# The Random Oracle Hypothesis is False

Richard Chang[1,2] Benny Chor[3,4] Oded Goldreich[3,5]

Juris Hartmanis[1] Johan Håstad[6] Desh Ranjan[1,7]

Pankaj Rohatgi[1]

July 7, 1992

### Abstract

The Random Oracle Hypothesis, attributed to Bennett and Gill, essentially states that the relationships between complexity classes which hold for almost all relativized worlds must also hold in the unrelativized case. Although this paper is not the first to provide a counterexample to the Random Oracle Hypothesis, it does provide a most compelling counterexample by showing that for almost all oracles $A$, $\mathrm{IP}^A \neq \mathrm{PSPACE}^A$. If the Random Oracle Hypothesis were true, it would contradict Shamir's result that IP = PSPACE. In fact, it is shown that for almost all oracles $A$, $\text{co-NP}^A \not\subseteq \mathrm{IP}^A$. These results extend to the multi-prover proof systems of Ben-Or, Goldwasser, Kilian and Wigderson. In addition, this paper shows that the Random Oracle Hypothesis is sensitive to small changes in the definition. A class IPP, similar to IP, is defined. Surprisingly, the IPP = PSPACE result holds for all oracle worlds.

[1] Department of Computer Science, Cornell University, Ithaca, NY 14853, U.S.A. Supported in part by NSF Research Grant CCR-88-23053.

[2] Supported in part by an IBM Graduate Fellowship.

[3] Department of Computer Science, Technion, Haifa 32000, Israel.

[4] Supported by grant No. 88-00282 from the United States – Israel Binational Science Foundation (BSF), Jerusalem, Israel.

[5] Supported by grant No. 88-00301 from the United States – Israel Binational Science Foundation (BSF), Jerusalem, Israel.

[6] Department of Computer Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden.

[7] Supported in part by a Mathematical Sciences Institute (MSI) Fellowship.

# 1 Introduction

Computational complexity theory studies the quantitative laws which govern computing. It seeks a comprehensive classification of problems by their intrinsic difficulty and an understanding of what makes these problems hard to compute. The key concept in classifying the computational complexity of problems is the complexity class which consists of all the problems solvable on a given computational model and within a given resource bound.

Structural complexity theory is primarily concerned with the relations among various complexity classes and the internal structure of these classes. Figure 1 shows some major complexity classes. Although much is known about the structure of these classes, there have not been any results which separate any of the classes between P and PSPACE. We believe that all these classes are different and regard the problem of proving the exact relationships among these classes as the Grand Challenge of complexity theory.

The awareness of the importance of P, NP, PSPACE, etc, has led to a broad investigation of these classes and to the use of relativization. Almost all of the major results in recursive function theory also hold in relativized worlds. Quite the contrary happens in complexity theory. It was shown in 1975 [3] that there exist oracles $A$ and $B$ such that

$$\mathrm{P}^A = \mathrm{NP}^A \text{ and } \mathrm{P}^{\mathrm{B}} \neq \mathrm{NP}^{\mathrm{B}}.$$

This was followed by an extensive investigation of the structure of complexity classes under relativization. An impressive set of techniques was developed for oracle constructions and some very subtle and interesting relativization results were obtained. For example, for a long time it was not known if the Polynomial-time Hierarchy (PH) can be separated by oracles from PSPACE. In 1985, Yao [35] finally resolved this problem by constructing an oracle $A$, such that

$$\mathrm{PH}^A \neq \mathrm{PSPACE}^A.$$

Håstad [23] simplified this proof and constructed an oracle $B$, such that

$$\forall k,\ \mathrm{PH}^B \neq \Sigma_k^{\mathrm{P},B}.$$

These methods were refined by Ko [28] to show that for every $k \geq 0$ there is an oracle which collapses PH to exactly the $k^{th}$ level and keeps the first $k-1$ levels of PH distinct. That is, for all $k$, there exists an $A$ such that

$$\Sigma_0^{\mathrm{P},A} \neq \Sigma_1^{\mathrm{P},A} \neq \cdots \neq \Sigma_k^{\mathrm{P},A} \text{ and } \Sigma_k^{\mathrm{P},A} = \Sigma_{k+i}^{\mathrm{P},A},\ i \geq 0.$$

Another aspect of relativized computations was studied by Bennett and Gill who wanted to measure the set of oracles which separate certain complexity classes. They showed that $\mathrm{P}^A \neq \mathrm{NP}^A$ for almost all oracles. In addition, they showed that for almost all oracles $A$ the following relationships hold [5]:

$$\mathrm{P}^A \neq \mathrm{NP}^A \neq \text{co-NP}^A$$
$$\mathrm{SPACE}^A[\log n] \neq \mathrm{P}^A$$
$$\mathrm{PSPACE}^A \neq \mathrm{EXP}^A$$
$$\mathrm{P}^A = \mathrm{RP}^A = \mathrm{BPP}^A.$$

