# Another proof that $\mathcal{BPP} \subseteq \mathcal{PH}$ (and more)

Oded Goldreich*
Department of Computer Science
Weizmann Institute of Science
Rehovot, ISRAEL.
E-mail: oded@wisdom.weizmann.ac.il

David Zuckerman†
Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712-1188.
E-mail: diz@cs.utexas.edu

September 29, 1997

**Abstract**

We provide another proof of the Sipser–Lautemann Theorem by which $\mathcal{BPP} \subseteq \mathcal{MA}$ ($\subseteq \mathcal{PH}$). The current proof is based on strong results regarding the amplification of $\mathcal{BPP}$, due to Zuckerman. Given these results, the current proof is even simpler than previous ones. Furthermore, extending the proof leads to two results regarding $\mathcal{MA}$: $\mathcal{MA} \subseteq \mathcal{ZPP}^{\mathcal{NP}}$ (which seems to be new), and that two-sided error $\mathcal{MA}$ equals $\mathcal{MA}$. Finally, we survey the known facts regarding the fragment of the polynomial-time hierarchy which contains $\mathcal{MA}$.

**Keywords:** BPP, The Polynomial-Time Hierarchy, Interactive Proof Systems (AM and MA), Randomness–Efficient Error Reduction (Amplification).

*Work done while visiting LCS, MIT.

†Supported in part by NSF NYI Grant No. CCR-9457799, a David and Lucile Packard Fellowship for Science and Engineering, and an Alfred P. Sloan Research Fellowship.

# 1 Introduction

Non-trivial results, showing containment of fundamental complexity classes in one another, are quite rare. One of the first such results is Sipser's Theorem [12] by which $\mathcal{BPP}$ is contained in the Polynomial-Time Hierarchy. A simpler proof, placing $\mathcal{BPP}$ even lower in this hierarchy, was presented by Lautemann [10]. Although not stated in these (subsequently introduced) terms, Lautemann's proof actually establishes –

**Theorem 1** (The Sipser–Lautemann Theorem): $\mathcal{BPP} \subseteq \mathcal{MA}$.

See definitions in next section.

In this note, we present an alternative proof of the Sipser–Lautemann Theorem. Our proof relies on powerful results regarding randomness–efficient error reduction (a.k.a amplification) for $\mathcal{BPP}$. Given these powerful results, our proof is almost a triviality.

Using similiar arguments, we show that $\mathcal{MA} \subseteq \mathcal{ZPP}^{\mathcal{NP}}$ (re-establishing a theorem of Zachos and Heller [14] by which $\mathcal{BPP} \subseteq \mathcal{ZPP}^{\mathcal{NP}}$). It follows that $\mathcal{NP}^{\mathcal{BPP}} \subseteq \mathcal{ZPP}^{\mathcal{NP}}$. To the best of our knowledge, these results were not known before.

The purpose of this note is three-fold: Firstly to demonstrate the power of the currently known results regarding randomness–efficient error reduction. We believe that these results have not been fully assimilated into complexity theory and are yet to be exploited by it. Secondly we wish to focus attention on the fragment of the polynomial-time hierarchy which contains $\mathcal{MA}$. It seems that this fragment gives rise to some challenges which may be within our current reach. Finally, we take the oppertunity to prove the new result claimed above.

# 2 Background

## 2.1 BPP and Amplification

**Definition 1** (The class BPP): *For any language $L$, we denote by $\chi_L$ the characteristic function of the language; that is, $\chi_L(x) = 1$ if $x \in L$ and $\chi_L(x) = 0$ otherwise. A language $L$ is in $\mathcal{BPP}$ if there exists a probabilistic polynomial-time machine $M$ such that for every $x \in \{0,1\}^*$*

$$\text{Prob}(M(x) \neq \chi_L(x)) \ \leq \ \frac{1}{3}$$

*where the probability is taken uniformly over the internal coin tosses of $M$.*

The error probability in the above procedure can be reduced by repetitions (a process hereafter referred to as *amplification*). The obvious way of doing so transforms a machine (as above) which, on input $x$, uses $p(|x|)$ coins into a machine having error probability at most $2^{-t(|x|)}$ which uses $O(t(|x|) \cdot p(|x|))$ coins (for any polynomial $t$). More efficient amplification procedures, utilizing Expander Random Walks and other tricks, yield the same error bound while using only $p(|x|) + (4 + o(1)) \cdot t(|x|)$ coins (see survey [6]). In particular, for any constant $c > 4$, using a sufficiently large polynomial $t$, we get a procedure which uses $c \cdot t(|x|)$ coins and has error probability bounded by $2^{-t(|x|)}$. An alternative construction due to Zuckerman provides, for any constant $c > 1$ and sufficiently large polynomial $t$, a procedure which uses $c \cdot t(|x|)$ coins and has error probability bounded by $2^{-t(|x|)}$. What is remarkable in the last procedure is that the number of coins used is essentially the logarithm of the error bound. Put in other words, the number of "bad" coin sequences can be made any (constant) root of the total number of coin sequences. In particular,

