP = AM.

For a more detailed survey on Lutz's resource bounded measure see [Lut97].

*Address: Theoretical computer science department, University of Geneva. Email: moser@cui.unige.ch

ISSN 1433-8092

Resource-bounded measure can be seen as a general framework which for many complexity classes C, yields a notion of "measure in C" which satisfies the following three axioms. First every singleton $\{L\}$ (where $L \in \mathsf{C}$) has measure zero in C, second the whole space C has measure one in C, and finally "easy infinite unions" of measure zero sets have measure zero in C. These axioms meet the essence of Lebegue's measure and ensure that it is impossible for a subset of C to have both measure zero and one in C.

Unfortunately, Lutz's formulation only works for measure in $\mathsf{C} \supseteq \mathsf{E}$. In [AS94] and [Str97] Allender and Strauss generalized Lutz's measure by introducing two measure notions in subexponential classes such as P and PSPACE. And what about probabilistic classes?

In [RS98], Regan and Sivakumar investigated the notion of measure on probabilistic classes. They introduced probabilistic martingales and defined a partial measure on probabilistic classes, unfortunately their partial measure does not satisfy the three measure axioms, indeed their theory missed out the finite union axioms, and henceforth the easy infinite union axioms.

It was thus left open whether it is possible to define a measure on probabilistic classes which satisfies Lutz's three measure axioms. We give an affirmative answer to this question by constructing measures on all probabilistic complexity classes BPP, BPE and BPEXP which satisfy all three Lutz's measure axioms.

The remainder of the paper is organized as follows. In Section 3 we introduce our measure on BPE. We first define probabilistic martingales after what we prove the so-called exact computation Lemma, which states that every probabilistic martingales which can be approximated, can also be computed exactly. Next we show that all three Lutz's measure axioms hold for our measure on BPE. All arguments in Section 3 also hold with E replaced by EXP thus one also obtains a measure on BPEXP.

For BPP we use both measure notions of Allender and Strauss in [AS94] and [Str97] on P, and adapt them to the probabilistic class BPP, thus obtaining two different measures on BPP. In Section 4 we use Allender and Strauss's polylogarithmic Turing machines model [AS94], whereas In Section 5 we use Strauss's dense betting strategies [Str97] and obtain another (stronger) measure on BPP.

Finally we investigate the problem of monic selection of random sampling (MSRS). A remarkable property of probabilistic Turing machines is that they can perform random sampling. The point is that such a sampling is multivalued, i.e. with high probability the machine outputs good approximations, but it might output different approximations depending on the output of the random coin tosses. An interesting question is whether it is possible to perform probabilistic sampling by a single valued probabilistic Turing machine, i.e. is it possible to output with high probability a single value, which is a good approximation of the sampling. This question is addressed in [OR93], and it is asked what consequences would follow from the assumption that MSRS is possible. We show that if MSRS is possible then the time hierarchy Theorem holds for probabilistic classes, i.e.

$$\mathsf{BPTIME}(O(n)) \subsetneq \mathsf{BPP} \subsetneq \mathsf{BPE} \subsetneq \mathsf{BPEXP}.$$

We also show that if MSRS is possible then both the class of $\leq_T^p$-autoreducible sets and the class of $\leq_T^p$-complete sets for EXP have measure zero in BPE. It is not known whether the same holds for measure in E, in fact this would have the nonrelativizing consequence $\mathsf{BPP} \neq \mathsf{EXP}$.

2 Preliminaries

We use standard notation for traditional complexity classes; see for instance the books of Balcazar, Diaz and Gabarro [BDG95], [BDG90], or the one from Papadimitriou [Pap94]. Let us fix some notations for strings and languages. Let $s_0, s_1, \ldots$ be the standard enumeration of the strings in $\{0,1\}^*$ in lexicographical order, where $s_0 = \lambda$ denotes the empty string. A sequence is an element of $\{0,1\}^\infty$. If w is a string or a sequence and $0 \leq i < |w|$ then $w[i]$ and $w[s_i]$ denotes the ith bit of w. Similarly $w[i \ldots j]$ and $w[s_i \ldots s_j]$ denote the ith through jth bits. We identify language L with its characteristic function χ_L, where χ_L is the sequence such that $\chi_L[i] = 1$ iff $s_i \in L$. If w_1 is a string and w_2 is a string or a sequence extending w_1, we write $w_1 \sqsubseteq w_2$.

2.1 Betting games

Lutz's [Lut92] measure on E is obtained by imposing appropriate resource-bound on a game theoretical characterization of the classical Lebesgue measure, via martingales. A martingale is a function $d : \{0,1\}^* \to \mathbb{R}_+$ such that,

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

for every $w \in \{0,1\}^*$.

This definition can be motivated by the following betting game in which a gambler puts bets on the successive membership bits of a hidden language A. The game proceeds in infinitely many rounds where at the end of round n, it is revealed to the gambler whether $s_n \in A$ or not. The game starts with capital 1. Then, in round n, depending on the first $n-1$ outcomes $w = \chi_A[0 \ldots n-1]$, the gambler bets a certain fraction $\epsilon_w d(w)$ of his current capital $d(w)$, that the nth word $s_n \in A$, and bets the remaining capital $(1 - \epsilon_w)d(w)$ on the complementary event $s_n \notin A$. The game is fair, i.e. the amount put on the correct event is doubled, the one put on the wrong guess is lost, as stated in Equation 1. The value of $d(w)$, where $w = \chi_A[0 \ldots n]$ equals the capital of the gambler after round n on language A. The player wins on a language A if he manages to make his capital arbitrarily large during the game. We say that a martingale d succeeds on a language A, if $d(A) := \limsup_{w \sqsubset A, w \to A} d(w) = \infty$, where we identify language A with its characteristic sequence χ_A. The success set $S^\infty[d]$ of a martingale d is the class of all languages on which d succeeds.

