More on BPP and the Polynomial-time Hierarchy*

Ran Canetti†

March 1996

Abstract

We simplify known proofs that BPP lies within the Polynomial-time Hierarchy. Furthermore, our simple proof shows that BPP is contained in a complexity class that presumably lies even lower in this Hierarchy than was previously known. Our technique also simplifies the proof that one-sided-error IP equals two-sided-error IP.

Keywords: theory of computation, computational complexity, randomness, interactive proofs.

1 Introduction

BPP is the class of languages recognizable by probabilistic polynomial time algorithms with two-sided, bounded probability of error. That is, a language $L \subseteq \{0,1\}^*$ is in BPP if there exists a probabilistic Turing machine M that runs in time polynomial in the length of its input, and such that for all inputs x:

$$\mathrm{Prob}(M(x) = \chi_L(x)) \geq \frac{3}{4}$$

where $\chi_L(x) = 1$ if $x \in L$ and $\chi_L(x) = 0$ otherwise, and the probability is taken over the random inputs of M. BPP is often perceived as the class of decision problems for which feasible algorithms exist. An interesting question is how efficiently can languages in BPP be recognized without use of randomness.

Answering this question, Sipser has shown that BPP lies within the Polynomial-time Hierarchy [Si]. The Polynomial-time Hierarchy may be defined as follows [St, W]. A language $L \subseteq \{0,1\}^*$ is in Σ_l^{P} if there exists an $(l+1)$-ary relation $R(\cdot,\ldots,\cdot)$ computable in time polynomial in the length of the first input, such that

$$x \in L \quad \Leftrightarrow \quad \exists y_1\ \forall y_2 \ldots \mathrm{Q} y_l \quad R(x, y_1, \ldots, y_l) = 1 \tag{1}$$

where Q can be either $\exists$ or $\forall$, and $R(\cdot,\ldots,\cdot) = 1$ denotes that $R(\cdot,\ldots,\cdot)$ is satisfied. Similarly, $L \in \Pi_l^{\mathsf{P}}$ if there exists an $(l+1)$-ary relation $R(\cdot,\ldots,\cdot)$ computable in time polynomial in the length of the first input, such that

$$x \in L \quad \Leftrightarrow \quad \forall y_1\ \exists y_2 \ldots \mathrm{Q} y_l \quad R(x, y_1, \ldots, y_l) = 1. \tag{2}$$

*Appears in *IPL 57*, 1996, pp. 237-241.

†Laboratory of Computer Science, Massachusetts Institute of Technology, Cambridge, MA.
Email: *canetti@theory.lcs.mit.edu*. Supported by grant No. 92-00226 from the United States - Israel Binational Science Foundation (BSF), Jerusalem, Israel.

Specifically, Sipser has shown that $\mathsf{BPP} \subseteq \Sigma_4^{\mathrm{p}}$. (Since BPP is closed under complement it follows that $\mathsf{BPP} \subseteq \Pi_4^{\mathrm{p}}$ as well.) Lautemann has improved on Sipser's result, showing that $\mathsf{BPP} \subseteq \Sigma_2^{\mathrm{p}}$ [L]. We simplify Lautemann's proof and show a slightly stronger result. We define a complexity class, denoted Φ_2^{p}, as follows: $L \in \Phi_2^{\mathrm{p}}$ if there exists a ternary relation $R(\cdot,\cdot,\cdot)$ computable in time polynomial in the length of the first input, such that for any $x \in \{0,1\}^*$,

$$\exists y \ \forall z \quad (R(x,y,z) = \chi_L(x)) \tag{3}$$

$$\exists z \ \forall y \quad (R(x,y,z) = \chi_L(x)) \tag{4}$$

The definition of Φ_2^{p} suggests the following interpretation. If $L \in \Phi_2^{\mathrm{p}}$ then there exists a (deterministic) polynomial-time verifier R that interacts with *two* provers, Y and Z. On input x, each of the provers hands R a "proof" (denoted y and z, respectively) to whether $x \in L$. Next R decides whether $x \in L$ based on y and z. Condition (3) above says that there exists a strategy for Y such that R decides correctly on x with any strategy of Z. Similarly, condition (4) says that there exists a strategy for Z such that R decides correctly on x with any strategy of Y.