**Theorem 2** (Zuckerman's efficient amplification of BPP [15]): *For any language $L$ in $\mathcal{BPP}$, there exists a polynomial-time recognizable binary relation $R$ and a polynomial $p$ such that*

$$|\{r \in \{0,1\}^{p(|x|)} : R(x,r) \neq \chi_L(x)\}| < 2^{p(|x|)/3}$$

### 2.2 The complexity class MA

**Definition 2** (The class MA): *A language $L$ is in $\mathcal{MA}$ if there exists a polynomial-time recognizable 3-ary relation $V$ and polynomials $p, q$ so that*

- *If $x \in L$ then there exists $w \in \{0,1\}^{q(|x|)}$ so that for every $r \in \{0,1\}^{p(|x|)}$, $V(x,w,r) = 1$.*
- *If $x \notin L$ then for every $w \in \{0,1\}^{q(|x|)}$*

$$\mathrm{Prob}_r(V(x,w,r) = 1) \ \leq \ \frac{1}{2}$$

*where the probability is taken uniformly over all $r \in \{0,1\}^{p(|x|)}$.*

The class $\mathcal{MA}$, introduced by Babai [1], consists of languages having a Merlin–Arthur proof system: The prover (Merlin) sends a certificate (denoted $w$ above) to the verifier (Arthur) who assesses it probabilistically (by tossing coins $r$ and applying the predicate $V$). Merlin–Arthur proof systems are a degenerate type of interactive proof systems (introduced by Goldwasser, Micali and Rackoff [7] and Babai [1]). Actually, in a Merlin–Arthur proof system there is no real interaction. Instead, it is instructive to view $\mathcal{MA}$ as *the* randomized version of $\mathcal{NP}$: Here the "certificates" (for membership) can be verified via a randomized procedure and errors may occur (alas with bounded probability).

# 3 A proof of the Sipser–Lautemann Theorem

### 3.1 The proof itself

Using Zuckerman's efficient amplification of BPP, we present the following MA proof system. Specifically, we will refer to the relation $R$ and the polynomial $p$ guaranteed in Theorem 2.

**The protocol.** On input $x$, both parties compute $m = p(|x|)$, and proceed as follows.

1. Merlin tries to select $r' \in \{0,1\}^{m/2}$ so that $R(x, r'r'') = 1$ for all $r'' \in \{0,1\}^{m/2}$. Merlin sends $r'$ to Arthur.
2. Arthur selects $r'' \in \{0,1\}^{m/2}$ uniformly and accepts if and only if $R(x, r'r'') = 1$.

**Analysis of the above protocol.** If $x \in L$ then there are at most $2^{m/3}$ possible $r$'s for which $R(x,r) = 0$. Thus there are at most $2^{m/3}$ prefixes $r' \in \{0,1\}^{m/2}$ for which some $r''$ exists so that $R(x, r'r'') = 0$. Merlin may just select any of the other $2^{m/2} - 2^{m/3}$ prefixes and make Arthur always accept. On the other hand, if $x \notin L$ then there are at most $2^{m/3}$ possible $r$'s for which $R(x,r) = 1$. Thus, for each $r' \in \{0,1\}^{m/2}$

$$\mathrm{Prob}_{r'' \in \{0,1\}^{m/2}}(R(x, r'r'') = 1) \ \leq \ \frac{2^{m/3}}{2^{m/2}} \ \ll \ \frac{1}{2}$$

## 3.2 Discussion

What we have done is partition the space of all ($2^m$) possible coin-tosses outcomes into ($2^{m/2}$) subsets of equal size. What we used is

1. The number of bad outcomes is smaller than the number of subsets (and so there exists a subset with no bad outcomes). This was used to analyze the case $x \in L$.

2. The number of bad outcomes is much smaller than the size of each subset (and so each subset contains a majority of good outcomes). This was used to analyze the case $x \notin L$.

Thus, what we have used is the fact that number of bad outcomes is much smaller than the square root of the total number of outcomes. The fact that any BPP-machine can be transformed into a machine for which the above hold is highly non-trivial. We believe that this fact (or known generalizations of it) may find further applications in complexity theory.

**Comparison to Lautemann's proof.** Recall that Lautemann's proof has the prover send the verifier $t = m/\log_2 m$ strings, $s_1, ..., s_t$, and the verifier tosses coins $r \in \{0,1\}^m$ and accepts iff $R(x, r \oplus s_i) = 1$ holds for some $i$. The existence of an appropriate sequence of strings is proven by an elementary probabilistic argument. Actually, $s_1$ may be any fixed string (e.g., $0^m$) and so needs not be sent (by the prover). We observe that IF we start with $R$ as guaranteed by Theorem 2, then $t = 2$ suffices. This gets us very close to the proof above. In fact, the probabilistic argument of Lautemann reduces to the trivial counting argument above. Thus, using Theorem 2 allows also a simplification of Lautemann's argument, although the proof presented earlier is believed to be simpler: Technically speaking, we have the prover send only $m/2$ bits (rather than $m$ required in the simplified Lautemann's argument), the verifier tosses only $m/2$ coins (again, rather than $m$), and the predicate $R$ is evaluated only once (rather than twice).

