The Role of Relativization in Complexity Theory

Lance Fortnow*
University of Chicago
Department of Computer Science
1100 East 58th Street
Chicago, Illinois 60637

Abstract

Several recent nonrelativizing results in the area of interactive proofs have caused many people to review the importance of relativization. In this paper we take a look at how complexity theorists use and misuse oracle results. We pay special attention to the new interactive proof systems and program checking results and try to understand why they do not relativize. We give some new results that may help us to understand these questions better.

1 Introduction

The recent result **IP** = **PSPACE** [LFKN92, Sha92] surprised the theoretical computer science community in more ways than one. A few years earlier, Fortnow and Sipser [FS88] created an oracle relative to which co-**NP** did not have interactive proofs. The **IP** = **PSPACE** result was honestly a nonrelativizing theorem.

Several questions immediately popped up: Why didn't this result relativize? What specific techniques were used that avoided relativization? How can we use these techniques to prove other nonrelativizing facts?

Also much older questions resurfaced: What exactly to oracle results mean? What should we infer, if anything, from a relativization?

Such questions gained even more importance when we discovered the amazing power of multiple prover interactive proof systems [BFL91], transparent proofs [BFLS91] and probabilistically checkable proofs [AS92, ALM+92].

We may not find satisfactory answers to these questions in the near future. However, in this paper we will give some intuition and some theorems about oracles that may help shed light on some of these issues.

In Section 3, we will see that relativization is an extremely powerful tool in helping complexity theorists direct their research.

In Section 4, we will look at early results of provable statements with negative relativization. We will look at these various examples and argue that they all lack a proper oracle access mechanism.

In section 5, we take a close look at the new unrelativizing results for interactive proof systems. We prove some new results that may help us understand these issues better. We also argue that the

*Email: fortnow@cs.uchicago.edu. Partially supported by NSF grant CCR 92-53582.

new interactive proof system techniques do not relativize because they take advantage of certain algebraic properties of complexity classes.

In section 5.1 we look at what happens when we add an oracle to probabilistically checkable proofs (**PCP**). Arora, Lund, Motwani, Sudan and Szegedy [ALM+92] show that every language in **NP** has a **PCP** using only a logarithmic number of random coin tosses and a constant number of queries to the proof. We show that in a relativized world, for all k, such a result does not hold even if we allow the verifier to use a polynomial number of random bits and n^k proof queries (Theorem 5.2).

Although Heller [Hel81] has created an oracle A relative to which $\mathbf{NP}^A = \mathbf{EXP}^A$, we show that $\mathbf{PCP}^A(\log n, 1) = \mathbf{EXP}^A$ would imply that $\mathbf{P} \neq \mathbf{NP}$ (Theorem 5.4). Thus finding such an oracle would be as hard as settling the $\mathbf{P} = \mathbf{NP}$ question.

Arora, Impagliazzo and Vazirani [AIV92] argue that the "local-checkability" property of complexity classes is a major reason that the results on interactive proofs do not relativize. In Section 5.2, we give negative evidence for this thesis by showing that under a reasonable access mechanism, local checkability does in fact relativize.

We give support instead to the thesis that it is the algebraic properties of complexity classes that do not relativize. In Section 5.3, we give evidence for this proposition by showing that $\mathbf{IP} = \mathbf{PSPACE}$ holds relative to algebraic extensions of arbitrarily complicated languages.

Finally, in Section 6, we give a brief arguement against infering any information from random oracle results. We argue that we should only use random oracles as a tool to combine oracle constructions. However, we believe that generic oracles are a much stronger and sharper tool for this purpose.

Caveats

This paper contains several opinions on the use and misuse of relativization results. We must caution the reader that other complexity theorists may have differing opinions on these matters.

In this paper, we have tried to give several examples to illustrate various points. However this paper is *not* meant to be a survey paper. Many important works in the area have not been mentioned due to lack of space.

2 Notation and Definitions

Most of the notation and definitions follow from the standard textbooks on the field [HU79, GJ79].

We use $\oplus$ to represent the join of two sets A and B, i.e., $A \oplus B = (\{0\} \times A) \cup (\{1\} \times B)$. We use **FP** to represent the polynomial-time computable functions.

It is a misnomer to relativize a complexity class $\mathcal{C}$. Instead suppose we take an enumeration of machines for $\mathcal{C}$ and give them some access mechanism to an oracle set A. We then say the relativized class $\mathcal{C}^A$ consists of the languages recognized by the acceptance criteria for $\mathcal{C}$ applied to the machines using the oracle A.

Of course this definition may depend greatly on the specific enumeration of the machines of $\mathcal{C}$ as well as the oracle access mechanism. The usual access mechanism consists of a separate oracle tape that the Turing machine can write down queries and learn the answers. However, as we shall see in Sections 4 and 5.2, such models may unduly handicap the machines.

2.1 Relativizing Formulae

It will be useful to have relativized **NP**-complete and **PSPACE**-complete sets.

