Electronic Colloquium on Computational Complexity, Report No. 53 (2004)

Polylogarithmic Round Arthur-Merlin Games and Random-Self-Reducibility

A. Pavan[*] N. V. Vinodchandran[†]

June 10, 2004

Abstract

We consider Arthur-Merlin proof systems where (a) Arthur is a probabilistic quasi-polynomial time Turing machine, denoted $\mathrm{AM}_{\mathrm{qpoly}}$, and (b) Arthur is a probabilistic exponential time Turing machine, denoted $\mathrm{AM}_{\mathrm{exp}}$. We prove two new results related to these classes.

- We show that if co-NP is in $\mathrm{AM}_{\mathrm{qpoly}}$ then the exponential hierarchy collapses to $\mathrm{AM}_{\mathrm{exp}}$.
- We show that if SAT is polylogarithmic round adaptive random-self-reducible, then $\overline{\mathrm{SAT}}$ is in $\mathrm{AM}_{\mathrm{qpoly}}$ with a polynomial advice.

The first result improves a recent result of Selman and Sengupta (2004) who showed that the hypothesis collapses the exponential hierarchy to $\mathrm{S}_2^{\mathrm{exp}}.\mathrm{P}^{\mathrm{NP}}$; a complexity class which contains $\mathrm{AM}_{\mathrm{exp}}$. The second result implies that if SAT is polylogarithmic round adaptive random-self-reducible, then the exponential hierarchy collapses. This partially answers a question posed by Feigenbaum and Fortnow (1993) who showed that if SAT is logarithmic round adaptive random-self-reducible then the polynomial hierarchy collapses.

1 Introduction

Interactive proof systems and random-self-reducibility are some of the well studied notions in complexity theory. Goldwasser, Micali, and Rackoff [GMR85] introduced interactive proof systems and Babai [Bab85] introduced Arthur-Merlin games to study the interaction between randomness and nondeterminism. Later Goldwasser and Sipser [GS89] showed that these two classes are indeed the same. It is known that some problems in co-NP such as Graph Non isomorphism, that are not known to be in NP, can be accepted by a two round Arthur-Merlin protocol [GMW86]. This raised the obvious question of whether every language in co-NP can be accepted by a two round Arthur-Merlin Protocol. Boppana, Hastad, and Zachos [BHZ87] showed that if every language in co-NP has a constant Arthur-Merlin protocol, then the polynomial-time hierarchy collapses. Breakthrough works by Lund *et al.* [LFKN90], and Shamir [Sha92] showed that entire PSPACE can be accepted by an Arthur-Merlin protocol if we allow polynomially many rounds. This leaves the question of what happens when the number of rounds are bounded below by constant and bounded above by polynomial. Very recently Selman and Sengupta [SS04] showed that if co-NP has polylogarithmic round Arthur-Merlin protocol, then the exponential hierarchy collapses to $\mathrm{S}_2^{\mathrm{exp}}.\mathrm{P}^{\mathrm{NP}}$,

[*] Department of Computer Science, Iowa State University, pavan@cs.iastate.edu.

[†] Department of Computer Science and Engineering, University of Nebraska-Lincoln, vinod@cse.unl.edu.

ISSN 1433-8092

where $\mathrm{S}_2^{\mathrm{exp}}$ is the exponential version of the class $\mathrm{S}_2^{\mathrm{P}}$ introduced by Russell and Sundaram [RS98] and Canetti [Can96].

Our first result improves Selman and Sengupta's result. We show that if co-NP has polylogarithmic round Arthur-Merlin protocol then the exponential hierarchy in fact collapses to $\mathrm{AM}_{\mathrm{exp}}$. $\mathrm{AM}_{\mathrm{exp}}$ is the exponential version of AM and is contained in $\mathrm{S}_2^{\mathrm{exp}}.\mathrm{P}^{\mathrm{NP}}$. In addition to the improved upperbound, our proof of this result is very simple. We observe that the proof of the implication co-NP $\subseteq$ AM $\Rightarrow$ PH $\subseteq$ AM due to Boppana, Hastad, and Zachos [BHZ87] also works for time bounds other than the polynomial range. In particular, the same proof using quantifier switching (with different parameters) shows that co-NP $\subseteq \mathrm{AM}_{\mathrm{qpoly}} \Rightarrow \mathrm{PH}_{\mathrm{qpoly}} \subseteq \mathrm{AM}_{\mathrm{qpoly}}$, where $\mathrm{PH}_{\mathrm{qpoly}}$ denotes the quasi-polynomial time hierarchy and $\mathrm{AM}_{\mathrm{qpoly}}$ is the quasi-polynomial time version of AM. By a standard padding argument we get $\mathrm{PH}_{\mathrm{qpoly}} \subseteq \mathrm{AM}_{\mathrm{qpoly}} \Rightarrow \mathrm{EH} \subseteq \mathrm{AM}_{\mathrm{exp}}$ where EH denotes the exponential hierarchy. The result follows from the fact that $\mathrm{AM}_{\mathrm{qpoly}}$ can simulate polylogarithmic round Arthur-Merlin games [GVW01].