## 3.3 Two-sided error equals one-sided error for MA

Both Lautemann's proof as well as ours can be extended to show that a two-sided error version of $\mathcal{MA}$ equals the one-sided error defined above. (This provides an alternative proof to the one presented in [13].) We mention that interactive proof systems with zero error collapse to $\mathcal{NP}$, whereas for all (higher than MA) levels of the interactive proof hierarcy, the two-sided error version equals the one-sided one [5].

**Definition 3** (Two-sided version of MA): *A language $L$ is in $\mathcal{MA}_2$ if there exists a polynomial-time recognizable 3-ary relation $V$ and polynomials $p, q$ so that*

- *If $x \in L$ then there exists $w \in \{0,1\}^{q(|x|)}$ so that*
$$\mathrm{Prob}_r(V(x,w,r) = 1) \geq \frac{2}{3}$$

- *If $x \notin L$ then for every $w \in \{0,1\}^{q(|x|)}$*
$$\mathrm{Prob}_r(V(x,w,r) = 0) \geq \frac{2}{3}$$

*In both cases, the probability is taken uniformly over all $r \in \{0,1\}^{p(|x|)}$.*

Note that $\mathcal{NP}^{\mathcal{BPP}} \subseteq \mathcal{MA}_2$ (as Merlin may send an accepting computation of the non-deterministic polynomial-time oracle-machine and Arthur may verifier the validity of the oracle answers by running a probabilistic decision procedure of negligible two-sided error).

**Theorem 3** [13, Thm 2(i)]: $\mathcal{MA} = \mathcal{MA}_2$.

**Proof:** For every $x \in L$ we consider $w$ as guaranteed by the first condition, whereas for $x \notin L$ we consider any $w \in \{0,1\}^{q(|x|)}$. Both Lautemann's proof and ours extend to the $\mathcal{BPP}$ promise problem, $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$, where

$$\begin{aligned} \Pi_{\text{YES}} &\stackrel{\text{def}}{=} \{(x,w) : \text{Prob}_r(V(x,w,r)=1) \ \geq \ \frac{2}{3}\} \\ \Pi_{\text{NO}} &\stackrel{\text{def}}{=} \{(x,w) : x \notin L\} \\ &\subseteq \{(x,w) : \text{Prob}_r(V(x,w,r)=0) \ \geq \ \frac{2}{3}\} \end{aligned}$$

In particular, the amplification technique of Zuckerman applies also to this case and so we obtain a predicate $V'$ and a polynomail $q'$ such that

$$\begin{array}{ll} \forall (x,w) \in \Pi_{\text{YES}} & |\{r \in \{0,1\}^{q'(|x|)} : V(x,w,r)=0\}| < 2^{q'(|x|)/3} \\ \forall (x,w) \in \Pi_{\text{NO}} & |\{r \in \{0,1\}^{q'(|x|)} : V(x,w,r)=1\}| < 2^{q'(|x|)/3} \end{array}$$

Thus, we augment the above MA-protocol, as follows. On input $x$, with $m = q'(|x|)$, Merlin sends $(w, r')$, where $|r'| = m/2$, and Arthur uniformly selects $r'' \in \{0,1\}^{m/2}$ and accepts if and only if $V'(x,w,r'r'') = 1$. As before, in case $x \in L$, Merlin can make Arthur accept for every choice of $r''$; whereas, in case $x \notin L$, for any choice of $w, r'$, Arthur accepts with negligible probability. ■

# 4 MA is contained in ZPP with an NP-oracle

**Definition 4** (The class ZPP): *A language $L$ is in $\mathcal{ZPP}$ if there exists a probabilistic polynomial-time machine $M$ such that for every $x \in \{0,1\}^*$*

$$\begin{aligned} \text{Prob}(M(x) = \chi_L(x)) &\geq \frac{1}{2} \\ \text{Prob}(M(x) = 1 - \chi_L(x)) &= 0 \end{aligned}$$

*where the probability is taken uniformly over the internal coin tosses of $M$.*

Thus, the ZPP machine either gives the correct answer or gives no answer at all. Clearly $\mathcal{ZPP} = \mathcal{RP} \cap \text{co}\mathcal{RP}$ (actualy, $\mathcal{ZPP}$ is sometimes defined this way). We start by providing an alternative proof to a result of Zachos and Heller

**Theorem 4** [14, P. 132, Cor. 3]: $\mathcal{BPP} \subseteq \mathcal{ZPP}^{\mathcal{NP}}$.

**Proof:** Using the same amplification as above, we construct a probabilistic polynomial-time oracle machine, $M$, which on input $x$ operates as follows (where $m = p(|x|)$):

1. Selects $\sigma \in \{0,1\}$ uniformly (as guess for $\chi_L(x)$);

2. Selects $r' \in \{0,1\}^{m/2}$ uniformly;
3. Queries the oracle on whether $(x, \sigma, r')$ is in the following co$\mathcal{NP}$ set

$$\{(y, \tau, s) : \forall w \in \{0,1\}^{|s|},\ R(y, sw) = \tau\}$$

4. If the oracle answers YES then the machine outputs $\sigma$. Otherwise it halts with no output.

Recall that by the above amplification, for any $x$,

- For any $r'$,

$$|\{r'' \in \{0,1\}^{m/2} : R(x, r'r'') \neq \chi_L(x)\}| < 2^{m/2}$$

  and so the oracle never answers YES on query $(x, r', 1 - \chi_L(x))$. Thus, the machine never outputs the wrong answer.