Let a *relativized 3CNF formula* be a CNF formula with clauses of the form:

$$x_{i1} \vee x_{i2} \vee x_{i3} \vee A(x_{j1}, \ldots, x_{jk})$$

where $A(x_{j1}, \ldots, x_{jk})$ is true if $x_{j1} \ldots x_{jk}$ is in A. Any of the variables or the $A(x_{j1}, \ldots, x_{jk})$ term may be negated.

Lemma 2.1 *Let ϕ_A be a relativized 3CNF formula over an oracle A. Let ψ_A be a closed relativized 3CNF formula with arbitrary first-order existential and universal quantifiers over the variables.*

1. *Determining whether ϕ_A is satisfiable is* $\mathbf{NP}^A$*-complete.*
2. *Determining the truth value of ψ_A is* $\mathbf{PSPACE}^A$*-complete.*

Furthermore the completeness reductions do not need access to the oracle.

Goldsmith and Joseph [GJ86] prove the first part of the lemma. An easy modification of their proof gives us the second part as well.

3 Uses of Oracle Results

In this section, we will discuss some legitimate uses of relativization results. As we will see, complexity theorists have used relativization as a powerful tool in studying complexity theory.

In Section 3.1 we will see how theorists use oracles to discover what techniques will not likely work to solve certain problems. In Section 3.2, we will see how relativization allows us to push at a problem in two different directions. In Section 3.3, we will argue that that lack of nonrelativizable techniques have caused theorists to look at new directions of research. Finally, in Section 3.4 we will see how old relativization results help us to recognize new techniques that can be applied to other problems.

Of course, theorists must execute extreme care in how one should interpret oracle results. In Sections 4 and 5, we look at some computational models where oracle results may not have the expected interpretation.

3.1 Limiting techniques

Suppose we can show for some statement S that there exists an oracle A such that S fails relative to A in some oracle model. Then any proof that S hold must not relativize in that model or otherwise that statement would also hold relative to A. If we can also find an oracle relative to which S holds then no relativizable technique can decide the truth of S.

Baker, Gill and Solovay [BGS75] noted this in their original oracle paper where they give a relativized world where $\mathbf{P} = \mathbf{NP}$ and another where $\mathbf{P} \neq \mathbf{NP}$. They also noted that essentially all the known complexity techniques at that time relativize. They concluded that current techniques would not solve the $\mathbf{P} = \mathbf{NP}$ question.

In the nearly two decades since the Baker-Gill-Solovay paper, there have been literally hundreds of results in complexity theory. With the exception of some results in interactive proof systems (see

Sections 3.4 and 5) all of the results in complexity theory have relativized. These include several important results such as $\mathbf{PH} \subseteq \mathbf{P}^{\#\mathbf{P}}$ [Tod91] and **PP** is closed under intersection [BRS91].

The techniques for interactive proofs have not yet proven fruitful towards proving *any* other theorems about complexity theory. Thus it really does appear that we still need to develop new techniques to settle the hundreds of complexity statements that relativize both ways.

Early on some people speculated that perhaps the Baker-Gill-Solovay result indicated that these questions about complexity theory may fall outside the axioms of set theory (see [Har78], chapter 7). However most researchers no longer subscribe to this viewpoint anymore because of lack of evidence and some of the examples in Sections 4 and 5.

3.2 Two directions

Often in complexity theory, one has a complexity statement S where one can easily show a relativized world where S holds but it is open whether there exists a relativizable proof of S. In order to tackle this problem, many complexity theorists look at trying to prove two opposite directions:

- Trying to prove S, or
- Creating a relativized world where S fails.

Often by working on a problem in two directions, one can often push a failure of a proof in one direction into a proof of the other direction.

Goldwasser and Sipser [GS89] used this method in their proof of the equivalence of public and private coins in interactive proof systems. Initially, Goldwasser and Sipser tried to prove that the private coin interactive proof hierarchy was infinite relative to some oracle. The failure of that attempt led to the equivalence result that implied the collapse of the private coin hierarchy.

3.3 New directions

Relativization results often lead researchers in new directions that prove extremely fruitful. This may happen in two different ways:

1. A specific problem may have large amount of time devoted to it because of the lack of a negative relativization result.
2. Whole new directions of research may develop in attempts to find new techniques to answer problems with negative relativizations.

Beigel, Reingold and Spielman [BRS91] concentrated their efforts on trying to show that **PP** is closed under intersection mainly because of the importance of the question and the fact that no negative relativization existed. They succeeded in finding a relativizable proof.

The whole area of circuit complexity was developed as a potential method for attacking the hard problems like $\mathbf{P} \neq \mathbf{NP}$. Although one can easily relativize circuits [Wil85], many researchers believed the structure of circuits would allow us to find nonrelativizable techniques to solve some basic complexity questions.

Circuit complexity still has a long way to go before it can fulfill this promise. However, circuit complexity has provided us with several interesting combinatorial problems and some other applications for machine-based complexity theory. In fact, circuit complexity has given us the tools to prove some important relativization results, such as a relativized world where the polynomial-time hierarchy is infinite (see [Hås89]).

