Nonrelativizing Separations

Harry Buhrman*
CWI
PO Box 94079
1090 GB Amsterdam
The Netherlands

Lance Fortnow†
Dept. of Computer Science
University of Chicago
1100 E. 58th St.
Chicago, IL 60637
USA

Thomas Thierauf‡
Abt. Theoretische Informatik
Universität Ulm
Oberer Eselsberg
89069 Ulm
Germany

Abstract

We show that $\mathbf{MA_{EXP}}$*, the exponential time version of the Merlin-Arthur class, does not have polynomial size circuits. This significantly improves the previous known result due to Kannan since we furthermore show that our result does not relativize. This is the first* separation *result in complexity theory that does* not *relativize. As a corollary to our separation result we also obtain that* **PEXP***, the exponential time version of* **PP** *is not in* $\mathbf{P}/poly$*.*

1 Introduction

Baker, Gill and Solovay [BGS75] noticed that the results proven in computational complexity theory relativize, i.e. the proofs go through virtually unchanged if all of the machines involved have access to the same information via an oracle. They then developed relativized worlds where $\mathbf{P} = \mathbf{NP}$ and $\mathbf{P} \neq \mathbf{NP}$ and thus argued that the current techniques in complexity theory could not settle this question.

More than two decades later, relativization still plays in important role in complexity theory. Though no longer believed to be any form of an independence result, they do help us decide where and how to put our efforts into solving problems in complexity theory. It is still true that virtually all of the theorems in computational complexity theory that have reasonable relativizations do relativize (see [For94]).

But we do have a small number of exceptions that arise from the area of interactive proofs. These results have previously always taken the form of collapses such as $\mathbf{IP} = \mathbf{PSPACE}$ [LFKN92, Sha92], $\mathbf{MIP} = \mathbf{NEXP}$ [BFL91] and $\mathbf{PCP}(O(1), O(\log n)) = \mathbf{NP}$ [ALM+92].

In this paper we give the first reasonable nonrelativizing *separation* results. Namely, we show that there exist languages in $\mathbf{MA_{EXP}}$, a one-round Merlin-Arthur game [BM88] with an exponential-time verifier, that cannot have polynomial-size circuits, in other words, $\mathbf{MA_{EXP}} \not\subseteq \mathbf{P}/poly$. On the other hand, we then create a relativized world where every $\mathbf{MA_{EXP}}$ language has polynomial-size circuits, i.e., $\mathbf{MA_{EXP}} \subseteq \mathbf{P}/poly$.

Paul, Pippenger, Szemerédi and Trotter [PPST83] show a separation of nondeterministic linear time from deterministic linear time where one can create a relativized world where these two classes coincide. However, both the separation and the oracle heavily depend on the machine model where our results are model independent.

We can strengthen our oracle to work for $\mathbf{MIP_{EXP}}$, the languages accepted by *multiple* prover interactive proof systems [BGKW88] with an exponential-time verifier. In fact, besides having small circuits, relative to our oracle $\mathbf{MIP_{EXP}}$ languages can be accepted in $\mathbf{P^{NP}}$ and in $\oplus\mathbf{P}$. Since these classes are contained in $\mathbf{MIP_{EXP}}$, we get a rel-

---

*Email: buhrman@cwi.nl. URL: http://www.cwi.nl/~buhrman. Partially supported by the Dutch foundation for scientific research (NWO) by SION project 612-34-002, and by the European Union through NeuroCOLT ESPRIT Working Group Nr. 8556, and HC&M grant nr. ERB4050PL93-0516.

†Email: fortnow@cs.uchicago.edu. URL: http://www.cs.uchicago.edu/~fortnow. Work done while on leave at CWI. Email: fortnow@cs.uchicago.edu. Supported in part by NSF grant CCR 92-53582, the Dutch Foundation for Scientific Research (NWO) and a Fulbright Scholar award.

