

# Polylogarithmic-round Interactive Proofs for coNP Collapse the Exponential Hierarchy\*

A. Pavan<sup>†</sup>   Alan L. Selman<sup>‡</sup>   Samik Sengupta<sup>§</sup>   N. V. Vinodchandran<sup>¶</sup>

March 2, 2006

## Abstract

If every language in coNP has constant round interactive proof system, then the polynomial-time hierarchy collapses [BHZ87]. On the other hand, the well-known LFKN protocol gives  $O(n)$ -round interactive proof systems for all languages in coNP [LFKN92]. We consider the question whether it is possible for coNP to have interactive proof systems with polylogarithmic round complexity. We show that this is unlikely by proving that if a coNP-complete set has a polylogarithmic-round interactive proof system then the exponential-time hierarchy collapses. We also consider exponential versions of the Karp-Lipton theorem and Yap's theorem.

## 1 Introduction

Babai introduced *Arthur-Merlin Games* to study the power of randomization in interaction [Bab85, BM88]. Soon afterward, Goldwasser and Sipser [GS89] showed that these classes are equivalent in power to *Interactive Proof Systems*, introduced by Goldwasser, Micali, and Rackoff [GMR85]. Study of interactive proof systems and Arthur-Merlin classes has been exceedingly successful [ZH86, BHZ87, ZF87, LFKN92, Sha92], eventually leading to the discovery of Probabilistically Checkable Proofs [BOGKW88, LFKN92, Sha92, BFL81, BFLS91, FGL<sup>+</sup>91, AS92, ALM<sup>+</sup>92].

Interactive proof systems are successfully placed relative to traditional complexity classes. In particular, it is known that for any constant  $k$ ,  $IP[k] \subseteq \Pi_2^P$  [BM88], and  $IP[\text{poly}] = PSPACE$  [Sha92], where  $IP[r]$  denotes the class of languages accepted by interactive proof systems with  $r$  rounds. However, the relationship between coNP and interactive proof systems is not entirely clear. On the one hand, Boppana, Håstad, and Zachos [BHZ87] proved that if every set in coNP has a constant-round interactive proof system, then the polynomial-time hierarchy collapses below the second level. On the other hand, the best interactive protocol for any language in coNP comes from the result of Lund et al. [LFKN92], who show that  $\#SAT$ , a problem hard for the entire

---

\*Some of the results of this paper were presented at the 19th IEEE Conference on Computational Complexity theory by second and third author.

<sup>†</sup>Department of Computer Science, Iowa State University, [pavan@cs.iastate.edu](mailto:pavan@cs.iastate.edu). Research supported in part by the NSF grant CCF-0430807.

<sup>‡</sup>Department of Computer Science and Engineering, University at Buffalo, [selman@cse.buffalo.edu](mailto:selman@cse.buffalo.edu). Research supported in part by the NSF grant CCR-0307077.

<sup>§</sup>Research done while at Department of Computer Science and Engineering, University at Buffalo

<sup>¶</sup>Department of Computer Science and Engineering, University of Nebraska-Lincoln, [vinod@cse.unl.edu](mailto:vinod@cse.unl.edu). Research supported in part by the NSF grant CCF-0430991.

polynomial-time hierarchy [Tod91], is accepted by an interactive proof system with  $O(n)$  rounds of interaction. Can every set in coNP be accepted by an interactive proof system with more than constant but sublinear number of rounds? In this paper we look at this question.

## Our Results

First we show that coNP cannot have a polylogarithmic-round interactive proof system unless the exponential-time hierarchy collapses to the second level. More specifically, we show that if coNP has a polylogarithmic-round interactive proof system, then  $\text{EH} = \text{AM}_{\text{exp}}$  (Theorem 3.8), where  $\text{EH} = \bigcup_k \text{NEXP}^{\Sigma_k^{\text{P}}}$  is the exponential-hierarchy and  $\text{AM}_{\text{exp}}$  is the class of languages accepted by Arthur-Merlin protocols with verifier running in exponential time. It is known that  $\text{AM}_{\text{exp}} \subseteq \Pi_2^{\text{exp}}$ . Thus if coNP has a polylogarithmic-round interactive proof system then the exponential hierarchy collapses to the second level.

Our second result shows that if every set in NP has a quasipolynomial-size family of circuits, then  $\text{EH} = \text{S}_2^{\text{exp}}$  (Theorem 4.1). Here  $\text{S}_2^{\text{exp}}$  is the exponential version of the symmetric-alternation class  $\text{S}_2^{\text{P}}$ [RS98] and is contained in  $\text{NEXP}^{\text{NP}} \cap \text{coNEXP}^{\text{NP}}$ . This improves the bound given by Buhrman and Homer in [BH92] where it is shown that the assumption collapses EH to  $\text{NEXP}^{\text{NP}} \cap \text{coNEXP}^{\text{NP}}$ . We also prove an exponential version of Yap's result. We show that if  $\text{NP} \subseteq \text{coNP}/\text{qpoly}$  then the exponential-time hierarchy collapses to  $\text{S}_2^{\text{qpoly}} \circ \text{P}^{\text{NP}}$  (Theorem 4.2).

## 2 Preliminaries

For definitions of standard complexity classes and machine models, we refer the reader to standard text books (Homer and Selman [HS01] or Balcázar, Díaz, and Gabarró [BDG95, BDG90]). In this paper we deal with complexity classes defined using general parameter ranges. We present these notations first.

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

**Definition 2.1** *We call a time constructible function  $l(n)$  nice if (1)  $l(n) \geq n$ , (2)  $l(l(n)) \geq nl(n)$ , and (3)  $l(cn) \geq cl(n)$  for any constant  $c > 1$ .*

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

We will next define exponential and quasipolynomial versions of the polynomial hierarchy.

For any class  $\mathcal{C}$ , the complement class  $\text{co}\mathcal{C} = \{L \mid \bar{L} \in \mathcal{C}\}$ . Let  $\Sigma_k[f(n)]$  denote the class of languages accepted by a  $\Sigma_k$ -machine (refer to [BDG90] for a definition) where the running time within a quantifier is bounded by  $f(n)$ . Let  $\Pi_k[f(n)] = \text{co}\Sigma_k[f(n)]$ . Using this definition, we can define the polynomial, quasipolynomial, and the exponential hierarchies as follows:

- $\Sigma_k^{\text{P}} = \cup_c \Sigma_k[n^c]$ ,  $\Sigma_k^{\text{qpoly}} = \cup_c \Sigma_k[2^{(\log n)^c}]$ , and  $\Sigma_k^{\text{exp}} = \cup_c \Sigma_k[2^{n^c}]$ .
- $\text{PH} = \cup_k \Sigma_k^{\text{P}}$ ,  $\text{PH}_{\text{qpoly}} = \cup_k \Sigma_k^{\text{qpoly}}$ , and  $\text{EH} = \cup_k \Sigma_k^{\text{exp}}$ .