We sometimes relax equality 1 by considering supermartingales. A supermartingale is a function $d : \{0,1\}^* \to \mathbb{R}_+$ such that,

$$d(w) \geq \frac{d(w0) + d(w1)}{2} \tag{2}$$

for every $w \in \{0,1\}^*$, i.e. the associated strategy is allowed to throw money away.

3 A Measure on BPE

Our measure on BPE will be defined via the following probabilistic martingales.

Definition 1 *A martingale $d : \{0,1\}^* \to \mathbb{R}_+$ is* BPESV *approximable if there exists a family of approximations $\{\hat{d}_k\}_{k\geq 0}$ (where $\hat{d}_k : \{0,1\}^* \to \mathbb{Q}_+$), and a probabilistic Turing machine M such that for every $w \in \{0,1\}^*, k, n \in \mathbb{N}$*

$$|\hat{d}_k(w) - d(w)| \leq 2^{-k}, \text{ and}$$
$$\Pr[M(w,k,n) = \hat{d}_k(w)] \geq 1 - 2^{-n}$$

where the probability is taken over the internal coin tosses of M and the running time of M is polynomial in $|w| + k + n$.

By using standard Chernoff bound arguments it is easy to show that Definition 1 is robust i.e. the error probability can range from $1/2 + 1/p(n)$ to $1 - 2^{q(n)}$ for any polynomials p, q without enlarging (resp. reducing) the class of functions defined in Definition 1.

A martingale $d : \{0,1\}^* \to \mathbb{Q}_+$ is said BPESV computable if there exists a probabilistic Turing machine M such that for every $w \in \{0,1\}^*, n \in \mathbb{N}$

$$\Pr[M(w,n) = d(w)] \geq 1 - 2^{-n}$$

where the probability is taken over the internal coin tosses of M and the running time of M is polynomial in $|w| + n$.

We will often consider indexed martingale. An indexed BPESV approximable martingale is a martingale d (where $d_i := d(i,\cdot)$) such that there exists a family of approximations $\{\hat{d}_{k,i}\}_{k,i\geq 0}$ (where $\hat{d}_{k,i} : \{0,1\}^* \to \mathbb{Q}_+$), and a probabilistic Turing machines M such that for every $w \in \{0,1\}^*, k, i, n \in \mathbb{N}$

$$|\hat{d}_{k,i}(w) - d_i(w)| \leq 2^{-k}, \text{ and}$$
$$\Pr[M(w,k,i,n) = \hat{d}_{k,i}(w)] \geq 1 - 2^{-n}$$

where the probability is taken over the internal coin tosses of M and the running time of M is polynomial in $|w| + k + i + n$.

Following Lutz [Lut92] we say that a set has measure zero if there is a single martingale that succeeds on it.

Definition 2 *$A \subseteq \mathsf{E}$ is said to have* BPESV *measure zero if there exists a* BPESV *approximable martingale $d : \{0,1\}^* \to \mathbb{R}_+$ such that $A \subseteq S^\infty[d]$.*

In order to formalize the third measure axiom, we need to define what we mean by "easy infinite union" of measure zero sets.

Definition 3 *$X = \bigcup_{i\in\mathbb{N}} X_i$ is a* BPESV *union of* BPESV *measure zero sets if there exists an indexed* BPESV *approximable martingale d such that $X_i \subseteq S^\infty[d_i]$.*

The so-called exact computation Lemma states that any BPESV approximable martingale can be replaced by a BPESV computable martingale with the same success set.

Lemma 1 *(Exact Computation Lemma)*
Let $d : \{0,1\}^* \to \mathbb{R}_+$ be a BPESV *approximable martingale. Then there exists a* BPESV *computable martingale $d' : \{0,1\}^* \to \mathbb{Q}_+$ such that $S^\infty[d] = S^\infty[d']$.*

Proof

Let $\hat{d}$ be an approximation of d, and let M be a probabilistic Turing machine computing $\hat{d}$. Let us define $c(w) := \hat{d}_{|w|}(w)$. We construct the following martingale d' recursively.

$$d'(\lambda) = c(\lambda) + 2$$
$$d'(wb) = d'(w) + \frac{c(wb) - c(w\bar{b})}{2} \text{ where } w \in \{0,1\}^* \text{ and } b \in \{0,1\}.$$

Claim d' is BPESV computable.

Indeed computing $d'(wb)$ requires computing $|w|$ recursive steps, each step requiring two computations of c. By computing c (via M) with error probability smaller than $2^{-s(n)}$ (where $s(n)$ is a polynomial which will be determined later), we obtain a total error probability smaller than $2|w|2^{-s(n)}$. Putting $s(n) = \log(|w|) + n + 1$ yields a total error probability smaller than 2^{-n}.

Let us check that d' defines a martingale. It is easy to check the average Equality 1. In order to check that $d'(w) \geq 0$ for every $w \in \{0,1\}^*$, we show by induction that

$$d'(w) \geq d(w) + 2^{-|w|}. \tag{3}$$

We have $d'(\lambda) = c(\lambda) + 2 \geq d(\lambda) - 2^0 + 2 \geq d(\lambda) + 2^0$. For $w \in \{0,1\}^*, b \in \{0,1\}$, we have

$$\begin{aligned} d'(wb) &= d'(w) + \frac{c(wb) - c(w\bar{b})}{2} \\ &\geq d(w) - 2^{-|w|} + \frac{c(wb) - c(w\bar{b})}{2} \\ &\geq d(w) + 2^{-|w|} + \frac{d(wb) - d(w\bar{b})}{2} - 2^{-|w|-1} \\ &= d(wb) + 2^{-|w|-1} \end{aligned}$$

where the first inequality holds by induction and the second holds because $|d(w) - c(w)| \leq 2^{-|w|}$ by definition of c. Next we show by induction that

$$|d(w) - d'(w)| \leq 4 - 2^{-|w|}. \tag{4}$$

For $w = \lambda$ we have $|d(\lambda) - c(\lambda) - 2| \leq 2^{-0} + 2 = 3$. For $w \in \{0,1\}^*, b \in \{0,1\}$ we have

$$\begin{aligned} |d'(wb) - d(wb)| &\leq |d'(w) - d(w)| + |d(w) + \frac{c(wb) - c(w\bar{b})}{2} - d(wb)| \\ &\leq 4 - 2^{|w|} + |\frac{d(w\bar{b}) - c(w\bar{b})}{2}| + |\frac{c(wb) - d(wb)}{2}| \\ &\leq 4 - 2^{-(|w|+1)}. \end{aligned}$$

Finally Equation 3 and 4 yield $S^\infty[d] = S^\infty[d']$. □

Let us prove that all three Lutz's measure axioms hold for our measure on BPE.