In the second part of the paper we study the notion of random-self-reducibility. Informally, a function f is *random-self-reducible*, if the value of f at any x can be computed by knowing the value of a few "random" instances y. Random-self-reducibility has played an important role in complexity theory. If f is random-self-reducible, then the complexity of computing f at a point is equivalent to computing the value of f on some randomly chosen points. This implies that the average-case complexity of f is same as the worst-case complexity of f. For example, Lipton [Lip91] showed that the Permanent function is random-self-reducible. From this it follows that the average-case complexity of Permanent is same as the worst-case complexity of Permanent [Lip91, GS92, FL92, CPS99, GRS00]. Random-self-reductions also play crucial role in program checkers [BK89, BLR90]. Ideas developed during the study of random-self-reducible functions were successfully used in the results of Lund *et al.* and Shamir [LFKN90, Sha92].

Since Permanent is #P complete, it follows that all #P-complete functions are random-self-reducible [FF93]. It is natural to ask whether NP-complete languages are random-self-reducible. Feigenbaum and Fortnow [FF93] studied this question. They showed that if any NP-complete language is *nonadaptive* random-self-reducible, then the polynomial-time hierarchy collapses to the third level. They also obtained the same consequence under the hypothesis that NP-complete languages are $O(\log n)$ *adaptive* random-self-reducible. They asked whether the result holds if we consider random-self-reductions that make more than $O(\log n)$ adaptive queries. Our second results gives a partial answer to their question.

We show, following Feigenbaum and Fortnow, that if a language L in NP is $\log^{O(1)} n$ adaptive random-self-reducible, then $\overline{L}$ is in $\mathrm{AM}_{\mathrm{qpoly}}^{\mathrm{poly}}$. $\mathrm{AM}_{\mathrm{qpoly}}^{\mathrm{poly}}$ denotes the class of languages accepted by a $\mathrm{AM}_{\mathrm{qpoly}}$ protocol when the Arthur has access to a polynomial advice string. Since $\mathrm{AM}_{\mathrm{qpoly}}^{\mathrm{poly}} \subseteq \mathrm{NP}/2^{\mathrm{polylog}}$, it follows that if NP-complete languages have polylog adaptive random-self-reductions, then the exponential hierarchy collapses. Finally we consider the relationship between random-self-reducibility and checkability [BK89]. We show that if SAT is nonadaptive random-self-reducible, then SAT is checkable with advice.

2 Preliminaries

We assume the definitions of standard complexity classes. Please refer to [BDG88, Pap94] for these and other standard complexity-theoretic definitions including the definitions of interactive complexity class Arthur-Merlin games. In this paper we deal with complexity classes defined using general parameter ranges. We present these notations first.

`lin` $= \bigcup_{c\geq 1} cn$ denotes the set of linear functions, `poly` $= \bigcup_{k\geq 1} n^k$ denotes the set of polynomials, `qpoly` $= \bigcup_{c\geq 1} 2^{(\log n)^c}$ denotes the set of quasi-polynomial functions, and `polylog` $= \bigcup_{k\geq 1} (\log n)^k$ denotes the set of polylogarithmic functions.

Definition. We call a time constructible function $l(n)$ *nice* if (a) $l(n) \geq n$, (b) $l(l(n)) \geq nl(n)$, and (c) $l(cn) \geq cl(n)$ for any constant $c > 1$

Notice that polynomials, quasi-polynomials, and exponentials are all nice functions. We will be dealing with only nice functions and use their properties implicitly in the proofs.

We consider several Arthur-Merlin classes for various parameter ranges.

- AM$[m(n), l(n)]$ denotes the class of languages accepted by $m(n)$ round Arthur-Merlin interactive protocol with maximum message length $l(n)$ by both Arthur and Merlin in each round.

- We denote AM$[2, l(n)]$ with AM$[l(n)]$.

- We consider polynomial, quasi-polynomial, and exponential versions of AM which are defined as follows.

  $\mathrm{AM} = \cup_k \mathrm{AM}[n^k]$, $\mathrm{AM}_{\mathrm{qpoly}} = \cup_c \mathrm{AM}[2^{\log^c n}]$, and $\mathrm{AM}_{\mathrm{exp}} = \cup_k \mathrm{AM}[2^{n^k}]$.

We also consider parameterized versions of the polynomial hierarchy.

- $\Sigma_k[f(n)]$ denotes the class of languages accepted by a Σ_k-machine where running time within a quantifier is bounded by $f(n)$, $\Pi_k[f(n)]$ is defined analogously.

- Polynomial, quasi-polynomial and exponential versions are defines as follows.

  $\Sigma_k^{\mathrm{P}} = \cup_c \Sigma_k[n^c]$, $\Sigma_k^{\mathrm{qpoly}} = \bigcup_c \Sigma_k[2^{(\log n)^c}]$, and $\Sigma_k^{\mathrm{exp}} = \cup_c \Sigma_k[2^{n^c}]$. These classes can also be defined using oracle Turing machines. For example $\Sigma_k^{\mathrm{exp}} = \mathrm{NEXP}^{\Sigma_{k-1}^{\mathrm{P}}}$.

