Circuit Lower Bounds Collapse Relativized Complexity Classes

Richard Beigel*
University of Illinois at Chicago

Alexis Maciel†
Clarkson University

November 13, 1998

Abstract

Since the publication of Furst, Saxe, and Sipser's seminal paper connecting AC^0 with the polynomial hierarchy [FSS84], it has been well known that circuit lower bounds allow you to construct oracles that separate complexity classes. We will show that similar circuit lower bounds allow you to construct oracles that collapse complexity classes. For example, based on Håstad's parity lower bound, we construct an oracle such that $\mathrm{P} = \mathrm{PH} \subset \oplus\mathrm{P} = \mathrm{EXP}$.

*Dept. of Electrical Engineering and Computer Science, University of Illinois at Chicago, Rm. 1120, SEO Building, M/C 154, 851 S. Morgan St., Chicago, IL 60607-7053. beigel@eecs.uic.edu. www.eecs.uic.edu/~beigel. Part of this work was performed while this author was at the Department of Computer Science, UMIACS, and the Human–Computer Interaction Laboratory, University of Maryland, College Park, on sabbatical from the Yale University Department of Computer Science. Supported in part by the National Science Foundation under grants CCR-8958528 and CCR-9415410, and by NASA under grant NAG 52895.

†Dept. of Mathematics and Computer Science, Clarkson University, Potsdam, NY 13699-5815. alexis@clarkson.edu. www.clarkson.edu/~alexis. Part of this work was done while this author was at the Department of Computer Science and UMIACS, University of Maryland, College Park. Supported in part by the National Science Foundation under grant CCR-9522084.

1 Introduction

Ever since Baker, Gill, and Solovay's [BGS75] seminal paper on relativized complexity class, the meaning and significance of oracle constructions has been hotly debated [Imp88, For94].

By definition, oracles tell us the limits of what can be proved by relativizable techniques. This is useful to researchers, because it tells us when not to give up on relativizable techniques (e.g., [Imm88, Sze88, BRS95]) and when an utterly novel technique will be required (e.g., [LFKN92, Sha92, BFL91, AS92]).

We really want to know what happens in the real world, and oracles don't answer this question. People have studied special classes of oracles such as random [BG81] and sparse [BBS86] but neither do these reflect what happens in the real world in general [Kur83, BBS86].

Despite all this, researchers still rely on oracles to support their intuitions. For example, many would say that Yao's separation of the polynomial hierarchy from PSPACE by oracles [Yao85, Hås86] provides "circumstantial evidence" for the conjecture that the polynomial hierarchy really is different from PSPACE.

Imagine that Yao had instead constructed an oracle that makes $\mathrm{P} = \mathrm{PH}$ and $\mathrm{PSPACE} = \mathrm{EXP}$. Arguably, this would not have supported our intuitions about the polynomial hierarchy. Such an oracle would make the polynomial hierarchy implausibly tiny and PSPACE implausibly large. But then, by the time hierarchy theorem, such an oracle would also make $\mathrm{PH} \neq \mathrm{PSPACE}$. Would it still be "circumstantial evidence" for $\mathrm{PH} \neq \mathrm{PSPACE}$ in the real world?

Recall that Yao's result builds on Furst, Saxe, and Sipser's paradigm [FSS84], which states that in general circuit lower bounds imply oracle separations. We show that in general exponential circuit lower bounds imply in fact a pair of collapses, which in turn imply Furst et al's separation. Our approach is based on combining oracle construction techniques of Beigel, Buhrman, and Fortnow [BBF98] with bounded-query techniques of Amir, Beigel, and Gasarch [ABG90].

2 Preliminaries

We assume that the reader is familiar with the basic notions of complexity theory. (See [Pap94], for example.) In this section, we recall some definitions that will be used in this article.

For any number r, a language L is in $\mathrm{MOD}_r\mathrm{P}$ if there is a polynomial-time nondeterministic Turing machine N such that for every input string x, $x \in L$ if and only if the number of accepting computations of N on x is not congruent to 0 modulo r. $\mathrm{MOD}_2\mathrm{P}$ is usually denoted $\oplus\mathrm{P}$.