These hierarchies can be defined using oracle Turing machines also.

- $\Sigma_0^{\text{exp}} = \text{EXP}, \Sigma_1^{\text{exp}} = \text{NEXP}, \Sigma_2^{\text{exp}} = \text{NEXP}^{\text{NP}}$ , and in general, for  $k \geq 0$ ,  $\Sigma_{k+1}^{\text{exp}} = \text{NEXP}^{\Sigma_k^{\text{exp}}}$ . Also for every  $k \geq 0$ ,  $\Pi_k^{\text{exp}} = \{L \mid \bar{L} \in \Sigma_k^{\text{exp}}\}$ .
- $\Sigma_0^{\text{qpoly}} = \text{QPOLY} = \bigcup_{c>0} \text{DTIME}(2^{\log^c n})$ ,  $\Sigma_1^{\text{qpoly}} = \text{NQPOLY} = \bigcup_{c>0} \text{NTIME}(2^{\log^c n})$ , and in general, for  $k \geq 1$ ,  $\Sigma_{k+1}^{\text{qpoly}} = \text{NQPOLY}^{\Sigma_k^{\text{qpoly}}}$ . Also for every  $k \geq 0$ ,  $\Pi_k^{\text{qpoly}} = \{L \mid \bar{L} \in \Sigma_k^{\text{qpoly}}\}$ .

Similar to the relationship between the polynomial and the linear-exponential-time hierarchy, there is a relationship between the quasipolynomial hierarchy and the exponential hierarchy. Given a set  $L$ , let  $\text{Tally}(L) = \{1^{n(w)} \mid w \in L\}$ , where  $w$  is the 2-adic representation of the integer  $n(w)$ . Clearly,  $|w| \leq c \log n(w)$  for some constant  $c > 0$ .

**Proposition 2.2** *For every  $k > 0$ ,*

$$L \in \Sigma_k^{\text{exp}} \Leftrightarrow \text{Tally}(L) \in \Sigma_k^{\text{qpoly}}.$$

As a consequence, there is no tally set in  $\Sigma_k^{\text{qpoly}} - \Sigma_{k-1}^{\text{qpoly}}$  if and only if  $\Sigma_k^{\text{exp}} = \Sigma_{k-1}^{\text{exp}}$ . Therefore, if the quasipolynomial hierarchy collapses at level  $k$ , then the exponential hierarchy collapses to the  $k$ -th level as well. The following proposition is easy to see. We note that the analogous result is not known for the exponential hierarchy.

**Proposition 2.3** *If  $\Sigma_k^{\text{qpoly}} = \Pi_k^{\text{qpoly}}$ , then the quasipolynomial hierarchy collapses to the  $k$ -th level.*

We will need classes with quasipolynomial-length advices.

**Definition 2.4** *Let  $\mathcal{C}$  be a complexity class. A set  $L \in \mathcal{C}/\text{qpoly}$  if there is a function  $s : 1^* \rightarrow \Sigma^*$ , some constant  $k > 0$ , and a set  $A \in \mathcal{C}$  such that*

1. *For every  $n$ ,  $|s(1^n)| \leq 2^{\log^k n}$ , and*
2. *For all  $x$ ,  $x \in L \Leftrightarrow (x, s(1^{|x|})) \in A$ . Here  $A$  is called the witness language.*

It is easy to see that  $\mathcal{D} \subseteq \mathcal{C}/\text{qpoly}$  if and only if  $\text{co}\mathcal{D} \subseteq \text{co}\mathcal{C}/\text{qpoly}$ .

## 2.1 Arthur-Merlin games

Babai [Bab85] introduced *Arthur-Merlin protocol*, a communication game that is played by Arthur, a probabilistic polynomial-time machine, and Merlin, a computationally unbounded Turing machine. Arthur can use random bits, but these bits are public, i.e., Merlin can see them and move accordingly.

Given an input string  $x$ , Merlin tries to convince Arthur that  $x$  belongs to some fixed language  $L$ . The game consists of a predetermined finite number of moves with Arthur and Merlin moving alternately. In each move Arthur (or Merlin) prints a finite string (a message) on a read-write communication tape. Arthur's moves depend on his random bits. After the last move, Arthur either accepts or rejects  $x$ .

**Definition 2.5** ([Bab85, BM88]) *Given two nice functions  $m(\cdot)$ , and  $l(\cdot)$ , a language  $L$  is in  $\text{AM}[m(n), l(n)]$  if there exists an Arthur-Merlin game such that for every string  $x$  of length  $n$  the following holds.*

- The game consists of  $m(n)$  moves and within each move the length of the message written on the communication tape is bounded by  $l(n)$ ;
- Arthur moves first;
- After the last move, Arthur behaves deterministically to either accept or reject the input string;
- If  $x \in L$ , then there exists a sequence of moves by Merlin that leads to the acceptance of  $x$  by Arthur with probability at least  $\frac{3}{4}$ ;
- if  $x \notin L$  then for all possible moves of Merlin, the probability that Arthur accepts  $x$  is less than  $\frac{1}{4}$ .

Babai and Moran showed that Arthur-Merlin games with many moves can be converted into games with only two moves at the expense of increasing the message length [BM88]. Thus games with two moves are important and we denote  $\text{AM}[2, l(n)]$  with  $\text{AM}[l(n)]$ . We consider polynomial, quasipolynomial, and exponential versions of AM which are defined as follows.  $\text{AM} = \cup_k \text{AM}[n^k]$ ,  $\text{AM}_{\text{qpoly}} = \cup_c \text{AM}[2^{\log^c n}]$  and  $\text{AM}_{\text{exp}} = \cup_k \text{AM}[2^{n^k}]$ .

We note the following standard proposition.

