Results on Resource-Bounded Measure

Harry Buhrman*[1], and Stephen Fenner**[2], and Lance Fortnow***[3]

[1] Centrum voor Wiskunde en Informatica
[2] University of Southern Maine
[3] CWI & The University of Chicago

Abstract. We construct an oracle relative to which NP has p-measure 0 but D^p has measure 1 in EXP. This gives a strong relativized negative answer to a question posed by Lutz [Lut96]. Secondly, we give strong evidence that BPP is small. We show that BPP has p-measure 0 unless EXP = MA and thus the polynomial-time hierarchy collapses. This contrasts with the work of Regan et. al. [RSC95], where it is shown that P/*poly* does *not* have p-measure 0 if exponentially strong pseudorandom generators exist.

1 Introduction

Since the introduction of resource-bounded measure by Lutz [Lut92], many researchers investigated the size (measure) of complexity classes in exponential time (EXP). A particular point of interest is the *hypothesis* that NP does not have p-measure 0. Recent results have shown that many reasonable conjectures in computational complexity theory follow from the hypothesis that NP is not small (i.e., $\mu_p(\mathrm{NP}) \neq 0$), and hence it seems to be a plausible scientific hypothesis [LM96, Lut96].

In [Lut96], Lutz shows that if $\mu_p(\mathrm{NP}) \neq 0$ then BPP is low for Δ_2^P. He shows that this even follows from the seemingly weaker hypothesis that $\mu_p(\Delta_2^P) \neq 0$. He asks whether the latter assumption is weaker or equivalent to $\mu_p(\mathrm{NP}) \neq 0$. In this paper we show that, relative to some oracle, the two assumptions are *not* equivalent.

We show a relativized world where $\mathrm{D}^p = \mathrm{EXP}$ whereas NP has no P-bi-immune sets. This immediately implies, via a result of Mayordomo [May94a], that in this relativized world, NP has p-measure 0 and D^p, and hence Δ_2^P, has measure 1 in EXP, and thus does not have p-measure 0, or even p_2-measure 0.

* URL: http://www.cwi.nl/cwi/people/Harry.Buhrman.html. E-mail: buhrman@cwi.nl. Partially supported by the Dutch foundation for scientific research (NWO) by SION project 612-34-002, and by the European Union through NeuroCOLT ESPRIT Working Group Nr. 8556, and HC&M grant nr. ERB4050PL93-0516.

** URL: http://www.cs.usm.maine.edu/~fenner/. Email: fenner@cs.usm.maine.edu. Partially supported by NSF grant CCR 92-09833.

*** URL: http://www.cs.uchicago.edu/~fortnow. Email: fortnow@cs.uchicago.edu. Supported in part by NSF grant CCR 92-53582, the Dutch Foundation for Scientific Research (NWO) and a Fulbright Scholar award.

This shows in a very strong way that relativized measure for NP and P^{NP} differ: $\mu_p(\mathrm{NP}) = 0$ whereas $\mu_p(\mathrm{P}^{\mathrm{NP}[2]}) \neq 0$. Here $\mathrm{P}^{\mathrm{NP}[2]}$ is the class of sets recognized by polynomial time Turing machines that are allowed two queries to an NP oracle. We show that our results cannot be improved to $\mathrm{P}^{\mathrm{NP}[1]}$.

Secondly, we investigate the possibility that BPP does not have p-measure 0. Intuitively BPP is a feasible complexity class close to P and therefore it should be the case that BPP is small. We give very strong evidence supporting this intuition. We show that $\mu_p(\mathrm{BPP}) = 0$ unless EXP = MA and thus the polynomial-time hierarchy collapses.

Since BPP $\subseteq$ P/*poly* our result contrasts with the one by Regan, Sivakumar and Cai [RSC95], where it is shown that $\mu_p(\mathrm{P}/poly) \neq 0$, unless exponentially strong pseudorandom generators do not exist.

2 Preliminaries

We let $\Sigma = \{0, 1\}$ and identify strings in Σ^* with natural numbers via the usual binary representation. We fix $N_1, N_2, \ldots$ to be a standard enumeration of all nondeterministic polynomial-time oracle Turing machines (NOTMs), where for each i and input of length n, N_i runs in time n^i for all oracles. All our machines run using symbols 0, 1 and blanks. Fix a deterministic oracle TM M which accepts some standard $\leq_m^p$-complete language for EXP^A for all $A \subseteq \Sigma^*$. We may assume that M runs in time 2^n. We let $\langle \cdot, \cdot \rangle$ be the standard pairing function, and we note that $x, y \leq \langle x, y \rangle$ for all $x, y \in \Sigma^*$. A set is in D^p if it can be expressed as the difference of two sets in NP.