A language L is in PP if there is a polynomial-time nondeterministic Turing machine N such that for every input string x, $x \in L$ if and only if more than half the computations of N on x are accepting.

A language is in EXP if it is recognized by a deterministic Turing machine in time $2^{n^{O(1)}}$.

3 The Main Oracle Construction

This section is devoted to the proof of the following theorem.

Theorem 1 *There is an oracle A such that* $\mathrm{P}^A = \mathrm{NP}^A$ *and* $\oplus\mathrm{P}^A = \mathrm{EXP}^A$.

Let M be a nondeterministic linear-time oracle Turing machine such that for all A, M^A accepts an NP^A complete language L^A. Let N be a deterministic oracle Turing machine that runs in time 2^n and such that for all A, N^A accepts an EXP^A complete language K^A.

Let c be a constant whose value will be determined later. Let x_0 be the first string (with respect to lexicographic ordering) whose length is at least c. We will construct A such that for all $x \geq x_0$,

$$x \in L^A \quad \Leftrightarrow \quad \langle 0, x, 1^{|x|^2} \rangle \in A$$

and

$$x \in K^A \quad \Leftrightarrow \quad |\{v : |v| = |x|^2 \text{ and } \langle 1, x, v \rangle \in A\}| \text{ is odd.}$$

The first condition will imply that $\mathrm{P}^A = \mathrm{NP}^A$ by allowing a deterministic Turing machine to recognize the NP^A complete language L^A in polynomial time. Similarly, the second condition will imply that $\oplus\mathrm{P}^A = \mathrm{EXP}^A$.

Strings of the form $\langle 0, x, 1^{|x|^2} \rangle$ are said to be of type 0. Strings of the form $\langle 1, x, v \rangle$ are said to be of type 1. By abuse of language, these strings are also called queries.

To each query $\langle 1, x, v \rangle$ of type 1, we associate a Boolean variable $A_{\langle x,v \rangle}$ whose value is the answer to the query, i.e., $A_{\langle x,v \rangle} = 1$ if $\langle 1, x, v \rangle \in A$ and $A_{\langle x,v \rangle} = 0$, otherwise. The set of queries $\{\langle 1, x, v \rangle : |v| = |x|^2\}$ is called the block of x. Denote by A_x the string of variables corresponding to the block of x. The condition on queries of type 1 can now be written as

$$x \in K^A \quad \Leftrightarrow \quad \mathrm{MOD}_2(A_x) = 1.$$

Notice that the condition on queries of type 0 implies that these queries are determined by shorter queries. Therefore, an assignment to the queries of type 0 that satisfies the condition on queries of type 0 will automatically follow from an assignment to the queries of type 1.

We therefore concentrate on the construction of an assignment to the queries of type 1. This will be done by constructing an infinite sequence $(A^{(x)})_{x \geq x_0}$ of assignments such that for every $x \geq x_0$, for every $y \in [x_0, x]$,

$$y \in K^{A^{(x)}} \quad \Leftrightarrow \quad \mathrm{MOD}_2(A_y^{(x)}) = 1. \tag{1}$$

An assignment A such that for every $x \geq x_0$,

$$x \in K^A \quad \Leftrightarrow \quad \mathrm{MOD}_2(A_x) = 1$$

can then be obtained by using a standard argument.

Consider an arbitrary $x \geq x_0$. This string x will be fixed for the remainder of the proof. Our goal is now to construct an assignment $A^{(x)}$ that satisfies the above condition. For convenience, we will drop the superscript and simply write A.

Consider the computation of M^A on an arbitrary input string z. This computation can be simulated by first guessing the answers to all the oracle queries and verifying them at the end. This implies that the computation of M^A on z can be represented by a depth-two circuit B'_z with an OR gate of fan-in $2^{|z|}$ at the output, AND gates of fan-in $|z|$ on level one, and whose inputs are either answers to queries of type 1 or answers to queries $\langle 0, w, 1^{|w|^2} \rangle$ of type 0 with $|w| \leq \sqrt{|z|}$. By the condition on queries of type 0, replace each such query $\langle 0, w, 1^{|w|^2} \rangle$ by the circuit B'_w that represents the computation of M_A on w. Repeat this recursively. The result is a circuit B_z whose inputs are only queries of type 1, whose size is at most $2^{2|z|}$ and whose depth is at most $\log \log |z|$. And, of course, B_z represents the computation of M^A on z.