- $$\mathrm{Prob}_{r'}(\forall r'' \in \{0,1\}^{m/2},\ R(x, r'r'') = \chi_L(x)) > \frac{1}{2}$$

  and so with probability at least 1/4, over the choices of $\sigma$ and $r'$, the oracle answers YES (and the machine produces a 0-1 output).

Using straightforward amplification, the theorem follows. ■

Combining ideas from the last two proofs, we obtain.

**Theorem 5** *$\mathcal{MA} \subseteq \mathcal{ZPP}^{\mathcal{NP}}$.*

Combining Theorems 3 and 5 and observing that $\mathcal{NP}^{\mathcal{BPP}} \subseteq \mathcal{MA}_2$ (see above), it follows that $\mathcal{NP}^{\mathcal{BPP}} \subseteq \mathcal{ZPP}^{\mathcal{NP}}$.

**Proof:** We consider the same promise problem, $\Pi$, as in the proof of Theorem 3. We construct a probabilistic polynomial-time oracle machine, $M$, which on input $x$ operates as follows (where $n = q(|x|)$ and $m = p(|x|)$):

1. Uniformly selects $r_1, ..., r_{2n} \in \{0,1\}^m$, and ask the NP-oracle whether there exists a $w \in \{0,1\}^n$ so that $\wedge_{i=1}^{2n} V(x, w, r_i) = 1$.
2. In case oracle answers NO then the machine halts with output 0.
3. Otherwise, the machine uses the self-reducibility of the NP-oracle in order to find $w$ as in Item 1. That is, the machine asks queries of the form "does there exists a $w'' \in \{0,1\}^{n-|w'|}$ so that $\wedge_{i=1}^{2n} V(x, w'w'', r_i) = 1$."
4. Once $w$ is found, the machine treats $(x, w)$ as an input to the $\mathcal{BPP}$ promise problem $\Pi$ and proceeds as in the proof of Theorem 4. Specifically, it considers a strong amplification of this promise problem, selects a random prefix, and queries whether all suffixes make the original predicate evaluate to 1. If the oracle answers YES then $M$ halts with output 1; otherwise, $M$ halts with no output. (We stress that we never output 0 in this step.)

Given the analysis in the proof of Theorem 4, it suffices to note the following

- For any $x$ (either in $L$ or not), if $\text{Prob}_r(V(x,w,r)=1) < \frac{2}{3}$ then

$$\text{Prob}_{r_1,\ldots,r_{2n}}(\wedge_{i=1}^{2n} V(x,w,r)=1) \;<\; (2/3)^{2n} \;<\; \frac{1}{2}\cdot 2^{-n}$$

  It follows that for $x \notin L$, with probability at least $1/2$ (over the choices of the $r_i$'s), the oracle answer in Step 2 is NO and machine $M$ outputs 0.

- It also follows that, with probability at least $1/2$, none of the $w$'s violating $(x,w) \in \Pi_{\text{YES}}$ will be reconstructed in Step 3.

- Thus, for any $x \in L$, with probability at least $1/2$, Step 4 is invoked with $(x,w) \in \Pi_{\text{YES}}$. In this case (by the analysis in the proof of Theorem 4) machine $M$ outputs 1 with probability at least $1/2$.

- On input $x \in L$, the machine never outputs 0 (since 0 is output only in Step 1 upon a condition which never holds when $x \in L$).

- On input $x \notin L$, the machine never outputs 1 (since 1 is output only in Step 4 upon a condition which, by the analysis in the proof of Theorem 4, never holds when $(x,w) \in \Pi_{\text{NO}}$).

Thus, we have seen that, for any $x$, the machine never errs and it produces output with probability at least $1/4$. ■

## 5 The bigger picture – the complexity classes around MA

### 5.1 Definitions

In the following definitions all relations hold only on arguments of polynomially related length (i.e., all tuples in a relation have arguments which are of length polynomial in the length of the first argument). Likewise, all quantifiers range over arguments of such lengths.