**Proposition 2.6** *Let  $E$  be an event that occurs with probability at least  $\frac{3}{4}$ . Then, for any polynomial  $p(\cdot)$  such that  $p(n) \geq n$ , there is a constant  $c$  such that within  $t \stackrel{\text{def}}{=} c \times p(n)$  independent trials,  $E$  occurs for more than  $\frac{t}{2}$  times with probability  $(1 - \frac{1}{2^{p(n)}})$ .*

## 2.2 Symmetric alternation

We define  $S_2^{\text{exp}}$  as the exponential version of the  $S_2$  operator defined by Russell and Sundaram [RS98] and Canetti [Can96]. A set  $L$  is in  $S_2^{\text{exp}} \circ \mathcal{C}$  if there is some  $k > 0$  and  $A \in \mathcal{C}$  such that for every  $x \in \{0, 1\}^n$ ,

$$\begin{aligned} x \in L &\implies \exists y \forall z (x, y, z) \in A, \text{ and} \\ x \notin L &\implies \exists z \forall y (x, y, z) \notin A, \end{aligned}$$

where  $|y|, |z| \leq 2^{n^k}$ . Similarly, we can define  $S_2^{\text{qpoly}}$  as the quasipolynomial version of the  $S_2$  operator.

Similar to  $S_2^{\text{P}} \stackrel{\text{def}}{=} S_2 \circ \text{P}$ , the class  $S_2^{\text{exp}} \circ \mathcal{C}$  ( $S_2^{\text{qpoly}} \circ \mathcal{C}$ ) can be thought of as a game between two provers and a verifier. Let  $L \in S_2^{\text{exp}} \circ \mathcal{C}$  (respectively, in  $S_2^{\text{qpoly}} \circ \mathcal{C}$ ). On any input  $x$  of length  $n$ , the *Yes-prover* attempts to show that  $x \in L$ , and the *No-prover* attempts to show that  $x \notin L$ . Both the proofs are at most exponentially (respectively, quasipolynomially) long in  $|x|$ . If  $x \in L$ , then there must be a proof by the yes-prover (called a *yes-proof*) that convinces the verifier that  $x \in L$  no matter what proof the no-prover (called a *no-proof*) provides; symmetrically, if  $x \notin L$ , then there must exist some no-proof such that the verifier rejects  $x$  irrespective of the yes-proof. For every input  $x$ , there is a yes-prover and a no-prover such that exactly one of them is correct. The verifier has the ability of the class  $\mathcal{C}$ ; for example, if  $\mathcal{C} = \text{P}$ , then the verifier is a deterministic polynomial-time Turing machine, and if  $\mathcal{C} = \text{P}^{\text{NP}}$ , then the verifier is a polynomial-time oracle Turing machine with SAT as the oracle. It is easy to see that if  $\mathcal{C}$  is closed under complement, then  $S_2^{\text{exp}} \circ \mathcal{C}$  (respectively,  $S_2^{\text{qpoly}} \circ \mathcal{C}$ ) is also closed under complement.

We concentrate on the classes  $S_2^{\text{exp}} \stackrel{\text{def}}{=} S_2^{\text{exp}} \circ \text{P}$ ,  $S_2^{\text{exp}} \circ \text{P}^{\text{NP}}$ , and  $S_2^{\text{qpoly}} \circ \text{P}^{\text{NP}}$ . The proofs of Russell and Sundaram can be easily modified to show the following.

**Proposition 2.7**

1.  $S_2^{\text{exp}} \subseteq \text{NEXP}^{\text{NP}} \cap \text{coNEXP}^{\text{NP}}$ .
2.  $\text{NEXP}^{\text{NP}} \cup \text{coNEXP}^{\text{NP}} \subseteq S_2^{\text{exp}} \circ \text{P}^{\text{NP}} \subseteq \text{NEXP}^{\Sigma_2^{\text{P}}} \cap \text{coNEXP}^{\Sigma_2^{\text{P}}}$ .
3.  $\text{AM}_{\text{exp}} \subseteq S_2^{\text{exp}} \circ \text{P}^{\text{NP}}$ .
4.  $\text{NQPOLY}^{\text{NP}} \cup \text{coNQPOLY}^{\text{NP}} \subseteq S_2^{\text{qpoly}} \circ \text{P}^{\text{NP}} \subseteq \text{NQPOLY}^{\Sigma_2^{\text{P}}} \cap \text{coNQPOLY}^{\Sigma_2^{\text{P}}}$ .

**Proof** We give a short proof of the second inclusion of item (2). Other inclusions are easy to verify. Note that since  $S_2^{\text{exp}} \circ \text{P}^{\text{NP}}$  is closed under complement, it suffices to show that  $S_2^{\text{exp}} \circ \text{P}^{\text{NP}}$  is a subset of  $\text{NEXP}^{\Sigma_2^{\text{P}}}$ . Let  $L \in S_2^{\text{exp}} \circ \text{P}^{\text{NP}}$ ; therefore,  $\exists k > 0, L' \in \text{P}^{\text{NP}}$  such that

$$\begin{aligned} x \in L &\implies \exists y \forall z (x, y, z) \in L', \text{ and} \\ x \notin L &\implies \exists z \forall y (x, y, z) \notin L', \end{aligned}$$

where  $|y|, |z| \leq 2^{|x|^k}$ . We define the language

$$A = \{(x, y, 0^{2^{|x|^k}}) \mid \exists z (x, y, z) \notin L'\}.$$

$A$  is in  $\Sigma_2^{\text{P}}$ . We define a NEXP machine  $N$  that decides  $L$  with  $A$  as an oracle. On input  $x$ ,  $N$  guesses  $y$ ,  $|y| \leq 2^{|x|^k}$ , and accepts  $x$  if and only if  $(x, y, 0^{2^{|x|^k}}) \notin A$ . If  $x \in L$ , then for the correctly guessed  $y$ ,  $(x, y, z) \in L'$  for every  $z$ ; therefore,  $N$  accepts  $x$ . On the other hand, if  $x \notin L$ , then there is a  $z$  such that for every  $y$ ,  $(x, y, z) \notin L'$ , and therefore,  $(x, y, 0^{2^{|x|^k}}) \in A$  and  $N$  rejects  $x$ . This completes the proof.  $\square$

**Proposition 2.8**

$$L \in S_2^{\text{exp}} \circ \text{P}^{\text{NP}} \Leftrightarrow \text{Tally}(L) \in S_2^{\text{qpoly}} \circ \text{P}^{\text{NP}}.$$

**Proof** We show the forward implication; the proof of the backward implication is similar. Let  $L \in S_2^{\text{exp}} \circ \text{P}^{\text{NP}}$ ; therefore, there exists  $k > 0$  and  $V \in \text{P}^{\text{NP}}$  such that

$$x \in L \implies \exists y \forall z (x, y, z) \in V$$

and

$$x \notin L \implies \exists z \forall y (x, y, z) \notin V,$$

where  $|y|, |z| \leq 2^{|x|^k}$ . If  $x \in L$ , let  $y_x$  be the string such that  $\forall z (x, y_x, z) \in V$ , and if  $x \notin L$ , let  $z_x$  be the string such that  $\forall y (x, y, z_x) \notin V$ .