3.4 Recognizing new techniques

Suppose we have a proof of a statement S but we also know that there exists a relativized world where S does not hold. We can then analyze the proof of S to find the technique used in that proof that does not relativize. We can then, hopefully, apply this technique to other negatively relativized problems.

In 1989, Noam Nisan found a multiple-prover interactive proof for $\overline{\mathbf{SAT}}$, a problem with a known negative relativization [FRS88]. Lund, Fortnow and Karloff analyzed this proof and using Nisan's techniques combined with some new ones showed a single-prover interactive proof system for $\overline{\mathbf{SAT}}$ [LFKN92]. Several other important papers on interactive proof theory followed from extensions of these techniques (e.g. [Sha92, BFL91, BFLS91, ALM+92]). Babai [Bab90] goes into more detail about the history of these developments.

4 When Oracles Fail

Almost since the first relativization results of Baker, Gill and Solovay [BGS75], complexity theorists have look for true mathematical statements that fail in some relativized world. Though researchers have published several papers along these lines (e.g. [Mor81, Kur83, Har85, HCKM88, Cha90]) most of these results hold because the machine model does not have proper access to the oracle.

By understanding the various types of failures, it will enhance our ability to interpret relativization results. One should not simply ignore relativization results that fall into the categories below but one should cautiously draw inferences from such results.

In the section we discuss several different categories of examples where oracles fail. In Section 5 we will look at the special case of interactive proofs.

Space-Bounded Computation

Should the oracle tape count towards the space bound? Either way can cause problems. If we do count the tape that would prevent a result like $\mathbf{ASPACE}(poly) = \mathbf{EXP}$ from relativizing because the space-bounded machine could not ask exponentially long queries. If we do not count the tape then a result like $\mathbf{ATIME}(poly) = \mathbf{PSPACE}$ will not relativize because the space-bounded machine could ask exponentially long queries. Hartmanis, Chang, Kadin and Mitchell [HCKM88] have further discussion on these oracle models.

Ladner and Lynch [LL76] suggest a reasonable alternative: Do not count the space on the oracle tape but make the oracle tape a write-only tape that is automatically erased after each query. While this suggestion works well below $\mathbf{P}$, it does not solve the quandary described in the previous paragraph.

Buss [Bus88] creates an oracle model that seems to get around this problem but is too cumbersome for use in practice. If one must relativize a space class we suggest to either use the corresponding relativized alternating time class or to use the Landner and Lynch model with a careful eye.

Partial Relativizations

In these results some, but not all, of the objects involved are allowed to have access to the oracle. Kurtz [Kur83] showed that $\mathbf{NP} \subseteq \mathbf{P}^{\mathbf{SAT}}$ but for a random R, $\mathbf{NP}^R \not\subseteq \mathbf{P}^{\mathbf{SAT}\oplus R}$ giving a counterexample to the Random Oracle Hypothesis (See Section 6). However, it uses heavily the fact that queries to the **SAT** oracle do not have access to R. In fact, for all A, $\mathbf{NP}^A \subseteq \mathbf{P}^{\mathbf{SAT}^A}$. Hartmanis [Har85] has a similar example.

Chang [Cha90] has a different example where a function $s(n)$ is not space-constructible in the unrelativized world but $s(n)$ is space-constructible in a relativized world. However, if the function $s(n)$ was computed using a relativized Turing machine, the theorem Chang states would relativize.

Insufficient Oracle Access

One of the basic theorems in complexity theory is linear speedup: if a program takes $t(n)$ time with $t(n)$ superlinear then for every c, for all but a finite n, there is a another Turing machine that takes only $t(n)/c$ time. Moran [Mor81] noted that this result does not relativize: we need only make the language dependent on strings in the oracle of length say $t(n)+1$.

This oracle failure occurs because of insufficient oracle access. The proof of the linear speed-up theorem works by encoding several tape square into single square by using a larger set of tape symbols. However, the relativized model does not allow this compression on the oracle tape. If we change the model to allow compressed queries to the oracle then we will indeed have a relativized linear speedup theorem. We will show another example of this type of failure in Section 5.2.

When interpreting a relativization result, one must be extremely careful in looking at how the computational model may access the oracle. If we can prove a statement S false relative to an oracle in a certain model then techniques that relativize *only in the model* will not work to prove S.

Recent nonrelativizing results on interactive proofs do not appear to fit in any of the above categories. In Section 5, we will explore as best we can why these techniques do not relativize.

5 Relativizations of Interactive Proofs

Interactive proofs were invented in 1985 simultaneously by Babai [Bab85, BM88] and Goldwasser, Micali and Rackoff [GMR89]. We refer the reader to these papers for descriptions and formal definitions of interactive proofs.

In 1986, Fortnow and Sipser [FS88] created a relativized world where some language in co-**NP** does not have an interactive proof. Fortnow and Sipser then conjectured that there exist languages in co-**NP** that do not have interactive proofs.

In 1989, Lund, Fortnow, Karloff and Nisan [LFKN92] showed that every co-**NP** language has an interactive proof system. By the Fortnow and Sipser oracle, their proof could not relativize. In this section, we will try to examine why that proof does not relativize.