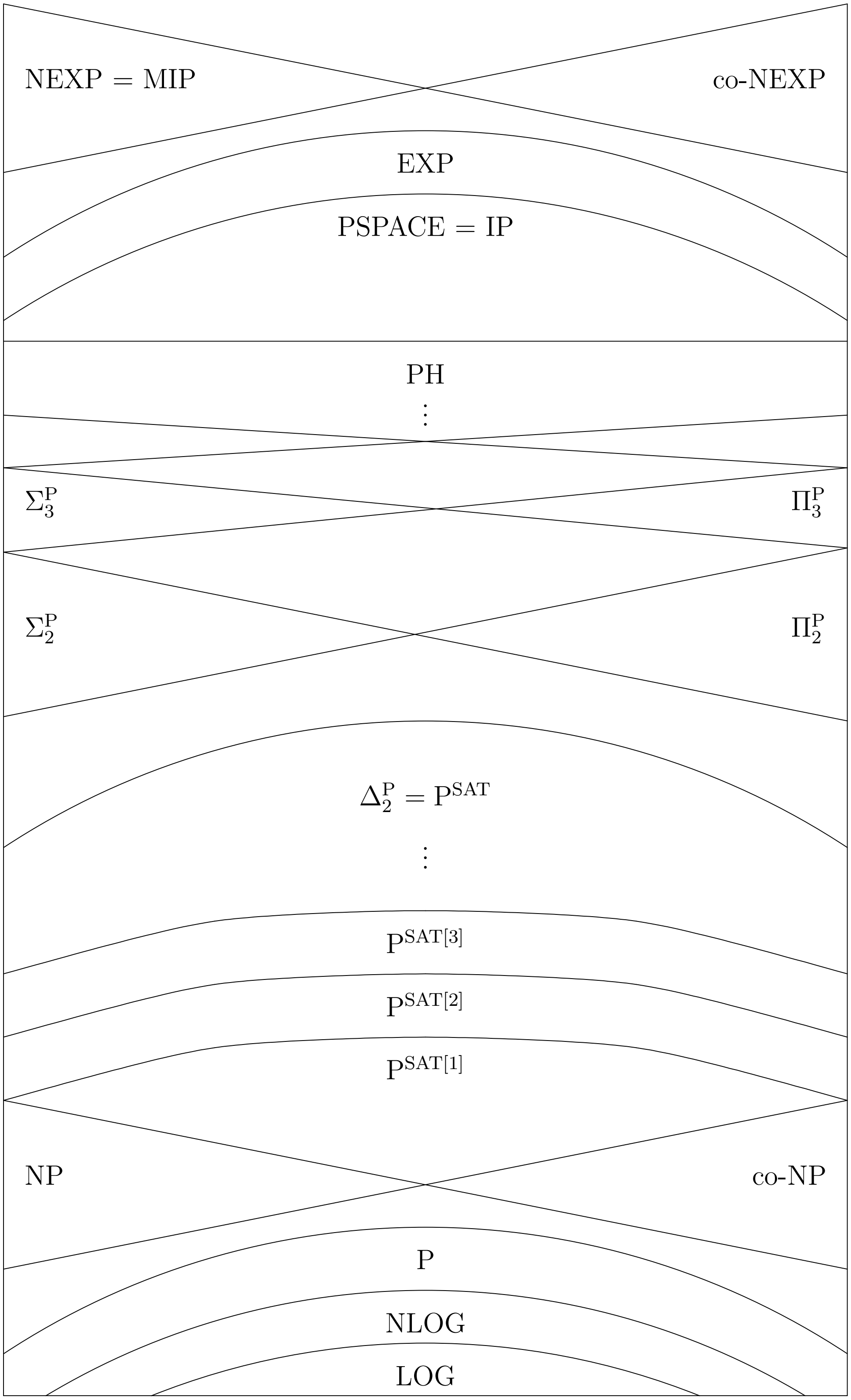

Figure 1: Some standard complexity classes.

Many other interesting random oracle results followed. For almost all oracles $A$ [8, 9, 30]:

- $\mathrm{PH}^A \subsetneq \mathrm{PSPACE}^A$.
- The Boolean Hierarchy relative to $A$, $\mathrm{BH}^A$, is infinite[5].
- The Berman-Hartmanis Conjecture fails relative to $A$.

The last result asserts that there exist non-isomorphic many-one complete sets for $\mathrm{NP}^A$ for random oracle $A$. It was conjectured that all NP many-one complete sets are polynomial-time isomorphic [6].

Surveying the rich set of relativization results, we can make several observations. First, almost all questions about the relationship between the major complexity classes have contradictory relativizations. That is, there exist oracles which separate the classes and oracles which collapse them. Furthermore, many of our proof techniques relativize and cannot resolve problems with contradictory relativizations. Finally, we have unsuccessfully struggled for over twenty years to resolve whether $\mathrm{P} =?\, \mathrm{NP} =?\, \mathrm{PSPACE}$.

These observations seemed to support the conviction that problems with contradictory relativizations are extremely difficult and may not be solvable by current techniques. This opinion was succinctly expressed by John Hopcroft [24]:

> This perplexing state of affairs is obviously unsatisfactory as it stands. No problem that has been relativized in two conflicting ways has yet been solved, and this fact is generally taken as evidence that the solutions of such problems are beyond the current state of mathematics.

How should complexity theorists remedy "this perplexing state of affairs"? In one approach, we assume as a working hypothesis that PH has infinitely many levels. Thus, any assumption which would imply that PH is finite is deemed incorrect. For example, Karp, Lipton and Sipser [27] showed that if $\mathrm{NP} \subseteq \mathrm{P/poly}$, then PH collapses to $\Sigma_2^{\mathrm{P}}$. So, we believe that SAT does not have polynomial sized circuits. Similarly, we believe that the Turing-complete and many-one complete sets for NP are not sparse, because Mahaney [32] showed that these conditions would collapse PH. One can even show that for any $k \geq 0$, $\mathrm{P}^{\mathrm{SAT}[k]} = \mathrm{P}^{\mathrm{SAT}[k+1]}$ implies that PH is finite [26]. Hence, we believe that $\mathrm{P}^{\mathrm{SAT}[k]} \neq \mathrm{P}^{\mathrm{SAT}[k+1]}$ for all $k \geq 0$. Thus, if the Polynomial Hierarchy is indeed infinite, we can describe many aspects of the computational complexity of NP.

A second approach used random oracles. Since most of the random oracle relativization results agreed with what complexity theorists believed to be true in the base case and since random oracles have no particular structure of their own, it seemed that the behavior of complexity classes relative to a random oracle should be the same as the base case behavior. This led Bennett and Gill to postulate the Random Oracle Hypothesis [5] which essentially

[5] The Boolean Hierarchy (BH) is the Hausdorff closure of NP— i.e., the smallest class containing NP that is also closed under union, intersection and complementation [10, 11]. BH is contained in the $\Delta_2^{\mathrm{P}}$ level of PH and it is known that if PH has infinitely many levels then so does BH [26]. This random oracle result is of particular interest when we contrast it with the case of PH because it is not known whether PH is infinite relative to a random oracle.

states that structural relationships which hold in almost all oracle worlds also hold in the unrelativized case — i.e., the real world.