We need to show that  $\text{Tally}(L)$  is in  $S_2^{\text{qpoly}} \circ \text{P}^{\text{NP}}$ . Let  $w = 1^{n(x)}$  be the input. Note that  $|x| \leq c \log |w|$  for some  $c > 0$ . On input  $(w, y, z)$ , the  $\text{P}^{\text{NP}}$  verifier constructs  $x$  from  $w$  (this requires time polynomial in  $|w| = n(x)$ ) and accepts if and only if  $(x, y, z) \in V$ . If  $w \in \text{Tally}(L)$ , then  $x \in L$  and  $y_x$  will convince the verifier; on the other hand, if  $w \notin \text{Tally}(L)$ , then  $x \notin L$ , and for  $z = z_x$ , the verifier will reject no matter what  $y$  is provided. Since  $|y_x|, |z_x| \leq 2^{|x|^k} \leq 2^{c^k \log^k |w|}$ , this defines an  $S_2^{\text{qpoly}} \circ \text{P}^{\text{NP}}$  protocol for  $\text{Tally}(L)$ .  $\square$

The following proposition follows immediately.

**Proposition 2.9**  $S_2^{\text{exp}} \circ \text{P}^{\text{NP}} = \text{NEXP}^{\Sigma_2^{\text{P}}}$  if and only if there is no tally set in  $\text{NQPOLY}^{\Sigma_2^{\text{P}}} - S_2^{\text{qpoly}} \circ \text{P}^{\text{NP}}$ .

### 3 Arthur-Merlin Games with Polylogarithmic Moves

In this section we show that if  $\text{coNP}$  has polylogarithmic-round Arthur-Merlin games then the exponential hierarchy collapses to  $\text{AM}_{\text{exp}}$ . Our result is proved in two steps. First, under the assumption that  $\text{coNP}$  has polylogarithmic round Arthur-Merlin games we show that  $\text{PH}_{\text{qpoly}}$ , the quasi-polynomial time hierarchy, collapses to  $\text{AM}_{\text{qpoly}}$ . Then we use simple padding to show that the lower collapse result  $\text{PH}_{\text{qpoly}} \subseteq \text{AM}_{\text{qpoly}}$  implies the collapse of  $\text{EH}$  to  $\text{AM}_{\text{exp}}$ . We first state a theorem that is proved using a standard padding technique. We omit the proof here.

**Theorem 3.1** *Let  $l(n) > n$ . If  $\Sigma_k[n] \subseteq \text{AM}[l(n)]$ , then  $\Sigma_k[f(n)] \subseteq \text{AM}[l(f(n))]$ .*

Babai and Moran [BM88] showed that Arthur-Merlin games with many rounds can be converted into games with only 2 rounds at the expense of increasing message complexity. We state such a collapse theorem involving general parameters which can be proved using probability amplification and quantifier swapping. See the paper of Goldreich, Vadhan, and Wigderson [GVW01] for a proof.

**Theorem 3.2** ([BM88])  $\text{AM}[m(n), l(n)] \subseteq \text{AM}[c^{m(n)}l(n)^{m(n)}]$  for some constant  $c$  independent of  $n$ .

As a corollary of the above theorem we get that polylog rounds of Arthur-Merlin games can be converted into 2-round Arthur-Merlin games with quasipolynomial message complexity at each round.

**Corollary 3.3**  $\text{AM}[\text{polylog}, \text{poly}] \subseteq \text{AM}_{\text{qpoly}}$ .

Boppana, Hastad, and Zachos [BHZ87] showed that if every language in  $\text{coNP}$  has a constant round Arthur-Merlin protocol, then the polynomial-time hierarchy collapses to  $\text{AM}$ . We first extend their proof to give a general result that 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 quasipolynomial range to show that if  $\text{coNP} \subseteq \text{AM}_{\text{qpoly}}$  then the polynomial hierarchy (or even quasipolynomial hierarchy) is in  $\text{AM}_{\text{qpoly}}$ .

**Theorem 3.4** *Let  $l$  be a nice function. Then for any constant  $k$ ,*

$$\text{coNTIME}[\mathbf{lin}] \subseteq \text{AM}[l(\mathbf{lin})] \Rightarrow \Sigma_k[\mathbf{lin}] \subseteq \text{AM}[l^{(2k)}(\mathbf{lin})]$$

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

We first need the following lemma.

**Lemma 3.5** *If  $\text{coNTIME}[\mathbf{lin}] \subseteq \text{AM}[l(\mathbf{lin})]$ , then  $\text{coAM}[\mathbf{lin}] \subseteq \text{AM}[l(\mathbf{lin})]$ .*

**Proof** Let  $L \in \text{coAM}[\mathbf{lin}]$ . Then, (by amplifying the probability by a constant amount) there exists a language  $A \in \text{coNTIME}[\mathbf{lin}]$  and a constant  $c_1$  so that for all  $x$ :

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

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

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

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

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

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

**Proof (of Theorem 3.4)** We can prove the theorem using induction. Assume that  $\text{coNTIME}[\mathbf{lin}] \subseteq \text{AM}[l(\mathbf{lin})]$ . Let  $L \in \Sigma_k[\mathbf{lin}]$ . Then there exists a language  $A \in \Pi_{k-1}[\mathbf{lin}]$  and a constant  $c$  so that for all  $x$  of length  $n$ :

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

Since  $A \in \Pi_{k-1}[\mathbf{lin}]$ , from the induction hypothesis and the assumption,  $A \in \text{coAM}[l^{(2k-2)}(\mathbf{lin})]$ . From Lemma 3.5 and the assumption that  $\text{coNTIME}[\mathbf{lin}] \subseteq \text{AM}[l(\mathbf{lin})]$ , we have  $\text{coAM}[\mathbf{lin}] \subseteq \text{AM}[l(\mathbf{lin})]$ . Using a padding argument, if  $\text{coAM}[\mathbf{lin}] \subseteq \text{AM}[l(\mathbf{lin})]$  then  $\text{coAM}[l^{(2k-2)}(\mathbf{lin})] \subseteq \text{AM}[l^{(2k-2)}(\mathbf{lin})] = \text{AM}[l^{(2k-1)}(\mathbf{lin})]$ . Therefore we have  $A \in \text{AM}[l^{(2k-1)}(\mathbf{lin})]$ .

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

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

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

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

With this amplified probabilities we can get that

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

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

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

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

An application of the above theorem gives the quasipolynomial version of the theorem due to Boppana et al.