‡Email: thierauf@informatik.uni-ulm.de. URL: http://hermes.informatik.uni-ulm.de/ti/Personen/tt.html.

ativized world where

$$\mathbf{MIP_{EXP}} \quad = \quad \mathbf{P}/poly \ \cap \ \mathbf{P^{NP}} \ \cap \ \oplus\mathbf{P}.$$

This contrasts greatly with the situation in the unrelativized world where we have

$$\begin{aligned} \mathbf{P^{NP}} \cup \oplus\mathbf{P} \quad &\subseteq \quad \mathbf{PSPACE} \\ &\subset \quad \mathbf{EXPSPACE} \\ &\subseteq \quad \mathbf{NEEXP} \\ &= \quad \mathbf{MIP_{EXP}}, \end{aligned}$$

where **NEEXP** is nondeterministic double exponential time.

Our proof that $\mathbf{MA_{EXP}}$ does not lie in $\mathbf{P}/poly$ uses the result of Babai, Fortnow, Nisan and Wigderson [BFNW93] that if **EXP** has polynomial-size circuits then **EXP** = **MA**. Since this is the only part of the proof that does not relativize, our paper gives the first oracle where this Babai-Fortnow-Nisan-Wigderson result does not relativize.

As a corollary to our separation result we also obtain the separation between **PEXP**, the exponential time version of **PP**, and $\mathbf{P}/poly$. Finding nonrelativizing results like these allows us to better understand the importance and limitations of relativization and gives us new ideas on how to prove other nonrelativizing results.

2 Preliminaries

We use $\langle x_1, \ldots, x_k\rangle$ to be any polynomial-time computable and invertible tupling function. For a language A, we denote the characteristic function of A as $A(.)$.

We assume the reader familiar with basic notations in complexity theory and classes such as **P** and **NP**. We let $\mathbf{EXP} = \mathbf{DTIME}[2^{n^{O(1)}}]$ and $\mathbf{NEXP} = \mathbf{NTIME}[2^{n^{O(1)}}]$. The class $\mathbf{P}/poly$ consists of languages accepted by a family of polynomial-size circuits or equivalently a polynomial-time Turing machine given a polynomially-length *advice* that depends only on the length of the input. More formally, $L \in \mathbf{P}/poly$ if there exist $A \in \mathbf{P}$ and a polynomially length bounded function $h : \mathbf{N} \mapsto \Sigma^*$ such that for all x

$$x \in L \iff \langle x, h(|x|)\rangle \in A.$$

The value $h(|x|)$ is called the *advice for strings of length* $|x|$.

Let $\#M(x)$ represent the number of accepting computations of a nondeterministic Turing machine $M(x)$, and $\#M_R(x)$ the number of rejecting computations of $M(x)$.

A language L is in $\oplus\mathbf{P}$ if there exists a polynomial-time nondeterministic Turing machine M such that for all x,

$$x \in L \iff \#M(x) \text{ is odd.}$$

A language L is in **PP** if there exists a polynomial-time nondeterministic Turing machine M such that for all x,

$$x \in L \iff \#M(x) > \#M_R(x).$$

PEXP is the exponential-time version of **PP**, i.e., the polynomial-time nondeterministic Turing machine in the definition of **PP** is nondeterministic exponential-time.

A language L is in **MA** if there exists a probabilistic polynomial time Turing machine M and a polynomial $q(n)$ such that

- $x \in L \Rightarrow \exists y \in \Sigma^{q(|x|)}\ \Pr(M(x,y) \text{ accepts}) \geq 2/3$, and
- $x \notin L \Rightarrow \forall y \in \Sigma^{q(|x|)}\ \Pr(M(x,y) \text{ accepts}) \leq 1/3$.

This corresponds to the Merlin-Arthur games due to Babai and Moran [BM88].

We also consider multiple-prover interactive proof systems as developed by Ben-Or, Goldwasser, Kilian and Wigderson [BGKW88]. In this model, the verifier can ask questions of several provers that are unable to communicate with each other. Babai, Fortnow and Lund [BFL91] show that the class, **MIP**, of languages provable by such systems is equal to **NEXP**.

