Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 7 (2004)

Polylogarithmic-round Interactive Proofs for coNP Collapse the Exponential Hierarchy

Alan L. Selman*
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260

Samik Sengupta†
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260

May 5, 2004

Abstract

It is known [BHZ87] that if every language in coNP has a constant-round interactive proof system, then the polynomial hierarchy collapses. On the other hand, Lund et al. [LFKN92] have shown that $\#\mathrm{SAT}$, the $\#\mathrm{P}$-complete function that outputs the number of satisfying assignments of a Boolean formula, can be computed by a linear-round interactive protocol. As a consequence, the coNP-complete set $\overline{\mathrm{SAT}}$ has a proof system with linear rounds of interaction.

We show that if every set in coNP has a polylogarithmic-round interactive protocol then the exponential hierarchy collapses to the third level. In order to prove this, we obtain an exponential version of Yap's result [Yap83], and improve upon an exponential version of the Karp-Lipton theorem [KL80], obtained first by Buhrman and Homer [BH92].

1 Introduction

Bábai [Báb85] and Bábai and Moran [BM88] introduced *Arthur-Merlin Games* to study the power of randomization in interaction. 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$^+$91, AS92, ALM$^+$92].

Interactive proof systems are successfully placed relative to traditional complexity classes. In particular, it is known that for any constant k, $\mathrm{IP}[k] \subseteq \Pi_2^p$ [BM88], and $\mathrm{IP}[\mathrm{poly}] = \mathrm{PSPACE}$ [Sha92]. 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 $\#\mathrm{SAT}$, a problem hard for the entire polynomial-time hierarchy [Tod91], is accepted by an interactive proof system with $O(n)$ rounds of interaction on an input of length n. Can every set in coNP be accepted by an interactive

*Research partially supported by NSF grant CCR-0307077. Email: selman@cse.buffalo.edu
†Email: samik@cse.buffalo.edu

ISSN 1433-8092

proof system with more than constant but sublinear number of rounds? Answering this question has been the motivation for this paper.

We show in this paper that coNP cannot have a polylogarithmic-round interactive proof system unless the exponential hierarchy collapses to the third level, i.e., $\mathrm{NEXP}^{\Sigma_k^p} = \mathrm{NEXP}^{\Sigma_2^p}$ for any $k > 2$. Three principal steps lead to the proof of our main result. Although we use Arthur-Merlin protocols to obtain our results, our main theorem holds for general interactive proof systems as well, due to the result of Goldwasser and Sipser [GS89], who showed that an interactive proof system with m rounds can be simulated by an $2m + 4$-move Arthur-Merlin protocol.

- Using a result of Goldreich, Vadhan, and Wigderson [GVW02], we show that an Arthur-Merlin protocol with polylogarithmic moves can be simulated by a two-move Arthur-Merlin protocol where both Arthur and Merlin send at most quasipolynomial (2^{polylog}) number of bits (Corollary 3.3).
- If L is accepted by a two-move AM protocol where both Merlin and Arthur send quasipolynomially many bits, then L belongs to the advice class $\mathrm{NP}/\mathrm{qpoly}$ (Lemma 3.4).
- If $\mathrm{coNP} \subseteq \mathrm{NP}/\mathrm{qpoly}$ (equivalently $\mathrm{NP} \subseteq \mathrm{coNP}/\mathrm{qpoly}$) then the exponential hierarchy collapses to

$$\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{NEXP}^{\Sigma_2^p} \cap \mathrm{coNEXP}^{\Sigma_2^p} \text{ (Theorem 4.2).}$$

In addition to these results, we improve upon a result of Buhrman and Homer [BH92], showing that if every set in NP has a quasipolynomial-size family of circuits, then $\mathrm{NEXP}^{\mathrm{NP}} = \mathrm{coNEXP}^{\mathrm{NP}} = \mathrm{S}_2^{\mathrm{EXP}}$.

