Electronic Colloquium on Computational Complexity, Report No. 15 (2002)

ZPP is hard unless RP is small

Philippe Moser*

Abstract

We use Lutz's resource bounded measure theory to prove that either **RP** is small or **ZPP** is hard. More precisely we prove that if **RP** has not p-measure zero, then **EXP** equals **ZPP** on infinitely many input lengths, i.e. there are infinitely many input lengths on which **ZPP** is hard. Second we prove that if **NP** has not p-measure zero, then derandomization of **AM** is possible on infinitely many input length, i.e. there are infinitely many input lengths such that **NP** = **AM**. Finally we prove easiness versus randomness tradeoffs for classes in the polynomial time hierarchy. We show that it appears to every strong adversary that either, every $\mathbf{\Sigma}_i^{\mathbf{P}}$ algorithm can be simulated infinitely often by a subexponential co-nondeterministic time algorithm having oracle access to $\mathbf{\Sigma}_{i-2}^{\mathbf{P}}$, or $\mathbf{BP\Sigma}_i^{\mathbf{P}} = \mathbf{\Sigma}_i^{\mathbf{P}}$.

1 Introduction

Not much is known about the relationship between **ZPP**, **RP** and **BPP**, except the trivial inclusions $\mathbf{ZPP} \subseteq \mathbf{RP} \subseteq \mathbf{BPP}$. For instance it is not known whether **RP** being easy implies the easiness of **BPP**. A similar relation between **ZPP** and **RP** is also unknown. In fact there are relativized worlds in which $\mathbf{P} = \mathbf{RP}$, but $\mathbf{P} \neq \mathbf{BPP}$ [MV96]. The question whether assuming $\mathbf{P} = \mathbf{ZPP}$ yields any non-trivial easiness result for **RP** also remains open.

In this paper, we use Lutz measure theory [Lut97], to prove a relationship between the easiness of **ZPP** and **RP**, by showing that either **RP** is small, or **ZPP** is hard. More precisely we prove that if **RP** has not p-measure zero, then **EXP** is equal to **ZPP** infinitely often, i.e. **RP** being hard implies **ZPP** being hard.

One ingredient used in our proof is the easy witness technique from [Kab00]. We simulate a given **RP** algorithm using truth table of easy functions instead of purely random strings. If the simulation works, we can construct a martingale that succeeds on **RP**, which implies that **RP** has p-measure zero. On the other hand, if the simulation fails, we get a hardness test, which can be used to guess a hard function, yielding a pseudorandom generator by using Impagliazzo and Widgerson's [IW97] result. This combined with Melkebeek's zero-one law for **BPP** [Mel00], yields that **ZPP** is hard.

The derandomization of complexity classes beyond **BPP**, such as **AM**, has just started to be studied. Klivans and Melkebeek [KvM99] showed that the Nissan-Widgerson [NW94] approach relativizes to any oracles, and gave conditional derandomization results for **AM**. Miltersen and Vinodchandran [MV99] used hitting sets to derandomize **AM**, under weaker assumptions than Klivans and Melkebeek [KvM99].

*Address: Computer Science Department, University of Geneva. Email: moser@cui.unige.ch

ISSN 1433-8092

In [AK97] V. Arvind and J. Köbler proved that under the assumption that $\mathbf{NP}$ has not p-measure zero, partial derandomization of $\mathbf{AM}$ was possible. More precisely they proved that $\mathbf{NP}$ not having p-measure zero implies $\mathbf{NP}/\log n = \mathbf{AM}$. Using different techniques, we show that the assumption $\mathbf{NP}$ has not p-measure zero implies that $\mathbf{NP} = \mathbf{AM}$ infinitely often. Thus we get rid of the nonuniformity in V. Arvind and J. Köbler's result [AK97], but on the other hand we only obtain derandomization for infinitely many input lengths.