Now consider the computation of N^A on an arbitrary input string y. Again, this computation can be simulated by first guessing the answers to all the oracle queries and verifying them at the end. This implies that the computation of N^A on y can be represented by a depth-two circuit C'_y with an OR gate of fan-in $2^{2^{|y|}}$ at the output, AND gates of fan-in $2^{|y|}$ on level one, and whose inputs are either answers to queries of type 1 or answers to queries $\langle 0, w, 1^{|w|^2} \rangle$ of type 0 with $|w| \leq 2^{|y|}$. Replace each query $\langle 0, w, 1^{|w|^2} \rangle$ by the corresponding circuit B_w. The result is a circuit C_y whose inputs are only queries of type 1, whose size is at most $2^{2^{|y|+2}}$ and whose depth is at most $\log |y|$.

Since C_y represents the computation of N^A on input y, (1) can be rewritten as follows: for every $y \in [x_0, x]$,

$$C_y(A_{x_0}, \ldots, A_x, \ldots) = \mathrm{MOD}_2(A_y). \tag{2}$$

For every $z \geq x$, set $A_z = 0$. Now, by contradiction, suppose that an assignment cannot be found to satisfy (2). In other words, suppose that for every sequence of binary strings $\alpha_{x_0}, \ldots, \alpha_x$, there is $y \in [x_0, x]$ such that

$C_y(\alpha_{x_0},\ldots,\alpha_x) \neq \mathrm{MOD}_2(\alpha_y)$. The following lemma, adapted from a result of Amir, Beigel and Gasarch [ABG90, Theorem 17], states that this implies the existence of a not too large circuit computing the MOD_2 function on $|A_z|$ variables, for some $z \in [x_0, x]$.

Lemma 2 *If for every $\alpha_{x_0},\ldots,\alpha_x$, there is $y \in [x_0, x]$ such that*

$$C_y(\alpha_{x_0},\ldots,\alpha_x) \neq F(\alpha_y),$$

then there is a circuit of size $2^{2^{5\sqrt{\log N}}}$ and depth $\log\log N$ that computes the F function on N variables, for some $N \geq c$.

Proof Let σ_x denote a string of Boolean variables of length equal to the size of the block of x. We will first aim for a circuit that computes $\overline{F(\sigma_x)}$. For this purpose, we will try to construct sets of *advice* $H_{x_0},\ldots,H_{x-1}$ with the following property: for every α_x, there are $\beta_{x_0} \in H_{x_0},\ldots,\beta_{x-1} \in H_{x-1}$ such that for every $w \in [x_0, x-1]$,

$$C_w(\beta_{x_0},\ldots,\beta_{z-1},\alpha_x) = F(\beta_w).$$

By the hypothesis in the statement of the lemma, this will imply that for every α_x, there are $\beta_{x_0} \in H_{x_0},\ldots,\beta_{x-1} \in H_{x-1}$ such that

$$C_x(\beta_{x_0},\ldots,\beta_{z-1},\alpha_x) \neq F(\alpha_x).$$

This will be the basis for the construction of a circuit computing $\overline{F(\sigma_x)}$.

So we proceed with the construction of the advice. This will be done in stages, one for each of $H_{x_0},\ldots,H_{x-1}$.