In Section 5.1, we discuss a recent important result about probabilistically checkable proofs by Arora, Lund, Motwani, Sudan and Szegedy [ALM+92] and show that that result does not relativize in a strong way. We also discuss the possibility that one can use the probabilistically checkable proof result to show that $\mathbf{NP} \neq \mathbf{EXP}$. In Section 5.2, we discuss local checkability developed by Arora, Impagliazzo and Vazirani [AIV92] and argue that this notion does not give a satisfactory answer to the question of why the interactive proof results do not relativize. In Section 5.3, we argue that a more algebraic property of complexity classes is the root of this nonrelativization. Finally, in Section 5.4 we discuss some further questions about interactive proofs and relativization.

5.1 Probabilistically Checkable Proofs

In this section, we will show two results about probabilistically checkable proof systems. First we show that the result of Arora, Lund, Motwani, Sudan and Szegedy [ALM+92] does not relativize in a strong way. We then look at what happens if we look at oracles trying to relate **PCP** and **EXP**.

Arora and Safra [AS92] define a hierarchy of complexity classes **PCP**, corresponding to the number of random and query bits required to verify a proof of membership in the language, as follows:

A *verifier* M is a probabilistic polynomial-time Turing machine with random access to a string Π representing a membership proof; M can *query* any bit of Π. Call M an $(r(n), q(n))$-*restricted* verifier if, on an input of size n, it is allowed to use at most $O(r(n))$ random bits for its computation, and query at most $O(q(n))$ bits of the proof.

A language L is in $\mathbf{PCP}(r(n), q(n))$ if there exists an $(r(n), q(n))$-restricted verifier M such that for every input x:

1. If $x \in L$, there is a proof Π_x which causes M to accept for every random string, *i.e.* with probability 1.

2. If $x \notin L$, then for all proofs Π, the probability that M using proof Π accepts is bounded by 1/2.

In 1988, Ben-Or, Goldwasser, Kilian and Wigderson [BGKW88] defined *multiple prover interactive proof systems* where the verifier communicates with several provers that cannot communicate among themselves. Fortnow, Rompel and Sipser [FRS88] show that the languages accepted by multiple provers (**MIP**) and $\cup_{k>0}\mathbf{PCP}(n^k, n^k)$ are equivalent. Babai, Fortnow and Lund [BFL91] building on the work of [LFKN92] show that $\mathbf{NEXP} = \mathbf{MIP} = \cup_{k>0}\mathbf{PCP}(n^k, n^k)$.

Arora, Lund, Motwani, Sudan and Szegedy [ALM+92] building on techniques of Arora and Safra [AS92] show the following surprising and powerful theorem:

Theorem 5.1 $\mathbf{NP} = \mathbf{PCP}(\log(n), 1)$

One can easily create an oracle relative to which this result does not relativize because the verifier has only a polynomial number of computation paths. We use techniques of Fortnow and Sipser [FS88] and Fortnow, Rompel and Sipser [FRS88] to show a much stronger negative oracle result:

Theorem 5.2 *For some oracle A, $\mathbf{NP}^A$ is not contained in $\cup_{j>0}\mathbf{PCP}^A(n^j, n^k)$ for any fixed k.*

Proof: Let $L_k(A) = \{1^n | \text{There exists a string } x \text{ of length } n^{2k} \text{ in } A\}$. Clearly $L_k(A)$ is in $\mathbf{NP}^A$ for every A.

Let $M_1, M_2, \ldots$ be an enumerate of probabilistic verifiers where M_i runs in time n^i.

Terminology: M_i makes two kinds of queries: A proof query is a query to the membership proof Π; An oracle query is a query to A.

Requirement $R_{i,k}$: $L_k(A)$ is not accepted by verifier M_i^A where M_i^A makes at most n^k proof queries.

We will handle these countably many requirements one at a time. Initially A is the empty oracle. We will add a finite number of strings to A in each stage. The final A is the union of all the strings added in each stage.

Stage (i,k):

1. Pick n large enough so that it does not conflict with earlier stages and such that $2^n >> n^{2ik}$.
2. Look at $M_i^A(1^n)$. If there exists a membership proof that causes $M_i^A(1^n)$ to accept with probability greater than 1/4 then we have fulfilled $R_{i,k}$. Go on to the next stage.
3. If some finite extension to A would cause $M_i^A(1^n)$ to ask more than n^k proof queries then make that extension and $R_{i,k}$ is fulfilled. Go on to the next stage.
4. Put x from Lemma 5.3 into A. Suppose there existed some membership proof such that the probability of $M_i^A(1^n)$ accepts is one. Then this same membership proof will cause $A - \{x\}$ to accept with probability at least 3/4 because of Lemma 5.3. This contradicts step 2. Thus we have fulfilled $R_{i,k}$. Go on to the next stage.

Lemma 5.3 *There exists a string x of length n^{2k} such that*

$$\Pr(\textit{There exists a membership proof } \Pi \textit{ where } M_i^A(1^n) \textit{ has an oracle query to } x) < 1/4$$

The probability is taken over the random strings of $M_i^A(1^n)$.

