Electronic Colloquium on Computational Complexity, Report No. 4 (2026)

Yet Another Proof that BPP $\subseteq$ PH

Ilya Volkovich*

January 15, 2026

Abstract

We present a new, simplified proof that the complexity class BPP is contained in the Polynomial Hierarchy (PH), using k-wise independent hashing as the main tool. We further extend this approach to recover several other previously known inclusions between complexity classes. Our techniques are inspired by the work of Bellare, Goldreich, and Petrank (Information and Computation, 2000).

1 Introduction

The class BPP stands for the set of problems that could be solved efficiently using randomness. Given its importance, BPP has received considerable attention in complexity theory. In particular, some of the earliest results in the field [Sip83, Lau83] established that BPP is contained in the so-called Polynomial Hierarchy (PH). Since then, there have been several alternative proofs [Can96, RS98, CR06, GZ11] which, in particular, put BPP is some "lower" classes.

Indeed, all the above proofs operate by distinguishing a "large" set from a "small" one. Our approach captures this distinction through the following intuition: consider applying a shrinking (i.e. a hash) function to the domain of the set. Such a function naturally induces a disjoint partition of the domain by grouping together all inputs that map to the same value. If the set is "large", then for *any* shrinking function, at least one part of the partition will also be "large." Conversely, if the set is "small", then a random (or sufficiently structured) shrinking function will, with high probability, produce an approximately balanced partition in which all parts are "small."' One important aspect of this approach is that it inherently one-sided error! We note that this intuition was formalized by Bellare, Goldreich, and Petrank [BGP00], who employed it to sample a uniform NP witness using an NP oracle. Indeed, our techniques is inspired by their work.

1.1 Comparison to Previous Work

- The proof of [Lau83] puts BPP in the class Σ_2P. It operates by showing that a set is "large" if and only if it can "cover" the entire space $\{0,1\}^n$ via appropriate shifts. This is carried out using the probabilistic method. It is to be noted that that the first step of this proof requires amplification in order to reduce the error probability exponentially. This, in turn, typically relies on Chernoff (or other) concentration bounds for sums of independent random variables. Our proof also relies on a concentration bound, albeit for partially dependent variables.

*Computer Science Department, Boston College, Chestnut Hill, MA. Email: ilya.volkovich@bc.edu

ISSN 1433-8092

- The proofs of [Can96, RS98] put BPP is a lower complexity class known as $\mathsf{S}_2\mathsf{P}$. Yet, these proofs still use covers.
- The proof of [GZ11] is based on Zuckerman's efficient amplification of BPP [Zuc96].

2 Definition and Main Argument

In this section we give the relevant definition and prove the main technical lemma (Lemma 2.4) which will be used in the analysis all our algorithms. Some notations are taken from [BGP00].

Definition 2.1. *Let $k \in \mathbb{N}$. A set of random variables $\{Z_1, \ldots, Z_t\}$ is called* k-wise independent, *if every subset of size k of those variables is independent.*

Definition 2.2. *A family of functions $\mathcal{H}(n,m,k) = \left\{h : \{0,1\}^n \to \{0,1\}^m\right\}$ is called* k-wise independent hash family, *if for any* ***distinct*** *$x_1, \ldots, x_k \in \{0,1\}^n$ and any $y_1, \ldots, y_k \in \{0,1\}^m$ we have that:*

$$\Pr_{h \in \mathcal{H}(n,m,k)} [\forall i \in [k] : h(x_i) = y_i] = 2^{-mk}.$$

The following lemma provides a concentration bound for partially dependent random variables.

Lemma 2.3 ([BR94])**.** *Let $k \geq 4$ be even integer. Suppose $Z_1, \ldots Z_t$ are k-wise independent random variables taking values in $[0,1]$. Let $Z = Z_1 + \ldots Z_t$ and $\mu = \mathrm{EX}[Z]$, and let $A > 0$. Then*

$$\Pr\left[|Z - \mu| \geq A\right] \leq 8 \cdot \left(\frac{k\mu + k^2}{A^2}\right)^{k/2}.$$