We use $\mathbf{MA_{EXP}}$ and $\mathbf{MIP_{EXP}}$ to represent the Merlin-Arthur games and multiple-prover interactive proof systems where we allow the verifier to use time 2^{n^k} for some fixed k. In particular, the provers can send messages up to this length to the verifier.

3 A Nonrelativizable Separation

The computational power of polynomial-size circuits, $\mathbf{P}/poly$, is an interesting issue. In particular, whether one can solve all sets in **NP** within $\mathbf{P}/poly$ is still an open question, although there are strong indications that this is not possible: otherwise the polynomial hierarchy collapses to Σ_2^p [KL80].

With respect to absolute separations, Kannan [Kan82] showed that there are sets in $\mathbf{NEXP^{NP}}$ that cannot be computed by polynomial-size circuits.

Theorem 3.1 (Kannan)
$\mathbf{NEXP^{NP}} \cap \mathbf{coNEXP^{NP}} \not\subseteq \mathbf{P}/poly.$

We improve Kannan's result and show that there are sets in $\mathbf{MA_{EXP}}$ that cannot be computed by polynomial-size circuits. We use the following result of Babai, Fortnow, Nisan and Wigderson [BFNW93].

Theorem 3.2 (BFNW)
$\mathbf{EXP} \subseteq \mathbf{P}/poly \implies \mathbf{EXP} = \mathbf{MA}$.

In order to prove the separation result we will also need the following lemma.

Lemma 3.3
$\mathbf{NP^{NP}} = \mathbf{MA} \implies \mathbf{NEXP^{NP}} = \mathbf{MA_{EXP}}$.

Proof: Let A be a set in $\mathbf{NEXP^{NP}}$, and let this be witnessed by an alternating Turing machine that runs in time $2^{p(n)}$ for some polynomial p. Consider the following padded version of A:

$$A' = \{\langle x, 0^{2^{p(|x|)}} \rangle \mid x \in A\}.$$

It follows that A' is in $\mathbf{NP^{NP}}$ and by assumption is in $\mathbf{MA}$. This however implies that removing the padding yields that $A \in \mathbf{MA_{EXP}}$. □

We are now ready to prove our separation result.

Theorem 3.4 $\mathbf{MA_{EXP}} \not\subseteq \mathbf{P}/poly.$

Proof: Assume to the contrary that $\mathbf{MA_{EXP}} \subseteq \mathbf{P}/poly$. Since $\mathbf{EXP} \subseteq \mathbf{MA_{EXP}}$ it follows that

$$\mathbf{EXP} \subseteq \mathbf{P}/poly,$$

and we can apply Theorem 3.2. Hence, we have

$$\mathbf{EXP} = \mathbf{MA}.$$

Since $\mathbf{NP^{NP}} \subseteq \mathbf{EXP}$, we get that

$$\mathbf{NP^{NP}} \subseteq \mathbf{MA}.$$

By Lemma 3.3 this yields

$$\mathbf{NEXP^{NP}} \subseteq \mathbf{MA_{EXP}} \subseteq \mathbf{P}/poly.$$

But this contradicts Theorem 3.1. □

The same proof works for $\mathbf{MA_{EXP}} \cap \mathbf{coMA_{EXP}}$ instead of just $\mathbf{MA_{EXP}}$. Then we need the form of Kannan's result as stated in Theorem 3.1.

Corollary 3.5 $\mathbf{MA_{EXP}} \cap \mathbf{coMA_{EXP}} \not\subseteq \mathbf{P}/poly.$

Vereshchagin [Ver92] shows that $\mathbf{MA}$ is in $\mathbf{PP}$. By padding analogously to the proof of Lemma 3.3 this implies that $\mathbf{MA_{EXP}} \subseteq \mathbf{PEXP}$.

Corollary 3.6 $\mathbf{PEXP} \not\subseteq \mathbf{P}/poly.$

4 Relativized Collapse

In this section we show that our separation result from the previous section does *not* relativize. This is the first known example of a non-relativizing separation result.