In this work we show that $\mathsf{BPP} \subseteq \Phi_2^{\mathrm{p}}$. It is easy to see that $\Phi_2^{\mathrm{p}} \subseteq \Sigma_2^{\mathrm{p}} \cap \Pi_2^{\mathrm{p}}$. It can also be seen that $\mathsf{P}^{\mathsf{NP}} \subseteq \Phi_2^{\mathrm{p}}$ (we sketch a proof in the sequel.) Presumably the containment $\Phi_2^{\mathrm{p}} \subseteq \Sigma_2^{\mathrm{p}} \cap \Pi_2^{\mathrm{p}}$ is strict. This may be argued as follows: The requirements for membership in $\Sigma_2^{\mathrm{p}} \cap \Pi_2^{\mathrm{p}}$ are, syntactically, a strict subset of the requirements for membership in Φ_2^{p}. Furthermore, $\Sigma_2^{\mathrm{p}} \cap \Pi_2^{\mathrm{p}}$ allows that (1) and (2) be satisfied with respect to two different relations, whereas Φ_2^{p} requires that a single relation R will satisfy both (3) and (4).

We note that Russell and Sundaram have independently obtained a similar result [RS]. They define a class, called S_2; even though their definition is slightly different then ours, it can be shown that $S_2 = \Phi_2^{\mathrm{p}}$. They show, via a different (and more complex) proof than the proof presented here, that $\mathsf{BPP} \subseteq S_2$. They also prove that $\mathsf{P}^{\mathsf{NP}} \subseteq S_2$. For completeness, we present a slightly modified version of this proof.

Our technique can also be used to simplify the proof that any language that has a two-sided-error interactive proof (see [GMR]) also has a one-sided-error interactive proof. We recall the standard definitions. Say that $L \in \mathsf{IP}^1[q]$ if there exists a polynomial-time verifier V that asks at most q queries, such that

$$\begin{aligned} x \in L \quad &\Rightarrow \quad \text{there exists a prover } P \text{ such that} \quad \mathrm{Prob}((V,P)=1) = 1 \\ x \notin L \quad &\Rightarrow \quad \text{for any prover } P^* \quad \mathrm{Prob}((V,P^*)=1) \leq \frac{1}{2} \end{aligned}$$

Say that $L \in \mathsf{IP}^2[q]$ if:

$$\begin{aligned} x \in L \quad &\Rightarrow \quad \text{there exists a prover } P \text{ such that} \quad \mathrm{Prob}((V,P)=1) \geq \frac{3}{4} \\ x \notin L \quad &\Rightarrow \quad \text{for any prover } P^* \quad \mathrm{Prob}((V,P^*)=1) \leq \frac{1}{4} \end{aligned}$$

Clearly, $\mathsf{IP}^1[q] \subseteq \mathsf{IP}^2[q]$. Furer *et al.* show that $IP^2[q] \subseteq \mathsf{IP}^1[q+2]$ (see [FGMSZ]). We simplify their proof.

Note that Shamir's result implies that $\mathsf{IP}^1[poly] = \mathsf{PSPACE} = \mathsf{IP}^2[poly]$, where $[poly]$ stands for polynomial number of queries [Sh]. However, his proof does not conserve the number of queries, or the number of random or communication bits used. The [FGMSZ] proof, as well as ours, conserves these parameters.

That is, given a verifier and prover that have two-sided-error and use r random bits, m communication bits and q queries, we construct a verifier and prover that have one-sided-error, and use r random bits, $O(m \cdot r)$ communication bits, and $q+2$ queries. (Furer *et al.* use logarithmically more communication and randomness than our proof.)

Organization. In Section 2 we show that $\mathsf{BPP} \subseteq \Phi_2^{\mathsf{P}}$. In Section 3 we present our simplified proof that $IP^2[q] \subseteq \mathsf{IP}^1[q+2]$.