  $\mathrm{PH} = \bigcup_k \Sigma_k^{\mathrm{P}}$, $\mathrm{PH}_{\mathrm{qpoly}} = \bigcup_k \Sigma_k^{\mathrm{qpoly}}$, and $\mathrm{EH} = \bigcup_k \Sigma_k^{\mathrm{exp}}$ where EH is the exponential hierarchy.

Finally, for a complexity class $\mathcal{C}$, and a nice function l, let $\mathcal{C}[l(\texttt{lin})]$ denotes $\bigcup_{c\geq 1} \mathcal{C}[l(cn)]$.

Random-self-reducibility

A function f is $k(n)$*-nonadaptive random-self-reducible* if there exist probabilistic polynomial-time computable functions g and h such that

- $\forall x, \Pr[g[x, f(h(1,x)), f(h(2,x)), \cdots f(h(k(|x|), x))] = f(x)] \geq 3/4.]$

- For every n, for every i, $1 \leq i \leq k(n)$, if $|x| = |y| = n$, the random variables $h(i,x)$ and $h(i,y)$ are identically distributed.

A function f is $k(n)$*-adaptive random-self-reducible* if there exists a probabilistic polynomial-time oracle Turing machine M such that, M makes $k(n)$-rounds of adaptive queries such that

- $\forall x, \Pr[M^f(x) = f(x)] \geq 3/4,$

- Given x, let $M^f(i,x)$ denote the random-variable corresponding to the ith query generated by M^f on x. For every n, and for every i, $1 \leq i \leq k(n)$, if $|x| = |y| = n$, then the random variables $M^f(i,x)$ and $M^f(i,y)$ are identically distributed.

As usual, we can amplify the success probabilities to $1 - 1/2^n$.

Blum and Kannan [BK89] introduced the notion of checkability. A language L is *checkable* if there exists a probabilistic polynomial-time oracle Turing machine M such that for every program P

- If $P(x) \neq L(x)$, then $\Pr[M^P(x) = \text{incorrect}] \geq 3/4$.
- If $\forall x, P(x) = L(x)$, then $\Pr[M^P(x) = \text{correct}] \geq 3/4$.

We say a language L is *checkable with advice* if M has access to a polynomial amount of advice.

3 Polylogarithmic Round Arthur-Merlin Games and the Exponential Hierarchy

In this section we show that if co-NP has polylogarithmic rounds Arthur-Merlin games then the exponential hierarchy collapses to $\mathrm{AM}_{\mathrm{exp}}$. Since $\mathrm{AM}_{\mathrm{exp}} \subseteq \mathrm{S}_2^{\mathrm{exp}}.\mathrm{P}^{\mathrm{NP}}$, this improves a recent result of Selman and Sengupta [SS04] who show that under the assumption exponential hierarchy collapses to $\mathrm{S}_2^{\mathrm{exp}}.\mathrm{P}^{\mathrm{NP}}$. Our result is proved in two steps. First, under the assumption that co-NP has polylogarithmic round Arthur-Merlin games we show that $\mathrm{PH}_{\mathrm{qpoly}}$, the quasi-polynomial time hierarchy, collapses to $\mathrm{AM}_{\mathrm{qpoly}}$. Then we use simple padding to show that the lower collapse result $\mathrm{PH}_{\mathrm{qpoly}} \subseteq \mathrm{AM}_{\mathrm{qpoly}}$ implies the collapse of EH to $\mathrm{AM}_{\mathrm{exp}}$. We first state a theorem which is proved using standard padding technique. We omit the proof here.

Theorem 1. *Let $l(n) > n$. If $\Sigma_k[n] \subseteq \mathrm{AM}[l(n)]$ then $\Sigma_k[f(n)] \subseteq \mathrm{AM}[l(f(n))]$.*

Interactive proof systems with many rounds can be converted into proof systems with 2 rounds at the expense of increasing the message complexity. The following theorem can be proved using probability amplification and quantifier switching. In particular see Selman and Sengupta [SS04] or Goldreich, Vadhan, and Wigderson [GVW01] for a proof.

Theorem 2. $\mathrm{AM}[m(n), l(n)] \subseteq \mathrm{AM}[c^{m(n)} l(n)^{m(n)}]$ *for some constant c independent of n.*

As observed in [SS04], a corollary is that polylog rounds of Arthur-Merlin games, can be converted into 2-round Arthur-Merlin games with quasi-polynomial message complexity at each round.

Corollary 3 ([SS04]). $\mathrm{AM}[\mathtt{polylog}, \mathtt{poly}] \subseteq \mathrm{AM}_{\mathrm{qpoly}}$.