In the following, we first discuss a set of results about interactive proofs which provide dramatic counterexamples to the belief that problems with contradictory relativizations cannot be resolved with known techniques. Hence, contradictory relativizations should no longer be viewed as strong evidence that a problem is beyond our grasp. We continue by presenting our main results, which combined with what is known on interactive proofs, yield a striking new counterexample against the Random Oracle Hypothesis. There have previously been several counterexamples in the literature and in unpublished reports [15, 21, 25, 29]. Some of these counterexamples use double relativization and classes which are not closed under polynomial time reductions. While the results in this paper are not the first, the authors believe that they are the most natural and compelling. We conclude that random oracle relativizations should not discourage attempts to prove the opposite in the real world.

This paper reports results obtained independently by two sets of researchers. Preliminary versions of these works can be found in [12, 22].

## 2 A Review of IP

The class IP is the set of languages that have interactive proofs. IP was first defined as way to generalize NP [1, 19]. NP can be characterized as being precisely those languages for which one can present a polynomially long proof to certify that the input string is in the language. Moreover, the proof can be checked in polynomial time. It is this idea of presenting and checking the proof that the definition of IP generalizes.

Is there a way of giving convincing evidence that the input string is in a language without showing the whole proof to a verifier? Clearly, if we do not give a complete proof to a verifier which does not have the power or the time to generate and check a proof, then we cannot expect the verifier to be completely convinced. This leads us to a very fascinating problem: *how can the verifier be convinced with high probability that there is a proof? and how rapidly can this be done?*

This problem has been formulated and extensively studied in terms of *interactive proofs* [1, 18, 19]. Informally, an interactive proof consists of a *Prover* and a *Verifier*. The Prover is an all powerful Turing Machine (TM) and the Verifier is a TM which operates in time polynomial in the length of the input. In addition, the Verifier has a random source (e.g., a fair coin) not visible to the Prover. In the beginning of the interactive proof the Prover and the Verifier receive the same input string. Then, the Prover tries to convince the Verifier, through a series of queries and answers, that the input string belongs to a given language. The Prover succeeds if the Verifier accepts with probability greater than 2/3. The probability is computed over all possible coin tosses made by the Verifier. However, the Verifier must guard against imposters masquerading as the real Prover. The Verifier must not be convinced to accept a string not in the language with probability greater than 1/3 — even if the Prover lies.

**Definition** IP: Let $V$ be a probabilistic polynomial time TM and let $P$ be an arbitrary TM. $P$ and $V$ share the same input tape and communicate via a communication tape. $P$ and $V$ form an interactive proof for a language $L$ if

1. $x \in L \Longrightarrow$ Prob[ $P$-$V$ accepts $x$ ] $> \frac{2}{3}$.
2. $x \notin L \Longrightarrow \forall P^*$, Prob[ $P^*$-$V$ accepts $x$ ] $< \frac{1}{3}$.

A language $L$ is in IP if there exist $P$ and $V$ which form an interactive proof for $L$.

Clearly, IP contains all NP languages, because in polynomial time the Prover can give the Verifier the entire proof. In such a proof, the Verifier cannot be fooled and never accepts a string not in the language. To illustrate how randomness can generalize the concept of a proof, we look at an interactive proof for a language not known to be in NP. Consider GNI, the set of pairs of graphs that are not isomorphic. GNI is known to be in co-NP and it is believed not to be in NP. However, GNI does have an interactive proof [17]. The Verifier determines if two graphs $G_1$ and $G_2$ are non-isomorphic, using the following interactive proof:

1. The Verifier randomly selects $G_1$ or $G_2$ and a random permutation of the selected graph. This process is independently repeated $n$ times, where $n$ is the number of vertices in $G_2$. If the graphs do not have the same number of vertices, they are clearly not isomorphic. This sequence of $n$ randomly chosen, randomly permuted graphs is sent to the Prover. Recall that the Prover has not seen the Verifier's random bits. (With a more elaborate interactive proof, this assumption is not necessary [20].)

2. The Verifier asks the Prover to determine, for each graph in the sequence, which graph, $G_1$ or $G_2$, was the one selected. If the Prover answers correctly, then the Verifier accepts.

Suppose the two original graphs are not isomorphic. Then, only one of the original graphs is isomorphic to the permuted graph. The Prover simply answers by picking that graph. If the graphs are isomorphic, then the Prover has at best a $2^{-n}$ chance of answering all $n$ questions correctly. Thus, the Verifier cannot be fooled often. Therefore, GNI $\in$ IP.

Note that GNI is an incomplete language in co-NP (assuming that PH is infinite [7]). So, the preceding discussion does not show that co-NP $\subseteq$ IP. For a while, it was believed that co-NP is not contained in IP, because there are oracle worlds where co-NP $\not\subseteq$ IP [14]. In fact, the computational power of interactive proofs was not fully appreciated until Lund, Fortnow, Karloff and Nisan [31] showed that IP actually contains the entire Polynomial Hierarchy. This result then led Shamir [34] to completely characterize IP by showing that

$$\text{IP} = \text{PSPACE}.$$

Then, Babai, Fortnow and Lund [2] characterized the computational power of multi-prover interactive proofs

$$\text{MIP} = \text{NEXP}.$$

In both cases, it is interesting to see that interactive proof systems provide alternative definitions of classic complexity classes. Thus, they fit very nicely with the overall classification of feasible computations. Furthermore, both of these problems have contradictory relativizations [14]. That is, there exist oracles $A$ and $B$ such that

$$\mathrm{IP}^A = \mathrm{PSPACE}^A \quad \text{and} \quad \mathrm{IP}^B \neq \mathrm{PSPACE}^B,$$

and similarly for the multi-prover case. Thus, these results provide the first *natural* counterexamples to the belief that problems with contradictory relativizations are beyond our proof techniques.

# 3 The Random Oracle Hypothesis

In this section we observe that the proof of IP = PSPACE does not relativize and show that for almost all oracles $A$ the two relativized classes differ:

$$\mathrm{IP}^A \neq \mathrm{PSPACE}^A.$$

It is easily seen that

$$\mathrm{IP}^{\mathrm{PSPACE}} = \mathrm{PSPACE}^{\mathrm{PSPACE}}$$

and using standard methods [3] one can construct an $A$ such that

$$\mathrm{IP}^A \neq \mathrm{PSPACE}^A.$$

Thus, the IP =? PSPACE problem has contradictory relativizations and the IP = PSPACE proof does not relativize. Similarly, we can see that the MIP =? NEXP problem has contradictory relativizations. In the following we show that these theorems also supply counterexamples to the Random Oracle Hypothesis.