The notations $\mathcal{R}$, $\mathcal{Q}$, $\mathcal{R}^+$ and $\mathcal{Q}^+$ denote the real numbers, the rational numbers, the positive real numbers and the positive rational numbers respectively.

2.1 Resource Bounded Measure

Classical Lebesque measure is an unusable tool in complexity classes. As these classes are all countable, everything we define in such a class has measure 0. Yet, we might wish to have a notion of "abundance" and "randomness" in complexity classes. Lutz [Lut87, Lut90] introduced the notion of *resource bounded measure*, and gave a tool to talk about these notions inside complexity classes.

Definition 1. A *martingale* d is a function from Σ^* to $\mathcal{R}^+$ with the property that $d(w0) + d(w1) = 2d(w)$ for every $w \in \Sigma^*$.

Definition 2. A p-martingale is a martingale $d : \Sigma^* \mapsto \mathcal{Q}^+$ that is polynomial time computable.

Definition 3. A martingale d *succeeds* on a language A if

$$\limsup_{n \mapsto \infty} d(\chi_A[0 \ldots n-1]) = +\infty$$

We write $S^\infty[d] = \{A \mid d \text{ succeeds on } A\}$

Definition 4. Let $\mathcal{X}$ be a class of languages.

- $\mathcal{X}$ has p-measure 0 ($\mu_p(\mathcal{X}) = 0$) iff there exists a p-martingale d such that $\mathcal{X} \subseteq S^\infty[d]$.
- $\mathcal{X}$ has p-measure 1 ($\mu_p(\mathcal{X}) = 1$) iff $\mu_p(\overline{\mathcal{X}}) = 0$
- $\mathcal{X}$ has p-measure 0 in EXP ($\mu_p(\mathcal{X}|\mathrm{EXP}) = 0$) iff $\mu_p(\mathcal{X} \cap \mathrm{EXP}) = 0$
- $\mathcal{X}$ has p-measure 1 in EXP ($\mu_p(\mathcal{X}|\mathrm{EXP}) = 1$) iff $\mu_p(\overline{\mathcal{X}} \cap \mathrm{EXP}) = 0$

One often defines measure in EXP using p_2-measure where the martingale can use $2^{\log^{O(1)} n}$ time. All of our results also hold in this weaker model.

3 Measure of NP versus Measure of P^{NP}

In this section we concentrate on the question posed by Lutz [Lut96]. We show that relative to some oracle $\mu_p(\mathrm{NP}) = 0$ does not imply that $\mu_p(\mathrm{P}^{\mathrm{NP}}) = 0$. We do this in a very strong way by constructing an oracle such that NP does not contain P-bi-immune sets and $\mathrm{D}^p = \mathrm{EXP}$.

Theorem 5. *There exists an oracle A such that, relative to A,* NP *has no* P*-bi-immune sets and* $\mathrm{D}^p = \mathrm{EXP}$.

Proof. We will code EXP into D^p on one "side" of the oracle and prevent P-bi-immunity on the other, i.e., strings in $\Sigma^*0 = \{x0 \mid x \in \Sigma^*\}$ will be used to code EXP into D^p, while strings in $\Sigma^*1 = \{x1 \mid x \in \Sigma^*\}$ will code the information to find an infinite subset of each NP set or its complement. Some diagonalization will also be necessary to force certain NP computations.

To mix coding with diagonalization, we employ a simplified version of the trick used to construct an oracle for $\mathrm{P}^{\mathrm{NP}} = \mathrm{NEXP}$ [BT94, FF95]. For each x, we reserve two potential regions—*left* and *right*—in which to code $M^A(x)$, only one of which will actually be used. To code correctly in a region we must let exactly one string in the region enter A. We will code in the left region unless we have to diagonalize against some NP machine, which may necessitate adding several strings of the left region to A. If this happens, we scrap the left region and code in the right region, but we can do this only if our diagonalization hasn't already put strings of the *right* region into A.