Proof: Fix a random coin toss r. The number of oracle queries that $M_i^A(1^n)$ could make is at most $n^i 2^{n^k}$ for all possible membership proofs because there are at most n^k proof queries. So a randomly chosen string of length n^{2k} will have extremely low probability of being in this set of oracle queries. The lemma follows from the usual averaging argument. □

The oracle constructed by this algorithm will fulfill all the requirements and thus $L(A)$ is not contained in $\cup_{j>0}\mathbf{PCP}^A(n^j, n^k)$ for any fixed k. □

Since clearly $\mathbf{PCP}^A(\log n, 1) \subseteq \mathbf{NP}^A$ for all A, We can interpret Theorem 5.1 as a weak characterization of **NP**. Perhaps we can use this characterization to separate **NP** from higher complexity classes like **PSPACE** and **EXP** by separating $\mathbf{PCP}(\log n, 1)$ from these classes.

However $\mathbf{P}^A \subseteq \mathbf{PCP}^A(\log n, 1)$ for all A and for some B we have $\mathbf{P}^B = \mathbf{PSPACE}^B$ [BGS75], for this B we will have $\mathbf{P}^B = \mathbf{PCP}^B(\log n, 1) = \mathbf{NP}^B = \mathbf{PSPACE}^B$. Thus we would need additional nonrelativizable techniques to separate $\mathbf{PCP}(\log n, 1)$ from **PSPACE**.

The class **EXP** does not fall into the same trap. For every oracle A, we have $\mathbf{P}^A \neq \mathbf{EXP}^A$ since the deterministic time hierarchy theorem relativizes [HS65]. Heller [Hel81] showed that there exists an oracle A where $\mathbf{NP}^A = \mathbf{EXP}^A$. Since Theorem 5.1 does not relativize, Heller's theorem does not necessarily imply that $\mathbf{PCP}^A(\log n, 1) = \mathbf{EXP}^A$. In fact any such oracle will be hard to find:

Theorem 5.4 *If there exists an oracle A such that $\mathbf{PCP}^A(\log n, 1) = \mathbf{EXP}^A$ then $\mathbf{P} \neq \mathbf{NP}$ in the unrelativized world.*

Proof: Since a $\mathbf{PCP}^A(\log n, 1)$ verifier has only a polynomial number of computation paths, a polynomial time machine could query all of the oracle queries on these paths. Then the polynomial-time machine could determine whether a proof Π exists by a single unrelativized **NP** question. Thus we have for every A that $\mathbf{PCP}^A(\log n, 1) \subseteq \mathbf{P}^{A \oplus \mathbf{SAT}}$.

Assume that $\mathbf{P} = \mathbf{NP}$ and for some oracle A, $\mathbf{PCP}^A(\log n, 1) = \mathbf{EXP}^A$. We then have $\mathbf{EXP}^A = \mathbf{PCP}^A(\log n, 1) \subseteq \mathbf{P}^{A \oplus \mathbf{SAT}} = \mathbf{P}^A$ which contradicts the fact that the deterministic time hierarchy relativizes. $\square$

5.2 Local Checkability

Arora, Impagliazzo and Vazirani [AIV92] define the notion of proof checker as follows:

A *proof-checker* is a Turing machine M that uses universal quantification and which is provided, in addition to the input, a *proof string*. It is allowed random access to both the input and the proof string. It is said to *accept* an input x using proof-string Π (denoted $M^\Pi(x) = 1$) iff all branches created by its universal branching accept.

A language L is in the class $\mathbf{PFCHK}(t(n))$ iff there is a proof-checker M that runs in $O(t(n))$ time and has the property

- $\forall x \in L$, there exists a Π such that $M^\Pi(x) = 1$.
- $\forall x \notin L$, $M^\Pi(x) = 0$ for every Π.

Using the Cook-Levin theorem [Coo71, Lev73], Arora, Impagliazzo and Vazirani show that $\mathbf{NP} = \mathbf{PFCHK}(\log n)$. Arora, Impagliazzo and Vazirani call this fact the "Local Checkability Theorem".

Now suppose we relativize $\mathbf{PFCHK}$ to an oracle A by giving the proof-checker an oracle tape by writing a full oracle query z on the tape and magically entering a state q_y if $z \in A$ and a state q_n if $z \notin A$. However since only queries of length $O(\log n)$ can be asked by a $\mathbf{PFCHK}^A(\log n)$ proof checker, it is easy to construct oracles A such that $A \notin \mathbf{PFCHK}^A(\log n)$ and thus $\mathbf{NP}^A \notin \mathbf{PFCHK}^A(\log n)$.

Arora, Impagliazzo and Vazirani conclude that local checkability does not relativize. However, we feel that any oracle access mechanism that prevents a machine from querying its own input is an extremely weak access model. We will present a more robust access model and show that relative to this model, local checkability does relativize.

We will allow M to query the oracle as follows: When M wants to make an oracle query, M writes two pointers on the oracle tape. M will now go to state q_y if the string located between those two pointers on the proof tape is in the oracle and will go to state q_n otherwise.