2 Preliminaries

For definitions of standard complexity classes, we refer the reader to Homer and Selman [HS01]. The exponential hierarchy is defined as follows:

$$\mathrm{EXP} = \Sigma_0^{exp}, \mathrm{NEXP} = \Sigma_1^{exp}, \mathrm{NEXP}^{\mathrm{NP}} = \Sigma_2^{exp},$$

and in general, for $k \geq 0$,

$$\Sigma_{k+1}^{exp} = \mathrm{NEXP}^{\Sigma_k^p}.$$

For every $k \geq 0$,

$$\Pi_k^{exp} = \{L \mid \overline{L} \in \Sigma_k^{exp}\}.$$

We define $\mathrm{polylog} = \bigcup_{k>0} \log^k n$ and $\mathrm{qpoly} = 2^{\mathrm{polylog}} = \bigcup_{k>0} 2^{\log^k n}$.

The *quasipolynomial hierarchy* has been studied before [BH92]. Buhrman and Homer [BH92] call it the PL-hierarchy. Define

$$\Sigma_0^{\mathrm{qpoly}} = \mathrm{QPOLY} = \bigcup_{c>0} \mathrm{DTIME}(2^{\log^c n}),$$

$$\Sigma_1^{\mathrm{qpoly}} = \mathrm{NQPOLY} = \bigcup_{c>0} \mathrm{NTIME}(2^{\log^c n}),$$

and in general, for $k \geq 1$,

$$\Sigma_{k+1}^{\mathrm{qpoly}} = \mathrm{NQPOLY}^{\Sigma_k^p}.$$

For every $k \geq 0$,

$$\Pi_k^{\mathrm{qpoly}} = \{L \mid \overline{L} \in \Sigma_k^{\mathrm{qpoly}}\}.$$

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 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 2.1 *For every $k > 0$,*

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

As a consequence, there is no tally set in $\Sigma_k^{\mathrm{qpoly}} - \Sigma_{k-1}^{\mathrm{qpoly}}$ if and only if $\Sigma_k^{exp} = \Sigma_{k-1}^{exp}$. 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.

Proposition 2.2 *If $\Sigma_k^{\mathrm{qpoly}} = \Pi_k^{\mathrm{qpoly}}$, then the quasipolynomial hierarchy collapses to the k-th level.*

We note that the analogous result is not known for the exponential hierarchy.

Let $\mathcal{C}$ be a complexity class. A set $L \in \mathcal{C}/\mathrm{qpoly}$ if there is a function $s : 1^* \to \Sigma^*$, 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 $\mathcal{D} \subseteq \mathcal{C}/\mathrm{qpoly}$ if and only if $\mathrm{co}\mathcal{D} \subseteq \mathrm{co}\mathcal{C}/\mathrm{qpoly}$.

Bábai [Báb85] introduced *Arthur-Merlin protocol*, a combinatorial 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 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 on a read-write communication tape. Arthur's moves depend on his random bits. After the last move, Arthur either accepts or does not accept x.

Definition 2.3 ([Báb85, BM88]) *For any $m > 0$, a language L is in $\mathrm{AM}[m]$ (respectively $\mathrm{MA}[m]$) if for every string x of length n*

- *The game consists of m moves*
- *Arthur (resp., Merlin) moves first*
- *After the last move, Arthur behaves deterministically to either accept or not accept 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 $\frac{1}{4}$.*

Bábai and Moran [BM88] showed that $\mathrm{AM}[k]$, where $k > 1$ is some constant, is the same as $\mathrm{AM}[2] = \mathrm{AM}$. Note that $\mathrm{MA}[2] = \mathrm{MA}$, $\mathrm{AM}[1] = \mathrm{BPP}$, and $\mathrm{MA}[1] = \mathrm{M} = \mathrm{NP}$. Bábai [Báb85] proved that $\mathrm{MA} \subseteq \mathrm{AM}$.

We note the following standard proposition.

Proposition 2.4 *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 $t \stackrel{def}{=} c \times p(n)$ independent trials, E occurs for more than $\frac{t}{2}$ times with probability $(1 - \frac{1}{2^{p(n)}})$.*

We define S_2^{exp} as the exponential version of the S_2 operator defined by Russell and Sundaram [RS98] and Canetti [Can96]. A set L is in $\mathrm{S}_2^{exp} \circ \mathcal{C}$ 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 define $\mathrm{S}_2^{\mathrm{qpoly}}$ as the quasipolynomial version of the S_2 operator.