We associate a set $S \subseteq \{0,1\}^n$ with its characteristic function $C : \{0,1\}^n \to \{0,1\}$ in a natural way: $S = C^{-1}(1) \stackrel{\Delta}{=} \{x \mid C(x) = 1\}$. Given this correspondence, we define $|C| \stackrel{\Delta}{=} \left|C^{-1}(1)\right| = |S|$. For a function $h : \{0,1\}^n \to \{0,1\}^m$ and $\alpha \in \{0,1\}^m$ we define the part $C_{h,\alpha} \stackrel{\Delta}{=} \{x \mid C(x) = 1 \wedge h(x) = \alpha\}$. Indeed, for every h the collection $\{C_{h,\alpha}\}_{\alpha \in \{0,1\}^m}$ forms a partition of $C^{-1}(1)$. The following Lemma in inspired by Lemmas 3.3 and 3.6 of [BGP00].

Lemma 2.4. *Let $C : \{0,1\}^n \to \{0,1\}$ and $k \geq 4$ a power of 2. Set $m \stackrel{\Delta}{=} n - 2 - 2\log_2 k$. Then:*

- *If $|C| \geq \frac{3}{4} \cdot 2^n$ then for* any *function $h : \{0,1\}^n \to \{0,1\}^m$ there is $\alpha \in \{0,1\}^m$ s.t. $|C_{h,\alpha}| \geq 2k^2$*
- *If $|C| \leq \frac{1}{4} \cdot 2^n$ then $\Pr_{h \in \mathcal{H}(n,m,k)}\left[\exists \alpha \in \{0,1\}^m \text{ s.t. } |C_{h,\alpha}| \geq 2k^2\right] \leq 2^{n+3} \cdot \left(\frac{2}{k}\right)^{k/2}$.*

Proof.

- Suppose $\forall \alpha \in \{0,1\}^m$ we have $|C_{h,\alpha}| < 2k^2$. Then $|C| = \sum_{\alpha} |C_{h,\alpha}| < 2^m \cdot 2k^2 = 2^{n-1} < \frac{3}{4} \cdot 2^n$.
- Fix $\alpha \in \{0,1\}^m$. For any $x \in \{0,1\}^n$ we define a random variable:

$$Z_x = \begin{cases} 1 & \text{if } h(x) = \alpha \\ 0 & \text{otherwise.} \end{cases}$$

In addition, let

$$Z = \sum_{x \in C^{-1}(1)} Z_x.$$

Observe that $Z = |C_{h,\alpha}|$ and the set $\{Z_x\}_x$ of random variables is k-wise independent. In addition, since h is a random k-wise independent function, for any $x \in \{0,1\}^n$ we have that $\mathrm{EX}[Z_x] = \Pr_h[h(x) = \alpha] = 2^{-m}$. Therefore:

$$\mu \triangleq \mathrm{EX}[Z] = \sum_{x \in C^{-1}(1)} \mathrm{EX}[Z_x] = \frac{|C|}{2^m} \leq \frac{\frac{1}{4} \cdot 2^n}{2^{n-2-2\log_2 k}} = k^2.$$

Therefore, we obtain:

$$\Pr_h\left[|C_{h,\alpha}| \geq 2k^2\right] = \Pr_h\left[Z - \mu \geq 2k^2 - \mu\right] \leq \Pr_h\left[|Z - \mu| \geq A\right],$$

where $A = 2k^2 - \mu \geq k^2$. And subsequently by Lemma 2.4:

$$\Pr_h\left[|C_{h,\alpha}| \geq 2k^2\right] \leq \Pr_h\left[|Z - \mu| \geq A\right] \leq 8 \cdot \left(\frac{k\mu + k^2}{A^2}\right)^{k/2} \leq 8 \cdot \left(\frac{k^3 + k^2}{k^4}\right)^{k/2} \leq 8 \cdot \left(\frac{2}{k}\right)^{k/2}.$$

Finally, by the union bound:

$$\Pr_h\left[\exists \alpha \in \{0,1\}^m \text{ s.t. } |C_{h,\alpha}| \geq 2k^2\right] \leq \sum_{\alpha} \Pr_h\left[|C_{h,\alpha}| \geq 2k^2\right] \leq 2^{n+3} \cdot \left(\frac{2}{k}\right)^{k/2}. \qquad \square$$

3 The Proofs

In this section, we present simplified algorithms for various containments of complexity classes. While these containments were already known, our contribution is in their simplification. We begin by defining the relevant complexity classes. The interested reader can find more details in the standard textbook [AB09].