## 3.1 $\mathrm{IP}^A \neq \mathrm{PSPACE}^A$ with probability 1, ...

Before we begin the construction of the counterexamples to the Random Oracle Hypothesis, we need to establish some conventions. For every verifier $V$ and every oracle $A$, there exists a prover which maximizes the probability that the verifier will accept each input string. This *optimal prover* considers all possible coin tosses made by $V$ and makes the replies to $V$ which result in the maximum accepting probability. Hence, in our discussions it suffices to specify the verifier and the oracle (as the prover is implicitly determined by them).

**Convention:** Let $\mathrm{opt}_V(A, x)$ denote the probability that the verifier $V$ accepts when interacting with the optimal prover on common input $x$ and access to the oracle $A$. If $V$ is part of an interactive proof for some language $L$, then

$$x \in L \Longrightarrow \mathrm{opt}_V(A, x) > \tfrac{2}{3}$$
$$x \notin L \Longrightarrow \mathrm{opt}_V(A, x) < \tfrac{1}{3}.$$

**Notation:** For every set $X$, let $X^{=n}$ denote the set $X \cap \{0,1\}^n$. Similarly, let $X^{<n}$ be the set of strings in $X$ of length strictly less than $n$ and let $X^{\leq n} = X^{<n} \cup X^{=n}$. By abuse of notation, we let $\{0,1\}^{<n}$ denote $(\{0,1\}^*)^{<n}$.

**Theorem 1** *For almost all oracles $A$, $\mathrm{IP}^A \subsetneq \mathrm{PSPACE}^A$.*

**Proof:** For all oracles $A$, $\mathrm{IP}^A \subseteq \mathrm{PSPACE}^A$, so we only need to show that this containment is strict for almost all oracles. We show that for almost all oracles $A$ the candidate language $\mathcal{L}(A)$ is in $\mathrm{PSPACE}^A$, but not in $\mathrm{IP}^A$, where $\mathcal{L}(A)$ is defined as:

$$\mathcal{L}(A) = \{ \, 1^n \mid \text{the cardinality of } A^{=n} \text{ is odd} \, \}.$$

Clearly, for all $A$, $\mathcal{L}(A) \in \mathrm{PSPACE}^A$. Let $V$ be a fixed verifier. We will show that the set of oracles $A$ for which $V^A$ constitutes a relativized interactive proof for $\mathcal{L}(A)$ has measure 0. Since there is only a countable number of verifiers, the set of oracles $A$ for which *some* verifier $V$ correctly accepts $\mathcal{L}(A)$ also has measure 0.

Let $n^c$ be a strict upper bound on the running time of the verifier $V$ on inputs of length $n$. Then, for any oracle $A$, the computation of $V^A(x)$, $|x| = n$, depends only on strings in the oracle of length up to $n^c$. Thus, if $A^{<n^c} = B^{<n^c}$, then the computation of $V^B(x)$ and $V^A(x)$ are identical. Now, define $\mathrm{seg}(n) = \{ \, \beta \mid \beta \subseteq \{0,1\}^{<n^c} \, \}$. I.e., a set $\beta$ is in $\mathrm{seg}(n)$ if and only if it is a finite set and contains only strings of length strictly less than $n^c$. We define $\mathrm{seg}(n)$ in this way because the computation of $V^A(1^n)$ depends only on $A^{<n^c}$ which is a set in $\mathrm{seg}(n)$.

Consider the class $C(n)$ of finite sets $\beta \in \mathrm{seg}(n)$ for which $V^\beta$ correctly determines whether $1^n$ is in $\mathcal{L}(\beta)$. That is, $C(n)$ contains the oracles $\beta \in \mathrm{seg}(n)$ for which one of the two following conditions holds:

- $V^\beta$ accepts $1^n$ with probability greater than $\frac{2}{3}$ and $1^n \in \mathcal{L}(\beta)$.
- $V^\beta$ accepts $1^n$ with probability less than $\frac{1}{3}$ and $1^n \notin \mathcal{L}(\beta)$.

For $n$ large enough, we can show that $|C(n)| < \frac{2}{3}|\mathrm{seg}(n)|$. By standard techniques in the literature [5, Lemma 1, pp.98–99], this bound on the size of $C(n)$ would be sufficient to prove the statement of the theorem. For the sake of completeness, we include a complete proof.

Now, let $N$ be large enough so that $2^N > 18N^c$. This guarantees that for all $n \geq N$, $\frac{1}{2}(1 - 6n^c 2^{-n}) > \frac{2}{3}$. Also, for all $n$ and for all $\alpha \subseteq \{0,1\}^{<n}$, let $B(\alpha, n)$ be the collection of $\beta \in \mathrm{seg}(n)$ such that $\beta^{<n} = \alpha$. Intuitively, $B(\alpha, n)$ is a set of finite extensions of $\alpha$.

The rest of our analysis is a finite extension argument. Our main lemma, Lemma 2, guarantees that for each $\alpha \in C(n)$ at most 2/3 of all the oracles $\beta \in B(\alpha, n)$ can be in $C(n^c)$. Thus, as we shall see in the Lemma 3, the measure of the oracles $A$ for which $V^A$ correctly determines whether $1^n$ is in $\mathcal{L}(A)$ *for all* $n$ is bounded by $(\frac{2}{3})^i$ for all $i$.