Similar to $\mathrm{S}_2^{\mathrm{P}} \stackrel{def}{=} \mathrm{S}_2 \circ \mathrm{P}$, the class $\mathrm{S}_2^{exp} \circ \mathcal{C}$ ($\mathrm{S}_2^{\mathrm{qpoly}} \circ \mathcal{C}$) can be thought of as a game between two provers and a verifier. Let $L \in \mathrm{S}_2^{exp} \circ \mathcal{C}$ (respectively, in $\mathrm{S}_2^{\mathrm{qpoly}} \circ \mathcal{C}$). 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 $\mathcal{C} = \mathrm{P}$, then the verifier is a deterministic polynomial-time Turing machine, and if $\mathcal{C} = \mathrm{P}^{\mathrm{NP}}$, 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 $\mathrm{S}_2^{exp} \circ \mathcal{C}$ (respectively, $\mathrm{S}_2^{\mathrm{qpoly}} \circ \mathcal{C}$) is also closed under complement.

We concentrate on the classes $\mathrm{S}_2^{\mathrm{EXP}} \stackrel{def}{=} \mathrm{S}_2^{exp} \circ \mathrm{P}$, $\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}}$, and $\mathrm{S}_2^{\mathrm{qpoly}} \circ \mathrm{P}^{\mathrm{NP}}$. The proofs of Russell and Sundaram can be easily modified to show the following.

Proposition 2.5

1. $\mathrm{S}_2^{\mathrm{EXP}} \subseteq \mathrm{NEXP}^{\mathrm{NP}} \cap \mathrm{coNEXP}^{\mathrm{NP}}$.
2. $\mathrm{NEXP}^{\mathrm{NP}} \cup \mathrm{coNEXP}^{\mathrm{NP}} \subseteq \mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{NEXP}^{\Sigma_2^{\mathrm{P}}} \cap \mathrm{coNEXP}^{\Sigma_2^{\mathrm{P}}}$.
3. $\mathrm{NQPOLY}^{\mathrm{NP}} \cup \mathrm{coNQPOLY}^{\mathrm{NP}} \subseteq \mathrm{S}_2^{\mathrm{qpoly}} \circ \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{NQPOLY}^{\Sigma_2^{\mathrm{P}}} \cap \mathrm{coNQPOLY}^{\Sigma_2^{\mathrm{P}}}$.

Proof We give a short proof of the second inclusion of item (2). Other inclusions are easy to verify. Note that since $\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}}$ is closed under complement, it suffices to show that $\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}}$ is a subset of $\mathrm{NEXP}^{\Sigma_2^{\mathrm{P}}}$. Let $L \in \mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}}$; therefore, $\exists k > 0, L' \in \mathrm{P}^{\mathrm{NP}}$ 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 Σ_2^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.

□

Proposition 2.6

$$L \in \mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}} \Leftrightarrow \mathrm{Tally}(L) \in \mathrm{S}_2^{\mathrm{qpoly}} \circ \mathrm{P}^{\mathrm{NP}}.$$

Proof We simply show the if direction; the only if direction is similar. Let $L \in \mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}}$; therefore, there exists $k > 0$ and $V \in \mathrm{P}^{\mathrm{NP}}$ 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 Tally(L) is in $\mathrm{S}_2^{\mathrm{qpoly}} \circ \mathrm{P}^{\mathrm{NP}}$. 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 P^{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 $w \in \mathrm{Tally}(L)$, then $x \in L$ and y_x will convince the verifier; on the other hand, if $w \notin \mathrm{Tally}(L)$, 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 $\mathrm{S}_2^{\mathrm{qpoly}} \circ \mathrm{P}^{\mathrm{NP}}$ protocol for Tally(L). □

The following proposition follows immediately.

Proposition 2.7 $\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}} = \mathrm{NEXP}^{\Sigma_2^{\mathrm{P}}}$ *if and only if there is no tally set in* $\mathrm{NQPOLY}^{\Sigma_2^{\mathrm{P}}} - \mathrm{S}_2^{\mathrm{qpoly}} \circ \mathrm{P}^{\mathrm{NP}}$.

3 Arthur-Merlin Games with Polylogarithmic Moves

We apply a theorem of Goldreich, Vadhan, and Wigderson [GVW02, Theorem 2.3] to obtain Corollary 3.3, where we prove that if coNP has a polylogarithmic-move Arthur-Merlin protocol, then coNP can be accepted by a two-move Arthur-Merlin protocol where both Arthur and Merlin exchange quasipolynomially many bits. As a consequence, using Lemma 3.4, we obtain that if coNP has a polylogarithmic-move Arthur-Merlin protocol, then coNP can be solved by nondeterministic polynomial-time machines with quasipolynomial-length advice.