2 BPP and the Polynomial-time Hierarchy

We show that $\mathsf{BPP} \subseteq \Phi_2^{\mathsf{P}}$. First we prove a simple technical claim. We use this claim also in our proof that $\mathsf{IP}^1[q] = \mathsf{IP}^2[q]$. We say that a sequence $s_1, \ldots, s_k$, where each $s_i \in \{0,1\}^m$, covers a set $A \subseteq \{0,1\}^m$ if for any element $r \in \{0,1\}^m$, two thirds of the s_i's satisfy $s_i \oplus r \in A$ (where $\oplus$ denotes the bitwise exclusive or operator). We note that [L] and [FGMSZ] use a weaker notion of a covering set, where it is only required that a single s_i satisfies $s_i \oplus r \in A$.

Claim 1 *Every set $A \subseteq \{0,1\}^m$ of size at least $\frac{3}{4}2^m$ has a cover set of size $k = O(m)$.*

Proof: We modify the probabilistic argument of [L]. Consider a sequence $s = s_1, \ldots, s_k$ chosen uniformly and independently from $(\{0,1\}^m)^k$, where $k = O(m)$ is specified below. We show that s covers the set A with probability greater than zero:

$$\begin{aligned} \mathrm{Prob}(s \text{ does not cover } A) &= \mathrm{Prob}(\exists r \text{ s.t. } r \oplus s_i \notin A \text{ for } \left\lceil \frac{k}{3} \right\rceil \; s_i\text{'s}) \\ &\leq \sum_{r \in \{0,1\}^m} \mathrm{Prob}(r \oplus s_i \notin A \text{ for } \left\lceil \frac{k}{3} \right\rceil \; s_i\text{'s}) \\ &= 2^m \cdot \mathrm{Prob}(s_i \notin A \text{ for } \left\lceil \frac{k}{3} \right\rceil \; s_i\text{'s}) \end{aligned}$$

However, each s_i is in A with probability at least $\frac{3}{4}$. Applying Chernoff Bound [AS], we get

$$\mathrm{Prob}(s \text{ does not cover } A) \leq 2^m \cdot \mathrm{e}^{-\Omega(k)}.$$

Setting $k = O(m)$ is sufficient to make sure that s covers A with probability greater than zero, thus there exists a choice of s that covers A. □

Theorem 2 $\mathsf{BPP} \subseteq \Phi_2^{\mathsf{P}}$.

Proof: Let $L \in \mathsf{BPP}$ and let M be the corresponding probabilistic Turing machine. Assume without loss of generality that on all inputs of length n, the set of random inputs of M is $\{0,1\}^m$, where m is polynomial in n. Let $A_x \subseteq \{0,1\}^m$ denote the set of random inputs on which M correctly decides input $x \in \{0,1\}^n$. That is, $A_x = \{r | M_r(x) = \chi_L(x)\}$ where $M_r(x)$ denotes the output of M on input x and random input r.

To show that $L \in \Phi_2^{\mathsf{P}}$, construct a relation $R(x,y,z)$ as follows. Let $y = y_1, \ldots, y_k$ and $z = z_1, \ldots, z_k$ where each $y_i, z_i \in \{0,1\}^m$ and $k = O(m)$ is as in Claim 1. The relation $R(x,y,z)$ is satisfied if $M(x) = 1$ for two thirds of the k^2 random inputs $\{y_i \oplus z_j\}_{i,j \in \{1,\ldots,k\}}$.

Clearly R is computable in polynomial time. Furthermore, $|A_x| \geq \frac{3}{4}2^m$ for all $x \in \{0,1\}^n$. It follows from Claim 1 that for any x there exists a sequence $y = y_1, \ldots, y_k$ that covers A_x. That is, consider an arbitrary sequence $z = z_1, \ldots, z_k$ and let $z_j \in \{0,1\}^m$ be some specific element of z. Then $M(x) = 1$ on two thirds of the k random inputs $y_1 \oplus z_j, \ldots, y_k \oplus z_j$. Consequently, $M(x) = 1$ on two thirds of the k^2 random inputs $\{y_i \oplus z_j\}_{i,j \in \{1,\ldots,k\}}$. Thus $R(x, y, z)$ is satisfied for all z, and Condition (3) holds with respect to relation R.

We note that R is defined so that the roles of y and z are interchangeable. That is, $R(x, y, z)$ is satisfied iff $R(x, z, y)$ is satisfied. Thus the above argument can be repeated with the roles of y and z interchanged, and Condition (4) holds as well. $\square$

Theorem 3 $\mathsf{P}^{\mathsf{NP}} \subseteq \Phi_2^{\mathsf{P}}$.