Theorem 4.1 *There exists an oracle A such that*

$$\mathbf{MA}^A_{\mathbf{EXP}} \subseteq \mathbf{P}^A/poly.$$

Proof: Let M_i be an enumeration of potential verifiers, i.e., probabilistic Turing machines, that run in time 2^n. It is sufficient to encode just these machines: for verifiers that use time 2^{n^k} we can use padding to reduce the language to a verifier that uses 2^n time.

We first describe how to encode inputs of a single length. Later we will show how to combine lengths. For inputs of length n, we will encode languages proven to verifiers $M_1, \ldots, M_n$.

The strings in our oracle A will be of the form $\langle r, i, x\rangle$, where $r \in \Sigma^{5n}$, $1 \le i \le n$ and $x \in \Sigma^n$. Consider the following requirement:

$$(R_{r,i,x}): \qquad \exists y \in \Sigma^{2^n} \ \Pr(M_i^A(x,y) \text{ accepts}) \geq \frac{2}{3} \iff \langle r, i, x\rangle \in A.$$

Our construction of A will guarantee that for all i, one of the following two conditions is fulfilled:

(i) either for some n and $x \in \Sigma^n$,

$$\frac{1}{3} < \max_{y \in \Sigma^{2^n}} \Pr(M_i^A(x,y) \text{ accepts}) < \frac{2}{3}$$

(ii) or, for all n, there is some $r \in \Sigma^{5n}$ such that for all $x \in \Sigma^n$, $R_{r,i,x}$ holds.

Note that this suffices to prove the theorem. The r will act as the advice for strings of length n. To accept a language in $\mathbf{MA}^{A}_{\mathbf{EXP}}$ with verifier M_i, a polynomial-time machine on input x just needs to ask the oracle A about $\langle r, i, x\rangle$.

Let $S_r = \{\langle r, i, x\rangle \mid 1 \leq i \leq n, x \in \Sigma^n\}$. During the course of the construction we will mark some r as *frozen* meaning that we guarantee that $A(\langle r, i, x\rangle)$ will not change. We will also mark tuples $\langle i, x\rangle$ as *forced* if we guarantee that $M_i^A(x, y)$ accepts for some y.

The construction works in stages. Initially, in stage $s = 0$, $A = \emptyset$, none of the r is frozen and none of the $\langle i, x\rangle$ are forced.

Stage s. Pick the first unfrozen r. Does there exist an unforced $\langle i, x\rangle$, a set $B \subseteq S_r$, and a y such that

$$\Pr(M_i^{A\cup B}(x, y) \text{ accepts}) \;\geq\; \frac{2}{3}\,?$$

- If not, then encode all strings in S_r with this r according to condition (ii) from above. That is, put $\langle r, i, x\rangle$ into A for all forced $\langle i, x\rangle$. Then halt the construction. This r will serve as the advice.

- Otherwise pick the first such $\langle i, x\rangle$, B and y. Let $A = A \cup B$, mark $\langle i, x\rangle$ as forced and freeze r.

  Consider the computation paths of $M_i^A(x, y)$ for this new A. Assume we have T paths each occurring with equal probability. For each path p let Q_p be set of queries made along this path. Freeze all of the r such that
  $$|\{p \mid S_r \cap Q_p \neq \emptyset\}| \;\geq\; \frac{T}{6 \cdot 2^{2n}} \qquad (1)$$
  and go to the next stage.

We need to show that the construction will terminate and that either condition (i) or (ii) is fulfilled.

Since the sets S_r are disjoint and there are at most 2^n queries on each path, by Equation (1) at most $1 + 6 \cdot 2^{3n}$ r's can be frozen in each stage. There are $n2^n$ tuples so at most $(1 + 6 \cdot 2^{3n})n2^n < 2^{5n}$ r's are frozen. Since we have 2^{5n} r's, we will run out of tuples $\langle i, x\rangle$ before we run out of r's.