**Theorem 3.6** *If  $\text{coNP} \subseteq \text{AM}_{\text{qpoly}}$ , then  $\text{PH}_{\text{qpoly}} \subseteq \text{AM}_{\text{qpoly}}$ .*

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

Now by the application of the Theorem 3.4 with  $l(n) = 2^{\log^b n}$ ,  $\Sigma_k[\mathbf{lin}] \subseteq \text{AM}[2^{\log^d n}]$  for a constant  $d$ . By padding we have  $\Sigma_k[2^{\log^a n}] \subseteq \text{AM}[2^{\log^{d'} n}]$  for a constant  $d'$ .

The last step uses the fact that quasipolynomial 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 3.7** *If  $\text{coNP} \subseteq \text{AM}_{\text{qpoly}}$ , then  $\text{EH} \subseteq \text{AM}_{\text{exp}}$ .*

**Proof** By Theorem 3.6, if  $\text{coNP} \subseteq \text{AM}_{\text{qpoly}}$  then  $\text{PH}_{\text{qpoly}} \subseteq \text{AM}_{\text{qpoly}}$ . Therefore, for any constant  $k$ , there is a constant  $c$  so that  $\Sigma_k[n] \subseteq \text{AM}[2^{\log^c n}]$ . 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 3.1, we get that  $L \in \text{AM}[2^{n^{kc}}]$ . Hence the theorem. □

Now we arrive at the main theorem.

**Theorem 3.8** *If  $\text{coNP}$  has polylogarithmic round Arthur-Merlin games, then  $\text{EH} \subseteq \text{AM}_{\text{exp}}$ .*

**Proof** Under the assumption  $\text{coNP} \subseteq \text{AM}[\text{polylog}, \text{poly}]$ , by Theorem 3.2,  $\text{coNP} \subseteq \text{AM}_{\text{qpoly}}$ . We obtain the result by applying the previous theorem.  $\square$

We obtain the following corollary.

**Corollary 3.9** *If every set in NP has an interactive proof system where the prover sends a total of at most polylogarithmic bits, then the exponential hierarchy collapses to  $\text{AM}_{\text{exp}}$ .*

**Proof** Goldreich, Vadhan, and Wigderson [GVW01, Corollary 3.8] have shown that if a set  $L$  has an interactive proof system where the prover sends a total of at most polylog bits, then  $\bar{L} \in \text{AM}[\text{qpoly}]$ . Therefore, if every set in NP has such an interactive proof system, then  $\text{coNP} \subseteq \text{AM}[\text{qpoly}]$ . By Theorem 3.7, the exponential hierarchy collapses to  $\text{AM}_{\text{exp}}$ .  $\square$

We can prove a version of Theorem 3.8 for  $(\log n)^{\log \log n}$ -round interactive proof for coNP. Let

$$\text{eexp} = \bigcup_{k>0} \{f \mid \forall x f(x) < 2^{2^{|x|^k}}\}.$$

Define  $\Sigma_1^{\text{eexp}} = \text{NEEXP} = \text{NTIME}(\text{eexp})$ , and for  $k > 1$ ,

$$\Sigma_k^{\text{eexp}} = \text{NEEXP}^{\Sigma_{k-1}^p}.$$

Let  $\text{AM}_{\text{eexp}}$  denote Arthur-Merlin games with double-exponential proof complexity and  $\text{EEH} = \cup_k \Sigma_k^{\text{eexp}}$  denote the double-exponential hierarchy.

**Theorem 3.10** *If coNP has  $2^{(\log \log n)^{O(1)}}$  round Arthur-Merlin games, then  $\text{EEH} \subseteq \text{AM}_{\text{eexp}}$ .*

## 4 Quasipolynomial advice for NP

In this section, we study the consequences of the existence of quasipolynomial length (i.e.,  $2^{\text{polylog}}$ -length) advice for NP. This question was first studied by Buhrman and Homer [BH92]. They showed that if every set in NP has a quasipolynomial-size family of circuits, then the exponential hierarchy collapses to the second level (i.e.  $\text{NEXP}^{\text{NP}} = \text{coNEXP}^{\text{NP}}$ ). In Theorem 4.1, we improve this collapse to  $\text{S}_2^{\text{exp}}$ . In Theorem 4.3 we obtain an exponential version of Yap's theorem [Yap83]. We prove that if NP is contained in  $\text{coNP}/\text{qpoly}$ , then the exponential hierarchy collapses to  $\text{S}_2^{\text{exp}} \circ \text{P}^{\text{NP}}$ .

We note that Cai et al. [CCHO03] improved Yap's theorem. They use self-reducibility of a language in  $\text{NP}^A$  (for any set  $A$ ) to show that  $\text{NP} \subseteq \text{coNP}/\text{poly} \implies \text{PH} = \text{S}_2 \circ \text{P}^{\text{NP}}$ . Theorem 4.2 in this section is somewhat similar in form to the result of Cai et al. However, we use a completely different technique from theirs. Furthermore, in Theorem 4.4 below, we will use our technique to give an independent (and hopefully easier) proof of their result.

**Theorem 4.1** *If every set in NP has a quasipolynomial-size family of circuits, then the exponential hierarchy collapses to  $\text{S}_2^{\text{exp}} \subseteq \text{NEXP}^{\text{NP}} \cap \text{coNEXP}^{\text{NP}}$ .*

**Proof** Buhrman and Homer showed under the same assumption that the exponential hierarchy collapses to  $\text{NEXP}^{\text{NP}}$ . Since  $\text{S}_2^{\text{exp}} \subseteq \text{NEXP}^{\text{NP}} \cap \text{coNEXP}^{\text{NP}}$  (Proposition 2.7), it suffices to show that  $\text{NEXP}^{\text{NP}} = \text{S}_2^{\text{exp}}$ .

We can assume that any circuit for SAT outputs not only 1 or 0 indicating whether the input formula is satisfiable or not, but also outputs a satisfying assignment when it claims that the input formula is satisfiable. This can be done by a polynomial blow-up in the size of the circuit, and therefore, the size of the circuit still remains quasipolynomial.

Let  $L \in \text{NEXP}^{\text{NP}}$  be accepted by a nondeterministic machine  $N$  with SAT as an oracle. There is some  $k > 0$  such that  $N$  runs in time  $2^{n^k}$  on any input of length  $n$ . Therefore, the formulas queried by  $N$  on any input of length  $n$  are of size  $m \leq 2^{n^k}$ , and therefore, have circuit size  $2^{\text{polylog}(m)} = 2^{n^c}$ , for some  $c$ .