Definition 3.1 ([GVW02]) *A set* $L \in \mathrm{AM}[b(n), m(n)]$ *if for every string of length* n *there is an* $m(n)$*-move Arthur-Merlin protocol where Arthur moves first and Merlin sends a total of at most* $b(n)$ *bits. Note that the running time of Arthur is bounded by polynomial in* n *and* $b(n)$.

In this manner the notion of Arthur-Merlin protocols is modestly extended to allow for the possibility that Arthur is not polynomial-time-bounded. Below we will consider two-move Arthur-Merlin protocols where $l(n)$ is a quasipolynomial; that is, we will consider the class $\mathrm{AM}[\mathrm{qpoly}, 2]$.

Proposition 3.2 ([GVW02])

$$\mathrm{AM}[b(n), m(n)] \subseteq \mathrm{AM}[(b(n) \cdot m(n))^{O(m(n))}, 2].$$

We denote $\mathrm{AM}[f, 2]$ by $\mathrm{AM}(f)$.

Corollary 3.3 *For any* $k > 0$*, there is a* $c > 0$ *such that*

$$\mathrm{AM}[\log^k n] \subseteq \mathrm{AM}(2^{\log^c n}).$$

Proof Let $L \in \mathrm{AM}[\log^k n]$. Assume that Arthur and Merlin exchange at most n^d bits during every move of the protocol that accepts L, where $d > 0$ is some constant. Therefore, $L \in \mathrm{AM}[n^d \log^k n, \log^k n]$. By Proposition 3.2, there is a constant k' such that $\mathrm{AM}[n^d \log^k n, \log^k n] \subseteq \mathrm{AM}((n^d \log^k n \times \log^k n)^{k' \log^k n})$. Note that for large enough n,

$$(n^d \log^k n \times \log^k n)^{k' \log^k n} \leq (2^{d \log n + 2k \log \log n})^{k' \log^k n}$$
$$\leq (2^{(2k+d) \log n})^{k' \log^k n} = 2^{k'(2k+d) \log^{k+1} n} \leq 2^{\log^{k+2} n}.$$

Taking $c = k + 2$, we have that $L \in \mathrm{AM}[2^{\log^c n}, 2]$. This completes the proof. $\square$

The following lemma is an extension of the result $\mathrm{AM} \subseteq \mathrm{NP}/\mathrm{poly}$, which in turn is an extension of Adleman's result that $\mathrm{BPP} \subseteq \mathrm{P}/\mathrm{poly}$ [Adl78].

Lemma 3.4 $\mathrm{AM}(\mathrm{qpoly}) \subseteq \mathrm{NP}/\mathrm{qpoly}$.

Proof Let $L \in \mathrm{AM}(2^{\log^k n})$. Consider any input x of length n. There is a constant k and a polynomial-time predicate R such that

$$x \in L \implies \Pr_y[\exists z \; R(x, y, z)] \geq \frac{3}{4}$$

and

$$x \notin L \implies \Pr_y[\exists z \; R(x, y, z)] \leq \frac{1}{4},$$

where $|y|, |z| \leq 2^{\log^k n}$. Note that by repeating the above protocol $c_1 n$ times, for some constant c_1, we can reduce the probability of error to $\frac{1}{2^{n+1}}$. Therefore, for every $x \in \{0, 1\}^n$, we get

$$x \in L \implies \Pr_y[\exists z \; R(x, y, z)] \geq 1 - \frac{1}{2^{n+1}}$$

and

$$x \notin L \implies \Pr_y[\exists z \; R(x, y, z)] \leq \frac{1}{2^{n+1}},$$

where $|y| \leq c_1 n \times 2^{\log^k n} = 2^{\log^k n + \log c_1 + \log n} \leq 2^{\log^c n}$ for some appropriate $c > k$. There are at most 2^n many strings of length n, and for every y the error probability is at most $\frac{1}{2^{n+1}}$. Therefore any random y will be correct on every input string with probability at least $1 - (2^n \times \frac{1}{2^{n+1}}) > 0$. Hence there must be some $\hat{y}, |\hat{y}| \leq 2^{\log^c n}$ such that the following holds for every x of length n:

$$x \in L \implies \exists z \; R(x, \hat{y}, z)$$

and

$$x \notin L \implies \forall z \; \neg R(x, \hat{y}, z).$$

This shows that $L \in \mathrm{NP}/2^{\log^c n}$. $\square$

Corollary 3.5 *For any constant $k > 0$, there is a constant $c > 0$ such that*

$$\mathrm{coNP} \subseteq \mathrm{AM}[\log^k n] \implies \mathrm{coNP} \subseteq \mathrm{NP}/2^{\log^c n}.$$