Theorem 1 *Let L be any language in* BPE. *Then the singleton $\{L\}$ has* BPESV *measure zero.*

Proof

Let $L \in \mathsf{BPE}$ be any language and let M be a Turing machine computing it. We construct a probabilistic Turing machine T computing martingale d yielded by the following game strategy; On input S_N bet the current capital that the membership bit is the same as $L(s_N)$. d is BPESV computable since on input w, where $|w| = N$, T simply computes $L(s_0), L(s_1), \ldots, L(s_N)$, (with error probability smaller than $2^{-s(n)}$) and outputs $2^{|w|}$ if $w = \chi_L[1 \ldots N]$ and zero otherwise. Clearly d is BPESV computable since computing $d(w)$ requires N computations of L each being performed with error probability smaller than $2^{-s(n)}$, which makes a total error probability smaller than $N \cdot 2^{-s(n)}$. Putting $s(n) = \log N + n$ yields a total error probability smaller than 2^{-n}. □

The second axiom is proved using the Exact Computation Lemma.

Theorem 2 BPE *does not have* BPESV *measure zero.*

Proof

Let d be a BPESV approximable martingale. By Lemma 1 we can suppose that d is BPESV computable. We construct a language $L \in \mathsf{BPE}$ such that $d(\chi_L[0 \ldots N]) \leq d(\lambda)$ for every $N \geq 1$, i.e. $L \notin S^\infty[d]$. On word s_N, L is defined as follows,

$$L(s_N) = 1 \iff d(L(s_0)L(s_1)\ldots L(s_{N-1})1) \leq d(L(s_0)L(s_1)\ldots L(s_{N-1})0)$$

where d is computed with error probability $2^{-s(n)}$ (where s(n) is a polynomial which will be determined later). $L \in \mathsf{BPE}$ because each of the N recursive steps to compute $L(s_N)$ requires two computations of d. This yields a total error probability smaller than $2N2^{-s(n)}$. Putting $s(n) = \log(|w|) + n + 1$ yields a total error probability smaller than 2^{-n}. Moreover d never increases its initial capital along L which ends the proof. □

Finally let us prove the third axiom.

Theorem 3 *Let* $X = \bigcup_{i \geq 1} X_i$ *be a* BPESV *union of* BPESV *measure zero sets. Then* X *has* BPESV *measure zero.*

Proof

Let d be a BPESV approximable indexed martingale that wins on X, and let $\hat{d}$ be an approximation of d. We construct a BPESV approximable indexed martingale D such that for every $j \in \mathbb{N}$,

1. $S^\infty[D_j] = S^\infty[d_j]$
2. $D_j(\lambda) \leq 2^{-j}$.

For $w \in \{0,1\}^*$ and $j \in \mathbb{N}$ define

$$D_j(w) := 2^{\min(0, -\log(\hat{d}_{j,1}(\lambda)) - 2 - j)} d_j(w).$$

It is easy to see that D_j verifies both properties 1 and 2 for every $j \in \mathbb{N}$. For $w \in \{0,1\}^*, j, k \in \mathbb{N}$, consider

$$\hat{D}_{j,k}(w) := 2^{\min(0, -\log(\hat{d}_{j,1}(\lambda)) - 2 - j)} \hat{d}_{j,k}(w).$$

For $w \in \{0,1\}^*, j, k \in \mathbb{N}$ we have

$$|\hat{D}_{j,k}(w) - D_{j,k}(w)| \leq |\hat{d}_{j,k}(w) - d_{j,k}(w)| \leq 2^{-k}.$$

Since $\hat{d}_{j,k}$ is BPESV computable, so is $\hat{D}_{j,k}$ therefore D is BPESV approximable.

Consider the following martingale

$$d'(w) := \sum_{i=0}^{\infty} D_j(w) \text{ where } w \in \{0,1\}^*.$$

Since $d'(\lambda) \leq \sum_{j=0}^{\infty} 2^{-j} = 2$ and $d'(w) \leq \sum_{j=0}^{\infty} 2^{|w|} D_j(\lambda) \leq 2^{|w|} d'(\lambda)$ for $w \in \{0,1\}^*$, d' is a well defined martingale. It is clear that $X \subseteq S^{\infty}[d']$. Let us show that d' is BPESV approximable. For $w \in \{0,1\}^*$, consider

$$\hat{d'}_k(w) := \sum_{j=0}^{k+|w|+1} \hat{D}_{j,j+k+2}(w)$$

where each $\hat{D}_{j,j+k+2}(w)$ is computed with probability $2^{-s(n)}$ where $s(n)$ is a polynomial which will be determined later. For $w \in \{0,1\}^*, k \in \mathbb{N}$ we have

$$\begin{aligned} |\hat{d'}_k(w) - d'(w)| &\leq \sum_{j=0}^{k+|w|+1} |\hat{D}_{j,j+k+2}(w) - D_j(w)| + \sum_{j=|w|+k+2}^{\infty} |D_j(w)| \\ &\leq \sum_{j=0}^{k+|w|+1} 2^{-(j+k+2)} + \sum_{j=|w|+k+2}^{\infty} 2^{|w|} 2^{-j} \\ &\leq 2^{-(k+1)} + 2^{-(k+1)} \leq 2^{-k} \end{aligned}$$

Since computing $\hat{d'}_k(w)$ requires adding $k + |w| + 1$ terms $\hat{D}_{j,j+k+2}$, each being computed with error probability smaller than $2^{-s(n)}$, $\hat{d'}_k(w)$ can be computed with error probability smaller than $(k + |w| + 1)2^{-s(n)}$. Putting $s(n) := \log(k + |w| + 1) + n$ yields a total error probability smaller than $2^{-s(n)}$, thus d' is BPESV approximable which ends the proof. □

Throughout Section 3 E can be replaced by EXP thus yielding a measure on the probabilistic class BPEXP.