Let  $x, |x| = n$ , be an input. We define a polynomial-time relation  $V(x, y_1, y_2)$  as follows. It may help to think of  $y_1$  as the proof of the yes-prover, and  $y_2$  as the proof of the no-prover.

1.  $V(x, y_1, y_2)$  holds only if  $y_1$  encodes an accepting computation of  $N$  on  $x$ , with queries, their answers, and for every query  $\phi$  that is answered “yes”, the satisfying assignment of  $\phi$ .
2. If  $y_1$  is of the form specified in item 1, then  $V(x, y_1, y_2)$  holds unless all of the following are true:
  - (a)  $y_2$  encodes a circuit  $C_m$  for strings of length  $m$ . Recall that  $C_m$  should output a satisfying assignment when the input formula  $\phi$  is satisfiable
  - (b) There is a query  $\phi$  that is answered “no” in the path encoded by  $y_1$  but  $C_m(\phi)$  outputs an assignment that satisfies  $\phi$

It is easy to see that this relation requires at most polynomial time in  $(|x| + |y_1| + |y_2|)$ . If  $x \in L$ , then let  $y_1$  be the string encoding the correct accepting computation of  $N$  on  $x$ , including the queries and their answers. Since the “no” queries are answered correctly on this path, for every “no” query  $\phi$ ,  $\phi \notin \text{SAT}$ , and therefore, no circuit (correct or otherwise) can output a satisfying assignment of  $\phi$ . As a consequence,  $V(x, y_1, y_2)$  will hold.

On the other hand, if  $x \notin L$ , then let  $y_2$  be the encoding of a correct circuit  $C_m$  for formulas of length  $m$ . Any  $y_1$  that satisfies item 1 must be incorrect about some query  $q$  that is in SAT but is answered “no” on the computation path encoded in  $y_1$ . For any such  $\phi$ ,  $C_m(\phi)$  will output a satisfying assignment for  $\phi$ , and therefore,  $V(x, y_1, y_2)$  cannot hold.

Finally, we need to argue that the proofs are of exponential length. The length of a circuit is  $2^{n^c}$  for some constant  $c$ . Due to the exponential bound on the running time of  $N$ , on the number of queries made by  $N$ , on the length of each query made by  $N$ , and on the length of  $y_q$ , for any  $q$ , the length of  $y_1$  is at most exponential in  $n$  as well. This completes the proof.  $\square$

Now we consider the exponential version of Yap’s theorem.

**Theorem 4.2**  $\text{NP} \subseteq \text{coNP}/\text{qpoly} \implies \text{NQPOLY}^{\Sigma_2^P} = \text{coNQPOLY}^{\Sigma_2^P} = \text{S}_2^{\text{qpoly}} \circ \text{P}^{\text{NP}}$ .

**Proof** Since  $\text{S}_2^{\text{qpoly}} \circ \text{P}^{\text{NP}}$  is closed under complement, it suffices to show under the hypothesis that  $\text{NQPOLY}^{\Sigma_2^P} = \text{S}_2^{\text{qpoly}} \circ \text{P}^{\text{NP}}$ . Let  $L \in \text{NQPOLY}^{\Sigma_2^P}$  via some quasipolynomial-time nondeterministic oracle machine  $N$  that has some  $\Sigma_2^P$  language  $A$  as an oracle. For any input  $x \in \{0, 1\}^n$ ,  $N$  runs in  $2^{\log^k n}$  time. Therefore, any query that  $N$  makes to  $A$  is also of length  $2^{\log^k n}$ , and the number of queries is also bounded by  $2^{\log^k n}$ .

For any string  $q$ ,  $q \in A \Leftrightarrow \exists y_q \phi_{q,y_q} \notin \text{SAT}$ . Note that  $\phi_{q,y_q}$  can be constructed from  $q$  and  $y_q$  in time polynomial in  $|q|$ .

For any string  $q$  of length  $2^{\log^k n}$ , let  $|\phi_{q,y_q}|$  be denoted by  $m$  (some quasipolynomial in  $n$ ). By our assumption, SAT is in  $\text{coNP}/\text{qpoly}$ ; let us assume that  $w$  is a correct advice for strings of length  $m$ , such that  $|w| = 2^{\text{polylog}(m)} = 2^{\log^c n}$  for some constant  $c$ , and let  $B \in \text{coNP}$  be the witness language. For any string  $q$ ,

$$\begin{aligned} q \notin A &\Leftrightarrow \forall y_q \phi_{q,y_q} \in \text{SAT} \\ &\Leftrightarrow \forall y_q (\phi_{q,y_q}, w) \in B \\ &\Leftrightarrow (q, w) \in C, \end{aligned}$$

where  $C = \{(q, w) \mid \forall y_q (\phi_{q,y_q}, w) \in B\}$ .

We define a  $\text{P}^{\text{NP}}$ -definable relation  $V(x, y_1, y_2)$  as follows. It may help to think of  $y_1$  as the proof of the yes-prover, and  $y_2$  as the proof of the no-prover.

1.  $V(x, y_1, y_2)$  holds only if  $y_1$  encodes an accepting computation of  $N$  on  $x$ , with queries, their answers, and for every query  $q$  that is answered “yes”, the string  $y_q$  as described above. In addition, the formulas  $\phi_{q,y_q}$  for the yes answers must be unsatisfiable. (This requires making queries to the NP oracle that  $V$  can access.)
2. If  $y_1$  is of the form specified in item 1, then  $V(x, y_1, y_2)$  holds unless all of the following are true:
  - (a)  $y_2$  encodes an advice for strings of length  $m$
  - (b) There is a query  $q$  that is answered “no” in the path encoded by  $y_1$  but  $(q, y_2) \notin C$  (here also  $V$  requires access to the NP oracle)
  - (c) The search procedure described below yields a string  $y_q$  for this query  $q$  such that  $\phi_{q,y_q} \notin \text{SAT}$

Now we describe the search procedure. Assume that a query  $q$  has been answered “no” in the path encoded by  $y_1$ , but  $(q, y_2) \notin C$ . Recall that  $\overline{C} = \{(q, w) \mid \exists y_q (\phi_{q,y_q}, w) \notin B\}$ . Since  $\overline{C}$  is in NP,  $V$  uses a prefix search algorithm that accesses an NP oracle to construct  $y_q$ .

If  $x \in L$ , then let  $y_1$  be the string encoding the correct accepting computation of  $N$  on  $x$ , including the queries and their answers. Since the “no” queries are answered correctly on this path, for every “no” query  $q$ ,  $q \notin A$ , and therefore,  $\forall y_q \phi_{q,y_q} \in \text{SAT}$ . Therefore, the search procedure cannot yield any  $y_q$  for which  $\phi_{q,y_q} \notin \text{SAT}$ . As a consequence,  $V(x, y_1, y_2)$  will hold.