**Lemma 2** *For all $n \geq N$ and for all $\alpha \subseteq \{0,1\}^{<n}$, $|B(\alpha, n) \cap C(n)| \leq \frac{2}{3} \cdot |B(\alpha, n)|$. That is, the fraction of finite sets $\beta \in B(\alpha, n)$ for which $V^\beta$ correctly determines whether $1^n$ is in $\mathcal{L}(\beta)$ is at most $\frac{2}{3}$ of all $\beta \in B(\alpha, n)$.*

**Proof:** On input $1^n$ and access to an oracle $\beta \in \mathrm{seg}(n)$, the verifier $V$ interacts with the optimal prover and makes some queries to $\beta$ about some strings. Let $\mathcal{Q}(\beta, q)$ be the probability over the coin tosses of $V$ that $V^\beta(1^n)$ makes query $q$. Since $n^c$ is a strict upper bound on the running time of $V$, for every $n$ and for every sequence of coin tosses made by $V^\beta$ on input $1^n$, the machine $V$ makes less than $n^c$ queries. So, for every oracle $\beta \in \mathrm{seg}(n)$

$$\sum_{q\in\{0,1\}^n} \mathcal{Q}(\beta, q) \leq n^c.$$

Thus, there is a string $q \in \{0,1\}^n$ such that for all but a $3n^c\,2^{-n}$ fraction of the $\beta$'s in $B(\alpha, n)$, $\mathcal{Q}(\beta, q) \leq \frac{1}{3}$. Fix $q$ to be such a string.

Now let $\beta$ be an oracle in $B(\alpha, n)$ such that $1^n \in \mathcal{L}(\beta)$ and denote by $\beta^{(q)}$ the oracle which contains the same strings as $\beta$ except for $q$ (i.e., the symmetric difference of $\beta$ and $\beta^{(q)}$ equals $\{q\}$). Then,

$$\mathrm{opt}_V(\beta^{(q)}, 1^n) \geq \mathrm{opt}_V(\beta, 1^n) - \mathcal{Q}(\beta, q).$$

To see this, consider the prover $P'$ that uses the same strategy which the optimal prover uses on $V^\beta$ to convince $V^{\beta^{(q)}}(1^n)$ to accept. Then, on the computation paths of $V^{\beta^{(q)}}(1^n)$ which never asks about $q$, $P'$ will do as well as the optimal prover does on $V^\beta$. Since only a $\mathcal{Q}(\beta, q)$ fraction of the paths ask about $q$, and since the optimal prover will do at least as well as $P'$, the relationship above holds.

Finally, group all the $\beta \in B(\alpha, n)$ in pairs of the form $(\beta, \beta^{(q)})$ where $1^n \in \mathcal{L}(\beta)$ (and hence $1^n \notin \mathcal{L}(\beta^{(q)})$). We claim that whenever $\mathcal{Q}(\beta, q) < \frac{1}{3}$, the verifier $V$ is incorrect in determining the membership of $1^n$ in $\mathcal{L}(\beta)$ or in $\mathcal{L}(\beta^{(q)})$. To prove this, suppose that $V^\beta$ accepts $1^n$. (If $V^\beta(1^n)$ does not accept, we are done since $1^n \in \mathcal{L}(\beta)$.) Then, $\mathrm{opt}_V(\beta, 1^n)$ must be greater than $\frac{2}{3}$. So, $\mathrm{opt}_V(\beta^{(q)}, 1^n) > \frac{1}{3}$. However, $1^n \notin \mathcal{L}(\beta^{(q)})$, so $V$ fails to determine whether $1^n$ is in $\mathcal{L}(\beta^{(q)})$. By our choice of $q$, $\mathcal{Q}(\beta, q) < \frac{1}{3}$ for at least a $1-2\cdot 3n^c\,2^{-n}$ fraction of the pairs. Hence, $V$ fails to determine the membership of $1^n$ for at least $\frac{1}{2}\cdot(1-6n^c\,2^{-n}) > \frac{1}{3}$ of all $\beta \in B(\alpha, n)$. ▮

We now apply the standard extension technique [5].

**Lemma 3** *Let $n_i = N^{c^i}$ and let $R_i$ be the collection of the finite sets $\beta \in \mathrm{seg}(n_i)$ such that for all $r \leq n_i$, $V^\beta$ correctly determines whether $1^r$ is in $\mathcal{L}(\beta)$. Then, for all $i \geq 0$, $|R_i| \leq (\frac{2}{3})^i \cdot |\mathrm{seg}(n_i)|$.*

**Proof:** The proof is by induction on $i$. The base case, $i = 0$, is trivial since $R_i \subseteq \mathrm{seg}(n_i)$. So, $|R_0| \leq |\mathrm{seg}(n_0)|$. In the induction case, suppose that the theorem holds for $i = k$, we show that it also holds for $i = k+1$.

First, let $n = n_k$ and $m = n_{k+1}$. We partition $\mathrm{seg}(m)$ according to the initial segments up to length $n^c$. That is, $\mathrm{seg}(m) = \bigcup_{\alpha\in\mathrm{seg}(n)} B(\alpha, m)$. Now, suppose that $\alpha$ is not in $R_k$, for some $\alpha \in \mathrm{seg}(n)$. Then, for all $\beta \in B(\alpha, m)$, $\beta \notin R_{k+1}$. To see this, observe that in order for $\alpha \notin R_k$ to hold, there must be an $r \leq n$ such that $V^\alpha$ does not correctly determine whether $1^r$ is in $\mathcal{L}(\alpha)$. Since $\beta \in B(\alpha, n)$ and since $V^\alpha(1^r)$ only queries about strings of length strictly less than $r^c$, $V^\beta$ will also fail to determine whether $1^r$ is in $\mathcal{L}(\beta)$. Thus, $R_{k+1} \subseteq \bigcup_{\alpha\in R_k} B(\alpha, m)$.