Proof This follows directly from Corollary 3.3 and Lemma 3.4. $\square$

4 Quasipolynomial advice for NP

In this section, we study the consequences of the existence of quasipolynomial length (i.e., 2^{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. $\mathrm{NEXP}^{\mathrm{NP}} = \mathrm{coNEXP}^{\mathrm{NP}}$). In Theorem 4.6, we improve this collapse to $\mathrm{S}_2^{\mathrm{EXP}}$. In Theorem 4.2 we obtain an exponential version of Yap's theorem [Yap83]. We prove that if NP is contained in $\mathrm{coNP}/\mathrm{qpoly}$, then the exponential hierarchy collapses to $\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}}$. We use this theorem to obtain the central technical result of this paper, which is Theorem 4.3.

We note that Cai et al. [CCHO03] improved Yap's theorem . They use self-reducibility of a language in NP^A (for any set A) to show that $\mathrm{NP} \subseteq \mathrm{coNP}/\mathrm{poly} \implies \mathrm{PH} = \mathrm{S}_2 \circ \mathrm{P}^{\mathrm{NP}}$. Theorem 4.1 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 4.7 below, we will use our technique to give an independent (and hopefully easier) proof of their result.

Theorem 4.1 $\mathrm{NP} \subseteq \mathrm{coNP}/\mathrm{qpoly} \implies \mathrm{NQPOLY}^{\Sigma_2^{\mathrm{P}}} = \mathrm{coNQPOLY}^{\Sigma_2^{\mathrm{P}}} = \mathrm{S}_2^{\mathrm{qpoly}} \circ \mathrm{P}^{\mathrm{NP}}$.

Proof Since $\mathrm{S}_2^{\mathrm{qpoly}} \circ \mathrm{P}^{\mathrm{NP}}$ is closed under complement, it suffices to show under the hypothesis that $\mathrm{NQPOLY}^{\Sigma_2^p} = \mathrm{S}_2^{\mathrm{qpoly}} \circ \mathrm{P}^{\mathrm{NP}}$. Let $L \in \mathrm{NQPOLY}^{\Sigma_2^p}$ via some quasipolynomial-time nondeterministic oracle machine N that has some Σ_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, $q \in A \Leftrightarrow \exists y_q\ \phi_{q,y_q} \notin \mathrm{SAT}$. Note that ϕ_{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 $\mathrm{coNP}/\mathrm{qpoly}$; let us assume that a correct advice for strings of length m is w, where $|w| = 2^{\mathrm{polylog}(m)} = 2^{\log^c n}$ for some constant c, and let $B \in \mathrm{coNP}$ be the witness language. For any string q,

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

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

We define a P^{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 ϕ_{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 $\phi_{q,y_q} \notin \mathrm{SAT}$

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, $\forall y_q \ \phi_{q,y_q} \in \mathrm{SAT}$. Therefore, the search procedure cannot yield any y_q for which $\phi_{q,y_q} \notin \mathrm{SAT}$. 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 $\phi_{q,y_q} \notin \mathrm{SAT}$. 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 4.2 $\mathrm{NP} \subseteq \mathrm{coNP/qpoly}$ *implies that the exponential hierarchy collapses to* $\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{NEXP}^{\Sigma_2^{\mathrm{P}}} \cap \mathrm{coNEXP}^{\Sigma_2^{\mathrm{P}}}$.

Proof By Theorem 4.1, under the hypothesis, the quasipolynomial hierarchy collapses to $\mathrm{S}_2^{\mathrm{qpoly}} \circ \mathrm{P}^{\mathrm{NP}}$. As a consequence, the exponential hierarchy collapses to $\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}}$. □

Now we prove our main theorem.