**Definition 5** (Traditional classes – classes of the 1970's:)

- *$L$ is in $\Sigma_2^P = \mathcal{NP}^{\mathcal{NP}}$ (resp., $\Pi_2^P = \text{co}\mathcal{NP}^{\mathcal{NP}}$) if there exists a polynomial-time recognizable 3-ary relation $R$ so that*

$$\begin{aligned} L &= \{x : \exists y \forall z\; R(x,y,z)=1\} \\ (\textit{resp.},\; L &= \{x : \forall y \exists z\; R(x,y,z)=1\}) \end{aligned}$$

- *A language $L$ is in $\Delta_2^P = \mathcal{P}^{\mathcal{NP}}$ if there exists a deterministic polynomial-time oracle machine $M$ and an $\mathcal{NP}$-set $A$ such that $x \in L$ iff $M^A(x)=1$, for all $x$'s.*

- *A language $L$ is in $\mathcal{RP}$ if there exists a probabilistic polynomial-time machine $M$ such that*

$$\begin{aligned} x \in L &\implies \text{Prob}(M(x)=1) \;\geq\; \frac{1}{2} \\ x \notin L &\implies \text{Prob}(M(x)=1) \;=\; 0 \end{aligned}$$

**Definition 6** ($\mathcal{AM}$ [1] – a class of the 1980's:) *A language $L$ is in $\mathcal{AM}$ if there exists a polynomial-time recognizable 3-ary relation $V$ and polynomials $p, q$ so that*

- *If $x \in L$ then for every $r \in \{0,1\}^{p(|x|)}$ there exists $w \in \{0,1\}^{q(|x|)}$ so that $V(x,r,w)=1$.*
- *If $x \notin L$ then*
$$\mathrm{Prob}_r(\exists w \text{ s.t. } V(x,r,w)=1) \leq \frac{1}{2}$$
*where the probability is taken uniformly over all $r \in \{0,1\}^{p(|x|)}$.*

The class $\mathcal{AM}$, introduced by Babai [1], consists of languages having an Arthur–Merlin proof systems: The verifier (Arthur) challenges the prover (Merlin) with a random query, denoted $r$, and given the prover's answer (denoted $w$) makes a decision using the predicate $V$. In contrast to Merlin–Arthur systems, here we have a real interaction between the prover and the verifier. The class $\mathcal{AM}$ coincides with the class of languages having constant-round interactive proof systems [1, 8]. Thus, it is the lowest level of the hierarchy of "real" interactive proofs [1, 7] (i.e., interactive proofs, which unlike $\mathcal{NP}$ and $\mathcal{MA}$, are really interactive).

**Definition 7** ($\mathcal{S}_2^P$ [4, 11] – a class of the 1990's:) *$L$ is in $\mathcal{S}_2^P$ if there exists a polynomial-time recognizable 3-ary relation $R$ so that for every $x \in \{0,1\}^*$*

$$\exists y \forall z \qquad R(x,y,z) = \chi_L(x) \tag{1}$$
$$\exists z \forall y \qquad R(x,y,z) = \chi_L(x) \tag{2}$$

The class $\mathcal{S}_2^P$ was introduced independently by Canetti [4] and Russell and Sundaram [11] with the motivation of providing a low "symmetric alternation class" which contains $\mathcal{BPP}$. Indeed, Canetti [4] has extended Lautemann's proof to show that $\mathcal{BPP} \subseteq \mathcal{S}_2^P$, whereas Russell and Sundaram [11] showed that $\mathcal{MA} \subseteq \mathcal{S}_2^P$ (and thus $\mathcal{BPP} \subseteq \mathcal{S}_2^P$).

## 5.2 Known Inclusions

We recall known inclusions between the classes defined above. For sake of self-containment, we present proofs as well. Recall that we already have $\mathcal{BPP} \subseteq \mathcal{MA}$.

**Syntactical Facts:**

1. $\mathcal{P} \subseteq \mathcal{RP} \subseteq \mathcal{NP} \subseteq \mathcal{MA}$.
2. $\mathcal{RP} \subseteq \mathcal{BPP}$.
3. $\mathcal{RP} \subseteq \mathrm{co}\mathcal{MA}$. (This is a syntactical fact, although it can also be derived from $\mathcal{RP} \subseteq \mathcal{BPP}$ and $\mathcal{BPP} \subseteq \mathcal{MA}$.)
4. $\mathcal{NP} \cup \mathrm{co}\mathcal{NP} \subseteq \mathcal{P}^{\mathcal{NP}}$.
5. $\mathcal{AM} \subseteq \Pi_2^P$.
6. $\mathcal{S}_2^P \subseteq \Sigma_2^P \cap \Pi_2^P$. (Actually, the transparent syntactical facts are the inclusion $\mathcal{S}_2^P \subseteq \Sigma_2^P$ and the closure of $\mathcal{S}_2^P$ under complement.)
7. $\mathcal{ZPP}^{\mathcal{NP}} \subseteq \Sigma_2^P \cap \Pi_2^P$. (Here the transparent fact is $\mathcal{ZPP}^{\mathcal{NP}} \subseteq \mathcal{RP}^{\mathcal{NP}} \subseteq \mathcal{NP}^{\mathcal{NP}} = \Sigma_2^P$.)