Finally following Lu [Lu00], we show a similar result for classes in the polynomial time hierarchy. But instead of using pseudorandom generators as in [Lu00], we use the hitting set construction of Miltersen and Vinodchandran [MV99]. Thus we gain one level in the polynomial hierarchy, enabling us to prove that it appears to every nondeterministic adversary, having oracle access to $\mathbf{\Sigma}^{\mathbf{P}}_{i-1}$, that either every $\mathbf{\Sigma}^{\mathbf{P}}_{i}$ algorithm can be simulated by a co-nondeterministic subexponential time algorithm, having oracle access to $\mathbf{\Sigma}^{\mathbf{P}}_{i-2}$, infinitely often, or $\mathbf{BP\Sigma}^{\mathbf{P}}_{i} = \mathbf{\Sigma}^{\mathbf{P}}_{i}$.

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]. The polynomial hierarchy is the following sequence of classes: First, $\mathbf{\Sigma}^{\mathbf{P}}_{0} = \mathbf{\Pi}^{\mathbf{P}}_{0} = \mathbf{P}$, and for all $i \geq 0$, $\mathbf{\Sigma}^{\mathbf{P}}_{i+1} = \mathbf{NP}^{\mathbf{\Sigma}^{\mathbf{P}}_{i}}$ and $\mathbf{\Pi}^{\mathbf{P}}_{i+1} = \mathbf{coNP}^{\mathbf{\Sigma}^{\mathbf{P}}_{i}}$. We denote by QSAT_i the standard $\mathbf{\Sigma}^{\mathbf{P}}_{i}$-complete language. For $i \geq 2$, the two-sided probabilistic version of $\mathbf{\Sigma}^{\mathbf{P}}_{i}$ is equal to the one-sided error version. A proof of this result can be found in [BDG90].

Proposition 1 ([BDG90]) *Let $L \in \mathbf{BP\Sigma}^{\mathbf{P}}_{i}$, then there exists a relation $M \in \mathbf{\Sigma}^{\mathbf{P}}_{i}$, and a polynomial $m(n)$ such that for all $x \in \{0,1\}^n$,*

$$\begin{aligned} x &\in L \Rightarrow \Pr_{y \in \{0,1\}^m}[(x,y) \in M] = 1 \text{ , and} \\ x &\notin L \Rightarrow \Pr_{y \in \{0,1\}^m}[(x,y) \in M] \leq \frac{1}{4} \ . \end{aligned}$$

2.1 Refuters and Pseudo Classes

Let A be any language. We say that a multi-valued nondeterministic procedure, with oracle access to A produces some object, if there is a nondeterministic oracle Turing machine M^A with oracle access to A such that at the end of its computation, each nondeterministic branches either produces the desired object, or is marked with reject, and at least one of the branches produces the desired object.

Kabanets [Kab00] introduced the concept of refuters. A refuter is a length preserving Turing machine R such that on input 1^n, R outputs a string of length n. We will consider multi-valued nondeterministic refuters with oracle access to $\mathbf{\Sigma}^{\mathbf{P}}_{i}$.

Definition 1 *A multi-valued nondeterministic refuter with oracle access to $\mathbf{\Sigma}^{\mathbf{P}}_{i}$ (abbreviated $\mathbf{NPMV}^{\mathbf{\Sigma}^{\mathbf{P}}_{i}}$) is a nondeterministic polynomial time Turing machine M with oracle access to $\mathbf{\Sigma}^{\mathbf{P}}_{i}$, that on input 1^n, nondeterministically produces a string in $\{0,1\}^n$.*

Kabanets [Kab00] introduced zero-error probabilistic refuters, i.e. refuters that halt with great probability, and which whenever they halt, output a string. The class of such refuters will be denoted by **FZPP**.

For a complexity class of languages $\boldsymbol{C}$, we will consider the class of languages indistinguishable (for a certain class of refuters) from languages of $\boldsymbol{C}$. We say that two languages L and M are indistinguishable for a certain class of refuters $\mathcal{R}$, if, for every refuter $R \in \mathcal{R}$, we have that $R(1^n) \notin L \triangle M$ for almost every n, where $L \triangle M$ denotes the symmetric difference of L and M.