Proof: This proof is a slight modification of the proof in [RS]. Let $L \in \mathsf{P}^{\mathsf{NP}}$. That is, there exists a polynomial-time Turing machine M and a language $O \in \mathsf{NP}$ such that M with oracle access to O correctly decides any input $x \in \{0,1\}^*$. To show that $L \in \Phi_2^{\mathsf{P}}$, we construct a polynomial-time ternary relation R that satisfies (3) and (4). The validity of the construction can be easily verified.

R simulates the computation of M on input x and with oracle access to language O. This is done with the help of inputs y and z of R, as follows. R expects both y and z to contain the oracle answers to M's queries asked in its computation on input x. Whenever the answer to a query w is that $w \in O$, R also expects to find an NP witness that $w \in O$. Now, whenever M, in the simulated computation, asks an oracle query w, R checks with both y and z for the oracle answer. If y and z agree on the oracle answer then R adopts this answer and continues the simulation. Otherwise, R verifies the validity of the NP witness provided by the side claiming that $w \in O$. If the witness is valid then R adopts the answer '$w \in O$' and continues the simulation. Otherwise R adopts the answer '$w \notin O$' and continues the simulation. Finally R is satisfied iff M accepts. (If either inputs y or z has a wrong format then R adopts the answers provided by the other input. If both y and z have a wrong format then R is not satisfied.) $\square$

3 Interactive Proofs

Theorem 4 $\mathsf{IP}^2[q] \subseteq \mathsf{IP}^1[q+2]$.

Proof: Let $L \in \mathsf{IP}^2[q]$, and let P and V be the corresponding prover and verifier. We construct a prover P' and a verifier V' with one-sided-error. Our construction is identical to that of [FGMSZ], except for the definition of a cover sequence, and the fact that [FGMSZ] need to first reduce the error probability to be polynomial in $\frac{1}{|x|}$. Our definition of a cover sequence eliminates the need for error reduction and allows for a much simpler analysis.

Without loss of generality, we assume that the queries made by the verifier consist of independent random bits only. In other words, we first transform (V, P) to the Arthur-Merlin model [B]. This can be done using the transformation of Goldwasser and Sipser [GS]. We note that this transformation maintains the number of rounds, up to an addition of 2. We further assume, without loss of generality, that V uses at most m random bits on all inputs of length n (where m is polynomial in n), and that the number of bits in each of the V's queries depends only on the input and the query number. Let $A_x^{V,P} \subseteq \{0,1\}^m$ denote the set of random inputs on which V correctly decides input $x \in \{0,1\}^n$ after interacting with P. That is, $A_x^{V,P} = \{r | (V_r, P)(x) = \chi_L(x)\}$ where $(V_r, P)(x)$ denotes the output of V on input x and random input r, after interacting with P.

We construct a prover P' and a verifier V' with one-sided-error as follows. We have $|A_x^{V,P}| \geq \frac{3}{4}2^m$ for any x. Thus Claim 1 assures that there exists a sequence $s = s_1, \ldots, s_k$ that covers $A_x^{V,P}$, where $k = O(m)$. First P' hands V' such a sequence s. Next V' chooses r uniformly from $\{0,1\}^m$. Next, k copies of the interaction (V, P) are simulated in parallel, where the i^{th} copy has random input $r \oplus s_i$. This is carried out as follows. Partition r into q chunks $r = r^{(1)}, \ldots, r^{(q)}$ according to the lengths of the queries made by V, where q is the number of queries. (We assume that this partition of r is fixed, for each input x.) Partition each s_i to chunks $s_i = s_i^{(1)}, \ldots, s_i^{(q)}$ with lengths equal to the lengths of the corresponding chunks of r. Prover P' keeps k copies $P_1, \ldots, P_k$ of the original prover P, and V' keeps k copies $V_1, \ldots, V_k$ of the original verifier V (V_i has random input $r \oplus s_i$). The j^{th} query of V consists of k segments, where the i^{th} segment is the query of V_i (i.e., $r^{(j)} \oplus s_i^{(j)}$). (Note that, since the prover knows s, it suffices that the j^{th} query of V' be only $r^{(j)}$.) Prover P' sends back k different answers, where the i^{th} answer is P_i's answer to query $r^{(j)} \oplus s_i^{(j)}$. Verifier V' hands the i^{th} answer to V_i. Finally V' accepts (i.e., outputs 1) if at least two thirds of $V_1, \ldots, V_k$ accept.