**Proposition 6** [1]: *$\mathcal{MA} \subseteq \mathcal{AM}$.*

**Proof:** We use a naive amplification to reduce the error probability in the Merlin–Arthur game to obtain error which is substantially smaller than the reciprocal of the number of possible Merlin messages. Specifically, we obtain a polynomial-time recognizable 3-ary relation $V$ and polynomials $p, q$ so that

1. If $x \in L$ then there exists $w_0 \in \{0,1\}^{q(|x|)}$ so that for every $r \in \{0,1\}^{p(|x|)}$, $V(x, w_0, r) = 1$.
2. If $x \notin L$ then for every $w \in \{0,1\}^{q(|x|)}$

$$\mathrm{Prob}_r(V(x,w,r)=1) \;<\; \frac{1}{2}\cdot 2^{-q(|x|)}$$

Thus,

$$\begin{aligned}\mathrm{Prob}_r(\exists w \in \{0,1\}^{q(|x|)} : V(x,w,r)=1) \;&\leq\; \sum_{w\in\{0,1\}^{q(|x|)}} \mathrm{Prob}_r(V(x,w,r)=1)\\ &<\; \frac{1}{2}\end{aligned}$$

We construct an Arthur–Merlin proof system (defined by a new predicate $V'$) by merely reversing the order of moves in the above proof system, and using essentially the same decision predicate as above: That is, we let $V'(x,r,w) \stackrel{\rm def}{=} V(x,w,r)$. This potentially makes the task of Merlin easier, and so we need only worry about the case $x \notin L$ (which we handle easily using the above bound). Specifically, for the case $x \in L$, we may use the string $w_0$ (guaranteed in Item 1) as Merlin's response to any challenge $r$ (and so $V'(x,r,w_0) = V(x,w_0,r) = 1$ for all $r$'s). For the case $x \notin L$ we use the bound in Item 2 and so $\mathrm{Prob}_r(\exists w \in \{0,1\}^{q(|x|)} : V'(x,r,w)=1) < 0.5$. The proposition follows. ■

**Proposition 7** [11]: $\mathcal{MA} \subseteq \mathcal{S}_2^P$.

**Proof:** We use the same amplification as in the previous proof. Here we write the case $x \notin L$ as

$$\forall w \in \{0,1\}^{q(|x|)} \qquad |\{r \in \{0,1\}^{p(|x|)} : V(x,w,r)=1\}| \;<\; 2^{p(|x|)-q(|x|)} - 1$$

We define a relation $R$ (for the class $\mathcal{S}_2^P$) so that $R(x,y,z)=1$ if $|y|=|z|=q(|x|)$ and at least one of the following two conditions holds

1. $y = w0^{p(|x|)-q(|x|)}$ and $V(x,w,z)=1$.
2. $z = w0^{p(|x|)-q(|x|)}$ and $V(x,w,y)=1$.

Clearly, this predicate is symmetric with respect to $y$ and $z$. Thus, we only show, for any $x$, the existence of a string $y$ such that, for all $z$'s, $R(x,y,z) = \chi_L(x)$. Let us shorthand $m = p(|x|)$ and $n = q(|x|)$. For $x \in L$ there exists $w \in \{0,1\}^n$ such that for all $r \in \{0,1\}^m$, $V(x,w,r)=1$. Thus, there exists $y = w0^{m-n} \in \{0,1\}^m$ such that for all $z \in \{0,1\}^m$, $R(x,y,z)=1$. We now turn to the case where $x \notin L$: In this case,

$$|\{r : \exists w \;\text{ s.t. }\; V(x,w,r)=1\}| \;<\; 2^n\cdot(2^{n-m}-1) \;=\; 2^m - 2^n$$

Thus, there exists $r \in \{0,1\}^m \setminus \{0,1\}^n 0^{m-n}$ so that for every $w \in \{0,1\}^n$, $V(x,w,r)=0$. Given such an $r$, we prove that for all $z$'s $R(x,r,z)=0$. This holds since $R(x,r,z)=1$ requires either $r$ ending with $0^{m-n}$ (which does not hold by our choice) or $z = w0^{n-m}$ with $V(x,w,r)=1$ (which again cannot hold). ■

**Proposition 8** [11]: $\mathcal{P}^{\mathcal{NP}} \subseteq \mathcal{S}_2^P$.

**Proof:** Let $M$ be a (deterministic) oracle machine recognizing $L$ when given access to the NP-complete language $A$. We say that a string $T$ is a **valid transcript of** $M(x)$ if there exists *some* oracle so that $T$ describes the computation of $M$ on input $x$ and access to this oracle. Note that the oracle's answers in a valid transcript of $M(x)$ do *not* necessarily agree with the language $A$. A valid transcript is said to be **supported** by a sequence of pairs $\overline{s}$ if for each oracle query $q$ in $T$ which was answered by 1 there is a pair $(q, w)$ in $\overline{s}$, where $w$ is an NP-witness for membership of $q$ in $A$. A valid transcript is said to be **consistent** with a sequence of pairs $\overline{s}$ if for each oracle query $q$ in $T$ which was answered by 0 there is no pair $(q, w)$ in $\overline{s}$, where $w$ is an NP-witness for membership of $q$ in $A$. We consider a fixed parsing of strings into pairs $(T, \overline{s})$, where $\overline{s}$ is a sequence of pairs.