Boppana, Hastad, and Zachos [BHZ87] showed that if every language in co-NP has a constant round Arthur-Merlin protocol, then the polynomial-time hierarchy collapses to AM. We first extend their proof to give a general result which also works for parameters other than the polynomial range. The proof uses the standard technique of probability amplification followed by quantifier switching. We present the proof so as to get the parameters more accurately. Then we apply this result to quasi-polynomial range to show that if co-NP $\subseteq \mathrm{AM}_{\mathrm{qpoly}}$ then the polynomial hierarchy (or even quasi-polynomial hierarchy) is in $\mathrm{AM}_{\mathrm{qpoly}}$.

Theorem 4. *Let l be a nice function. Then for any constant k,*

$$\text{co-NTIME}[\texttt{lin}] \subseteq \text{AM}[l(\texttt{lin})] \Rightarrow \Sigma_k[\texttt{lin}] \subseteq \text{AM}[l^{(2k)}(\texttt{lin})]$$

Here $l^{(k)}(n)$ denotes k compositions of l.

We first need the following lemma.

Lemma 5. *If* co-NTIME[$\texttt{lin}$] $\subseteq$ AM[$l(\texttt{lin})$] *then* co-AM[$\texttt{lin}$] $\subseteq$ AM[$l(\texttt{lin})$].

Proof. Let $L \in$ co-AM[$\texttt{lin}$]. Then, (by amplifying the probability by a constant amount) there exists a language $A \in$ co-NTIME[$\texttt{lin}$] and a constant c_1 so that for all x:

$x \in L \Rightarrow \Pr_{y\in\{0,1\}^{c_1 n}}[\langle x, y\rangle \in A] \geq \frac{9}{10}$
$x \notin L \Rightarrow \Pr_{y\in\{0,1\}^{c_1 n}}(\langle x, y\rangle \in A) \leq \frac{1}{10}$.

Since $A \in$ co-NTIME[$\texttt{lin}$], from the assumption we have $A \in$ AM[$l(\texttt{lin})$]. That is, (again by amplifying the probability by a constant amount) there is a language $B \in$ NTIME[$\texttt{lin}$] and a constant c_2 so that for all $\langle x, y\rangle, y \in \{0,1\}^{c_1 n}$:

$\langle x, y\rangle \in A \Rightarrow \Pr_{z\in\{0,1\}^{l(c_2 n)}}[\langle x, y, z\rangle \in B] \geq \frac{9}{10}$
$\langle x, y\rangle \notin A \Rightarrow \Pr_{z\in\{0,1\}^{l(c_2 n)}}[\langle x, y, z\rangle \in B] \leq \frac{1}{10}$.

We can combine the two probabilities to get that for a suitable constant c, for all x:

$x \in L \Rightarrow \Pr_{\langle y,\, z\rangle\in\{0,1\}^{l(cn)}}[\langle x, y, z\rangle \in B] \geq \frac{8}{10}$
$x \notin L \Rightarrow \Pr_{\langle y,\, z\rangle\in\{0,1\}^{l(cn)}}(\langle x, y, z\rangle \in B) \leq \frac{2}{10}$.

Since $B \in$ NTIME[$\texttt{lin}$], the overall protocol in an AM protocol which accepts L and has a message complexity $l(\texttt{lin})$. Hence $L \in$ AM[$l(\texttt{lin})$]. □

Proof. (*of Theorem 4*) We can prove the theorem using induction. Assume that co-NTIME[$\texttt{lin}$] $\subseteq$ AM[$l(\texttt{lin})$]. Let $L \in \Sigma_k[\texttt{lin}]$. Then there exists a language $A \in \Pi_{k-1}[\texttt{lin}]$ and a constant c so that for all x of length n:

$x \in L \Rightarrow \exists y \in \{0,1\}^{cn}\langle x, y\rangle \in A$
$x \notin L \Rightarrow \forall y \in \{0,1\}^{cn}\langle x, y\rangle \notin A$.

Since $A \in \Pi_{k-1}[\texttt{lin}]$, from the induction hypothesis and the assumption, $A \in$ co-AM[$l^{(2k-2)}(\texttt{lin})$]. From Lemma 5 and the assumption that co-NTIME[$\texttt{lin}$] $\subseteq$ AM[$l(\texttt{lin})$], we have co-AM[$\texttt{lin}$] $\subseteq$ AM[$l(\texttt{lin})$]. Using a padding argument, if co-AM[$\texttt{lin}$] $\subseteq$ AM[$l(\texttt{lin})$] then co-AM[$l^{(2k-2)}(\texttt{lin})$] $\subseteq$ AM[$l(l^{(2k-2)}(\texttt{lin}))$] $=$ AM[$l^{(2k-1)}(\texttt{lin})$]. Therefore we have $A \in$ AM[$l^{(2k-1)}(\texttt{lin})$].