On the other hand, if  $x \notin L$ , then let  $y_2$  be a correct advice string for strings of length  $m$ . Any  $y_1$  that satisfies item 1 must be incorrect about some query  $q$  that is in  $A$  but is answered “no” on the computation path encoded in  $y_1$ . For any such  $q$ ,  $(q, y_2) \notin C$ , and the search procedure will yield some  $y_q$  such that  $\phi_{q,y_q} \notin \text{SAT}$ . Therefore,  $V(x, y_1, y_2)$  cannot hold.

Finally, we need to argue that the proofs are of quasipolynomial length. The length of an advice string is  $2^{\log^c n}$  for some constant  $c$ . Due to the quasipolynomial bound on the running time of  $N$ , on the number of queries made by  $N$ , on the length of each query made by  $N$ , and on the length of  $y_q$  for any  $q$ , the length of  $y_1$  is at most quasipolynomial in  $n$  as well. The relation  $V$  clearly takes time polynomial in  $|y_1|$  and  $|y_2|$ . This completes the proof. □

**Theorem 4.3**  $\text{NP} \subseteq \text{coNP}/\text{qpoly}$  implies that the exponential hierarchy collapses to  $\text{S}_2^{\text{exp}} \circ \text{P}^{\text{NP}} \subseteq \text{NEXP}^{\Sigma_2^{\text{P}}} \cap \text{coNEXP}^{\Sigma_2^{\text{P}}}$ .

**Proof** By Theorem 4.2, under the hypothesis, the quasipolynomial hierarchy collapses to  $\text{S}_2^{\text{qpoly}} \circ \text{P}^{\text{NP}}$ . As a consequence, the exponential hierarchy collapses to  $\text{S}_2^{\text{exp}} \circ \text{P}^{\text{NP}}$ .  $\square$

Using proof ideas from the above theorem, we obtain a different proof of the following result due to Cai et al. [CCHO03].

**Theorem 4.4** ([CCHO03]) *If  $\text{NP} \subseteq \text{coNP}/\text{poly}$ , then  $\text{PH} = \text{S}_2 \circ \text{P}^{\text{NP}}$ .*

**Proof** Since  $\text{S}_2 \circ \text{P}^{\text{NP}}$  is closed under complement, it suffices to show under the hypothesis that  $\text{NP}^{\Sigma_2^{\text{P}}} = \text{S}_2 \circ \text{P}^{\text{NP}}$ . Let  $L \in \text{NP}^{\Sigma_2^{\text{P}}}$  via some polynomial-time nondeterministic oracle machine  $N$  that has some  $\Sigma_2^{\text{P}}$  language  $A$  as an oracle. For any input  $x \in \{0, 1\}^n$ ,  $N$  runs in  $n^k$  time. Therefore, any query that  $N$  makes to  $A$  is also of length  $n^k$ , and the number of queries is also bounded by  $n^k$ .

For any string  $q$ ,  $q \in A \Leftrightarrow \exists y_q \phi_{q,y_q} \notin \text{SAT}$ . Note that  $\phi_{q,y_q}$  can be constructed from  $q$  and  $y_q$  in time polynomial in  $|q|$ .

For any string  $q$  of length  $n^k$ , let  $|\phi_{q,y_q}|$  be denoted by  $m$  (some polynomial in  $n$ ). By our assumption,  $\text{SAT}$  is in  $\text{coNP}/\text{poly}$ ; let us assume that  $w$  is a correct advice for strings of length  $m$ , where  $|w| = \text{poly}(m) = n^c$  for some constant  $c$ , and let  $B \in \text{coNP}$  be the witness language. For any string  $q$ ,

$$\begin{aligned} q \notin A &\Leftrightarrow \forall y_q \phi_{q,y_q} \in \text{SAT} \\ &\Leftrightarrow \forall y_q (\phi_{q,y_q}, w) \in B \\ &\Leftrightarrow (q, w) \in C, \end{aligned}$$

where  $C = \{(q, w) \mid \forall y_q (\phi_{q,y_q}, w) \in B\}$ .

We define a  $\text{P}^{\text{NP}}$ -definable relation  $V(x, y_1, y_2)$  as follows. It may help to think of  $y_1$  as the proof of the yes-prover, and  $y_2$  as the proof of the no-prover.

1.  $V(x, y_1, y_2)$  holds only if  $y_1$  encodes an accepting computation of  $N$  on  $x$ , with queries, their answers, and for every query  $q$  that is answered “yes”, the string  $y_q$  as described above. In addition, the formulas  $\phi_{q,y_q}$  for the yes answers must be unsatisfiable. (This requires making queries to the  $\text{NP}$  oracle that  $V$  can access.)
2. If  $y_1$  is of the form specified in item 1, then  $V(x, y_1, y_2)$  holds unless all of the following are true:
  - (a)  $y_2$  encodes an advice for strings of length  $m$
  - (b) There is a query  $q$  that is answered “no” in the path encoded by  $y_1$  but  $(q, y_2) \notin C$  (here also  $V$  requires access to the  $\text{NP}$  oracle)
  - (c) The search procedure described below yields a string  $y_q$  for this query  $q$  such that  $\phi_{q,y_q} \notin \text{SAT}$

Now we describe the search procedure. Assume that a query  $q$  has been answered “no” in the path encoded by  $y_1$ , but  $(q, y_2) \notin C$ . Recall that  $\overline{C} = \{(q, w) \mid \exists y_q (\phi_{q, y_q}, w) \notin B\}$ . Since  $\overline{C}$  is in NP,  $V$  uses a prefix search algorithm that accesses an NP oracle to construct  $y_q$ .

If  $x \in L$ , then let  $y_1$  be the string encoding the correct accepting computation of  $N$  on  $x$ , including the queries and their answers. Since the “no” queries are answered correctly on this path, for every “no” query  $q$ ,  $q \notin A$ , and therefore,  $\forall y_q \phi_{q, y_q} \in \text{SAT}$ . Therefore, the search procedure cannot yield any  $y_q$  for which  $\phi_{q, y_q} \notin \text{SAT}$ . As a consequence,  $V(x, y_1, y_2)$  will hold.