Theorem 4.3 *For every constant k, if* $\mathrm{coNP} \subseteq \mathrm{AM}[\log^k n]$, *then the exponential hierarchy collapses to* $\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{NEXP}^{\Sigma_2^{\mathrm{P}}} \cap \mathrm{coNEXP}^{\Sigma_2^{\mathrm{P}}}$.

Proof If every language in coNP has an Arthur-Merlin proof system with $\log^k n$ moves for any $k > 0$, then by Corollary 3.5, we obtain that $\mathrm{coNP} \subseteq \mathrm{NP}/2^{\log^c n}$ for some constant $c > 0$. This is equivalent to saying that $\mathrm{NP} \subseteq \mathrm{coNP}/2^{\log^c n}$. By Theorem 4.2, we get the consequence that the exponential hierarchy collapses to $\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{NEXP}^{\Sigma_2^{\mathrm{P}}} \cap \mathrm{coNEXP}^{\Sigma_2^{\mathrm{P}}}$. This completes the proof. □

Corollary 4.4 *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* $\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{NEXP}^{\Sigma_2^{\mathrm{P}}} \cap \mathrm{coNEXP}^{\Sigma_2^{\mathrm{P}}}$.

Proof Goldreich, Vadhan, and Wigderson [GVW02, 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 $\overline{L} \in \mathrm{AM(qpoly)}$. Therefore, if every set in NP has such an interactive proof system, then $\mathrm{coNP} \subseteq \mathrm{AM(qpoly)}$, and therefore, by Lemma 3.4, $\mathrm{coNP} \subseteq \mathrm{NP/qpoly}$. This is equivalent to saying that $\mathrm{NP} \subseteq \mathrm{coNP/qpoly}$. By Theorem 4.2, we obtain the consequence that the exponential hierarchy collapses to $\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{NEXP}^{\Sigma_2^{\mathrm{P}}} \cap \mathrm{coNEXP}^{\Sigma_2^{\mathrm{P}}}$.
□

We can prove a version of Theorem 4.3 for $(\log n)^{\log \log n}$-round interactive proof for $\overline{\mathrm{SAT}}$. Let NEEXP be the set of languages that can be decided by a nondeterministic Turing machine that takes at most $2^{2^{n^k}}$ time on an input of length n, and let $\mathrm{coNEEXP} = \{L \mid \overline{L} \in \mathrm{NEEXP}\}$.

Let

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

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

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

Theorem 4.5 *If* $\overline{\mathrm{SAT}} \in \mathrm{AM}[(\log n)^{\log\log n}]$, *then* $\Sigma_4^{\mathrm{eexp}} = \Sigma_3^{\mathrm{eexp}}$, *and therefore, the double exponential hierarchy collapses to the third level.*

Proof Suppose $L \in \Sigma_4^{\mathrm{eexp}}$ via a NEEXP machine M that has some oracle $A \in \Sigma_3^p$. We will design a machine N that accepts L with a Σ_2^p oracle. Let us assume that the running time of M on an input of length n is $m = 2^{2^{n^c}}$.

On an input x, $|x| = n$, N guesses an accepting path of M with queries $q_1, q_2, \cdots$. The number of queries, as well as the length of each query, is bounded by m. We know that there is a polynomial $p(\cdot)$ and some polynomial-time predicate R such that $\forall i$, $i \leq m$,

$$\begin{aligned} q_i \in A &\Leftrightarrow \exists y \; \forall z \; \exists v \; R(q_i, y, z, v) \\ &\Leftrightarrow \exists y \; \forall z \; (q_i, y, z) \in B, \end{aligned}$$

where $B \in \mathrm{NP}$. Let $|(q_i, y, z)| = m'$, where m' is some polynomial in m. We know from the hypothesis that $\mathrm{NP} \subseteq \mathrm{coNP}/2^{\log n^{c \log\log n}}$. Let the witness language be $C \in \mathrm{coNP}$, and let w be an advice string for words of length m'. Since $m' = 2^{2^{n^d}}$, for some $d > 0$, we have $|w| = 2^{(2^{n^d})^{cn^d}} \leq 2^{2^{n^a}}$, for some constant a. Then,

$$q_i \in A \Leftrightarrow \exists y \; \forall z \; (q_i, y, z, w) \in C,$$

and therefore,

$$q_i \in A \Leftrightarrow \exists y \; (q_i, y, w) \in D,$$

where D is in coNP. Therefore,

$$q_i \in A \Leftrightarrow (q_i, w) \in D',$$

where $D' \in \Sigma_2^p$. As a consequence, if N is given a correct advice string w, then N can simply make queries to the Σ_2^p set D' to obtain the answers to the queries q_i. The time taken by N is polynomial in $|w|$, which is still doubly exponential in n.