Theorem 5.5 *For all oracles A,* $\mathbf{NP}^A = \mathbf{PFCHK}^A(\log n)$.

Proof: Let ϕ_A be a relativized 3CNF formula as described in Section 2.1. A proof Π will consist of a satisfying assignment as well as a list of $x_{i1}, \ldots, x_{ik}$ for each $A(x_{i1}, \ldots, x_{ik})$ occurring in a clause. Such a proof can be universally verified in $O(\log n)$ time using the oracle access mechanism described above.

Let $L \in \mathbf{NP}^A$. By Lemma 2.1 there is an unrelativized reduction f mapping an input x to some relativized 3CNF formula ϕ_A. Part of Π will contain the formula ϕ_A as well as as proof that $\phi_A = f(x)$. $\square$

5.3 Algebraic Oracles

Babai and Fortnow [BF91] give an algebraic characterization of various complexity classes and argue that the interactive proof take advantage of this characterization. This algebraic characterization also does not seem to relativize.

In this section, we will give additional evidence that it is the algebraic properties of complexity classes that prevent the relativization of interactive proof results.

Let A be any function mapping $\{0,1\}^*$ to $\{0,1\}$. Let A_n be that function restricted to $\{0,1\}^n$. Let f_n to be the unique multilinear extension to A_n.

Let $\langle y_1,\ldots,y_k\rangle$ be a standard pairing function such that $|\langle y_1,\ldots,y_k\rangle| > |y_1|+|y_2|+|y_3|+\cdots+|y_k|$.

For any set $L \subseteq \Sigma^*$ we define the *algebraic* extension A of L inductively in n as follows:

1. Let $f(x_1,\ldots,x_n)$ be the unique multilinear extension of $A(x_1\ldots x_n)$.
2. Let $\langle 0,y_1,\ldots,y_n\rangle$ be in A iff $y_1\ldots y_n$ is in L.
3. Let $\langle 1,x_1,\ldots,x_n\rangle$ be in A if $f(x_1,\ldots,x_n) > 0$.
4. Let $\langle i+2,x_1,\ldots,x_n\rangle$ be in A if the ith bit of $f(x_1,\ldots,x_n)$ is one.

Note that L is many-one reducible to A.

Lund, Fortnow, Karloff and Nisan [LFKN92] and Shamir [Sha92] show that every language in **PSPACE** has an interactive proof. This result does not relativize, Fortnow and Sipser [FS88] show that relative to some oracle A, even co-**NP** does not have interactive proofs. However, the **IP** = **PSPACE** result does hold for algebraic extensions:

Theorem 5.6 *For A an algebraic extension for any set $L \subseteq \Sigma^*$, $\mathbf{IP}^A = \mathbf{PSPACE}^A$.*

Proof Sketch:

Instead of repeating the entire proof in [Sha92], we will just describe how to modify it.

We use the relativized formulae described in Section 2.1. We arithmetized the formulas in the same way as in [Sha92] replacing $A(x_{j1},\ldots,x_{jk})$ with $f(x_{j1},\ldots,x_{jk})$. Thus the arithmetized degree remains low as required. At the end of the protocol the verifier can read off the values of f using the encoding in the oracle A. □

From Theorem 5.6 we immediately have the have the following corollary:

Corollary 5.7 *For any set L there is an oracle A such that*

1. *L is many-one reducible to A and*
2. *$\mathbf{IP}^A = \mathbf{PSPACE}^A$.*

5.4 Further Questions about Interactive Proofs

We would like to see the ideas of Section 5.3 applied to other classes based on the interactive proof system such as multiple prover interactive proof systems and probabilistically checkable proofs. This may lead to even more evidence of an algebraic property that is the main cause of the nonrelativizing nature of these results.

However, given what we have seen in Section 5.2, we may question the usual oracle access mechanisms used in these models. We think the oracle access mechanism used in $\mathbf{PCP}(\log n, 1)$

works fine because the verifier is allowed to run in polynomial time. However, time may prove us wrong.

The access mechanism used by Fortnow, Rompel and Sipser [FRS88] is almost surely the wrong one. If one thinks about multiple-provers as $\cup_{j>0}\mathbf{PCP}(n^j, n^j)$ then we see the proof might have exponential size and thus describe exponential strings in the oracle. It is not clear how to extend the access mechanism. One simple but workable suggestion is to have the verifier run in exponential time.

It should be noted however that even if the verifier is given access to exponentially long strings in the oracle, there will still be an oracle A such that co-$\mathbf{NP}^A \not\subseteq \mathbf{MIP}^A$. We can easily modify the proof of Fortnow, Rompel and Sipser [FRS88] so all the diagonalizations occur on exponentially distant lengths with the oracle empty in between.

Because of the difficulty in creating an oracle A such that $\mathbf{PCP}(\log n, 1)^A = \mathbf{EXP}^A$, we should continue to try to show that $\mathbf{PCP}(\log n, 1) \neq \mathbf{EXP}$ and thus $\mathbf{NP} \neq \mathbf{EXP}$. We should also see if some different assumption, like a suitably strong one-way function, would allow us to find an oracle relative to which $\mathbf{PCP}(\log n, 1)^A = \mathbf{EXP}^A$.