We now proceed with the formal treatment. For every $x \in \Sigma^*$ with $|x| = n$ and $b \in \Sigma$, we call s an (x, b, left)*-coding string* (respectively, an (x, b, right)*-coding string*) if $s = xyb00$ (respectively, $s = xyb10$) for some $y \in \Sigma^*$ of length $3n$. We identify *left* and *right* with 0 and 1, respectively. We build the oracle A in stages, each successive stage extending a finite portion of A's characteristic function. If $\alpha: \Sigma^* \to \Sigma$ is some partial characteristic function, N an oracle machine, and $x \in \Sigma^*$, then the computation $N^\alpha(x)$ is defined as usual, except that when N makes any query outside domain(α), it is answered negatively. As is customary, we regard α as a set of ordered pairs. If β is another characteristic function, we write $\beta \succeq \alpha$ to mean that β extends α. Finally, define the "tower of 2's" function $t(n)$ for $n \geq 0$ by

$$\begin{aligned} t(0) &= 1 \\ t(n+1) &= 2^{t(n)}. \end{aligned}$$

Stage -1.
$\alpha_{-1}{:=}\emptyset$.
End Stage.

Stage $n \geq 0$.
We are given α_{n-1}. Set $\alpha{:=}\alpha_{n-1}$.

1. (*Forcing an* NP *computation*) If $n \neq t(k)$ for any k, then set

$$d_n{:=} \begin{cases} \text{right if } \alpha(s) = 1 \text{ for some } (x, b, \text{left})\text{-coding string } s \text{ with } |x| = n, \\ \text{left} \quad \text{otherwise,} \end{cases}$$

and go to step 2. Otherwise, let $n = t(k)$ for some $k = \langle i, j \rangle$. If there exists a minimal $\beta \succeq \alpha$ such that both
(a) $N_i^\beta(0^n)$ has an accepting path in which all queries are in domain(β), and
(b) for no x with $|x| \geq n$ and no (x, b, right)-coding string s does $\beta(s) = 1$,
then set $\alpha{:=}\beta \cup \{(0^{n^i}1, 1)\}$ and set $d_n{:=}$right (note that β is only defined on strings no longer than n^i). Otherwise, set $\alpha{:=}\alpha \cup \{(0^{n^i}1, 0)\}$ and set $d_n{:=}$left.

2. (*Preserving computations of M*) For all x of length n, run $M^\alpha(x)$, and extend α with just enough 0's to "cover" all queries made by $M^\alpha(x)$ not in domain(α).
3. (*Coding computations of M*) For all $x \in \Sigma^*$ of length n, let $y \in \Sigma^*$ be the lexicographically least string (if one exists) such that $|y| = 3n$ and neither the $(x, 0, d_n)$-coding string nor the $(x, 1, d_n)$-coding string corresponding to y is in domain(α). If M^α accepts, set $\alpha{:=}\alpha \cup \{(xy1d_n0, 1)\}$; otherwise, set $\alpha{:=}\alpha \cup \{(xy0d_n0, 1)\}$.
4. Set α_n to be α extended with just enough 0's to cover all remaining (x, b, d)-coding strings for all $b \in \Sigma$, $d \in \{\text{left}, \text{right}\}$, and x of length n.

End Stage.

Let A be such that χ_A extends α_n for all n ($\chi_A(x) = 0$ for any $x \notin \bigcup_n \alpha_n$). For any $B \subseteq \Sigma^*$, define the language L^B by

$$L^B(x) = \begin{cases} 1 & \begin{array}{l} \text{if either } B \text{ contains an } (x, 1, \text{right})\text{-coding string, or} \\ B \text{ contains no } (x, 0, d)\text{-coding strings for any } d \in \{\text{left}, \text{right}\}, \end{array} \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, $L^B \in \text{coD}^{p,B}$. We now show that $L^A(x) = M^A(x)$ for all $x \in \Sigma^*$, and hence $\text{coD}^{p,A} = \text{EXP}^A = \text{D}^{p,A}$.

Pick an n large enough, and fix an input x of length n. In Step 3 of Stage n, such a y must exist: there are at most $2^n \cdot (2^{n+1} - 1)$ (x, b, d)-coding strings queried by M on inputs of length $\leq n$, because of the running time of M, and less than $n \cdot n^{\log^* n} < 2^{(\log n)^2}$ total strings queried by the N_i in Step 1 of Stages 0 through n. Thus there are less than 2^{3n} (x, b, d)-coding strings in domain(α) at Step 3 of Stage n.