We now show how N can obtain a correct advice string w for words of length m'. This will complete the proof.

We describe an NP oracle machine N' with oracle SAT and we let

$$\text{INCORR-ADVICE} = L(N'^{\mathrm{SAT}}).$$

Thus, the set INCORR-ADVICE is in Σ_2^p.

On input w of length l, N' guesses a formula ϕ of length l' such that l is the length of the advice for formulas of length l'. Note that

$$\phi \in \mathrm{SAT} \Leftrightarrow (\phi, w) \in C.$$

N' makes two queries: whether $\phi \in \mathrm{SAT}$, and whether $(\phi, w) \in \overline{C}$. N' accepts w if and only if both the queries are answered identically.

We claim that w is not in the set INCORR-ADVICE if and only if w is a correct advice string. If w is the correct advice string, for every formula ϕ, $\phi \in \mathrm{SAT} \Leftrightarrow (\phi, w) \in C$. Therefore, for no formula ϕ, will both the queries be answered the same. On the other hand, if w is not the correct advice string, there must be a formula ϕ such that either $\phi \in$ SAT and $(\phi, w) \notin C$, or $\phi \notin$ SAT and $(\phi, w) \in C$. For the path of N' that guesses this formula ϕ, the path will accept. This shows that w is a correct advice string if and only if w is not in INCORR-ADVICE. Recall that INCORR-ADVICE is in Σ_2^p.

To generate a correct advice string w, N simply guesses a string w of appropriate length and asks the Σ_2^p oracle whether w is in INCORR-ADVICE. Since there is at least one correct advice string, at least one of the guessed strings will not be in INCORR-ADVICE, and therefore, will be identified by N to be a correct advice string. This completes the proof.

□

In the following theorem, we improve the result of Buhrman and Homer [BH92, Theorem 1], who showed under the same hypothesis that the exponential hierarchy collapses to $\mathrm{NEXP}^{\mathrm{NP}}$.

Theorem 4.6 *If every set in* NP *has a quasipolynomial-size family of circuits, then the exponential hierarchy collapses to* $\mathrm{S}_2^{\mathrm{EXP}} \subseteq \mathrm{NEXP}^{\mathrm{NP}} \cap \mathrm{coNEXP}^{\mathrm{NP}}$.

Proof Buhrman and Homer showed under the same assumption that the exponential hierarchy collapses to $\mathrm{NEXP}^{\mathrm{NP}}$. Since $\mathrm{S}_2^{\mathrm{EXP}} \subseteq \mathrm{NEXP}^{\mathrm{NP}} \cap \mathrm{coNEXP}^{\mathrm{NP}}$ (Proposition 2.5), it suffices to show that $\mathrm{NEXP}^{\mathrm{NP}} = \mathrm{S}_2^{\mathrm{EXP}}$.

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 $L \in \mathrm{NEXP}^{\mathrm{NP}}$ 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 $2^{\mathrm{polylog}(m)} = 2^{n^c}$, 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 ϕ that is answered "yes", the satisfying assignment of ϕ.

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 ϕ is satisfiable

   (b) There is a query ϕ that is answered "no" in the path encoded by y_1 but $C_m(\phi)$ outputs an assignment that satisfies ϕ

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 ϕ, $\phi \notin$ SAT, and therefore, no circuit (correct or otherwise) can output a satisfying assignment of ϕ. 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 ϕ, $C_m(\phi)$ will output a satisfying assignment for ϕ, 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 exponenial 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 improve Yap's theorem.

Theorem 4.7 *If* $\mathrm{NP} \subseteq \mathrm{coNP/poly}$, *then* $\mathrm{PH} = \mathrm{S}_2 \circ \mathrm{P}^{\mathrm{NP}}$.