4 Measure on BPP with Sparse Dependencies

4.1 Supermartingales with small dependency set

To define a measure on BPP we will consider supermartingales. Our supermartingales will be computed by Turing machines with random access to their inputs i.e. on input w, the machine can query any bit of w to its oracle. We will consider polylogarithmic time Turing machines. In order to allow such Turing machines to compute the lengths of their inputs w without querying their oracles, we also provide them with $s_{|w|}$. For such a Turing machine M running on input w, we denote this convention by $M^w(s_{|w|})$. Since these Turing machines will need to approximate real valued martingales, we will suppose that these Turing machines output their results as two binary number (a, b) corresponding to the rational number a/b. It is easy to check that under this convention these Turing machines can perform the usual operations such as $+, -, \cdot, \div$ and $\leq$. The point is that with this convention, rationals such as $1/3$ can be said to be computed exactly.

It is widely believed that polylogarithmic Turing machines are too strong to define a measure on polynomial time classes, because it is not clear whether the whole class has not measure zero relatively to polylogarithmic computed supermartingales. Therefore Allender and Strauss [AS94] weakened the concept by bounding the number of recursive queries such a Turing machine is allowed to make. Let M be a polylogarithmic Turing machine and $n \in \mathbb{N}$. Define the dependency set $G_{M,n} \subseteq \{1, 2, \ldots, 2^{n+1} - 1\}$ such that for every string $w \in \{0,1\}^*$ coding for words of size up to n, M can compute $M^w(s_{|w|})$ querying only input bits in $G_{M,n}$.

To define our measure on BPP we will consider polylogarithmic probabilistic Turing machines with polynomial size dependency sets.

Definition 4 *A supermartingale $d : \{0,1\}^* \to \mathbb{R}_+$ is* BPPSV *approximable if there exists a family of approximations $\{\hat{d}_k\}_{k\geq 0}$ (where $\hat{d}_k : \{0,1\}^* \to \mathbb{Q}$), a family of polynomial sized dependency sets $\{G_{M,n}\}_{n\geq 0}$ and a probabilistic Turing machine M such that for every $w \in \{0,1\}^*, k, n \in \mathbb{N}$*

$$|\hat{d}_k(w) - d(w)| \leq 2^{-k}, \text{ and}$$
$$\Pr[M^w(s_{|w|}, k, n) = \hat{d}_k(w)] \geq 1 - 2^{-n}$$

where the probability is taken over the internal coin tosses of M, the running time of M is polynomial in $\log |w| + k + n$ and M only queries input bits in $G_{M,|s_{|w|}|}$.

BPPSV computable supermartingales and indexed supermartingales are defined as in Section 3.

Similarly to Section 3, a set is said to have measure zero if there is a single martingale that succeeds on it.

Definition 5 *$A \subseteq$ BPP is said to have* BPPSV *measure zero if there exists a* BPPSV *approximable supermartingale d such that $A \subseteq S^\infty[d]$.*

As in Section 3, we need to formalize the concept of easy infinite unions of small sets.

Definition 6 *$X = \bigcup_{i\in\mathbb{N}} X_i$ is a* BPPSV *union of* BPPSV *measure zero sets if there exists an indexed* BPPSV *approximable supermartingale d such that $X_i \subseteq S^\infty[d_i]$.*

The Exact Computation Lemma also holds for BPPSV approximable martingale.

Lemma 2 *(Exact Computation Lemma)*
Let $d : \{0,1\}^* \to \mathbb{R}_+$ be a BPPSV *approximable supermartingale. Then there exists a* BPPSV *computable supermartingale $d' : \{0,1\}^* \to \mathbb{Q}_+$ such that $S^\infty[d] = S^\infty[d']$.*

Proof
Let $\hat{d}$ be an approximation of d, such that for every $w \in \{0,1\}^*$ we have $|d(w) - \hat{d}_{|w|}(w)| \leq \frac{1}{|w|^2+1}$, and let M be a probabilistic Turing machine computing $\hat{d}$. Consider the following function

$$F(|w|) = \begin{cases} \frac{4}{|w|} & \text{if } |w| > 0, \\ 6 & \text{if } |w| = 0. \end{cases}$$

F can be computed in time polynomial in $\log |w|$. Consider the following supermartingale

$$d'(w) = \frac{\hat{d}_w(w) + F(|w|)}{2 + F(0)}.$$

Since $\hat{d}$ is BPPSV computable, so is d'. We have $S^\infty[d] = S^\infty[d']$ and d and d' have the same dependency set. Let us show that d' is a supermartingale, i.e. satisfies Equation 2. For $w \in \{0,1\}^*$ we have (we omit the constant factor)

$$\begin{aligned}
d'(w) = \hat{d}_{|w|}(w) + F(|w|) &\geq \frac{d(w0) + d(w1)}{2} - \frac{1}{|w|^2+1} + F(|w|) \\
&\geq \frac{\hat{d}_{|w0|}(w0) - 1/(|w0|^2+1)}{2} + \frac{\hat{d}_{|w1|}(w1) - 1/(|w1|^2+1)}{2} - \frac{1}{|w|^2+1} + F(|w|) \\
&= \frac{\hat{d}_{|w0|}(w0) + \hat{d}_{|w1|}(w1)}{2} - \frac{1}{(|w|+1)^2+1} - \frac{1}{|w|^2+1} + F(|w|) \\
&\geq \frac{\hat{d}_{|w0|}(w0) + \hat{d}_{|w1|}(w1)}{2} + \underbrace{F(|w|+1)}_{=(F(|w0|)+F(|w1|))/2} = \frac{d'(w0) + d'(w1)}{2}
\end{aligned}$$