The fact that

$$M^A(x) = L^A(x) \tag{1}$$

is now easily seen: first we observe that no (x, b, right)-coding string (for any $b \in \Sigma$) gets into A in Steps 1 or 2 of any stage. Thus we have two cases:

d_n = left: For any $b \in \Sigma$ and $d \in \{\text{left}, \text{right}\}$, the only (x, b, d)-coding string that ever enters A does so in Step 3 of Stage n. This unique string is an $(x, 1, \text{left})$-coding string if $M^A(x)$ accepts, and is otherwise an $(x, 0, \text{left})$-coding string; thus, (1) is satisfied.

d_n = right: Exactly one (x, b, right)-coding string enters A. It is an $(x, 1, \text{right})$-coding string iff $M^A(x)$ accepts. Again, (1) is satisfied.

It remains to show that NP^A has no P^A-bi-immune sets. This will be done if we can show that for any $L \in \mathrm{NP}^A$, there exist P^A sets Q and R with Q infinite, such that $L \cap Q = R$ (or at least the symmetric difference of $L \cap Q$ and R is finite). Let $L = L(N_i^A)$ for some fixed i. Let

$$Q = \{0^n \mid (\exists j) n = t(\langle i, j\rangle)\},$$
$$R = Q \cap \{0^n \mid 0^{n^i}1 \in A\}.$$

The sets Q and R are clearly in P^A. Pick $n = t(\langle i, j\rangle)$ for j large enough so that $t(\langle i, j\rangle + 1) = 2^n > n^i$, and consider Step 1 of Stage n. If β exists, then $N_i^A(0^n)$ accepts and $0^{n^i}1 \in A$, so $0^n \in R$. If no such β exists, then $0^n \notin R$. To see that $N_i^A(0^n)$ rejects, we simply observe that $d_n = d_{n+1} = \cdots = d_{n^i-1} = d_{n^i} =$ left, so no (x, b, right)-coding strings enter A in any of the stages n through n^i. Therefore, A preserves our conditions on the nonexistence of β, and so $N_i^A(0^n)$ rejects.

Corollary 6. *There exists an oracle relative to which* NP *has p-measure 0 and* D^p = EXP *(and thus has p-measure 1 in* E *and in* EXP*).*

We actually get something more from the construction above: relative to A, we have EXP $\subseteq$ (NP $\cap$ coNP)/1. That is, EXP can be computed in NP $\cap$ coNP with one bit of advice for strings of length n, namely d_n. On input x of length n, an NP^A machine accepting $L(M^A)$ (respectively $\overline{L(M^A)}$) simply checks if there is some $(x, 1, d_n)$-coding string (respectively, some $(x, 0, d_n)$-coding string) in A.

A natural question is whether Theorem 5 and Corollary 6 are tight. It could still happen that $\mu_p(\mathrm{NP}) = 0$ and $\mu_p(\mathrm{P}^{\mathrm{NP}[1]}) \neq 0$. The next theorem discards this possibility.

Theorem 7. *If* $\mu_p(\mathrm{P}^{\mathrm{NP}[1]}) \neq 0$ *then* $\mu_p(\mathrm{NP}) \neq 0$.

Proof. $\mu_p(\mathrm{P}^{\mathrm{NP}[1]}) \neq 0$ implies that SAT is weakly $\leq_{1tt}^p$-complete for EXP. Ambos-Spies, Mayordomo, and Zheng [ASMZ96] have shown that the weakly $\leq_{1tt}^p$-completeness notion coincides with weakly $\leq_m^p$-completeness for EXP. Hence SAT is weakly $\leq_m^p$-complete for EXP and thus $\mu_p(\mathrm{NP}) \neq 0$.

Corollary 8. *Relative to the oracle constructed in Theorem 5 it holds that* $\mathrm{D}^p = \mathrm{coD}^p \neq \mathrm{P}^{\mathrm{NP}[1]}$.

4 BPP likely has measure 0

In this section we investigate the consequences of BPP not having p-measure 0. We will see that this is unlikely since it would collapse the polynomial-time hierarchy. Hence we provide strong evidence that $\mu_p(\mathrm{BPP}) = 0$.

Theorem 9. *If* $\mu_p(\mathrm{BPP}) \neq 0$ *then* EXP = MA.

Since MA $\in \Sigma_2^p \cap \Pi_2^p$ [BM89], EXP = MA implies that PH = Σ_2^p.

We use the following Theorem from Babai, Fortnow, Nisan and Wigderson [BFNW93] stating that if EXP $\neq$ MA then BPP can be simulated in subexponential time for infinitely many input lengths.