Definition 2 *Let $\boldsymbol{C}$ be a complexity class of languages, and let A be any language. We define :* $[\text{pseudo}_{\mathbf{NPMV}^A}]\boldsymbol{C} = \{L \subseteq \{0,1\}^* | \exists M \in C$ *such that any multi-valued nondeterministic refuter with oracle access to A fails to distinguish L_n from M_n for all but finitely many n*$\}$.

Analogously to definition 2, we define the class i.o.$[\text{pseudo}_{\mathbf{NPMV}^A}]\boldsymbol{C}$, which is the class of languages equal to languages of $[\text{pseudo}_{\mathbf{NPMV}^A}]\boldsymbol{C}$ on infinitely many words lengths. If the refuters are zero-error probabilistic Turing machines, we denote by $[\text{pseudo}_{\mathbf{FZPP}^A}]\boldsymbol{C}$ and i.o.$[\text{pseudo}_{\mathbf{FZPP}^A}]\boldsymbol{C}$ the corresponding pseudo classes.

2.2 Hitting Sets

We will need the following definitions of nondeterministic and co-nondeterministic circuits.

Definition 3 *A nondeterministic Boolean circuits C contains, in addition to the usual AND, OR and NOT gates, choice gates of fan-in* 0. *The circuit evaluates to 1 on input x, and we say that* $C(x) = 1$, *if there is some assignment of truth values to the choice-gates that makes the circuit evaluate to 1. Otherwise* $C(x) = 0$.

A co-nondeterministic circuit C is defined similarly: The circuit evaluates to 0 on input x, and we say that $C(x) = 0$, if there is some assignment of truth values to the choice-gates that makes the circuit evaluate to 0. Otherwise $C(x) = 1$.

Similarly, a single-valued (abbreviated SV) nondeterministic circuit C computing a function f has, in addition to its usual output, an extra output bit, called the flag. For any input x, and any setting of the choice-gates, if the flag is on, the circuit should output the correct value of $f(x)$. Furthermore, for any x, there should be some setting of the choice-gates that turn the flag on.

Pseudorandom generators are used to derandomize two-sided error algorithms. The tool used to derandomize one-sided error algorithms is a hitting set. In particular, we will need hitting sets for co-nondeterministic circuits.

Definition 4 *Let A be any language. A subset $H \subseteq \{0,1\}^n$ is a $\frac{1}{2}$ -hitting set for Boolean co-nondeterministic circuits of size $s(n)$, with oracle gates to A, if for any such circuit C^A on n inputs, the following holds:*

If $\Pr_{x \in \{0,1\}^n}[C^A(x) = 1] \geq \frac{1}{2}$, *then* $\exists h \in H$ *such that* $C^A(h) = 1$.

The following hardness-randomness tradeoffs are from [MV99]

Theorem 1 (Miltersen, Vinodchandran) *Let A be any language. For any $\epsilon > 0$, there is a $\gamma > 0$ so that the following holds. There is a deterministic polynomial time procedure which, given as input the truth table of a $\log m$-variables Boolean function $f : \{0,1\}^{\log m} \longrightarrow \{0,1\}$ with circuit complexity greater than m^ϵ for nondeterministic oracle circuits with oracle gates for A, outputs a hitting set in $\{0,1\}^n$, with threshold $\frac{1}{2}$, for co-nondeterministic oracle circuits having oracle access to A of size n, where $n = m^\gamma$.*

Thanks to Proposition 1 producing a hitting set with a multi-valued nondeterministic procedure is enough to derandomize $\mathbf{BP\Sigma_i^P}$, more precisely:

Proposition 2 *If there is a multi-valued nondeterministic procedure with oracle access to $\mathbf{\Sigma_{i-1}^P}$ which on input 1^n outputs a hitting set in $\{0,1\}^n$, with threshold $\frac{1}{2}$, for co-nondeterministic oracle circuits with oracle gates for $\mathbf{\Sigma_{i-1}^P}$, of size n, then $\mathbf{BP\Sigma_i^P} = \mathbf{\Sigma_i^P}$.*

2.3 p-measure

In this section we describe the fragment of Lutz's measure theory for the class **EXP** that we will need. For a more detailed presentation of this theory we refer the reader to the survey by Lutz [Lut97].