3.1 Complexity Classes

Definition 3.1 (The class BPP)**.** *A language $L \in \mathsf{BPP}$, if there exists a polynomial $p(n)$ and polynomial-time computable predicate $V(x,r)$, where $|r| = p(|x|)$, such that:*

- $x \in L \implies \Pr_r[V(x,r) = 1] \geq \frac{3}{4}$
- $x \notin L \implies \Pr_r[V(x,r) = 1] \leq \frac{1}{4}$

The complexity classes MA and AM were introduced in the seminal work of Babai [Bab85] and admit two variants: *perfect* completeness (zero error) and *imperfect* completeness. Although these variants were later shown to be equivalent (see e.g. [FGM+89]), we present definitions of both, as one of our results provides an alternative proof of this equivalence.

Definition 3.2 (The classes MA and MA_0)**.** *A language $L \in \mathsf{MA}$ (resp. MA_0), if there exists a polynomials $p(n)$ and $q(n)$ and polynomial-time computable predicate $V(x,w,r)$, where $|r| = p(|x|)$ and $|w| = q(|x|)$, such that:*

- $x \in L \implies \exists w \Pr_r[V(x,w,r)=1] \geq \frac{3}{4}$ $(resp. = 1)$
- $x \notin L \implies \forall w \Pr_r[V(x,w,r)=1] \leq \frac{1}{4}$

Definition 3.3 (The classes AM and AM_0)**.** *A language $L \in \mathsf{AM}$ $(resp.\ \mathsf{AM}_0)$, if there exists polynomials $p(n)$ and $q(n)$ and polynomial-time computable predicate $V(x,w,r)$, where $|r| = p(|x|)$ and $|w| = q(|x|)$, such that:*

- $x \in L \implies \Pr_r[\exists w : V(x,w,r)=1] \geq \frac{3}{4}$ $(resp. = 1)$
- $x \notin L \implies \Pr_r[\exists w : V(x,w,r)=1] \leq \frac{1}{4}$

3.2 Main Results

In this section we state and prove our main results. As was mentioned before, one of features of the proofs is that they are inherently one-sided error.

The following notation will be useful for us: for $n \in \mathbb{N}$, we denote by $\delta_2(n)$ the smallest power of 2 greater or equal to n. Given that, observe that for $k = \delta_2(n)$ the bounds from the second case of Lemma 2.4 becomes $2^{n+3} \cdot \left(\frac{2}{k}\right)^{k/2} \ll 0.25$, which we will use implicitly.

Our first result establishes the containment $\mathsf{BPP} \subseteq \mathsf{PH}$ by showing that BPP belongs to the class AM_0. It is to be noted that the containment in AM (the imperfect version) follows from definition.

Theorem 3.4. $\mathsf{BPP} \subseteq \mathsf{AM}_0$ $(\subseteq \mathsf{PH})$.

Proof. Let $L \in \mathsf{BPP}$. By definition, there exist $V(x,r)$ and a polynomial $p(n)$ such that $|r| = p(|x|)$. Consider the following algorithm:

1. Set $n = |x|$, $k = \delta_2(p(n))$ and $m = p(n) - 2 - 2\log_2 k$.
2. Arthur picks $h \in \mathcal{H}(p(n), m, k)$ at random and sends it to Merlin.
3. Merlin sends $\alpha \in \{0,1\}^m$ and strings $r_1, \ldots, r_{2k^2} \in \{0,1\}^{p(n)}$.
4. Arthur accepts iff $\forall i \in [2k^2] : V(x,r_i) = 1 \wedge h(r_i) = \alpha$.

Algorithm 1: $\mathsf{BPP} \subseteq \mathsf{AM}_0$

Analysis: Define $C_x(r) \stackrel{\Delta}{=} V(x,r) : \{0,1\}^{p(n)} \to \{0,1\}$.

- $x \in L \implies |C_x| \geq \frac{3}{4} \cdot 2^{p(n)}$. By Lemma 2.4, for any h there exists α s.t. $|C_{x,h,\alpha}| \geq 2k^2$ i.e. $|\{r \mid V(x,r) = 1 \wedge h(r) = \alpha\}| \geq 2k^2$.
- $x \notin L \implies |C_x| \leq \frac{1}{4} \cdot 2^{p(n)}$. By Lemma 2.4,

$$\Pr_h\left[\exists \alpha \in \{0,1\}^m \text{ s.t. } |C_{x,h,\alpha}| \geq 2k^2\right] \leq 2^{p(n)+3} \cdot \left(\frac{2}{k}\right)^{k/2} \leq 2^{p(n)+3} \cdot \left(\frac{2}{p(n)}\right)^{p(n)/2} \ll 0.25.$$

□

By slightly extending this algorithm, we can show that the two variants of AM are, indeed, equal.