> **Begin Stage z.** Notice that for every $w \leq z$, C_w makes queries of length at most $2^{|w|}$. So the value of every $C_w(\sigma_{x_0},\ldots,\sigma_x)$, for $w \leq z$, is determined by $\sigma_{x_0},\ldots,\sigma_{2^z}$, where 2^z denotes the last string of length $2^{|z|}$.
>
> Say that β_z is *advice* for $\gamma \in \{0,1\}^{|\sigma_{z+1}|+\cdots+|\sigma_{2^z}|}$ if there are $\beta_{x_0} \in H_{x_0},\ldots,\beta_{z-1} \in H_{z-1}$ such that for every $w \in [x_0, z]$,
>
> $$C_w(\beta_{x_0},\ldots,\beta_{z-1},\beta_z,\gamma) = F(\beta_w).$$
>
> Let advisees(β_z) be the set of $\gamma \in \{0,1\}^{|\sigma_{z+1}|+\cdots+|\sigma_{2^z}|}$ for which β_z is advice.
>
> Let $T_z = \{0,1\}^{|\sigma_{z+1}|+\cdots+|\sigma_{2^z}|}$. Let $H_z = \emptyset$.
>
> While there is β_z such that $|\text{advisees}(\beta_z) \cap T_z| \geq \frac{1}{4}|T_z|$:
>
> 1. Choose such an β_z.

2. Let $H_z = H_z \cup \{\beta_z\}$.
3. Let $T_z = T_z - \text{advisees}(\beta_z)$.

If $T_z \neq \emptyset$, halt.

If $T_z = \emptyset$ and $z = x - 1$, halt.

If $T_z = \emptyset$ and $z < x - 1$, proceed to Stage $z + 1$. **End Stage z.**

Before we continue, let us bound the size of H_z. We have that $|H_z| \leq 4 \log |\{0,1\}^{|\sigma_{z+1}|+\cdots+|\sigma_{2^z}|}| \leq 2^{2^{2|z|+1}}$. In particular, this implies that $|H_{x_0}| \cdots |H_{z-1}| \leq 2^{2^{4|z|}}$.

There are now two cases to consider. First, suppose that the construction of the advice halted with $T_z = \emptyset$ and $z = x-1$. This means that we successfully found an advice for every string of length $|\sigma_x|$. In other words, for every α_x, there are $\beta_{x_0} \in H_{x_0}, \ldots, \beta_{x-1} \in H_{x-1}$ such that for every $w \in [x_0, x-1]$,

$$C_w(\beta_{x_0}, \ldots, \beta_{x-1}, \alpha_x) = F(\beta_w),$$

which implies that

$$C_x(\beta_{x_0}, \ldots, \beta_{x-1}, \alpha_x) \neq F(\alpha_x). \tag{3}$$

The function $\overline{F(\sigma_x)}$ can now be computed with a depth-two circuit as follows. For each sequence $\beta_{x_0} \in H_{x_0}, \ldots, \beta_{x-1} \in H_{x-1}$, construct an AND gate that will test whether for every $w \in [x_0, x-1]$,

$$C_w(\beta_{x_0}, \ldots, \beta_{x-1}, \sigma_x) = F(\beta_w).$$

These AND gates have fan-in $2^{|x|+1}$ and their inputs are instances of the circuits $C_{x_0}, \ldots, C_x$. Add to each of these AND gates the input $\overline{C_x(\beta_{x_0}, \ldots, \beta_{x-1}, \sigma_x)}$. For every α_x, by (3), each of these AND gates will output either 0 or $\overline{F(\alpha_x)}$. Feed all of these AND gates into an OR gate of fan-in $|H_{x_0}| \cdots |H_{x-1}| \leq 2^{2^{4|x|}}$. This OR gate clearly computes $\overline{F(\sigma_x)}$. Therefore, $F(\sigma_x)$ can be computed by a circuit of size $2^{2^{4|x|+1}}$ and depth $\log |x| + 2$. Since the length of the input is $N_x = |\sigma_x| = 2^{|x|^2}$, we have a circuit of size $2^{2^{4\sqrt{\log N_x}+1}} \leq 2^{2^{5\sqrt{\log N_x}}}$ and depth $\frac{1}{2} \log \log N_x + 2 \leq \log \log N_x$.