6 Random and Generic Oracles

In 1981, Bennet and Gill [BG81] looked at what happens when we choose the oracles randomly: decide for each string whether or not it should be in the oracle independently with probability one-half. We say a statement S *holds with probability one* if the set of oracles relative to which S holds has measure one. From measure theory we have a wonderful zero-one law: for virtually any complexity theory statement S, we know that S holds with either probability zero or probability one.

Bennet and Gill conjectured the *random oracle hypothesis* roughly stated as "if a complexity statement holds with a random oracle with probability one then it holds in the unrelativized world". The random oracle hypothesis has great appeal. Intuitively it seems right: We ought to be able to simulate a random oracle with a suitably strong pseudorandom number generator.

Kurtz [Kur83] showed that the formulation of the random oracle hypothesis given by Bennet and Gill was false. Other counterexamples come from the area of interactive proofs [CGH90, HCRR90]. Despite these examples, many complexity theorists still believe that some version of the random oracle hypothesis. We however would like to argue that we should not have ever believed the random oracle hypothesis in the first place.

Kolmogorov complexity tells us that a "random" set R will contain lots of information. It is true that for a fixed set L, looking at a random R may not greatly affect the complexity of L. However the theory of random oracles works in the other direction. First we fix the set R. Then we can look at a language L designed to take advantage of the information in R. For example, Bennet and Gill [BG81] create a language $L \in \mathbf{NP}^R - \mathbf{P}^R$ that takes advantage of the fact that some of the information in R can be accessed nondeterministically but not deterministically.

Since an empty oracle does not contain any information, there is no reason to believe that results about random oracles should carry over to the unrelativized world. If we know that a statement S holds with probability one, we should not infer anything about the statement S other than what we can infer from the fact that there exists an oracle where S holds (see Section 3).

We know that all the definable statements that hold with probability one all hold simultaneously with probability one. Thus random oracles give us a nice relativized world where several interesting complexity theory statements all hold at the same time. However, a much more powerful tool for such purposes is the theory of generic oracles.

Generic oracles allow us to combine different oracle requirements in a clean manner. They give us very powerful tools in developing oracles. Fenner, Fortnow and Kurtz [FFK92] use generic oracles to develop a relativized world where the isomorphism conjecture holds, answering a long-standing open question. Space limitations prevent us from giving more details about generic oracles here but for the interested reader we recommend [BI87, FFKL93, FR93].

7 Conclusions and Other Questions

Hopefully, this paper will convince the reader of the many and varied uses of relativization results if done properly. The area of relativization remains a very important and active area of complexity theory. We caution researchers in the area though to keep in mind the limitations mentioned in Sections 4 and 5. Also, theorists must remember that oracles results are a tool. Theorems strictly about the structure of oracles should be discouraged.

Although we do not know how to settle many important complexity theory statements, the opposite is true in relativized worlds. For most important complexity theory statements S, we either know how to prove S or show that S does not hold in some relativized world. We thus would like to end this paper with two interesting exceptions:

1. Does $\mathbf{P} = \mathbf{UP}$ and $\mathbf{NP} = \text{co-}\mathbf{NP}$ imply that $\mathbf{P} = \mathbf{NP}$? (See [HS92])
2. Does the isomorphism conjecture imply that there are no one-way functions? (See [FFK92])

Acknowledgments

This paper grew out of an informal debate with Russell Impagliazzo on relativization results held at the Eighth Annual Structures in Complexity Theory Conference that was part of the Federated Computing Research Conference in San Diego in May, 1993. I thank the Structures program and conference committees, particularly Steve Homer, in organizing the debate. I would also like to thank Russell Impagliazzo for interesting discussion before and during the debate.

I have also had several interesting discussion about oracles and proof checking with many people including Sanjeev Arora, László Babai, Richard Beigel, Joan Feigenbaum, Stuart Kurtz, Carsten Lund, Muli Safra and Mike Sipser. Stuart Kurtz was particularly helpful in Section 5.2. The author based Section 3.2 on discussions with Mike Sipser.

References

[AIV92] S. Arora, R. Impagliazzo, and U. Vazirani. Relativizing versus nonrelativizing techniques: The role of local checkability. Manuscript, University of California, Berkeley, 1992.

[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.

[AS92] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. In *Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science*, pages 2–13. IEEE, New York, 1992.

[Bab85] L. Babai. Trading group theory for randomness. In *Proceedings of the 17th ACM Symposium on the Theory of Computing*, pages 421–429. ACM, New York, 1985.

[Bab90] L. Babai. E-mail and the unexpected power of interaction. In *Proceedings of the 5th IEEE Structure in Complexity Theory Conference*, pages 30–44. IEEE, New York, 1990.

[BF91] L. Babai and L. Fortnow. Arithmetization: A new method in structural complexity theory. *Computational Complexity*, 1(1):41–66, 1991.