The measure on **EXP** is obtained by imposing appropriate resource-bound on a game theoretical characterization of the classical Lebesgue measure.

A martingale is a function $d : \{0,1\}^* \to [0, \infty[$ such that,

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

for every $w \in \{0,1\}^*$. d is a p-martingale if d is computable in time polynomial in $|w|$.

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. Denote by $s_0, s_1, \ldots$ the enumeration of all Boolean strings in lexicographic order. 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 $\alpha_w d(w)$ of his current capital $d(w)$, that the nth word $s_n \in A$, and bets the remaining capital $(1 - \alpha_w)d(w)$ 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. 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.

Definition 5 *A class $\boldsymbol{C}$ has p-measure zero if there is a single p-martingale d that succeeds on every language A of $\boldsymbol{C}$.*

This property is monotone in the following sense: If class $\boldsymbol{D}$ is contained in a class $\boldsymbol{C}$ of p-measure zero, then $\boldsymbol{D}$ also has p-measure zero.

Definition 6 *A class $\boldsymbol{C}$ has p-measure one if the complement of $\boldsymbol{C}$ has p-measure zero.*

Lutz showed in [Lut92] that the class **E** does not have p-measure zero, which he called the measure conservation property. Since finite unions of null classes is a null class, it's impossible for a class to have both measure zero and one.

Lutz also proved in [Lut92] that "easy" infinite union of null classes is null.

Theorem 2 (Lutz) *Suppose $\{d_i\}_{i\geq 1}$ is a set of martingales, each covering class $\boldsymbol{C}_i$; where $d(i, w) := d_i(w)$ is computable in time $q = (i, |w|)$ for a certain polynomial q. Then $\cup_{i\geq 1}\boldsymbol{C}_i$ has p-measure zero.*

We will need the following zero-one law for **BPP**, stating that **BPP** has either measure zero or one.

Theorem 3 (Melkebeek [Mel00]) **BPP** *has either p-measure zero or else has p-measure one.*

3 ZPP is hard unless RP is small

The following result shows that no derandomization of **ZPP** is possible unless **RP** is small. More precisely it states that **ZPP** is as hard as **EXP** on infinitely many input lengths, unless **RP** has p-measure zero.

Theorem 4 **EXP** = **ZPP** *on infinitely many input lengths, unless* **RP** *has p-measure zero.*

Proof

It is easy to see that the results in [Kab00] yield the following Theorem.

Theorem 5 *At least one of the following statements holds.*

1. $\forall\epsilon > 0$, $\mathbf{RP} \subseteq [\text{pseudo}_{\mathbf{FZPP}}]\mathbf{DTIME}(2^{n^\epsilon})$, *for a.e. length n or,*
2. $\mathbf{BPP} =_{i.o.} \mathbf{ZPP}$.

Now suppose the first statement of Theorem 5 holds. Taking $\epsilon = 1$, we have that for every language $A \in \mathbf{RP}$, there exists a language $B \in [\text{pseudo}_{\mathbf{FZPP}}]\mathbf{DTIME}(2^n)$, such that every **FZPP** refuter fails to distinguish A from B; i.e. for every **FZPP** refuter R and for a.e. length n, there is a y produced by $R(1^n)$, such that $A(y) = B(y)$.