Now we argue that for each input x and machine i we properly encoded $\langle r, i, x\rangle$ using the r from the construction above. We have two cases.

Case 1: tuple $\langle i, x\rangle$ is *not* forced. Let A' be the value of A right before the last stage of the construction. Since $\langle i, x\rangle$ is not forced, there is no set $B \subseteq S_r$ and y that causes $M_i^{A'\cup B}(x, y)$ to accept. But the final A does equal $A' \cup B$ for some $B \subseteq S_r$. So $M_i^A(x, y)$ still will not accept and remains properly encoded.

Case 2: tuple $\langle i, x\rangle$ is forced. Let A' be the value of A at the end of the stage when $\langle i, x\rangle$ was forced. We have that some y will cause $M_i^{A'}(x, y)$ to accept. Each stage can only add strings to A in one S_r and there are at most $n2^n + 1$ stages. After the stage where $\langle i, x\rangle$ was forced, we will only add strings in S_r that affect the probability of $M_i^A(x, y)$ accepting by at most $\frac{1}{6 \cdot 2^{2n}}$. Thus the total probability of acceptance can go down by at most

$$\frac{1}{6 \cdot 2^{2n}}(n2^n + 1) \;<\; \frac{1}{6}.$$

Hence for the final oracle A we have that y causes $M_i^A(x, y)$ to accept with probability at least 1/2. Thus we have

$$\max_{y\in\Sigma^{2^n}} \Pr(M_i^A(x, y)) \geq \frac{1}{2}$$

fulfilling either condition (i) or (ii).

We still need to make our oracle work for all input lengths. For this, we replace r with $r_1\#r_2\#\ldots\#r_n$, where, for $i < n$, r_i plays the role of r for strings of length i, and r_n is what we vary for this proof. □

We can generalize this result.

Theorem 4.2 *There exists a relativized world where*

$$\mathbf{MIP_{EXP}} = \mathbf{P}/poly \;\cap\; \mathbf{P^{NP}} \;\cap\; \oplus\mathbf{P}$$

Proof Sketch: To get $\mathbf{MIP_{EXP}}$, we just replace the single prover message y in the above proof by the strategy of the provers.

To get $\oplus\mathbf{P}$ we encode $M_i^A(x)$ accept if there are an odd number of r such that $\langle r, i, x\rangle$ is in A.

To get $\mathbf{P^{NP}}$ we pick our r in lexicographical order and then find r in $\mathbf{P}^{\mathbf{NP}^A}$ by using binary search to find the largest tuple $\langle r, i, x\rangle$ in A. We then use r as our advice. □

5 Conclusion

We give the first reasonable nonrelativizing separation showing that $\mathbf{MA_{EXP}}$ is not computable by polynomial-size circuits. We believe our techniques give us a foot in the door that may open to many other exciting separations.

References

[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 IEEE Symposium on Foundations of Computer Science*, pages 14–23. IEEE, New York, 1992.

[BFL91] L. Babai, L. Fortnow, and C. Lund. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1991.

[BFNW93] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[BGKW88] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the 20th ACM Symposium on the Theory of Computing*, pages 113–131. ACM, New York, 1988.

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P = NP question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

[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(2):254–276, 1988.

[For94] L. Fortnow. The role of relativization in complexity theory. *Bulletin of the European Association for Theoretical Computer Science*, 52:229–244, February 1994.

[Kan82] R. Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55:40–56, 1982.

[KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th ACM Symposium on the Theory of Computing*, pages 302–309. ACM, New York, 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.

[PPST83] W. Paul, N. Pippenger, E. Szemerédi, and W. Trotter. On determinism versus nondeterminism and related problems. In *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science*, pages 429–438. IEEE, New York, 1983.

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

[Ver92] N. Vereshchagin. On the power of PP. In *Proceedings of the 7th IEEE Structure in Complexity Theory Conference*, pages 138–143. IEEE, New York, 1992.