We are now ready to define a relation $R$ (for the class $\mathcal{S}_2^P$): For $y = (T, \overline{s})$ and $z = (T', \overline{s}')$, we let $R(x, y, z) \stackrel{\text{def}}{=} \sigma$ if at least one of the following two conditions holds

1. $T$ is a valid transcript of $M(x)$ with output $\sigma$, supported by $\overline{s}$ and consistent with $\overline{s}'$.

2. $T'$ is a valid transcript of $M(x)$ with output $\sigma$, supported by $\overline{s}'$ and consistent with $\overline{s}$.

In case none of the conditions hold, $R(x, y, z)$ may be defined arbitrarily. Observe that $R$ is well-defined (i.e., it can not be the case that $T$ and $T'$ are both valid, supported and consistent but with different outputs). Here we use the fact that $M$ is deterministic and so given the same oracle answers it must yield the same output. Also, given two valid transcripts which differ on some oracle answer it cannot be that both transcripts are supported and consistent with the same two sequences of pairs.[1] Finally, observe that for every $x$ there exists a pair $(T, \overline{s})$ with output $\chi_L(x)$ so that $T$ is a valid transcript of $M(x)$, supported by $\overline{s}$ and consistent with any possible sequence of pairs. ■

### 5.3 Conjectured Separations

Below we present some common conjectures.

**Conjecture 1** (The leading conjecture of TOC): $\mathcal{P} \neq \mathcal{NP}$.

**Conjecture 2** (most widely believed): $\mathcal{NP} \not\subseteq \mathcal{BPP}$.

**Conjecture 3** (most widely believed): $\mathcal{NP} \neq \text{co}\mathcal{NP}$.

**Conjecture 4** (widely believed): *The Polynomial-Time Hierarchy does not collapse.*

Conjecture 4 implies the following (see [3]):

**Conjecture 5** (widely believed): $\text{co}\mathcal{NP} \not\subseteq \mathcal{AM}$.

We believe that Conjecture 5 is interesting on its own.

[1] Consider the first conflicting answer and suppose, without loss of generality, that the transcript $T$ the answer is 1. Since $T$ is supported by a sequence of pairs $\overline{s}$, it cannot be the case that $T'$ (in which the answer to the same query is 0) is consistent with $\overline{s}$.

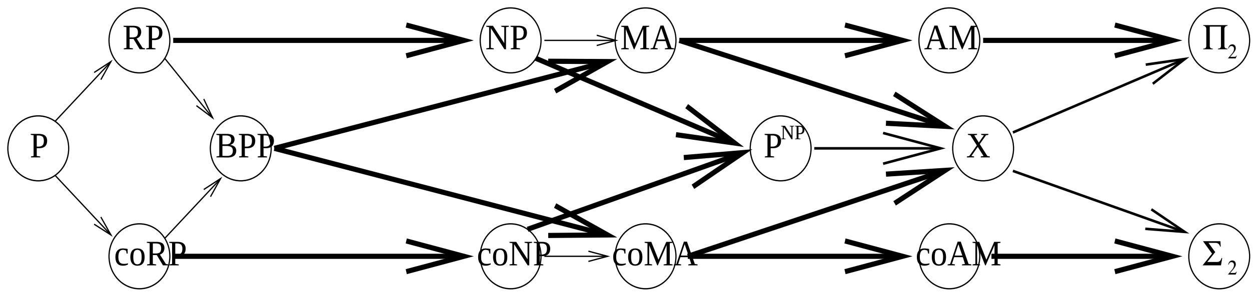


Figure 1: Arrows indicate containment between classes, with $\mathcal{C}_1 \to \mathcal{C}_2$ indicating that $\mathcal{C}_1 \subseteq \mathcal{C}_2$. Bolder (and bigger) arrows indicate conjectured gaps between the classes. $X$ stands for either $\mathcal{S}_2^P$ or $\mathcal{ZPP}^{\mathcal{NP}}$ (we do not know how these two classes are related).

### 5.4 Conjectured Inclusions

What we know combined with what is widely believed is depicted in Figure 1. We note that some of the inclusions which were not conjectured to be separations are believed to be equalities or "close to it". In particular, it is widely believed that $\mathcal{BPP}$ is very close to $\mathcal{P}$. This belief is supported, among other things, by the conjecture that (uniform) exponential-time cannot be computed by subexponential-size (non-uniform) circuits [2, 9]. We note that the latter conjecture holds provided there exist strong one-way functions (i.e., ones which cannot be inverted by subexponential-sized circuits).

**The derandomization of BPP versus the derandomization of MA.** A trivial fact, rarely noted, is that results about derandomization of $\mathcal{BPP}$ are likely to imply results on the derandomization of $\mathcal{MA}$. This holds provided that the former results extend also to a generalization of $\mathcal{BPP}$ to promise problems. We note that all known derandomization results have this property. In the next proposition co$\mathcal{RP}$ is the class of promise problems of the form $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$, where there exists a probabilistic polynomial time machine $M$ so that

$$x \in \Pi_{\text{YES}} \implies \text{Prob}(M(x)=1) \;=\; 1$$
$$x \in \Pi_{\text{NO}} \implies \text{Prob}(M(x)=1) \;\leq\; \frac{1}{2}$$

**Proposition 9** (folklore): *Suppose that* co$\mathcal{RP} \subseteq \text{DTIME}(c(n))$*, for a time constructible function* $c : \mathsf{N} \mapsto \mathsf{N}$*. Then,* $\mathcal{MA} \subseteq \cup_{i\in\mathsf{N}} \text{NTIME}(c(n^i))$.