Since $A \in$ AM[$l^{(2k-1)}(\texttt{lin})$], there is a language $B \in$ NTIME[$\texttt{lin}$] so that for all x:

$x \in L \Rightarrow \exists y \in \{0,1\}^{cn}\left[\Pr_{z\in\{0,1\}^{l^{(2k-1)}(cn)}}[\langle x, y, z\rangle \in B] \geq \frac{9}{10}\right]$
$x \notin L \Rightarrow \forall y \in \{0,1\}^{cn}\left[\Pr_{z\in\{0,1\}^{l^{(2k-1)}(cn)}}[\langle x, y, z\rangle \in B] \leq \frac{1}{10}\right]$

We can amplify the probability (inside the square brackets) by repeating on $10cn$ random zs and taking a majority vote. This will yield that for a language $B' \in$ NTIME[$\texttt{lin}$] (majority language of B), for all x:

$$x \in L \Rightarrow \exists y \in \{0,1\}^{cn} \left[\Pr_{z \in \{0,1\}^{l^{(2k-1)}(cn) \times 10cn}} [\langle x, y, z \rangle \in B'] \geq 1 - \tfrac{1}{2^{cn+2}} \right]$$
$$x \notin L \Rightarrow \forall y \in \{0,1\}^{cn} \left[\Pr_{z \in \{0,1\}^{l^{(2k-1)}(cn) \times 10cn}} [\langle x, y, z \rangle \in B'] \leq \tfrac{1}{2^{cn+2}} \right]$$

With this amplified probabilities we can get that

$$x \in L \Rightarrow \Pr_{z \in \{0,1\}^{l^{(2k-1)}(cn) \times 10cn}} [\exists y \in \{0,1\}^{cn} \langle x, y, z \rangle \in B'] \geq 1 - \tfrac{1}{2^{cn+2}}$$
$$x \notin L \Rightarrow \Pr_{z \in \{0,1\}^{l^{(2k-1)}(cn) \times 10cn}} [\exists y \in \{0,1\}^{cn} \langle x, y, z \rangle \in B'] \leq \tfrac{1}{4}$$

Now consider the language $B'' = \{\langle x, z \rangle \mid \exists y \in \{0,1\}^{cn} \langle x, y, z \rangle \in B'\}$. Then $B'' \in \mathrm{NTIME}[\mathtt{lin}]$. Therefore we have that for all x:

$$x \in L \Rightarrow \Pr_{z \in \{0,1\}^{l^{(2k-1)}(cn) \times 10cn}} [\langle x, z \rangle \in B''] \geq 1 - \tfrac{1}{2^{cn+2}}$$
$$x \notin L \Rightarrow \Pr_{z \in \{0,1\}^{l^{(2k-1)}(cn) \times 10cn}} [\langle x, z \rangle \in B''] \leq \tfrac{1}{4}$$

Thus $L \in \mathrm{AM}[l^{(2k-1)}(cn) \times 10cn] \subseteq \mathrm{AM}[l^{(2k)}(dn)]$ for a suitable constant d, since l is a nice function. Hence $L \in \mathrm{AM}[l^{(2k)}(\mathtt{lin})]$.

□

An application of the above theorem gives the quasi-polynomial version of Boppana et al's theorem.

Theorem 6. *If* co-NP $\subseteq \mathrm{AM}_{\mathrm{qpoly}}$ *then* $\mathrm{PH}_{\mathrm{qpoly}} \subseteq \mathrm{AM}_{\mathrm{qpoly}}$.

Proof. Let $L \in \mathrm{PH}_{\mathrm{qpoly}}$. Then $L \in \Sigma_k[2^{\log^a n}]$ for some constants k and a. Under the assumption that co-NP $\subseteq \mathrm{AM}_{\mathrm{qpoly}}$ we also have that co-NTIME$[\mathtt{lin}] \subseteq \mathrm{AM}[2^{\log^b n}]$, for a fixed constant b. This is because since co-NP $\subseteq \mathrm{AM}_{\mathrm{qpoly}}$, the co-NP complete problem TAUT (SAT complement) is in $\mathrm{AM}[2^{\log^c n}]$ for some fixed c. Now using Cook's reduction, for any $L \in$ co-NTIME$[\mathtt{lin}]$, an instance of length n is reduced to $O(n^2)$ length formula. Hence $L \in \mathrm{AM}[2^{\log^{c+1} n}]$.

Now by the application of the Theorem 4 with $l(n) = 2^{\log^b n}$, $\Sigma_k[\mathtt{lin}] \subseteq \mathrm{AM}[2^{\log^d n}]$ for a constant d. Now by padding we have $\Sigma_k[2^{\log^a n}] \subseteq \mathrm{AM}[2^{\log^{d'} n}]$ for a constant d'.

The last step uses the fact that quasi-polynomial functions are closed under a finite number of compositions: if $f(n) = 2^{\log^a n}$ and $g(n) = 2^{\log^b n}$ then $f(g(n)) = 2^{\log^{ab} n}$. □

Theorem 7 (Main Result 1). *If* co-NP $\subseteq \mathrm{AM}_{\mathrm{qpoly}}$ *then* $\mathrm{EH} \subseteq \mathrm{AM}_{\mathrm{exp}}$.