We analyze (V', P'). It is easy to see that if $x \in L$ then $\text{Prob}((P', V')(x) = 1) = 1$. Next, assume $x \notin L$. We use a simple counting argument to show that $\text{Prob}((V', P^*)(x) = 1) \leq \frac{3}{8}$ for any prover P^*. Let P^* be an arbitrary prover. Consider the $2^m \times k$ table where the $(r,i)^{\text{th}}$ entry is '1' iff $r \oplus s_i \notin A_x^{V_i,P^*}$. (That is, the $(r,i)^{\text{th}}$ entry is '1' iff the simulated V_i accepts x on random input $r \oplus s_i$, and after interacting with P^*.) Consider the ith column in the table. The 2^m entries in each column of the table correspond to the outputs of the original V, after an interaction with P^*, on all 2^m random inputs in $\{0,1\}^m$. However, V accepts x, after interacting with P^*, with probability at most $\frac{1}{4}$. Thus each column contains at most $\frac{1}{4}2^m$ '1' entries. Thus at most a quarter of the entries in the table are '1'. Thus at most $\frac{3}{8}2^m$ rows have at least $\left\lceil \frac{2k}{3} \right\rceil$ '1' entries. However, V' accepts on random input r, after interacting with P^*, iff the r^{th} row contains at least $\left\lceil \frac{2k}{3} \right\rceil$ '1' entries. Thus V' accepts with probability at most $\frac{3}{8}$, for any P^*. $\square$

We note that V' uses the same number of random bits as V. The number of communication bits used by P' is $k = O(m)$ times the number of communication bits used by P. A similar proof shows that the class of one-sided-error Probabilistically Checkable Proofs (PCP) equals two-sided-error PCP. In the PCP model the number of queries is also multiplied by an $O(r)$ factor.

Acknowledgments. I am grateful to Oded Goldreich for his devoted assistance in this work. In particular, it was Oded who suggested that the roles of y and z be interchangeable in the definition of the relation R in the proof of Theorem 2. I also thank Alex Russell and Ravi Sundaram for sharing their independently discovered results with me, and for agreeing that I include their proof of Theorem 3 in this paper.

References

[AS] N. Alon and J. Spencer, *The Probabilistic Method*, Wiley, 1992.

[B] L. Babai and S. Moran. Arthur-Merlin Games: A Randomized Proof System, and a Hierarchy of Complexity Classes. *J. Comput. System Sci.,* No. 36, 1988, pp. 254-276.

[FGMSZ] M. Furer, O. Goldreich, Y. Mansour, M. Sipser and S. Zachos. On Completeness and Soundness in Interactive Proof Systems. *Advances in Computing Research,* V. 5, 1989, pp. 429-442.

[GMR] Shafi Goldwasser, Silvio Micali and Charlie Rackoff. The knowledge Complexity of Interactive Proofs. *SIAM J. on Computing,* V. 18, No. 1, 1989, pp 186-208.

[GS] Shafi Goldwasser and Mike Sipser. Private Coins versus Public Coins in Interactive Proof Systems. *Advances in Computing Research,* V. 5, 1989, pp. 73-90.

[L] Clemens Lautemann. BPP and the Polynomial Hierarchy. *IPL* No. 17, 1983, pp. 215-217.

[RS] Alex Russell and Ravi Sundaram. Symmetric Alternation Captures BPP. *Technical Report MIT-LCS-TM-541,* 1995.

[Si] Mike Sipser. A Complexity Theoretic Approach to Randomness. *15th STOC,* 1983, pp. 330-335.

[St] Larry J. Stockmeyer. The polynomial Time Hierarchy. *Theoretical Computer Science* No. 3, 1976, pp. 1-22.

[Sh] Adi Shamir. $IP = PSPACE$. *31st FOCS,* 1990. pp. 11-15.

[W] Celia Wrathall. Complete Sets and the Polynomial-Time Hierarchy. *Theoretical Computer Science* No. 3, 1976, pp. 23-33.