So let A be any language in **RP** and let B be as above. The refuter $R(1^n) := 1^n$ is a **FZPP** refuter, therefore we have,

$$A(1^n) = B(1^n) \text{ for a.e. } n. \tag{1}$$

Consider the following martingale d_B, that for each length n, only bets on the membership bit of the string 1^n, and for each of these strings puts $\frac{1}{2}$ of its capital on the outcome that the membership is the same as for B. To compute $d_B(w)$, where w is the characteristic sequence of some language for words up to length t, one only needs to compute whether $1^n \in B$ for $n = 1, 2, \ldots, t$. Since $B \in \mathbf{DTIME}(2^n)$, d_B is computable in time $t2^t \leq |w|^2$. Moreover if A is a language such that 2 holds, then d_B doubles its capital on a.e. bets it makes, and loses a fraction of its capital on finitely many bets. Therefore $d_B(\chi_A) = \infty$.

Denote by L_B the class of languages A such that equation 2 holds.

Let $M_1, M_2, \ldots$ be a standard enumeration of Turing machines running in deterministic time 2^n, where M_i runs in time polynomial in $i + 2^n$, and denote by B_i the language decided by M_i. The martingale defined by $d(i, w) = d_{B_i}(w)$ is computable in time polynomial in $i + |w|$. Therefore the class $\boldsymbol{C} = \bigcup_{i \geq 1} L_i$ has p-measure zero, by Theorem 2. Since $\mathbf{RP} \subseteq \boldsymbol{C}$, $\mathbf{RP}$ has p-measure zero. Now if $\mu_p(\mathbf{RP}) \neq 0$, we have that $\mathbf{BPP} =_{i.o.} \mathbf{ZPP}$. Moreover, since $\mathbf{RP}$ has not p-measure zero, $\mathbf{BPP}$ has not p-measure zero, therefore $\mathbf{EXP} = \mathbf{BPP}$, by Theorem 3. Therefore $\mu_p(\mathbf{RP}) \neq 0$ implies $\mathbf{EXP} =_{i.o.} \mathbf{ZPP}$, which ends the proof.

□

We also obtain a similar result for **AM**, namely that the assumption **NP** has not p-measure zero, implies derandomization of **AM**.

Theorem 6 **NP** = **AM** *on infinitely many input lengths, unless* **NP** *has p-measure zero.*

Proof

It is easy to see that the results in [Lu00] yield the following Theorem.

Theorem 7 *At least one of the following statements holds.*

1. $\forall \epsilon > 0$, $\mathbf{NP} \subseteq [\text{pseudo}_{\mathbf{NPMV}}]\mathbf{DTIME}(2^{n^\epsilon})$, *for a.e. length n or,*

2. $\mathbf{AM} =_{i.o.} \mathbf{NP}$.

Now suppose the first statement of Theorem 7 holds. Taking $\epsilon = 1$, we have that for every language $A \in \mathbf{NP}$, there exists a language $B \in [\text{pseudo}_{\mathbf{NPMV}}]\mathbf{DTIME}(2^n)$, such that every **NPMV** refuter fails to distinguish A from B; i.e. for every **NPMV** refuter R and for a.e. length n, there is a y produced by $R(1^n)$, such that $A(y) = B(y)$.

So let A be any language in **NP** and let B be as above. The refuter $R(1^n) := 1^n$ is a **NPMV** refuter, therefore we have,

$$A(1^n) = B(1^n) \text{ for a.e. } n. \tag{2}$$

Consider the following martingale d_B, that for each length n, only bets on the membership bit of the string 1^n, and for each of these strings puts $\frac{1}{2}$ of its capital on the outcome that the membership is the same as for B. To compute $d_B(w)$, where w is the characteristic sequence of some language for words up to length t, one only needs to compute whether $1^n \in B$ for $n = 1, 2, \ldots, t$. Since $B \in \mathbf{DTIME}(2^n)$, d_B is computable in time $t2^t \leq |w|^2$. Moreover if A is a language such that 2 holds, then d_B doubles its capital on a.e. bets it makes, and loses a fraction of its capital on finitely many bets. Therefore $d_B(\chi_A) = \infty$.

Denote by L_B the class of languages A such that equation 2 holds.