Let us check the last inequality. Putting $t = |w|$, we have to check whether

$$F(t) - F(t+1) \geq \frac{1}{(t+1)^2+1} + \frac{1}{t^2+1}. \tag{5}$$

The left term of Equation 5 is equal to $\frac{4}{t(t+1)}$. Denote by R the right term of Equation 5. Equation 5 holds iff

$$4 \geq R \cdot t(t+1) \iff 4 \geq \frac{t^2+t}{(t+1)^2+1} + \frac{t^2+t}{t^2+1}$$

which is true for every positive t. □

Let us check that all three measure axioms hold.

Theorem 4 *Let $L \in$ BPP be any language. Then the singleton $\{L\}$ has* BPPSV *measure zero.*

Proof

Let $L \in$ BPP be any language and let M be a probabilistic Turing machine deciding it. Consider the following supermartingale d. d only bets on words in $\{0\}^*$ and on those words, d bets all its current capital according to M. d is BPPSV computable because computing $d(w)$ (where $w \in \{0,1\}^N$) requires computing $a_t := M(0^t)$ for $t = 1, 2, \ldots, |s_N|$ (with error probability smaller than $2^{-s(n)}$) and checking whether $a_t = w[0^t]$ for every $t = 1, 2, \ldots, |s_N|$, where $w[0^t]$ denotes the bit of w corresponding to word 0^t, in which case $d(w)$ equals $2^{|s_N|}$ otherwise $d(w)$ equals 0. The total error probability is smaller than $|s_N| \cdot 2^{-s(n)}$, which is smaller than 2^{-n} by an appropriate choice of $s(n)$. The dependency set is $\{0^t\}_{t \leq |s_N|}$ which is polynomial sized. □

The proof of the second axiom relies on the Exact Computation Lemma.

Theorem 5 BPP *does not have* BPPSV *measure zero.*

Proof Suppose there exists a BPPSV approximable supermartingale d such that d beats BPP. By Lemma 2 there is a BPPSV computable supermartingale d' with the same success set as d. Denote this supermartingale by d. Consider the following language $L \in$ BPP.

$$L(s_N) = \begin{cases} 1 & \text{if } d(\chi_L[0 \dots N-1]1) \leq d(\chi_L[0 \dots N-1]), \\ 0 & \text{otherwise.} \end{cases}$$

Since the dependency set is polynomial there are at most $q(|s_N|)$ recursive calls, for some polynomial q. Each recursive call requires two computations of d, each being performed with error probability smaller than $2^{-s(n)}$. Thus the total error probability is smaller than $q(|s_N|)2^{-s(n)}$ which is smaller than 2^{-n} for an appropriate choice of $s(n)$. Since d never increases its initial capital along L, we have $L \notin S^\infty[d]$. □

Finally let us prove the third axiom.

Theorem 6 *Let $X = \bigcup_{i \geq 1} X_i$ be a* BPPSV *union of* BPPSV *measure zero sets. Then X has* BPPSV *measure zero.*

Proof

Let d be an indexed BPPSV approximable supermartingale that beats X. By Lemma 2 we can suppose that d is BPPSV computable. Consider the following supermartingale

$$D_j(w) = 2^{-p(j)} d_j(w)$$

where p is a polynomial which will be determined later. We have

1. $D_j(\lambda) \leq 2^{-p(j)}$ (Wlog $d_j(\lambda) \leq 1$ for every $j \in \mathbb{N}$)
2. $S^\infty[D_j] = S^\infty[d_j]$ for every $j \in \mathbb{N}$.

For $w \in \mathbb{N}$ consider the following supermartingale

$$d'(w) = \sum_{j=0}^{\infty} D_j(w).$$

d' is well defined because

$$d'(\lambda) = \sum_{j=0}^{\infty} D_j(\lambda) \leq \sum_{j=0}^{\infty} 2^{-p(j)} \leq 2.$$

It is clear that $X \subseteq S^\infty[d']$. Let us check that d' is BPPSV approximable. Consider the following approximation

$$\hat{d'}_k(w) = \sum_{j=0}^{k+\log|w|+1} D_j(w).$$

Let us show that $\hat{d'}$ is BPPSV computable. First the dependency set of $\hat{d'}$ is the union of $k+\log|w|+1$ dependency sets, which still is polynomial sized. Second computing $\hat{d'}_k$ requires computing $k+\log|w|+1$ terms D_j, each being computed with error probability smaller than $2^{-s(n)}$. This yields a total error probability smaller than $(k+\log|w|+1)2^{-s(n)}$ which is smaller than 2^{-n} for an appropriate choice of $s(n)$. Since d_j is BPPSV computable its dependency

set is polynomial sized, therefore $d_j(w) \leq 2^{q(\log|w|+j)}$ for some polynomial q. Consider the following polynomial $p(x) = q(2x) + x$. For $w \in \{0,1\}^*$ we have

$$\begin{aligned}
|\hat{d'}_k(w) - d'(w)| &\leq \sum_{j=k+\log|w|+2}^{\infty} D_j(w) &&= \sum_{j=k+\log|w|+2}^{\infty} 2^{-p(j)} d_j(w) \\
&\leq \sum_{j=k+\log|w|+2}^{\infty} 2^{q(\log|w|+j)} 2^{-p(j)} d_j(\lambda) &&\leq \sum_{j=k+\log|w|+2}^{\infty} 2^{q(\log|w|+j)} 2^{-p(j)} \\
&\leq \sum_{j=k+\log|w|+2}^{\infty} 2^{q(\log|w|+j)} 2^{-q(2j)} 2^{-j} &&\leq \sum_{j=k+\log|w|+2}^{\infty} 2^{-j} \\
&\leq 2^{-k}
\end{aligned}$$

Thus d' is BPPSV approximable which ends the proof. □

5 Measure on BPP with Large Dependency Sets

5.1 Betting strategies

To define a second measure on BPP, we consider betting strategies instead of supermartingales.