**Proof:** Each languages $L \in \mathcal{MA}$ gives rise to a promise problem $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$, where

$$\Pi_{\text{YES}} \stackrel{\text{def}}{=} \{(x,w) : \forall r \in \{0,1\}^{p(|x|)} \quad V(x,w,r)=1\}$$
$$\Pi_{\text{NO}} \stackrel{\text{def}}{=} \{(x,w) : x \notin L\}$$

with $V$ and $p$ as in Definition 2. Note that for every $x \in L$ there exists $w \in \{0,1\}^{q(|x|)}$ so that $(x,w) \in \Pi_{\text{YES}}$, and that for every $x \notin L$ and every $w \in \{0,1\}^{q(|x|)}$, $(x,w) \in \Pi_{\text{NO}}$. Also, for every $(x,w) \in \Pi_{\text{NO}}$ it holds

$$\text{Prob}_{r\in\{0,1\}^{p(|x|)}}(V(x,w,r)=1) \;\leq\; \frac{1}{2}$$

and so $\Pi \in$ co$\mathcal{RP}$. Using the hypothesis, we have $\Pi \in \text{DTIME}(c(n+q(n)))$, and so $L \in \text{NTIME}(c(n+q(n)))$. The proposition follows. ■

### 5.5 Challenges

1. Try to put $\mathcal{BPP}$ in $\mathcal{P}^{\mathcal{NP}}$. (Recall that $\mathcal{BPP}$ in in $\mathcal{ZPP}^{\mathcal{NP}}$.)
2. Try to put $\mathcal{MA}$ in $\mathcal{P}^{\mathcal{NP}}$. (This certainly implies (1).)
3. Try to put $\mathcal{RP}$ in co$\mathcal{NP}$. (Recall that $\mathcal{RP}$ is in co$\mathcal{MA}$.)
4. Try to put $\mathcal{AM}$ in $\Sigma_2^P \cap \Pi_2^P$.

## References


[1] L. Babai. Trading Group Theory for Randomness. In *17th STOC*, pages 421–429, 1985.

[2] L. Babai, L. Fortnow, N. Nisan and A. Wigderson. BPP has Subexponential Time Simulations unless EXPTIME has Publishable Proofs. *Complexity Theory*, Vol. 3, pages 307–318, 1993.

[3] R. Boppana, J. Håstad, and S. Zachos. Does Co-NP Have Short Interactive Proofs? *IPL*, 25, May 1987, pages 127–132.

[4] R. Canetti. On BPP and the Polynomial-time Hierarchy. *IPL*, 57, pages 237–241, 1996.

[5] M. Fürer, O. Goldreich, Y. Mansour, M. Sipser and S. Zachos. On Completeness and Soundness in Interactive Proof Systems. *Advances in Computing Research*, Vol. 5 (Randomness and Computation, S. Micali, ed.), pages 429–442, 1989.

[6] O. Goldreich. A Sample of Samplers – A Computational Perspective on Sampling. *ECCC*, TR97-020, May 1997.

[7] S. Goldwasser, S. Micali and C. Rackoff. The knowledge Complexity of Interactive Proofs. *SIAM J. on Computing,* V. 18, No. 1, 1989, pp 186–208.

[8] S. Goldwasser and M. Sipser. Private Coins versus Public Coins in Interactive Proof Systems. *Advances in Computing Research*, Vol. 5 (Randomness and Computation, S. Micali, ed.), pages 73–90, 1989.

[9] R. Impagliazzo and A. Wigderson. P=BPP if E requires exponential circuits: Derandomizing the XOR Lemma. In *29th STOC*, pages 220–229, 1997.

[10] C. Lautemann. BPP and the Polynomial Hierarchy. *IPL*, 17, pages 215–217, 1983.

[11] A. Russell and R. Sundaram. Symmetric Alternation Captures BPP. *Journal of Computational Complexity*, to appear. Preliminary version in Technical Report MIT-LCS-TM-541, 1995.

[12] M. Sipser. A Complexity Theoretic Approach to Randomness. *15th STOC,* 1983, pages 330–335.

[13] S. Zachos and M. Fürer. Probabilistic Quantifiers vs. Distrustful Adversaries. In *Proc. FST-TCS*, Springer-Verlag, Lecture Notes in Computer Science (Vol. 287), pages 443–455, 1987.

[14] S. Zachos and H. Heller. A decisive characterization of BPP. *Information and Control*, Vol. 69 (1-3), pages 125–135, 1986.

[15] D. Zuckerman. Simulating BPP Using a General Weak Random Source. *Algorithmica*, Vol. 16, pages 367–391, 1996.