Proof. By Theorem 6, if co-NP $\subseteq \mathrm{AM}_{\mathrm{qpoly}}$ then $\mathrm{PH}_{\mathrm{qpoly}} \subseteq \mathrm{AM}_{\mathrm{qpoly}}$. Therefore, for any constant k, there is a constant c so that $\Sigma_k[n] \subseteq \mathrm{AM}[2^{\log^c n}]$. Now let L be a language in the exponential hierarchy. That is $L \in \Sigma_k[2^{n^k}]$ for some constant k. Substituting $f(n) = 2^{n^k}$ and $l(n) = 2^{\log^c n}$ in Theorem 1, we get that $L \in \mathrm{AM}[2^{n^{kc}}]$. Hence the theorem. □

Corollary 8. *If* co-NP *has polylogarithmic round Arthur-Merlin games then* $\mathrm{EH} \subseteq \mathrm{AM}_{\mathrm{exp}}$.

Proof. Under the assumption co-NP $\subseteq \mathrm{AM}[\mathrm{polylog}, \mathrm{poly}]$, by Theorem 2, co-NP $\subseteq \mathrm{AM}_{\mathrm{qpoly}}$. Corollary follows from the main theorem. □

Theorem 9 ([SS04]). *If* co-NP $\subseteq$ NP/$\mathtt{qpoly}$ *then* $\mathrm{EH} \subseteq \mathrm{S}_2^{\mathrm{exp}}.\mathrm{P}^{\mathrm{NP}}$.

Since $\mathrm{AM}^{\mathrm{poly}}_{\mathrm{qpoly}} \subseteq$ NP/$\mathtt{qpoly}$, we have the following theorem. We use this theorem in the next section.

Theorem 10. *If* co-NP $\subseteq \mathrm{AM}^{\mathrm{poly}}_{\mathrm{qpoly}}$ *then* $\mathrm{EH} \subseteq \mathrm{S}_2^{\mathrm{exp}}.\mathrm{P}^{\mathrm{NP}}$.

4 Polylogaritmic Round Adaptive Random-self-reducibility

Feigenbaum and Fortnow [FF93] showed that if SAT is $O(\log n)$-adaptive random-self-reducible then the polynomial hierarchy collapses. Their proof can be directly extended to show that if SAT is $\log^{O(1)} n$-adaptive random-self-reducible then the exponential hierarchy collapses. For the sake of completeness here we present a proof. We prove the following theorem from which the result about SAT follows.

Theorem 11 (Main Result 2). *If L is in* NP *and L is $O(\log^k n)$-adaptive random-self-reducible, then $\overline{L}$ is in* $\mathrm{AM}^{\mathrm{poly}}_{\mathrm{qpoly}}$.

Proof. The proof follows Feigenbaum and Fortnow's proof. Let n be any given length. Let R be the random-self reduction for L that makes $k = O(\log^c n)$ adaptive queries. Let with $q_1, \cdots q_k$ be the k queries made by R. (We will consider the case where only one query is made at each round. The proof can be extended to the case where there are polynomially many queries in each round). We assume that the error probability of rsr R is at most $1/2^n$. This can be achieved using standard methods. Note that each q_i is a randomly generated query and the distribution of each query q_i depends only on the input length n, and is independent of the input x. Let p_i denote the probability that the ith query belongs to L. The verifier has $p_1, p_2, \cdots, p_k$ as advice. Consider the following protocol for $\overline{L}$. In this protocol, the values of e and m will be set later.

1. Input: $x, |x| = n$.
2. Verifier randomly chooses m sequences $r_1, r_2, \cdots, r_m$ and sends to the prover.
3. For $1 \leq i \leq m$, the prover uses r_i as random seed to the reduction R and generates queries $q_{i1}, q_{i2}, \cdots q_{ik}$ along with answers $b_{i1}, b_{i2}, \cdots, b_{ik}$ for each query. If $b_{ij} = 1$ then the prover also generates a witness w_{ij} to the fact that $q_{ij} \in L$. Prover sends (q_{ij}, b_{ij}, w_{ij}) for all $1 \leq i \leq m, 1 \leq j \leq k$, to the verifier.
4. Verifier checks the following conditions.
   (a) For each r_i, verifier checks the consistency of the reduction assuming that the answers to the queries are correct.
   (b) For each $b_{ij} = 1$, verifier checks whether w_{ij} is a witness to the claim that $q_{ij} \in L$.
   (c) For $1 \leq i \leq m$, the reduction R rejects x when r_i is used as random string and b_{ij}'s as answers to the queries.
   (d) For $1 \leq j \leq k$, let $S_j = \{q_{1j}, q_{2j}, \cdots, q_{mj}\}$. For $1 \leq j \leq k$, the Verifier checks at least $p_j m - 2^{j+2} e$ strings from S_j are in L according to the prover.

   If all of the above conditions are satisfied, then the verifier accepts x, else rejects x.

We claim that the above protocol correctly accepts $\overline{L}$. We need the following lemma, which easily follows from Chernoff's bound.

Lemma 12. *For $1 \leq j \leq k$,*

$$Pr[p_j m - e \leq |L \cap S_j| \leq p_j m + e] > 1 - 2^{-\epsilon}$$

where $\epsilon = \frac{e^2}{4p_j m} - 1$. i.e., number of q_{ij}'s that belong to L lie between $p_j m - \epsilon$ and $p_j m + \epsilon$ with high probability.