Theorem 3.5. $\mathsf{AM}_0 = \mathsf{AM}$.

Proof. Let $L \in \mathsf{AM}$. By definition, there exists polynomials $p(n)$ and $q(n)$, and $V(x, w, r)$ such that $|r| = p(|x|)$ and $|w| = q(|x|)$. Consider the following algorithm:

1. Set $n = |x|$, $k = \delta_2(p(n))$ and $m = p(n) - 2 - 2\log_2 k$.
2. Arthur picks $h \in \mathcal{H}(p(n), m, k)$ at random and sends it to Merlin.
3. Merlin sends $\alpha \in \{0,1\}^m$ and strings $r_1, \ldots, r_{2k^2} \in \{0,1\}^{p(n)}$ and $w_1, \ldots, w_{2k^2} \in \{0,1\}^{q(n)}$.
4. Arthur accepts iff $\forall i \in [2k^2] : V(x, w_i, r_i) = 1 \wedge h(r_i) = \alpha$.

Algorithm 2: $\mathsf{AM}_0 = \mathsf{AM}$

Analysis: Define $C_x(r) : \{0,1\}^{p(n)} \to \{0,1\}$ as $C_x(r) = 1 \iff \exists w V(x, w, r) = 1$. Hence $|C_x| = |\{r \mid \exists w \text{ s.t. } V(x, w, r) = 1\}|$. Equivalently, one can think about $C_x(r)$ as a "non-deterministic" function.

- $x \in L \implies |C_x| \geq \frac{3}{4} \cdot 2^{p(n)}$. By Lemma 2.4, for any h there exists α s.t. $|C_{x,h,\alpha}| \geq 2k^2$.
- $x \notin L \implies |C_x| \leq \frac{1}{4} \cdot 2^{p(n)}$. By Lemma 2.4:

$$\Pr_h \left[\exists \alpha \text{ s.t. } |C_{x,h,\alpha}| \geq 2k^2\right] \leq 2^{p(n)+3} \cdot \left(\frac{2}{k}\right)^{k/2} \ll 0.25.$$

□

The next result shows that containment of MA in AM. The previously existing proofs typically first show that $\mathsf{MA} = \mathsf{MA}_0$ and then proceed by showing that $\mathsf{MA}_0 \subseteq \mathsf{AM}_0$. As was noted before, our proof immediately yields a one-sided error procedure.

Theorem 3.6. $\mathsf{MA} \subseteq \mathsf{AM}_0$.

Proof. Let $L \in \mathsf{MA}$. By definition, there exists polynomials $p(n)$ and $q(n)$, and $V(x, w, r)$ such that $|r| = p(|x|)$ and $|w| = q(|x|)$. Consider the following algorithm:

1. Set $n = |x|$, $k = \delta_2(p(n) + q(n))$ and $m = p(n) - 2 - 2\log_2 k$.
2. Arthur picks $h \in \mathcal{H}(p(n), m, k)$ at random and sends it to Merlin.
3. Merlin sends $\alpha \in \{0,1\}^m$, $w \in \{0,1\}^{q(n)}$ and strings $r_1, \ldots, r_{2k^2} \in \{0,1\}^{p(n)}$.
4. Arthur accepts iff $\forall i \in [2k^2] : V(x, w, r_i) = 1 \wedge h(r_i) = \alpha$.

Algorithm 3: $\mathsf{MA} \subseteq \mathsf{AM}_0$

Analysis: Define $C_{x,w}(r) \stackrel{\Delta}{=} V(x, w, r) : \{0,1\}^{p(n)} \to \{0,1\}$.

- $x \in L \implies \exists w : |C_{x,w}| \geq \frac{3}{4} \cdot 2^{p(n)}$. By Lemma 2.4, for any h there exists α s.t. $|C_{x,w,h,\alpha}| \geq 2k^2$.