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

[BFLS91] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd ACM Symposium on the Theory of Computing*, pages 21–31. ACM, New York, 1991.

[BG81] C. Bennet and J. Gill. Relative to a random oracle, $P^A \neq NP^A \neq co-NP^A$ with probability one. *SIAM Journal on Computing*, 10:96–113, 1981.

[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.

[BI87] M. Blum and R. Impagliazzo. Generic oracles and oracle classes. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pages 118–126. IEEE, New York, 1987.

[BM88] L. Babai and S. Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988.

[BRS91] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. In *Proceedings of the 23rd ACM Symposium on the Theory of Computing*, pages 1–9. ACM, New York, 1991.

[Bus88] J. Buss. Relativized alternation and space-bounded computations. *Journal of Computer and System Sciences*, 36(3):351–378, 1988.

[CGH90] B. Chor, O. Goldreich, and J. Håstad. The random oracle hypothesis is false. Manuscript, Technion, Haifa, Israel, 1990.

[Cha90] R. Chang. An example of a theorem that has contradictory relativizations and a diagonalization proof. *Bulletin of the European Association for Theoretical Computer Science*, 42:172–173, October 1990.

[Coo71] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd ACM Symposium on the Theory of Computing*, pages 151–158. ACM, New York, 1971.

[FFK92] S. Fenner, L. Fortnow, and S. Kurtz. The isomorphism conjecture holds relative to an oracle. In *Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science*, pages 30–39. IEEE, New York, 1992.

[FFKL93] S. Fenner, L. Fortnow, S. Kurtz, and L. Li. An oracle builder's toolkit. In *Proceedings of the 8th IEEE Structure in Complexity Theory Conference*, pages 120–131. IEEE, New York, 1993.

[FR93] L. Fortnow and J. Rogers. Separability and one-way functions. Technical Report CS 93-14, University of Chicago Department of Computer Science, 1993.

[FRS88] L. Fortnow, J. Rompel, and M. Sipser. On the power of multi-prover interactive protocols. In *Proceedings of the 3rd IEEE Structure in Complexity Theory Conference*, pages 156–161. IEEE, New York, 1988.

[FS88] L. Fortnow and M. Sipser. Are there interactive protocols for co-NP languages? *Information Processing Letters*, 28:249–251, 1988.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and intractability. A Guide to the theory of NP-completeness.* W. H. Freeman and Company, New York, 1979.

[GJ86] J. Goldsmith and D. Joseph. Three results on the polynomial isomorphism of complete sets. In *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pages 390–397. IEEE, New York, 1986.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[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, Greenwich, 1989.

[Har78] J. Hartmanis. *Feasible Computations and Provable Complexity Properties*, volume 30 of *CBMS-NSF Regional Conference Series in Mathematics.* Society for Industrial and Applied Mathematics, Philadelphia, 1978.

[Har85] J. Hartmanis. Solvable problems with conflicting relativizations. *Bulletin of the European Association for Theoretical Computer Science*, 27:40–49, October 1985.

[Hås89] J. Håstad. Almost optimal lower bounds for small depth circuits. In S. Micali, editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 143–170. JAI Press, Greenwich, 1989.

[HCKM88] J. Hartmanis, R. Chang, J. Kadin, and S. Mitchell. Some observations about relativizations of space bounded computations. *Bulletin of the European Association for Theoretical Computer Science*, 35:82–92, June 1988.

[HCRR90] J. Hartmanis, R. Chang, D. Ranjan, and P. Rohatgi. Structural complexity theory: Recent surprises. In *SWAT 90: 2nd Scandinavian Workshop on Algorithm Theory*, volume 447 of *Lecture Notes in Computer Science*, pages 1–12. Springer, Berlin, 1990.

[Hel81] H. Heller. *Relativized polynomial hierarchy extending two levels*. PhD thesis, Universität München, 1981.

[HS65] J. Hartmanis and R. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[HS92] S. Homer and A. Selman. Oracles for structural properties: The isomorphism problem and public-key cryptography. *Journal of Computer and System Sciences*, 44(2):287–301, 1992.

[HU79] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, Mass., 1979.

[Kur83] S. Kurtz. On the random oracle hypothesis. *Information and Control*, 57(1):40–47, April 1983.

[Lev73] L. Levin. Universal'nyĭe perebornyĭe zadachi (Universal search problems: in Russian). *Problemy Peredachi Informatsii*, 9(3):265–266, 1973. Corrected English translation in [Tra84].

[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.

[LL76] R. Ladner and N. Lynch. Relativization of questions about log-space reducibility. *Mathematical Systems Theory*, 10:19–32, 1976.

[Mor81] S. Moran. Some results on relativized deterministic and nondeterministic time hierarchies. *Journal of Computer and System Sciences*, 22:1–8, 1981.

[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(5):865–877, 1991.

[Tra84] R. Trakhtenbrot. A survey of Russian approaches to *Perebor* (brute-force search) algorithms. *Annals of the History of Computing*, 6(4):384–400, 1984.

[Wil85] C. Wilson. Relativized circuit complexity. *Journal of Computer and System Sciences*, 31:169–181, 1985.