Finally, if $\beta \in R_{k+1}$, then $V^\beta$ must correctly determine whether $1^m$ is in $\mathcal{L}(\beta)$. So, $\beta$ must be in $C(m)$, and $R_{k+1} \subseteq \bigcup_{\alpha \in R_k} C(m) \cap B(\alpha, m)$. By Lemma 2, we know that for all $\alpha$, $|C(m) \cap B(\alpha, m)| \leq \frac{2}{3} \cdot |B(\alpha, m)|$. Also, since for all $\alpha \in \text{seg}(n)$, $|B(\alpha, m)| = \frac{|\text{seg}(m)|}{|\text{seg}(n)|}$,

$$|R_{k+1}| \leq \frac{2}{3} \cdot \sum_{\alpha \in R_k} |B(\alpha, m)| = \frac{2}{3} \cdot |R_k| \cdot \frac{|\text{seg}(m)|}{|\text{seg}(n)|} \leq \left(\frac{2}{3}\right)^{k+1} \cdot |\text{seg}(m)| \ .$$

∎

To finish the proof of the theorem, simply note that for a random oracle $A$, the probability that $V^A$ correctly determines whether $1^n$ is in $\mathcal{L}(A)$ for all $n$ is bounded by the probability that $A^{<n_i^c} \in R_i$. This probability is in turn equal to $|R_i|/|\text{seg}(n_i)|$, which by Lemma 3 is bounded by $(2/3)^i$ for all $i$. Hence,

$$\text{Prob}_A[\ \text{IP}^A = \text{PSPACE}^A\ ] = 0.$$

□

Using standard techniques [3, 5, 14], the proof of Theorem 1 can be modified to yield the following theorem.

**Theorem 4** *For almost all oracles $A$,* co-NP$^A \not\subseteq$ IP$^A$.

**Proof:** We will use a different candidate language, $\mathcal{L}_1(A)$, for this proof. First, for each length $n$, we define $2^{n/2}$ disjoint segments, $S_1(n), S_2(n), S_3(n), \ldots, S_{2^{n/2}}(n)$, each containing $n/2$ contiguous strings of length $n$. Then,

$$\mathcal{L}_1(A) = \{\ 1^n \mid \forall i,\ 1 \leq i \leq 2^{\frac{n}{2}},\ S_i(n) \not\subseteq A\ \}.$$

Clearly,

$$\overline{\mathcal{L}_1(A)} = \{\ 1^n \mid \exists i,\ 1 \leq i \leq 2^{\frac{n}{2}},\ S_i(n) \subseteq A\ \},$$

so $\mathcal{L}_1(A) \in$ co-NP$^A$ for any $A$. We will prove that $\mathcal{L}_1(A) \notin$ IP$^A$ with oracle measure 1 by the same outline as the previous proof. Again, we fix a verifier $V$ with running time $n^c$. We also fix a length $n$ and a prefix $\alpha$ then consider only oracles from $B(\alpha, n)$, the set of $\beta \in \text{seg}(n)$ which extend $\alpha$.

In the following, let $N$ be large enough so that for all $n \geq N$, $0.36 < (1 - 1/n)^n < e^{-1}$ and $3n^c < 0.01 \cdot 2^n$. (For the first condition, $N \geq 25$ suffices.) As in the previous theorem, let $C(n)$ be the set of $\beta \in \text{seg}(n)$ such that $V^\beta$ correctly determines whether $1^n$ is in $\mathcal{L}_1(\beta)$. We show that $C(n) \cap B(\alpha, n)$ contains at most $\frac{2}{3}$ of all the sets in $B(\alpha, n)$.

**Lemma 5** *For all $n \geq N$ and for all $\alpha \subseteq \{0,1\}^{<n}$, $|C(n) \cap B(\alpha, n)| \leq \frac{2}{3} \cdot |B(\alpha, n)|$. That is, the fraction of finite sets $\beta \in B(\alpha, n)$ such that $V^\beta$ correctly determines whether $1^n$ is in $\mathcal{L}_1(\beta)$ is at most $\frac{2}{3}$ of all $\beta \in B(\alpha, n)$.*

**Proof:** Call $\beta \in B(\alpha, n)$ *accepting* if none of the first $2^{n/2}$ segments of $\{0,1\}^n$ is contained in $\beta$ and call $\beta$ *uniquely rejecting* if exactly one of these segments is contained in $\beta$. Observe that the fraction of accepting oracles converges quickly to $e^{-1}$ *from below*. By our choice of $N$, this fraction is bounded below by 0.36 and above by $e^{-1}$. The same holds for the fraction of uniquely rejecting oracles.

For any accepting $\beta$, let $\beta^{(i)} = \beta \cup S_i(n)$. Then, $\beta^{(i)}$ is a uniquely rejecting oracle and by an obvious extension of the argument used in Lemma 2, the following relation holds:

$$\mathrm{opt}_V(\beta^{(i)}, 1^n) \geq \mathrm{opt}_V(\beta, 1^n) - \sum_{q \in S_i(n)} \mathcal{Q}(\beta, q).$$

Thus, if $\mathrm{opt}_V(\beta, 1^n) \geq \frac{2}{3}$, then, for all but $3n^c$ of the $i$'s, $\mathrm{opt}_V(\beta^{(i)}, 1^n) > \frac{1}{3}$. Hence, for each accepting $\beta$ where $\mathrm{opt}_V(A, 1^n) \geq \frac{2}{3}$, there exists $2^{n/2} - 3n^c$ uniquely rejecting oracles $\beta^{(i)}$ such that $V^{\beta^{(i)}}$ fails to determine the membership of $1^n$ in $\mathcal{L}_1(\beta^{(i)})$. Moreover, each uniquely rejecting oracle $\beta^{(i)}$ can be obtained in this manner from at most $2^{n/2} - 1$ accepting oracles (one for each proper subset of $S_i(n)$). Let $\delta$ be the fraction of $\beta \in B(\alpha, n)$ for which $\beta$ is accepting and where $\mathrm{opt}_V(\beta, 1^n) \geq \frac{2}{3}$. Then, the fraction of oracles $\beta \in B(\alpha, n)$ for which $V^\beta$ fails to determine the membership of $1^n$ in $\mathcal{L}_1(\beta)$ is at least

$$(0.36 - \delta) + \left(\frac{2^{n/2} - 3n^c}{2^{n/2} - 1}\right)\delta = 0.36 - \left(\frac{3n^c - 1}{2^{n/2} - 1}\right)\delta \geq 0.36 - 0.01 > \frac{1}{3}.$$

This completes the proof of the Lemma. ▮

To finish the proof of the theorem, we simply use a lemma analogous to Lemma 3 to show that for a random oracle $A$, the probability that $V^A$ correctly determines the membership of $1^n$ in $\mathcal{L}_1(A)$ for all $n$ is bounded by $(2/3)^i$ for all $i$. Hence,

$$\mathrm{Prob}_A[\ \text{co-NP}^A \subseteq \mathrm{IP}^A\ ] = 0.$$

□

These results easily extend to the multi-prover interactive proof systems of Ben-Or, Goldwasser, Kilian and Wigderson [4]. For the sake of brevity, we omit the proofs.