For the rest of the proof we will set $e = 2\sqrt{km}$. For this setting we will get that $\epsilon \geq k$.

Assume x is not in L, then we show that the honest prover causes the verifier to accept x with high probability. The honest prover provides correct answers to all the queries q_{ij}. So Conditions 4a and 4b are always satisfied. Recall that rsr R correctly decides x, when all the queries are answered correctly, with probability bigger than $1 - 1/2^n$. The probability that the reduction R rejects x for all random sequences $r_1, \cdots, r_m$ is at least $(1 - m/2^n)$. Thus Condition 4c is satisfied with high probability. By Lemma 12, for each j, with high probability, at least $p_j m - e$ strings from S_j belong L. Thus Condition 4d is satisfied with probability at least $(1 - k2^{-\epsilon})$. Thus the probability that the any of the conditions are not satisfied is at most $m/2^n + k2^{-\epsilon}$. Thus the verifier accepts x with high probability.

We now consider the case when the prover is dishonest. Assume x is in L. We show that the verifier rejects x with high probability. For the verifier to accept x, all the four conditions in the protocol must be satisfied. Since the optimal prover never violates Conditions 4a and 4b, we concentrate on Conditions 4c, and 4d. There are two ways the verifier accepts x when x is in L. In the first case, the verifier has chosen "bad" random strings, i.e, in this case the prover provides all correct answers and yet the reduction R says x is not in L. However, this can happen with probability at most $m/2^n$.

In the second case, the prover provides wrong answers to some queries q_{ij} which cause the reduction R to reject x. Note that if a query q_{ij} does not belong to L, then the prover can not claim otherwise, since it will violate Condition 4b. So the prover can provide wrong answer to a query q_{ij} only when q_{ij} belongs to L. This means the prover has to claim that the number of queries that belong L are far less than they actually are. Condition 4d ensures that prover can do this with only a small probability. We now give formal details for this case.

From now we assume that the verifier pricks "good" random strings, i.e, given correct answers to the queries the reduction R produces correct answer. Thus for each $1 \leq i \leq m$, the prover has to provide at least one wrong answer to the queries among $q_{i1}, q_{i2}, \cdots q_{ik}$. Thus in total, the prover has to provide at least m wrong answers. Let $S_j = \{q_{1j}, q_{2j}, \cdots, q_{mj}\}$.

Lemma 13. *The prover can provide at most $2^{j+1}e$ wrong answers to the queries from S_j.*

Proof. We prove the claim by induction. Consider the case $j = 1$. By Lemma 12, with high probability, at most $p_1 m + e$ queries from S_1 belong to L. The verifier checks if at least $p_1 m - e$ queries from S_1 belongs to L. Thus, with high probability, the prover can provide at most $2e$ wrong answers to queries from S_1.

Assume that the claim is true for all $j < l - 1$. Consider a query q_{il} that belongs to S_l. This query was produced by the reduction R with queries $q_{i1}, q_{i2}, \cdots q_{il-1}$ and the respective answers $b_{i1}, b_{12}, \cdots b_{il-1}$. If all the answers are correct, then q_{il} is "valid" query that R would have produced. However, it is possible that the prover might have provided wrong answers to some of the queries $q_{i1}, \cdots q_{il-1}$. In this case we call q_{il} "invalid" query. Partition S_l into "valid" queries and "invalid" queries. Our goal is to estimate the number of queries from S_l for which the prover can provide wrong answers. Recall that the prover can give a wrong answer only if a query belongs to L. Consider the worst possible case; All the invalid queries actually belong to L.

We first put a bound on the number of invalid queries. A query is invalid only if one of the preceding queries is answered wrong by the prover. By our induction hypothesis, the number queries that are answered wrong is at most $\sum_1^{l-1} 2^j e < 2^l e$. Thus the total number of invalid queries is at most $2^l e$.

Consider the set of valid queries. All of them are produced when given correct answers to the previous queries. Thus, by Lemma 12, with high probability, at most $p_l m + e$ of these valid queries belong to L. Thus, with high probability, at most $p_l + e + 2^l e$ queries from S_l belong to L. Since the verifier checks whether at least $p_l m - e$ queries from S_l belong to L, the prover can provide wrong answers to at most $2^l + e + e = 2^{l+1}e$ queries from S_l.

□

Thus the total number of wrong answers that the prover can give is at most $2^{k+2}e$. However, the prover has to provide at least m wrong answers to make the verifier accept x. If we choose $m > 2^{k+2}e = 2^{k+2} \times 2\sqrt{km}$, then the probability that the prover can make the verifier accept x is extremely small. Since $k = \log^r n$, we can choose a value for m which is quasi-polynomial in n. Thus there is a $\mathrm{AM}^{\mathrm{poly}}_{\mathrm{qpoly}}$ proof system that accepts $\overline{L}$.

□

We have the following corollary, partially answering a question by Feigenbaum and Fortnow [FF93].