Let $M_1, M_2, \ldots$ be a standard enumeration of Turing machines running in deterministic time 2^n, where M_i runs in time polynomial in $i + 2^n$, and denote by B_i the language decided by M_i. The martingale defined by $d(i, w) = d_{B_i}(w)$ is computable in time polynomial in $i + |w|$. Therefore the class $\boldsymbol{C} = \bigcup_{i \geq 1} L_i$ has p-measure zero, by Theorem 2. Since $\mathbf{NP} \subseteq \boldsymbol{C}$, $\mathbf{NP}$ has p-measure zero. Now if $\mathbf{NP}$ has not p-measure zero, we have that $\mathbf{NP} =_{i.o.} \mathbf{AM}$, which ends the proof.

□

4 Derandomization of PH in a Uniform Setting

The following result is an easiness versus randomness tradeoff for classes in the polynomial hierarchy.

Theorem 8 *For every $i \in \mathbb{N}$ at least one of the following statements holds.*

1. $\forall \epsilon > 0$, $\mathbf{\Sigma_i^P} \subseteq$ i.o.[pseudo$_{\mathbf{NPMV}^{\mathbf{\Sigma_{i-1}^P}}}$]$\mathbf{coNTIME}^{\mathbf{\Sigma_{i-2}^P}}(2^{n^\epsilon})$, *or*

2. $\mathbf{BP\Sigma_i^P} = \mathbf{\Sigma_i^P}$.

Proof.
Suppose inclusion 1 is false, i.e. there exists $\epsilon_0 > 0$ and a language $A \in \mathbf{\Sigma_i^P}$, such that $A \notin$ i.o.[pseudo$_{\mathbf{NPMV}^{\mathbf{\Sigma_{i-1}^P}}}$]$\mathbf{coNTIME}^{\mathbf{\Sigma_{i-2}^P}}(2^{n^{\epsilon_0}})$.

Since $A \in \mathbf{\Sigma_i^P}$, there exists a relation $M \in \mathbf{\Pi_{i-1}^P}$, and a polynomial $m(n)$, such that for every $x \in \{0,1\}^n$,

$$x \in A \iff \exists y \in \{0,1\}^m \text{ such that } (x, y) \in M.$$

For $m \in \mathbb{N}$ and $\delta > 0$, let S_m^δ be the set of truth tables of all $\log m$ variables Boolean functions with circuit complexity smaller than m^δ, for nondeterministic oracle circuits with oracle gates for QSAT_{i-1}. We have : $|S_m^\delta| \leq 2^{m^{O(\delta)}}$.

Consider the following procedure B_m^δ, which accepts x iff there exists a truth table $y \in S_m^\delta$, such that $(x, y) \in M$.

Claim The procedure B_m^δ is in $\mathbf{coNTIME}^{\mathbf{\Sigma_{i-2}^P}}(2^{n^{\epsilon_0}})$.

Indeed here is a description of the procedure B_m^δ.

1. Construct all $2^{m^{O(\delta)}}$ single-valued nondeterministic oracle gates circuits with oracle gates for QSAT_{i-1} of size at most m^δ.

2. Given one such circuit, compute its truth table.

3. Check for each $y \in S_m^\delta$ whether $(x, y) \in M$.

Since simulating a QSAT_{i-1} gate of fan in at most m^{δ} is in $\mathbf{DTIME}^{\mathbf{\Sigma}_{i-2}^{\mathbf{P}}}(2^{O(m^{\delta})})$, and since there are at most m^{δ} nondeterministic choice gates, the running time of step 2 is in $\mathbf{DTIME}^{\mathbf{\Sigma}_{i-2}^{\mathbf{P}}}(2^{O(m^{\delta})})$. For step 3, since $M \in \Pi_{i-1}$, step 3 can be executed in $\mathbf{coNTIME}^{\mathbf{\Sigma}_{i-2}^{\mathbf{P}}}(2^{m^{O(\delta)}})$. Thus B_m^{δ} is in $\mathbf{coNTIME}^{\mathbf{\Sigma}_{i-2}^{\mathbf{P}}}(2^{m^{c\delta}})$, for some constant $c > 0$. Choosing δ such that $m^{c\delta} \leq n^{\epsilon_0}$ proves the claim.