**Theorem 6** *For almost all oracles $A$* $\mathrm{MIP}^A \subsetneq \mathrm{NEXP}^A$.

## 3.2 . . . , but $\mathrm{IPP}^A = \mathrm{PSPACE}^A$ always.

The IP = PSPACE and MIP = NEXP results provided natural examples against the Random Oracle Hypothesis. To give a more complete understanding of the behavior of these classes with random oracles, we define a less restrictive acceptance criterion for interactive proofs and denote the class of such languages by IPP. This class is a slight variant[6] of the class PPSPACE defined by Papadimitriou [33]. We show that

$$\forall A,\ \mathrm{IPP}^A = \mathrm{PSPACE}^A.$$

[6] The difference between the two definitions is that IPP uses private coins and PPSPACE uses public coins. However, the language classes can be shown to be identical using standard techniques [13].

Using the theorem in the previous section, we can provide both an example and a counterexample to the Random Oracle Hypothesis, because for almost all oracles $A$

$$\mathrm{IP}^A \neq \mathrm{PSPACE}^A \quad \text{and} \quad \mathrm{PSPACE}^A = \mathrm{IPP}^A,$$

whereas IP = PSPACE = IPP. This severely damages the already battered hypothesis because it shows that the Random Oracle Hypothesis is sensitive to small changes in the definition of complexity classes. Thus, it cannot be used to predict what happens in the real world.

**Definition** IPP: Let $V$ be a probabilistic polynomial time machine and let $P$ be an arbitrary TM. $P$ and $V$ share the same input tape and they communicate via a communication tape. $V$ forms an *unbounded interactive proof* for a language $L$ if

1. $x \in L \Longrightarrow$ Prob[ $P$-$V$ on $x$ accept ] $> \frac{1}{2}$.
2. $x \notin L \Longrightarrow \forall P^*$, Prob[ $P^*$-$V$ on $x$ accept ] $< \frac{1}{2}$.

A language $L$ is said to be in the class IPP if it has an unbounded interactive proof.

As in the case with IP, we only need to consider the interaction of the IPP verifier with the optimal prover. Again, we denote the probability that the verifier $V$ with access to an oracle $A$ accepts a string $x$ by $\mathrm{opt}_V(A, x)$. The interaction between the verifier and the optimal prover can be represented by a computation tree with alternating "maximizing" nodes (prover's move) and "averaging" nodes (verifier's move). We rely on this observation in the proof of the following theorem.

**Theorem 7** *For all oracles $A$,* $\mathrm{IPP}^A = \mathrm{PSPACE}^A$.

**Proof:**
$\mathrm{IPP}^A \subseteq \mathrm{PSPACE}^A$: Let $L$ be a language in $\mathrm{IPP}^A$, and $V$ be a verifier for an unbounded interactive proof for $L$. It suffices to show that on input $x$ and access to oracle $A$, the value $\mathrm{opt}_V(A, x)$ can be computed using space polynomial in $|x|$. This is done by recursively computing the value $\mathrm{opt}_V(A, x, h)$, which we define to be the residual accepting probability of $V$ on input $x$ and access to oracle $A$, given the contents $h$ of some previous messages sent between prover and verifier. In case the last message in $h$ is a verifier message, $\mathrm{opt}_V(A, x, h)$ is computed by enumerating all possible prover messages, $m$, and taking the maximum over all $\mathrm{opt}_V(A, x, h \cdot m)$'s. In case the last message in $h$ is a prover message, $\mathrm{opt}_V(A, x, h)$ is computed by enumerating all possible sequences of verifier coin tosses ($r$) which are consistent with the history $h$, computing for each such sequence the verifier message, $m_r$, and taking the average over all $\mathrm{opt}_V(A, x, h \cdot m_r)$'s.

$\mathrm{PSPACE}^A \subseteq \mathrm{IPP}^A$: This proof is similar to the proof that NP $\subseteq$ PP [16] (see also [33]). Let $L$ be a language in $\mathrm{PSPACE}^A$. Then there is a machine $M^A$ accepting $L$ which runs in space $p(n)$ and halts in exactly $2^{q(n)}$ steps for some polynomials $p$ and $q$. Consider the verifier $V$ which attempts to find out if a string $x$ is in $L$ by running the CHECKCOMP subroutine (Figure 2) on input $(I, F, 2^{q(n)})$, where $I$ and $F$ are the unique initial and final configurations of $M^A(x)$. Now, if $x \in L$, then the optimal prover can always convince $V$ to

```
procedure CHECKCOMP($C_1, C_2, s$) ;
{ This procedure tries to detect if $M^A$ can reach configuration $C_2$ from configuration $C_1$ in
$s$ steps. }
begin
    if $s = 1$ then
        { This may involve querying the oracle. }
        if $C_1 \to C_2$ in one step of $M^A$ then accept else reject
    else
        Ask the prover for the middle configuration $C_3$ between $C_1$ and $C_2$.
        Toss a coin.
        if the coin toss is heads then
            CHECKCOMP($C_1, C_3, s/2$)
        else
            CHECKCOMP($C_3, C_2, s/2$)
end { procedure }
```

Figure 2: Pseudo-code for procedure CHECKCOMP.

accept. On the other hand, if $x \notin L$, then the probability that the verifier rejects is at least $2^{-q(n)}$ from the following lemma.