For the second case, suppose that the construction of the advice halted with $T_z \neq \emptyset$. Instead of aiming for a circuit that computes $\overline{F(\sigma_x)}$, we will now aim for circuit that computes $\overline{F(\sigma_z)}$. Every β_z is advice for less than $\frac{1}{4}$ of the γ in the resulting T_z. This means that for every β_z, at least $\frac{3}{4}$ of the elements γ of T_z satisfy the following: for every $\beta_{x_0} \in H_{x_0}, \ldots, \beta_{z-1} \in H_{z-1}$ there is $w \in [x_0, z]$ such that

$$C_w(\beta_{x_0}, \ldots, \beta_{z-1}, \beta_z, \gamma) \neq F(\beta_w).$$

On the other hand, by the previous stage of the construction (Stage $z-1$), which terminated with $T_{z-1} = \emptyset$, for every β_z and every $\gamma \in \{0,1\}^{|\sigma_{z+1}|+\cdots+|\sigma_{2^z}|}$, there are $\beta_{x_0} \in H_{x_0}, \ldots, \beta_{z-1} \in H_{z-1}$ such that for every $w \in [x_0, z-1]$,

$$C_w(\beta_{x_0}, \ldots, \beta_{z-1}, \beta_z, \gamma) = F(\beta_w).$$

Therefore, for every β_z and for at least $\frac{3}{4}$ of the elements γ of T_z, there are $\beta_{x_0} \in H_{x_0}, \ldots, \beta_{z-1} \in H_{z-1}$ such that

$$C_z(\beta_{x_0}, \ldots, \beta_{z-1}, \beta_z, \gamma) \neq F(\beta_z).$$

So given β_z, choose γ uniformly at random in T_z. With probability at least $\frac{3}{4}$, there are $\beta_{x_0} \in H_{x_0}, \ldots, \beta_{z-1} \in H_{z-1}$ such that

$$C_z(\beta_{x_0}, \ldots, \beta_{z-1}, \beta_z, \gamma) \neq F(\beta_z).$$

This implies that $\overline{F(\sigma_x)}$ can be computed with probability of error no greater than $\frac{1}{4}$ by a probabilistic depth-two circuit with an OR gate of fan-in $|H_{x_0}| \cdots |H_{z-1}| \leq 2^{2^{4|z|}}$ at the output, AND gates of fan-in $2^{|z|+1}$ on level one and whose inputs are instances of the circuits $C_{x_0}, \ldots, C_z$. Therefore, $F(\sigma_z)$ can be computed by a probabilistic circuit of size $2^{2^{4|z|+1}}$ and depth $\log|z|+2$. Since the length of the input is now $N_z = |\sigma_z| = 2^{|z|^2}$, we have a probabilistic circuit of size $2^{2^{4\sqrt{\log N_z}+1}}$ and depth $\frac{1}{2}\log\log N_z + 2$. By a technique of [ABO84], this probabilistic circuit can be transformed into a deterministic circuit of size $2^{2^{5\sqrt{\log N_z}}}$ and depth $\log\log N_z$.

Therefore, in both cases, we obtain a circuit of size $2^{2^{5\sqrt{\log N}}}$ and depth $\log\log N$ that computes the F function on N variables, for some $N \geq c$. $\square$

Returning to the proof of Theorem 1, we now choose the value of c to be at least the value of the constant in the following easy corollary of Håstad's AC^0 lower bound [Hås86, Theorem 1]. The value of c must also be larger than some other small constant as required by some of the inequalities used in the calculations in the proof of Lemma 2.

Lemma 3 *There is a constant N_0 such that if $N \geq N_0$, then there are no circuits of size $2^{2^{5\sqrt{\log N}}}$ and depth $\log\log N$ that compute the MOD_2 function on N variables.*

By combining the two lemmas, we get that there are $\alpha_{x_0}, \ldots, \alpha_x$ such that for every $y \in [x_0, x]$,

$$C_y(\alpha_{x_0}, \ldots, \alpha_x) = \mathrm{MOD}_2(\alpha_y).$$

This completes the proof of Theorem 1.

4 Generalizations

The following generalization of Lemma 3 is also an easy corollary of Håstad's AC^0 lower bound [Hås86, Theorem 1].