Definition 7 *A betting strategy is a function $\beta : \{0,1\}^* \to \mathbb{R}_+$ such that for every $w \in \{0,1\}^*$, $\beta(w0) + \beta(w1) \leq 0$ and $\sum_{z \sqsubseteq w} \beta(z) \geq 0$.*

Intuitively $\beta(w0)$ is the amount of money the strategy wins or looses while betting on the last bit of $w0$; the first inequality guarantees that on every bet the possible won amount is not larger than the possible lost amount, whereas the second inequality guarantees that the strategy never bets more money than its current capital.

For a betting strategy β, the function

$$d(w) = \sum_{z \sqsubseteq w} \beta(z) \tag{6}$$

is a supermartingale, on the other hand starting with a supermartingale d, one obtains a betting strategy by putting $\beta(wi) = \frac{d(wi)-d(w)}{2}$, where $w \in \{0,1\}^*, i \in \{0,1\}$. Therefore in large complexity classes such as E and EXP betting strategies and supermartingales are equivalent since the exponentially large sum of Equation 6 can be computed in E. On the contrary, Strauss has shown in [Str97] that for polynomial time classes, betting strategies define a stronger measure than supermartingales.

We therefore use betting strategies to define our second measure on BPP.

Our second measure on BPP will use a more general notion of dependency set as in Section 4. This dependency notion is from Strauss [Str97].

Definition 8 *A betting strategy $\beta : \{0,1\}^* \to \mathbb{R}_+$ is dense computable if for every $x \in \{0,1\}^*$ there exists a set $\tilde{G}_{\beta,x} \subseteq \{s_0, s_1, \ldots, x\}$ (called dependency set) such that*

1. *If $y \in \tilde{G}_{\beta,x}$ then $\tilde{G}_{\beta,y} \subseteq \tilde{G}_{\beta,x}$*

2. *$\beta(w[s_0 \dots x])$ can be computed by querying w only on words in $\tilde{G}_{\beta,x}$*

3. *there is a polynomial p such that for every $x \in \{0,1\}^*$ the dependency set $\tilde{G}_{\beta,x}$ is printable in time $p(|x|)$.*

As noted by Strauss in [Str97], for a dense computable strategy β, $\beta(w[\lambda \dots s_i])$ can differ from $\beta(w[\lambda \dots s_{i-1}])$ for every i, as opposed to supermartingales of Section 4 which move only for $s_i \in G_d$. The transitive closure property together with the polynomial size condition, insures that given a betting strategy β, one can diagonalize against β in polynomial time.

Our second measure on BPP will be defined via the following probabilistic betting strategies.

Definition 9 *A betting strategy $\beta : \{0,1\}^* \to \mathbb{R}_+$ is BPPSV_d computable if β is dense computable and there exists a probabilistic Turing machine M such that for every $w \in \{0,1\}^*, n \in \mathbb{N}$*

$$\Pr[M^w(s_{|w|}, n) = \beta(w)] \geq 1 - 2^{-n}$$

where the probability is taken over the internal coin tosses of M and the running time of M is polynomial in $\log |w| + n$.

BPPSV_d indexed betting strategies are defined as in Section 3.

The following technical definitions of quotients of languages will be useful to prove the second measure axiom.

For a language L and a string y the quotient of L by y is

$$L/y = \{x | xy \in L\}.$$

A class A is closed under quotient if for every L in A and every string x, we have $L/x \in A$. For a class of sets $\{L_i\}_{i \geq 0}$ define its direct product by

$$\bigotimes_{i \geq 0} L_i = \{x 10^i | x \in L_i\}.$$

Quotient and direct product are essentially inverse operations. We have $(\bigotimes_j L_j)/10^i = L_i$ and $\bigotimes_i (L/10^{i-1}) = L \backslash \{0\}^*$. For convenience we write L/i for $L/10^i$ where $i \in \mathbb{N}$. Quotients can be composed and we have $(L/x)/y = L/(yx)$, for every strings $x, y \in \{0,1\}^*$. Note that the characteristic sequence $\chi_{L/y}$ is a subsequence of χ_L obtained by taking the bits indexed by an arithmetic progression of difference $2^{|y|}$.

Similarly to Strauss [Str97], we define measure zero sets as sets contained in union of small subsets.

Definition 10 *A set A is a basic null set if $A = \bigcup_{i \geq 0} A_i$ where A_i is closed under quotient for every i, and there exists an indexed BPPSV_d computable strategy β such that β_i beats A_i.*

The A_i's in Definition 10 are called subbasic null sets. We say that $B \subseteq \mathsf{BPP}$ has BPPSV_d measure zero if $B \subseteq A$ for some basic null set A.

5.2 Verifying the axioms

The third axiom holds by definition. The proof of the first axiom follows.