**Lemma:** Let WRONG($C_1, C_2, s$) be the proposition that configuration $C_2$ does not follow from configuration $C_1$ in exactly $s$ steps. Then for all $A$, $C_1$, $C_2$, $C_3$, $s$ and $u$ with $0 \leq u \leq s$,

$$\begin{aligned}&\mathrm{WRONG}(C_1, C_2, s) \Longrightarrow \\ &\quad \forall u,\ 0 \leq u \leq s,\ C_3,\ \mathrm{WRONG}(C_1, C_3, u) \bigvee \mathrm{WRONG}(C_3, C_2, s-u).\end{aligned}$$

Now, the verifier described above does not define an IPP proof for the language $L$, because

$$\begin{aligned}&x \in L \Longrightarrow \mathrm{opt}_V(A, x) = 1\\ &x \notin L \Longrightarrow \mathrm{opt}_V(A, x) < 1 - 2^{-q(n)}.\end{aligned}$$

However, these probability bounds can be normalized and centered around 1/2. To do this, our new verifier $V'$ tosses $q(n) + 2$ coins and naively rejects with probability $1/2 - 2^{-q(n)-2}$ (one less than half the possible coin tosses). When $V'$ does not reject outright, it simulates $V$. Now, if $x \in L$, then $V'$ accepts whenever it simulates $V$ — i.e., with probability $1/2+2^{-q(n)-2}$, which is strictly greater than 1/2. On the other hand, if $x \notin L$, $V'$ accepts with probability less than $(1 - 2^{-q(n)}) \cdot (1/2 + 2^{-q(n)-2}) < 1/2$. Thus,

$$\begin{aligned}&x \in L \Longrightarrow \mathrm{opt}_{V'}(A, x) > \tfrac{1}{2},\\ &x \notin L \Longrightarrow \mathrm{opt}_{V'}(A, x) < \tfrac{1}{2},\end{aligned}$$

and $L \in$ IPP. □

# 4 Conclusion

We have shown that random oracle results do not reliably predict the base case behavior of complexity classes. On the other hand, the meaning of random oracle results needs to be clarified and remains an interesting problem. It would be very interesting to know if there are identifiable problem classes for which the random oracle results do point in the right direction.

In addition, we would like to note that the IP = PSPACE and MIP = NEXP results demonstrated *equality* in the base case. In many other problems with contradictory relativizations, we expect the unrelativized complexity classes to be different (e.g., we expect that P $\neq$ NP $\neq$ PSPACE, etc). The next big challenge for complexity theorists is to resolve one of these problems and *separate* — if not P and NP — any two classes which have contradictory relativizations or which are equal relative to a random oracle.

# References

[1] L. Babai. Trading group theory for randomness. In *ACM Symposium on Theory of Computing*, pages 421–429, 1985.

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

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

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

[5] C. Bennett and J. Gill. Relative to a random oracle A, $\mathrm{P}^{\mathrm{A}} \neq \mathrm{NP}^{\mathrm{A}} \neq \text{co-NP}^{\mathrm{A}}$ with probability 1. *SIAM Journal on Computing*, 10(1):96–113, February 1981.

[6] L. Berman and J. Hartmanis. On isomorphism and density of NP and other complete sets. *SIAM Journal on Computing*, 6:305–322, June 1977.

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

[8] J. Cai. With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy. In *ACM Symposium on Theory of Computing*, pages 21–29, 1986.

[9] J. Cai. Probability one separation of the Boolean hierarchy. In *4th Annual Symposium on Theoretical Aspects of Computer Science*, volume 247 of *Lecture Notes in Computer Science*, pages 148–158. Springer-Verlag, 1987.

[10] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The Boolean hierarchy I: Structural properties. *SIAM Journal on Computing*, 17(6):1232–1252, December 1988.

[11] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The Boolean hierarchy II: Applications. *SIAM Journal on Computing*, 18(1):95–111, February 1989.

[12] B. Chor, O. Goldreich, and J. Håstad. The random oracle hypothesis is false. Technical Report 631, Department of Computer Science, Technion, 1990.

[13] A. Condon. *Computational Models of Games.* An ACM Distinguished Dissertation. MIT Press, 1988.

[14] L. Fortnow and M. Sipser. Are there interactive protocols for co-NP languages? *Information Processing Letters*, 28(5):249–251, August 1988.

[15] W. Gasarch. More on the random oracle hypothesis: What is true almost always is not necessarily so. Technical Report TR-1956, Department of Computer Science, University of Maryland—College Park, 1985.

[16] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, December 1977.

[17] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, July 1991.

[18] S. Goldwasser. Interactive proof systems. In *Computational Complexity Theory*, volume 38 of *Proceedings of Symposia in Applied Mathematics*, pages 108–128. American Mathematical Society, 1989.

[19] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[20] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *ACM Symposium on Theory of Computing*, pages 59–68, 1986.

[21] J. Hartmanis. Solvable problems with conflicting relativizations. *Bulletin of the European Association for Theoretical Computer Science*, 27:40–49, Oct 1985.

[22] J. Hartmanis, R. Chang, D. Ranjan, and P. Rohatgi. Structural complexity theory: Recent surprises. In *Proceedings of the 2nd Scandinavian Workshop on Algorithm Theory*, volume 447 of *Lecture Notes in Computer Science*, pages 1–12. Springer-Verlag, 1990.

[23] J. Håstad. Almost optimal lower bounds for small depth circuits. In Silvio Micali, editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 143–170. JAI Press Inc, 1989.

[24] J. E. Hopcroft. Turing machines. *Scientific American*, pages 86–98, May 1984.

[25] R. Impagliazzo. NP in zero-knowledge fails with respect to random oracle. Personal communication, 1990.

[26] J. Kadin. The polynomial time hierarchy collapses if the Boolean hierarchy collapses. *SIAM Journal on Computing*, 17(6):1263–1282, December 1988.

[27] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *ACM Symposium on Theory of Computing*, pages 302–309, 1980.

[28] K. Ko. Relativized polynomial time hierarchies having exactly $k$ levels. In *ACM Symposium on Theory of Computing*, pages 245–253, 1988.

[29] S. A. Kurtz. On the random oracle hypothesis. In *ACM Symposium on Theory of Computing*, pages 224–230, 1982.

[30] S. A. Kurtz, S. R. Mahaney, and J. S. Royer. The isomorphism conjecture fails relative to a random oracle. In *ACM Symposium on Theory of Computing*, pages 157–166, 1989.

[31] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 2–10, 1990.

[32] S. Mahaney. Sparse complete sets for NP: Solution of a conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences*, 25(2):130–143, 1982.

[33] C. H. Papadimitriou. Games against nature. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 446–450, 1983.

[34] A. Shamir. IP = PSPACE. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 11–15, 1990.

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