Theorem 10 [BFNW93]. *If* EXP $\neq$ MA *then for all* $L \in$ BPP, *and for all* ϵ *there exists a set* $L' \in \mathrm{DTIME}(2^{n^\epsilon})$ *such that for infinitely many* n, $L \cap \Sigma^n = L' \cap \Sigma^n$.

We will see that if BPP can be simulated in subexponential time for infinitely many input lengths, then it has p-measure 0. Taking this together with Theorem 10 yields that EXP $\neq$ MA implies that $\mu_p(\mathrm{BPP}) = 0$, which proves Theorem 9.

Theorem 11. *If for all languages* $L \in$ BPP *there exists an* $\epsilon < 1$ *and a set* $L' \in \mathrm{DTIME}(2^{n^\epsilon})$ *such that for infinitely many* n, $L \cap \Sigma^n = L' \cap \Sigma^n$, *then* $\mu_p(\mathrm{BPP}) = 0$.

Proof. (Sketch) We will construct a martingale that succeeds on all sets in BPP that runs in time n^k for some fixed k. Let $L \in$ BPP and let $M_{L'}$ be the machine that runs in subexponential time and accepts L'. If we are betting on strings of length n such that $L \cap \Sigma^n = L' \cap \Sigma^n$ then we can use $M_{L'}$ to predict exactly the next bit, and hence we win 2^n times. The problem however is that we do not know for which n, $M_{L'}$ is going to be correct. We overcome this problem by the following strategy.

Assume that our initial capital is 1. We reserve 2^{-n} to bet against the strings of length n, using $M_{L'}$ to predict the next bit (i.e. whether the next string of length n is in L'). We bet everything won so far on the strings of length n to the outcome of $M_{L'}$. At the last string of length n we set aside what (if any) we have won betting on the strings of length n.

Observe that if n is a length such that $L \cap \Sigma^n = L' \cap \Sigma^n$ then we win $2^{2^n} * 2^{-n}$ and this is greater than n. So for infinitely many n we add n to our capital and hence the lim-inf of this martingale goes to infinity.

To make the construction work uniformly for all $L \in$ BPP we simulate all the DTIME(2^n) machines with a single DTIME(2^{2n}) machine allocating 2^{-i} of our initial capital to machine i (see [Lut92, May94b]).

Acknowledgment

We thank Leen Torenvliet for comments on an earlier version and Dieter van Melkebeek for helpful discussions on the writeup of the proof of Theorem 11.

References

[ASMZ96] K. Ambos-Spies, E. Mayordomo, and Xizhong Zheng. A comparison of weak completeness notions. In *Proeceedings of Eleventh Annual Conference on Computational Complexity*, pages 171 - 178, 1996.

[BFNW93] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[BM89] László Babai and Shlomo Moran. Proving properties of interactive proofs by a generalized counting technique. *Information and Computation*, 82(2):185-197, August 1989.

[BT94] Buhrman and Torenvliet. On the cutting edge of relativization: The resource bounded injury method. In *Annual International Colloquium on Automata, Languages and Programming*, pages 263–273, 1994.

[FF95] S. Fenner and L. Fortnow. Beyond $P^{NP} = NEXP$. In *STACS 95*, volume 900 of *Lecture Notes in Computer Science*, pages 619–627. Springer, 1995.

[LM96] J. Lutz and E. Mayordomo. Cook versus Karp-Levin: Separating completeness notions if NP is not small. *Theoretical Computer Science*, 164(1-2):141–163, 1996.

[Lut87] J. Lutz. *Resource-Bounded Category and Measure in Exponential Complexity Classes.* PhD thesis, Department of Mathematics, California Institute of Technology, 1987.

[Lut90] J. Lutz. Category and measure in complexity classes. *SIAM J. Comput.*, 19(6):1100–1131, December 1990.

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

[Lut96] J. Lutz. Observations on measure and lowness for Δ_2^P. In *STACS 96*, volume 1046 of *Lecture Notes in Computer Science*, pages 87 - 98. Springer, 1996.

[May94a] E. Mayordomo. Almost every set in exponential time is p-bi-immune. *Theoretical Computer Science*, 136(2):487-506, 1994.

[May94b] E. Mayordomo. *Contributions to the study of resource-bounded measure.* PhD thesis, Universitat Politècnica de Catalunya, 1994.

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