Lemma 4 *For every number $r \geq 2$, there is a constant N_0 such that if $N \geq N_0$, then there are no circuits of size $2^{2^{5\sqrt{\log N}}}$ and depth $\log\log N$ that compute either the MOD_r function or the majority function on N variables.*

Theorem 5 *For every set of numbers $r_1, \ldots, r_k \geq 2$, there is an oracle A such that $\mathrm{P}^A = \mathrm{NP}^A$ and $\mathrm{MOD}_{r_1}\mathrm{P}^A = \cdots = \mathrm{MOD}_{r_k}\mathrm{P}^A = \mathrm{PP}^A = \mathrm{EXP}^A$.*

Proof The proof is similar to the proof of Theorem 1. We will only indicate the main differences.

The idea is to construct an oracle A such that for all $x \geq x_0$,

$$x \in L^A \quad \Leftrightarrow \quad \langle 0, x, 1^{|x|^2} \rangle \in A,$$

$$x \in K^A \quad \Leftrightarrow \quad |\{v : |v| = |x|^2 \text{ and } \langle 1, x, v \rangle \in A\}|/2^{|x|^2} > 1/2$$

and for every $i \in [1, k]$,

$$x \in K^A \quad \Leftrightarrow \quad |\{v : |v| = |x|^2 \text{ and } \langle 1^{r_i}, x, v \rangle \in A\}| \not\equiv 0 \pmod{r_i}.$$

Each input string x will have $k+1$ blocks $A_{x,1}, A_{x,r_1}, \ldots, A_{x,r_k}$ so the condition on strings of type 1 can be written as follows:

$$x \in K^A \quad \Leftrightarrow \quad \text{majority}(A_{x,1}) = 1$$

and for every $i \in [1, k]$,

$$x \in K^A \quad \Leftrightarrow \quad \mathrm{MOD}_{r_i}(A_{x,r_i}) = 1.$$

As before, the goal is to construct, for an arbitrary $x \geq x_0$, an oracle A that works for all input strings $y \in [x_0, x]$.

Since C_y still represents the computation of N^A on input y, the above condition can be rewritten as follows: for every $y \in [x_0, x]$ and every $r \in \{1, r_1, \ldots, r_k\}$,

$$C_y(A_{x_0,1}, A_{x_0,r_1}, \ldots, A_{x_0,r_k}, \ldots, A_{x,1}, A_{x,r_1}, \ldots, A_{x,r_k}, \ldots) = F_r(A_{y,r})$$

where $F_r =$ majority if $r = 1$, and $F_r = \mathrm{MOD}_r$ if $r \geq 2$.

Lemma 2 and its proof can be easily adapted to this new setting. The proof initially aims at the construction of a set of advice $H_{z,r}$ for each pair $(z, r) \in [x_0, x-1] \times \{1, r_1, \ldots, r_k\}$. □

The oracle construction can be generalized further by using the following lemma which can be easily obtained from the proof Smolensky's lower bound for $\mathrm{ACC}^0[p]$ circuits [Smo87].

Lemma 6 *For every power q of a prime p there is a constant N_p such that for every number r divisible by some other prime, if $N \geq N_p$, then there are no circuits of size $2^{2^{5\sqrt{\log N}}}$ and depth $\log\log N$ composed of* AND, OR *and* MOD_q *gates that compute either the* MOD_r *function or the majority function on N variables.*

Theorem 7 *For every power q of a prime p and for every set of numbers $r_1, \ldots, r_k$ divisible by some other prime, there is an oracle A such that* $\mathrm{P}^A = \mathrm{NP}^A = \mathrm{MOD}_q\mathrm{P}^A$ *and* $\mathrm{MOD}_{r_1}\mathrm{P}^A = \cdots = \mathrm{MOD}_{r_k}\mathrm{P}^A = \mathrm{PP}^A = \mathrm{EXP}^A$.

Proof To the proof of the previous theorem, we add an additional condition on the oracle A:

$$x \in L^A \quad \Leftrightarrow \quad \langle 0^q, x, 1^{|x|^2} \rangle \in A.$$

The rest of the proof is as before except that the circuit B_z, and hence the circuit C_y, may now contain MOD_q gates. □

5 Conclusions and Related Work

Since the same underlying techniques that yield "expected" separations also yield unexpected collapses, we believe that our work should shake up anyone's faith in oracles for providing circumstantial evidence about the real world. We have also exposed a new and deep connection between upper bounds and lower bounds. Similar connections have been noted in other contexts. See, for example, [PSZ97] and [BI87].