Proof Since $\mathrm{S}_2 \circ \mathrm{P}^{\mathrm{NP}}$ is closed under complement, it suffices to show under the hypothesis that $\mathrm{NP}^{\Sigma_2^p} = \mathrm{S}_2 \circ \mathrm{P}^{\mathrm{NP}}$. Let $L \in \mathrm{NP}^{\Sigma_2^p}$ via some polynomial-time nondeterministic oracle machine N that has some Σ_2^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, $q \in A \Leftrightarrow \exists y_q\ \phi_{q,y_q} \notin \mathrm{SAT}$. Note that ϕ_{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, SAT is in $\mathrm{coNP/poly}$; let us assume that w is a correct advice for strings of length m, where $|w| = \mathrm{poly}(m) = n^c$ for some constant c, and let $B \in \mathrm{coNP}$ be the witness language. For any string q,

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

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

We define a P^{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 ϕ_{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 $\phi_{q,y_q} \notin \mathrm{SAT}$

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, $\forall y_q\ \phi_{q,y_q} \in \mathrm{SAT}$. Therefore, the search procedure cannot yield any y_q for which $\phi_{q,y_q} \notin \mathrm{SAT}$. 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 $\phi_{q,y_q} \notin \mathrm{SAT}$. 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.

□

4.1 Interactive Proof Systems

Let $\mathrm{IP}[g(n)]$ denote an interactive proof system with $g(n)$ rounds in the Goldwasser, Micali and Rackoff [GMR85] formalization. Goldwasser and Sipser [GS89] proved that $\mathrm{IP}[g(n)] \subseteq \mathrm{AM}[2g(n) + 4]$ as long as $g(n)$ is bounded by a polynomial. Thus, if $L \in \mathrm{IP}[\log^k n]$, then $L \in \mathrm{AM}[\log^{k+1} n]$. So the following corollary follows immediately from Theorem 4.3.

Corollary 4.8 *If every set in* coNP *has a polylogarithmic-round interactive proof system, then the quasipolynomial hierarchy collapses to*

$$\mathrm{S}_2^{\mathrm{qpoly}} \circ \mathrm{P}^{\mathrm{NP}} = \mathrm{NQPOLY}^{\Sigma_2^{\mathrm{P}}} \cap \mathrm{coNQPOLY}^{\Sigma_2^{\mathrm{P}}}.$$

Hence, under the same hypothesis, the exponential hierarchy collapses to

$$\mathrm{S}_2^{exp} \circ \mathrm{P}^{\mathrm{NP}} = \mathrm{NEXP}^{\Sigma_2^{\mathrm{P}}} \cap \mathrm{coNEXP}^{\Sigma_2^{\mathrm{P}}}.$$

5 Conclusions

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

Aiello, Goldwasser and Håstad [AGH90] have shown that AM is properly included in AM[polylog] in a relativized world. Goldreich, Vadhan, and Wigderson [GVW02, Theorem 3.10] showed that AM is a proper subset of AM[polylog] unless #SAT has a two-move Arthur-Merlin protocol where Merlin can send at most subexponentially many bits.

6 Acknowledgments

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

References

[Adl78] L. Adleman. Two theorems on random polynomial time. In *Procedings 19th Symposium on Foundations of Computer Science*, pages 75–83. IEEE Computer Society Press, 1978.

[AGH90] W. Aiello, S. Goldwasser, and J. Håstad. On the power of interaction. *Combinatorica*, 10(1):3–25, 1990.

[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+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.

[Báb85] L. Bábai. Trading group theory for randomness. In *Proceedings of the 17th Symposium on Theory of Computing*, pages 421–429. ACM Press, 1985.

[BFL81] L. Bábai, L. Fortnow, and C. Lund. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1981.

[BFLS91] L. Bábai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symopsium 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. Bábai 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. $\mathrm{S}_2^{\mathrm{P}} \subseteq \mathrm{ZPP}^{\mathrm{NP}}$. 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 of the 20th Symposium on Theoretical Aspects of Computer Science*, pages 535–546, 2003.

[FGL+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.

[GH98] O. Goldreich and J. Håstad. On the complexity of interactive proofs with bounded communication. *Information Processing Letters*, 67(4):205–214, 1998.

[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.

[GVW02] O. Goldreich, S. Vadhan, and A. Wigderson. On interactive proofs with laconic provers. *Computational Complexity*, 11(1–2):1–53, 2002.

[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 ACM*, 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 ACM*, 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.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'