Corollary 14. *If* SAT *is* $O(\log^k n)$*- adaptive random-self-reducible, then* $\mathrm{EH} \subseteq \mathrm{S}_2^{\mathrm{exp}}.\mathrm{P}^{\mathrm{NP}}$.

Next we consider the relation between random-self-reducibility and checkability. Program checkers for many languages use random-self-reducibility. This raises the question of whether random-self-reducibility implies checkability. We observe that if SAT is $k(n)$-nonadaptive random-self-reducible, then SAT is checkable with advice.

Theorem 15. *If* SAT *is* $k(n)$*-nonadaptive random-self-reducible, then* SAT *is checkable with advice.*

Proof. Blum and Kannan [BK89] characterized checkability using interactive proof systems. A language L is in FRIP (short form for Function restricted Interactive Protocol) if there is a polynomial-round AM protocol for L where the power of the prover is the complexity of L, i.e., the verifier only asks questions of the form $q \in L$?. Blum and Kannan showed the following theorem.

Theorem 16 ([BK89]). *A language L is checkable if and only if both L and $\overline{L}$ are in* FRIP*.*

Their proof can be easily modified to show that a language L is checkable with advice if and only if, L and $\overline{L}$ are in $\mathrm{FRIP}^{\mathrm{poly}}$, where $\mathrm{FRIP}^{\mathrm{poly}}$ is similar to FRIP except that the verifier has access to a polynomial advice.

Let SAT is $k(n)$-nonadaptive random-self-reducible. Feigenbaum and Fortnow [FF93] showed that this implies $\overline{\mathrm{SAT}}$ is in AM^{poly}. Moreover, the verifier asks only types of questions i) $q \in \mathrm{SAT}$?, and ii) if $q \in \mathrm{SAT}$, then what is the witness?. Since search reduces to decision for SAT, all Type (ii) questions can be converted into Type (i) questions. Thus $\overline{\mathrm{SAT}}$ is in $\mathrm{FRIP}^{\mathrm{poly}}$. Note that SAT is trivially in FRIP. Thus SAT is checkable with advice. □

References

[Bab85] L. Babai. Trading group theory for randomness. In *Proc. 17th Annual ACM Symp. on Theory of Computing*, pages 421–429, 1985.

[BDG88] J. Balcázar, J. Diaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, Berlin, 1988.

[BHZ87] R. Boppana, J. Hastad, and S. Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 1987.

[BK89] M. Blum and S. Kannan. Designing programs that check their work. In *Proc. 21st Symp. ACM Symp. Theory of Computing*, pages 86–97, 1989.

[BLR90] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting programs with applications to numerical problems. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1990.

[Can96] R. Canetti. More on BPP and the Polynomial-time Hierarchy. *Information Processing Letters*, 57(5):237–241, March 1996.

[CPS99] J. Cai, A. Pavan, and D. Sivakumar. On the hardness of permanent. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, volume LNCS, 1627, pages 90–99, 1999.

[FF93] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22(5):994–1005, October 1993.

[FL92] U. Feige and C. Lund. On the hardness of computing permanent of random matrices. In *Proceedings of 24th Annual ACM Symposium on Theory of Computing*, pages 643–654, 1992.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *ACM Symposium on Theory of Computing (STOC '85)*, pages 291–304. ACM Press, May 1985.

[GMW86] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *27th Annual Symposium on Foundations of Computer Science (FOCS '86)*, pages 174–187, Los Angeles, Ca., USA, October 1986. IEEE Computer Society Press.

[GRS00] O. Goldreich, D. Ron, and M. Sudan. Chinese remaindering with errors. *IEEE Transactions on Information Theory*, 46, 2000.

[GS89] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In S. Micali, editor, *Randomness and Computation, Advances in Computing Research.* Jai Press, 1989.

[GS92] P. Gemmel and M. Sudan. Higly resilient correctors for polynomials. *Information Processing Letters*, 43:169–174, May 1992.

[GVW01] O. Goldreich, S. Vadhan, and A. Wigderson. On interactive proofs with laconic provers. In *Proceedings of the 28th International Colloquium on Automata, Languages, and Programming*, volume LNCS 2076, pages 334–345, 2001.

[LFKN90] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. In IEEE, editor, *Proceedings: 31st Annual Symposium on Foundations of Computer Science: October 22–24, 1990, St. Louis, Missouri*, volume 1, pages 2–10. IEEE Computer Society Press, 1990.

[Lip91] R. Lipton. New directions in testing. In *Distributed Computing and Cryptography*, volume 2 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 191–202. American Mathematics Society, 1991.

[Pap94] C. Papadimitriou. *Computational Complexity.* Addison-Wesley Publishing Company, 1994.

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

[Sha92] A. Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, October 1992.

[SS04] A. Selman and S. Sengupta. Polylogarithmic-round interactive proofs for CoNP collapses the exponential hierarchy. In *Proceedings of the 19th IEEE Conference on Computational Complexity*, 2004. To Appear. Also ECCC Technical Report TR04-007, 2004.