- $x \notin L \implies \forall w : |C_{x,w}| \leq \frac{1}{4} \cdot 2^{p(n)}$. By Lemma 2.4, for any fixed $w \in \{0,1\}^{q(n)}$:

$$\Pr_h \left[\exists \alpha \text{ s.t. } |C_{x,w,h,\alpha}| \geq 2k^2\right] \leq 2^{p(n)+3} \cdot \left(\frac{2}{k}\right)^{k/2}.$$

Therefore, by the union bound:

$$\Pr_h \left[\exists w, \alpha \text{ s.t. } |C_{x,w,h,\alpha}| \geq 2k^2\right] \leq \sum_{w \in \{0,1\}^{q(n)}} \Pr_h \left[\exists \alpha \text{ s.t. } |C_{x,w,h,\alpha}| \geq 2k^2\right]$$

$$\leq 2^{p(n)+q(n)+3} \cdot \left(\frac{2}{k}\right)^{k/2} \ll 0.25. \qquad \square$$

We conclude our results by showing containments in the class $\mathsf{ZPP}^{\mathsf{NP}}$.

Theorem 3.7. $\mathsf{BPP} \subseteq \mathsf{ZPP}^{\mathsf{NP}}$.

Proof. Let $L \in \mathsf{BPP}$. By definition, there exist $V(x,r)$ and a polynomial $p(n)$ such that $|r| = p(|x|)$. Consider the following algorithm:

1. Set $n = |x|$, $k = \delta_2(p(n))$ and $m = p(n) - 2 - 2\log_2 k$.
2. Define $C_x(r) \stackrel{\Delta}{=} V(x,r)$ and $D_x(r) \stackrel{\Delta}{=} \neg V(x,r)$
3. Pick $h \in \mathcal{H}(p(n), m, k)$ at random.
4. Using the NP oracle compute:
   i. $a = 1$ iff $\exists \alpha \in \{0,1\}^m$ s.t. $|C_{x,h,\alpha}| \geq 2k^2$
   ii. $b = 1$ iff $\exists \beta \in \{0,1\}^m$ s.t. $|D_{x,h,\beta}| \geq 2k^2$
5. If $a = 1 \wedge b = 0$ then accept
6. If $b = 1 \wedge a = 0$ then reject
7. Otherwise, output $\perp$

Algorithm 4: $\mathsf{BPP} \subseteq \mathsf{ZPP}^{\mathsf{NP}}$

Analysis:

- $x \in L$. By Lemma, 2.4 for any h there exists α s.t. $|C_{x,h,\alpha}| \geq 2k^2$. At the same time, $\Pr_h \left[\exists \beta \in \text{ s.t. } |D_{x,h,\beta}| \geq 2k^2\right] \ll 0.25$.
  Therefore: $a = 1$ and in addition, $b = \begin{cases} 1, & \text{w.p. } \ll 0.25 \implies \perp \\ 0, & \text{w.p. } \gg 0.75 \implies \text{accept.} \end{cases}$
- $x \notin L$. Similarly, by Lemma 2.4, for any h there exists β s.t. $|D_{x,h,\beta}| \geq 2k^2$ and at the same time, $\Pr_h \left[\exists \alpha \in \text{ s.t. } |C_{x,h,\alpha}| \geq 2k^2\right] \ll 0.25$.
  Therefore: $b = 1$ and in addition, $a = \begin{cases} 1, & \text{w.p. } \ll 0.25 \implies \perp \\ 0, & \text{w.p. } \gg 0.75 \implies \text{reject.} \end{cases}$

We remark that the algorithm uses only two queries to the NP oracle. $\square$

Theorem 3.8. $\mathsf{MA} \subseteq \mathsf{ZPP}^{\mathsf{NP}}$.

Proof. Let $L \in \mathsf{MA}$. By definition, there exists polynomials $p(n)$ and $q(n)$, and $V(x,w,r)$ such that $|r| = p(|x|)$ and $|w| = q(|x|)$. Consider the following algorithm:

1. Set $n = |x|$, $k = \delta_2(p(n) + q(n))$ and $m = p(n) - 2 - 2\log_2 k$.
2. Define $C_{x,w}(r) \triangleq V(x,w,r)$ and $D_{x,w}(r) \triangleq \neg V(x,w,r)$.
3. Pick $h \in \mathcal{H}(p(n), m, k)$ at random.
4. Using the NP oracle ask if $\exists w, \alpha$ such that $|C_{x,w,h,\alpha}| \geq 2k^2$.
5. If "no" - reject; Otherwise, compute such w using NP with a search-to-decision reduction.
6. Using the NP oracle ask if $\exists \beta$ such that $|D_{x,w,h,\beta}| \geq 2k^2$.
7. If "no" - accept; Otherwise, output $\perp$.