Theorem 7 *Let $L \in \mathsf{BPP}$ be a language. Then the singleton $\{L\}$ has BPPSV_d measure zero.*

Proof

Let $L \in \mathsf{BPP}$ be any language and let T be a probabilistic Turing machine deciding it. We show that $\{L\} \subseteq A$, where A is a subbasic null set, i.e A is closed under quotient and there is a BPPSV_d computable supermartingale d' which beats A. Consider $A = \bigcup_{i \in \mathbb{N}} L/s_i$. It is clear that A is closed under quotient. Consider the following probabilistic Turing machine M such that $M(i, \cdot)$ decides L/s_i. On input (i, x) M simply simulates T on input xs_i and outputs T's answer. Consider the following indexed supermartingale d. For $i \in \mathbb{N}$, d_i only bets on words in $\{0\}^*$ and on those words bets all its current capital according to $M(i, \cdot)$. d is BPPSV_d computable because computing $d_i(w)$ (where $|w| = N$) requires computing $a_t = M(i, 0^t)$ for $t = 1, 2, \ldots, |s_N|$ (each with error probability smaller than $2^{-s(n)}$), and checking whether $a_t = w[0^t]$ for every $t = 1, 2, \ldots, |s_N|$, in which case $d_i(w)$ equals $2^{|s_N|}$, otherwise $d_i(w)$ equals 0. The total error probability is smaller than $|s_N| \cdot 2^{-s(n)}$, which is smaller than 2^{-n} by an appropriate choice of $s(n)$. For $w \in \{0,1\}^*$ consider the following supermartingale

$$d(w) = \sum_{i \in \mathbb{N}} 2^{-i} d_i(w).$$

It is clear that $A \subseteq S^\infty[d]$. Consider the following approximation of d,

$$\hat{d}_k(w) = \sum_{i=0}^{log(|w|)+k} 2^{-i} d_i(w).$$

$\hat{d}$ is BPPSV computable, indeed the dependency set of $\hat{d}$ is in $\{0\}*$, and computing $\hat{d}_k$ requires computing $k + \log|w|$ terms d_i, each being computed with error probability smaller than $2^{-s(n)}$, which yields a total error probability smaller than 2^{-n} for an appropriate choice of $s(n)$. Since for every $i \in \mathbb{N}$ and $w \in \{0,1\}^*$ we have $d_i(w) \leq 2^{\log|w|} d_i(\lambda)$ (remember that d_i only bets on words in $\{0\}^*$), we have for every $k \in \mathbb{N}$

$$\begin{aligned}
|\hat{d}_k(w) - d(w)| &\leq \sum_{i=\log|w|+k+1}^{\infty} 2^{-i} d_i(w) \\
&\leq \sum_{i=\log|w|+k+1}^{\infty} 2^{-i} 2^{\log|w|} d_i(\lambda) \\
&\leq 2^{\log|w|} 2^{-(\log|w|+k+1)} \sum_{i=0}^{\infty} 2^{-i} \\
&\leq 2^{-k}
\end{aligned}$$

Therefore d is BPPSV approximable. By Lemma 2 there is a BPPSV and hence BPPSV_d computable supermartingale d' with the same success set as d, which implies $A \subseteq S^\infty[d']$. □

Next we prove the second measure axiom.

Theorem 8 BPP *does not have* BPPSV_d *measure zero.*

Proof

Suppose $\mathsf{BPP} \subseteq A$ where $A = \bigcup_{i\in\mathbb{N}} A_i$ is a basic null set, let β be the corresponding BPPSV_d indexed betting strategy and let M be a probabilistic Turing machine computing β. For every $i \in \mathbb{N}$ we define a language L_i such that $L_i \notin S^\infty[\beta_i]$. For $i \in \mathbb{N}$ we define L_i on word s_N as follows,

$$L_i(s_N) = 1 \iff M^{\chi_{L_i}[0\ldots N-1]1}(s_N, i, p(n)) \leq 0$$

where p is some polynomial to be determined later. $L_i \in \mathsf{BPP}$ because the number of recursive calls is smaller than $q(|s_N|)$ for some polynomial q, and each recursive call requires one computation of β_i (via M), which is performed with error probability smaller than $2^{-p(n)}$. Thus the total error probability is smaller than $q(|s_N|) \cdot 2^{-p(n)}$, which is smaller than 2^{-n} for an appropriate choice of $p(n)$.

Consider $L = \bigotimes_{i\in\mathbb{N}} L_i$. It is easy to check that $L \in \mathsf{BPP}$.

Suppose $L \in \bigcup_{i\in\mathbb{N}} A_i$, then there exists an index j such that $L \in A_j$. Since A_j is closed under quotient we have $L/j \in A_j$. Since $L/j = L_j$ we have $L_j \in A_j$ which is a contradiction. □

6 Monic Selection of Sampling

Consider the following problem called circuit acceptance probability (CAP) : On input a polynomial size Boolean circuit C with n inputs, output its probability of acceptance i.e. $\Pr_{y\in\{0,1\}^n}[C(y) = 1]$. A probabilistic polynomial Turing machine can easily approximate the acceptance probability of a circuit C by a random sampling. The point is that such a Turing machine M is in general multivalued, i.e. with high probability M will output a good approximation of the CAP problem, but M might output different values depending on the output of the random coin tosses. It is shown in [OR93] that CAP can be solved by a probabilistic two-valued probabilistic Turing machines, i.e. there is a probabilistic Turing machine which outputs with high probability a number a or b, and both values a and b are good approximation of CAP. The question whether CAP can be solved by a single valued probabilistic polynomial Turing machine is addressed in [OR93] and is known as the monic selection of random sampling (MSRS). Formally MSRS is possible if there exists a probabilistic Turing machine such that for every polynomial Boolean circuit C and every $k, n \in \mathbb{N}$, there exists a number $a \in [0, 1]$ such that

$$|a - \Pr_x[C(x) = 1]| \leq 2^{-k}, \text{ and}$$
$$\Pr[M(C, k, n) = a] \geq 1 - 2^{-n}$$

where the probability is taken over the internal coin tosses of M, and M runs in time polynomial in $|C| + k + n$.

The following results states that if MSRS is possible then the time hierarchy Theorem holds for probabilistic classes, in an abundance like form.