References

[ABG90] Amihood Amir, Richard Beigel, and William I. Gasarch, *Some connections between bounded query classes and nonuniform complexity*, Proc. Fifth Ann. Structure in Complexity Theory Conf., 1990, pp. 232–243.

[ABO84] Ajtai and Ben-Or, *A theorem on probabilistic constant depth computations*, Proceedings of the 16th ACM Symposium on Theory of Computing, 1984.

[AS92] Sanjeev Arora and Shmuel Safra, *Probabilistic checking of proofs*, Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science, 1992, pp. 2–13.

[BBF98] R. Beigel, H. Buhrman, and L. Fortnow, *NP might not be as easy as detecting unique solution,*, Proceedings of the 30th ACM Symposium on Theory of Computing, 1998.

[BBS86] José L. Balcázar, Ronald V. Book, and Uwe Schöning, *The polynomial-time hierarchy and sparse oracles*, J. ACM **33** (1986), no. 3, 603–617.

[BFL91] Laszlo Babai, Lance Fortnow, and Carsten Lund, *Nondeterministic exponential time has two-prover interactive protocols*, Comput. Complexity **1** (1991), 3–40.

[BG81] Charles H. Bennett and John Gill, *Relative to a random oracle A,* $\mathrm{P}^A \neq \mathrm{NP}^A \neq \text{co-NP}^A$ *with probability 1*, SIAM J. Comput. **10** (1981), 96–112.

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

[BI87] Blum and Impagliazzo, *Generic oracles and oracle classes*, Proceedings of the 28th IEEE Symposium on Foundations of Computer Science, 1987.

[BRS95] R. Beigel, N. Reingold, and D. Spielman, *PP is closed under intersection*, J. Comput. System Sci. **50** (1995), no. 2, 191–202.

[For94] Lance Fortnow, *The role of relativization in complexity theory*, Bull. Europ. Assoc. Theor. Computer Sci. **52** (1994).

[FSS84] M. Furst, J.B. Saxe, and M. Sipser, *Parity, circuits, and the polynomial-time hierarchy*, Math. Systems Theory **17** (1984), 13–27.

[Hås86] J. Håstad, *Computational limitations of small depth circuits*, MIT Press, Cambridge, MA, U.S.A., 1986.

[Imm88] Neil Immerman, *Nondeterministic space is closed under complementation*, Proc. Third Ann. Structure in Complexity Theory Conf., 1988, pp. 260–277.

[Imp88] Russell Impagliazzo, *On the power of proofs which relativize*, Manuscript, 1988.

[Kur83] Stuart A. Kurtz, *On the random oracle hypothesis*, Inform. and Control **57** (1983), no. 1, 40–47.

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan, *Algebraic methods for interactive proof systems*, J. ACM **39** (1992), no. 4, 859–868.

[Pap94] Christos H. Papadimitriou, *Computational complexity*, Addison-Wesley, 1994.

[PSZ97] Paturi, Saks, and Zane, *Exponential lower bounds for depth 3 boolean circuits*, Proceedings of the 29th ACM Symposium on Theory of Computing, 1997.

[Sha92] Adi Shamir, *IP = PSPACE*, J. ACM **39** (1992), no. 4, 869–877.

[Smo87] R. Smolensky, *Algebraic methods in the theory of lower bounds for boolean circuit complexity*, Proceedings of the 19th ACM Symposium on Theory of Computing, 1987, pp. 77–82.

[Sze88] Szelepcsenyi, *The method of forced enumeration for nondeterministic automata*, Acta Infor. **26** (1988).

[Yao85] A.C.-C. Yao, *Separating the polynomial-time hierarchy by oracles*, Proceedings of the 26th IEEE Symposium on Foundations of Computer Science, 1985, pp. 1–10.