Now since $A \notin$ i.o.[pseudo$_{\mathbf{NPMV}^{\mathbf{\Sigma}_{i-1}^{\mathbf{P}}}}$]$\mathbf{coNTIME}^{\mathbf{\Sigma}_{i-2}^{\mathbf{P}}}(2^{n^{\epsilon_0}})$, there is a multi-valued nondeterministic refuter R with oracle access to QSAT_{i-1}, such that for almost every n, a string x produced by R is in the symmetrical difference $A \triangle L(B_m^{\delta})$ (where $L(B_m^{\delta})$ is the language decided by procedure B_m^{δ}).

Since $L(B_m^{\delta}) \subseteq A$, we have $A \triangle L(B_m^{\delta}) = A \setminus L(B_m^{\delta})$, therefore for each such string x we have: For every $y \in S_m^{\delta}$ it holds that $(x, y) \notin M$, but there exists $y \in \{0,1\}^m \backslash S_m^{\delta}$ such that $(x, y) \in M$.

Now let $\epsilon = \delta$, and let $\gamma > 0$ be as in Theorem 1, and let $k = m^{\gamma}$. The following multi-valued nondeterministic procedure with oracle access to $\mathbf{\Sigma}_{i-1}^{\mathbf{P}}$ produces the truth table of a Boolean function with $\log m$ variables with circuit complexity greater than m^{ϵ} for nondeterministic oracle circuits with oracle gate for QSAT_{i-1}.

Procedure PRODUCE-FUNCTION:

1. Use R to nondeterministically produce a string x in $A \setminus L(B_m^{\delta})$.

2. Nondeterministically guess a string y of length m and output it if $(x, y) \in M$.

Once we obtain the truth table of a hard function, we use Theorem 1 to produce in time polynomial in k a hitting set in $\{0,1\}^k$ with threshold $\frac{1}{2}$, for co-nondeterministic circuits with oracle gates for QSAT_{i-1} of size k. Applying Proposition 2, we get $\mathbf{BP\Sigma_i^P} = \mathbf{\Sigma}_i^{\mathbf{P}}$.

□

5 Final Remarks

It would be interesting to see whether Theorem 4 could be improved to prove a zero-one measure law for **RP**. Since it is possible that **EXP** and **ZPP** are equal on infinitely many input lengths and still **ZPP** has not p-measure one, it seems that stronger refuters than **FZPP** refuters are needed.

References

[AK97] V. Arvind and J. Köbler. On pseudorandomness and resource-bounded measure. *Proceedings 17th Conference of the Foundations of Software Technology and Theorical Computer Science*, 1346:235–249, 1997.

[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.

[IW97] R. Impagliazzo and A. Widgerson. P = BPP if E requires exponential circuits: derandomizing the XOR lemma. *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 220–229, 1997.

[Kab00] V. Kabanets. Easiness assumptons and hardness test: Trading time for zero error. *Proceedings of the Fifteenth Annual IEEE Conference on Computational Complexity*, pages 150–157, 2000.

[KvM99] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial hierarchy collapses. *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 659–667, 1999.

[Lu00] C.-J. Lu. Derandomizing arthur-merlin games under uniform assumptions. *Proceedings of the Eleventh Annual International Symposium on Algorithms and Computation*, 2000.

[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.

[Mel00] D. Melkebeek. The zero one law holds for BPP. *Theoretical Computer Science*, 244(1-2):283–288, 2000.

[MV99] P. Miltersen and N. Vinodchandran. Derandomizing Arthur-Merlin games using hitting sets. *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pages 71–80, 199.

[MV96] A.A. Muchnik and N.K. Vereshchagin. A general method to construct oracles realizing given relationships between complexity classes. *Theoretical Compuetr Science*, 157:227–258, 1996.

[NW94] N. Nissan and A. Widgerson. Hardness vs. randomness. *Journal of Computer and System Science*, 49:149–167, 1994.

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