Theorem 9 *If MSRS is possible then*

$$\mu_{\mathsf{BPPSV}}(\mathsf{BPTIME}(O(n))) = \mu_{\mathsf{BPESV}}(\mathsf{BPP}) = \mu_{\mathsf{BPEXPSV}}(\mathsf{BPE}) = 0$$

Proof

We show that $\mu_{\mathsf{BPESV}}(\mathsf{BPTIME}(2^{cn})) = 0$ where $c > 0$. The other results are similar. Let $M_1, M_2, \ldots$ be a standard enumeration of probabilistic Turing machines running in time 2^{cn} obtained by adding an alarm clock. Consider the following BPESV computable indexed martingale d. d divides its initial capital into shares $\{s_k\}_{k>0}$ where $s_k = 1/k^2$, the idea is that the strategy associated to d_k will play share s_k against machine P_k. Suppose the strategy needs to bet on string x, then it constructs the Boolean circuit C where $C(y) = M_{k,y}(x)$. The strategy then computes a single valued approximation of the acceptance probability of C with high probability, denote this approximation by a. The strategy then bets its current capital that the membership bit of x is 1 iff $a > 1/2$. It is easy to see that d_k is a BPESV computable indexed martingale, and that whenever P_k is a $\mathsf{BPTIME}(2^{cn})$ machine $L(P_k) \subseteq S^\infty[d_k]$. Thus by Theorem 3 it follows that $\mu_{\mathsf{BPESV}}(\mathsf{BPTIME}(2^{cn})) = 0$. □

Corollary 1 *If MSRS is possible, then*

$$\mathsf{BPTIME}(O(n)) \subsetneq \mathsf{BPP} \subsetneq \mathsf{BPE} \subsetneq \mathsf{BPEXP}.$$

Next we show that every betting game of [BvMR$^+$98] can be replaced by a BPESV approximable martingale if MSRS is possible. This implies that both both the class of $\leq_T^p$-autoreducible sets and the class of $\leq_T^p$-complete sets for EXP have BPESV measure zero if MSRS is possible.

Theorem 10 *If MSRS is possible then the class of $\leq_T^p$-autoreducible sets has* BPESV *measure zero, and the class of $\leq_T^p$-complete sets for* EXP *has* BPESV *measure zero.*

Proof

It is shown in [BvMR$^+$98] that for every betting game G running in time $t(n)$ there exists a martingale d_G with the same success set i.e. $S^\infty[G] \subseteq S^\infty[d_G]$, and d_G is given by the following formula

$$d_G(w) = \mathrm{E}_{|y|=t(n)}(f_G(w, y))$$

where f_G is computable in time $O(|w| + t(n))$, where n is the size of the largest string in the domain of w. Thus by a monic selection of random sampling, the martingale d_G is probabilistically single valued computable and we have the following result.

Theorem 11 *If MSRS is possible then for every* E *betting game G there exists a* BPESV *computable martingale d_G such that $S^\infty[G] \subseteq S^\infty[d_G]$.*

It is shown in [BvMR$^+$98] the existence of E betting games winning on the class of $\leq_T^p$-autoreducible sets, and the class of $\leq_T^p$-complete sets for EXP, therefore by Theorem 11 both class have BPESV measure zero. □

7 Conclusion

We presented a measure notion on all standard probabilistic classes, which unlike former attempts satisfy all three Lutz's measure axioms. Our work links the question of monic selection of random sampling to the longstanding open problem of the existence of a time hierarchy for BPTIME classes, and to the betting games of [BvMR$^+$98]. Note that under the hypothesis that MSRS is possible, the martingales of [KL88] and [LSW98] (which require the assumption

$\mathsf{MA} \neq \mathsf{EXP}$) can also be simulated by our probabilistic martingales. Therefore it would be interesting to see under which hypothesis MSRS is possible. Since the existence of pseudorandom generator implies that MSRS is possible, it would be interesting to see whether there are some weaker assumptions than those implying the existence of pseudorandom generators that still imply that MSRS is possible.

References

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

[BDG90] J. L. Balcazar, J. Diaz, and J. Gabarro. *Structural Complexity II.* EATCS Monographs on Theorical Computer Science Volume 22, Springer Verlag, 1990.

[BDG95] J. L. Balcazar, J. Diaz, and J. Gabarro. *Structural Complexity I.* EATCS Monographs on Theorical Computer Science Volume 11, Springer Verlag, 1995.

[BvMR+98] H. Buhrman, D. van Melkebeek, K. Regan, D. Sivakumar, and M. Strauss. A generalization of resource-bounded measure, with an application. *Proc. 15th Annual Symposium on Theoretical Aspects of Computer Science*, 1373:161–171, 1998.

[IM02] R. Impagliazzo and P. Moser. A zero-one law for RP. *to be published*, 2002.

[JL93] D. Juedes and J. Lutz. The complexity and distribution of hard problems. *Proceedings of the 34th FOCS Conference*, pages 177–185, 1993.

[KL88] J. Köbler and W. Lindner. On the resource bounded measure of P/poly. *Proc. 13th Annual IEEE Conference on Computational Complexity*, pages 182–185, 1988.

[LM94] J. Lutz and E. Mayordomo. Measure, stochasticity, and the density of hard languages. *SIAM Journal on Computing*, 23:762–779, 1994.

[LSW98] W. Lindner, R. Schuler, and O. Watanabe. Resource bounded measure and learnability. *Proc. 13th Annual IEEE Conference on Computational Complexity*, pages 261–270, 1998.

[Lut90] J.H. Lutz. Category and measure in complexity classes. *SIAM Journal on Computing*, 19:1100–1131, 1990.

[Lut92] J.H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Science*, 44:220–258, 1992.

[Lut97] J.H. Lutz. The quantitative structure of exponential time. In L.A. Hemaspaandra and A.L. Selman, editors, *Complexity Theory Retrospective II*, pages 225–260. Springer, 1997.

[OR93] M. Ogiwara and K. Regan. Random polynomial time computable functions. *Manuscript*, 1993.

[Pap94] C. Papadimitriou. *Computational complexity.* Addisson-Wesley, 1994.

[RS98] K. Regan and D. Sivakumar. Probabilistic martingales and BPTIME classes. *In Proc. 13th Annual IEEE Conference on Computational Complexity*, pages 186–200, 1998.

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

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'