Algorithm 5: $\mathsf{MA} \subseteq \mathsf{ZPP}^{\mathsf{NP}}$

Analysis: By Lemma 2.4, for any x:

$$\Pr_h \left[\exists w \text{ s.t. } |C_{x,w}| \leq \frac{1}{4} \cdot 2^{p(n)} \wedge \alpha \text{ s.t. } |C_{x,w,h,\alpha}| \geq 2k^2 \right] \leq 2^{p(n)+q(n)+3} \cdot \left(\frac{2}{k}\right)^{k/2} \ll 0.01.$$

- $x \in L \implies \exists w : |C_{x,w}| \geq \frac{3}{4} \cdot 2^{p(n)}$. By Lemma 2.4, for any h there exists α s.t. $|C_{x,w,h,\alpha}| \geq 2k^2$. Therefore, the algorithm will never reject at Line 5. Based on the above, w.h.p the witness w computed in Line 5 satisfies: $|C_{x,w}| \geq \frac{3}{4} \cdot 2^{p(n)}$ or equivalently $|D_{x,w}| \leq \frac{1}{4} \cdot 2^{p(n)}$. Applying Lemma 2.4 again, the probability that there exists β s.t. $|D_{x,w,h,\beta}| \geq 2k^2$ is very small.
- $x \notin L \implies \forall w : |C_{x,w}| \leq \frac{1}{4} \cdot 2^{p(n)}$. By the above, $\Pr_h \left[\exists w, \alpha \text{ s.t. } |C_{x,w,h,\alpha}| \geq 2k^2\right]$ is very small. Nonetheless, if the event does occur, the algorithm computes a witness w such that $|D_{x,w}| \geq \frac{3}{4} \cdot 2^{p(n)}$. And therefore by Lemma 2.4, there always exists β s.t. $|D_{x,w,h,\beta}| \geq 2k^2$ which will lead to $\perp$. □

4 Discussion

We present a new, intuitive approach to the "set-size" problem: a "large" set always contains a "large part" whereas in a "small" set all parts will, with high probability, be small. This yields new simulations for some complexity classes by inherent one-sided error procedures. We hope that this approach can be applied to establish other and new containments as well.

Acknowledgments

The author would like to thank the anonymous referees for their comments.

References

[AB09] S. Arora and B. Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[Bab85] L. Babai. Trading group theory for randomness. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing (STOC)*, pages 421–429, 1985.

[BGP00] M. Bellare, O. Goldreich, and E. Petrank. Uniform generation of np-witnesses using an np-oracle. *Inf. Comput.*, 163(2):510–526, 2000.

[BR94] M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, November 20-22, 1994*, pages 276–287. IEEE Computer Society, 1994.

[Can96] R. Canetti. More on BPP and the polynomial-time hierarchy. *Inf. Process. Lett.*, 57(5):237–241, 1996.

[CR06] V. T. Chakaravarthy and S. Roy. Oblivious symmetric alternation. In *STACS*, pages 230–241, 2006.

[FGM+89] M. Fürer, O. Goldreich, Y. Mansour, M. Sipser, and S. Zachos. On completeness and soundness in interactive proof systems. *Adv. Comput. Res.*, 5:429–442, 1989.

[GZ11] O. Goldreich and D. Zuckerman. Another proof that bpp $\subseteq$ ph (and more). *Studies in Complexity and Cryptography*, pages 40–53, 2011.

[Lau83] C. Lautemann. BPP and the polynomial hierarchy. *Inf. Process. Lett.*, 17(4):215–217, 1983.

[RS98] A. Russell and R. Sundaram. Symmetric alternation captures BPP. *Comput. Complex.*, 7(2):152–162, 1998.

[Sip83] M. Sipser. A complexity theoretic approach to randomness. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 330–335. ACM, 1983.

[Zuc96] D. Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4/5):367–391, 1996.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il