On the other hand, if  $x \notin L$ , then let  $y_2$  be a correct advice string for strings of length  $m$ . Any  $y_1$  that satisfies item 1 must be incorrect about some query  $q$  that is in  $A$  but is answered “no” on the computation path encoded in  $y_1$ . For any such  $q$ ,  $(q, y_2) \notin C$ , and the search procedure will yield some  $y_q$  such that  $\phi_{q, y_q} \notin \text{SAT}$ . Therefore,  $V(x, y_1, y_2)$  cannot hold.

Finally, we need to argue that the proofs are of polynomial length. The length of an advice string is  $n^c$  for some constant  $c$ . Due to the polynomial bound on the running time of  $N$ , on the number of queries made by  $N$ , on the length of each query made by  $N$ , and on the length of  $y_q$  for any  $q$ , the length of  $y_1$  is at most polynomial in  $n$  as well. The relation  $V$  clearly takes time polynomial in  $|y_1|$  and  $|y_2|$ . This completes the proof. □

## 5 Concluding Remarks

We have shown that if coNP has polylogarithmic-round interactive proofs then the exponential hierarchy collapses to the third level. An obvious extension would be to obtain consequences of  $\overline{\text{SAT}}$  having  $n^\epsilon$ -round interactive proof systems for some  $\epsilon < 1$ .

One longstanding open problem in this area is to show that if SAT has polynomial-sized circuits, then PH collapses to AM. Since  $\text{coNP} \subseteq \text{AM}$  implies that PH collapses to AM, it suffices to show under this hypothesis that  $\text{coNP}$  is included in AM. Moreover, Arvind et al. [AKSS95] have shown that if SAT has a polynomial-size family of circuits, then  $\text{MA} = \text{AM}$ . Since  $\text{MA} \subseteq \text{S}_2^P$ , this would improve the best-known version of the Karp-Lipton theorem [KL80] (by Sengupta, reported in Cai [Cai01]).

After Babai introduced the class AM in 1985 [Bab85], the evidence that coNP does not have constant round interactive proofs unless the polynomial hierarchy collapses first appeared in 1987 [BHZ87]. It is interesting to note that a weaker collapse could be obtained by Yap’s 1983 result [Yap83] and the fact that  $\text{AM} \subseteq \text{NP/poly}$ . Yap showed that if  $\text{coNP} \subseteq \text{NP/poly}$  then the polynomial hierarchy collapses to its third level [Yap83]. Since  $\text{AM} \subseteq \text{NP/poly}$ , this directly implies that if  $\text{coNP} \subseteq \text{AM}$  then the polynomial hierarchy collapses to its third level.

## 6 Acknowledgments

The authors thank D. Sivakumar for suggesting the question that we address in this paper. The second and the third authors thank the program committee of the 19th IEEE Conference on Computational Complexity for their comments and Salil Vadhan for pointers to his paper.

## References

- [AKSS95] V. Arvind, J. Köbler, U. Schöning, and R. Schuler. If NP has polynomial-size circuits, then MA = AM. *Theoretical Computer Science*, 137(2):279–282, 1995.
- [ALM<sup>+</sup>92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, pages 14–22. IEEE Computer Society Press, 1992.
- [AS92] S. Arora and S. Safra. Approximating clique is NP-complete. In *Proceedings of the 33rd Annual IEEE Symposium on Foundations on Computer Science*, pages 2–13. IEEE Computer Society Press, 1992.
- [Bab85] L. Babai. Trading group theory for randomness. In *Proceedings of the 17th Symposium on Theory of Computing*, pages 421–429. ACM Press, 1985.
- [BDG90] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II*. EATCS Monographs on Theoretical Computer Science. Springer Verlag, Berlin Heidelberg, 1990.
- [BDG95] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*. EATCS Monographs on Theoretical Computer Science. Springer Verlag, Berlin Heidelberg, 2nd edition, 1995.
- [BFL81] L. Babai, L. Fortnow, and C. Lund. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1981.
- [BFLS91] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 21–31, 1991.
- [BH92] H. Buhrman and S. Homer. Superpolynomial circuits, almost sparse oracles, and the exponential hierarchy. In *Foundations of Software Technology and Theoretical Computer Science, 12th Conference, New Delhi, India, December 18-20, 1992, Proceedings*, volume 652 of *Lecture Notes in Computer Science*, pages 116–127. Springer-Verlag, 1992.
- [BHZ87] R. B. Boppana, J. Håstad, and S. Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 1987.
- [BM88] L. Babai and S. Moran. Arthur-merlin games : a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.
- [BOGKW88] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multiprover interactive proofs: How to remove the intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 113–131, 1988.
- [Cai01] J-Y. Cai.  $S_2^P \subseteq ZPP^{NP}$ . In *Proceedings of the 42nd IEEE Conference on Foundations of Computer Science*, pages 620–629, 2001.

- [Can96] R. Canetti. On BPP and the polynomial-time hierarchy. *Information Processing Letters*, pages 237–241, 1996.
- [CCHO03] J-Y. Cai, V. Chakaravarthy, L. Hemaspaandra, and M. Ogihara. Competing provers yield improved karp-lipton collapse results. In *Proceedings 20th Symposium on Theoretical Aspects of Computer Science*, pages 535–546, 2003.
- [FGL<sup>+</sup>91] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proceedings 32nd Symposium on Foundations of Computer Science*, pages 2–12. IEEE Computer Society Press, 1991.
- [GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proofs. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 291–304, 1985.
- [GS89] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In S. Micali, editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 73–90. JAI Press, 1989.
- [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 2076 of *Lecture Notes in Computer Science*, pages 334–345. Springer Verlag, 2001.
- [HS01] S. Homer and A. Selman. *Computability and Complexity Theory*. Springer-Verlag, 2001.
- [KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 302–309, 1980.
- [LFKN92] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the Association of Computing Machines*, 39(4):859–868, 1992.
- [RS98] A. Russell and R. Sundaram. Symmetric alternation captures BPP. *Journal of Computational Complexity*, 7(2):152–162, 1998.
- [Sha92] A. Shamir.  $IP = PSPACE$ . *Journal of the Association of Computing Machines*, 39(4):869–877, 1992.
- [Tod91] S. Toda. PP is as hard as the polynomial time hierarchy. *SIAM Journal on Computing*, 20:865–877, 1991.
- [Yap83] C. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26(3):287–300, 1983.
- [ZF87] S. Zachos and M. Fürer. Probabilistic quantifiers vs distrustful adversaries. In *Foundations of Software Technology and Theoretical Computer Science, 1987, Proceedings*, volume 287 of *Lecture Notes in Computer Science*, pages 449–455. Springer-Verlag, 1987.

- [ZH86] S. Zachos and H. Heller. A decisive characterization of BPP. *Information and Control*, 69:125–135, 1986.