Oracles versus Proof Techniques that Do Not Relativize[1]

Eric Allender[2]
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

ABSTRACT *Oracle constructions have long been used to provide evidence that certain questions in complexity theory cannot be resolved using the usual techniques of simulation and diagonalization. However, the existence of nonrelativizing proof techniques seems to call this practice into question. This paper reviews the status of nonrelativizing proof techniques, and argues that many oracle constructions still yield valuable information about problems in complexity theory.*

1 Introduction

One of the most exciting theorems of this past winter was proved by Adi Shamir in [Sh-89], where it is shown that PSPACE is the class of sets having interactive proof systems. This result is significant for many reasons, but this paper will focus on one aspect of this work: it does not relativize. The results of [Sh-89], along with the related work of [LFKN-89, BFL-90], are the first truly compelling examples of theorems about complexity classes that are known to be true in the unrelativized case, but are false relative to some oracles.

Of course, other examples of nonrelativizing proof techniques have been known for quite some time. These earlier results are surveyed in Section 2.

The survey will be followed in Section 3 by a discussion about what is novel in the proof techniques of [Sh-89, LFKN-89, BFL-90], and how they differ from the nonrelativization results surveyed in Section 2.

In Section 4, we discuss what significance, if any, should be attached to oracle results, in light of the fact that nonrelativizing proof techniques now seem to be available. In particular, arguments will be presented explaining why some oracle results (including some of the author's own results) should still be of interest.

[1]This was an invited paper at the SIGAL International Symposium on Algorithms, August 16-18, 1990, in Tokyo.
[2]Supported in part by National Science Foundation grants CCR-8810467 and CCR-9000045, and by The International Information Science Foundation under grant number 90 1 3 227.

Finally, we conclude with a brief discussion and summary in Section 5.

2 A Long History of Nonrelativizing Proofs

Oracle constructions have never been very useful as a tool for talking about possible relationships among space complexity classes, for the simple reason that there is no obvious "right" way to talk about a space-bounded oracle Turing machine. If the oracle tape is subjected to a sublinear space bound, then there are sets that are not even reducible to themselves (because the input can't be written on the oracle tape); this is clearly an anomalous situation. On the other hand, if the oracle tape is not subjected to any space bound, then extremely long queries can be posed to the oracle, which can also lead to anomalies because it represents a way in which the space bound can be circumvented. For instance, there are oracles relative to which NLOG is not contained in P, using this notion of relativization.

These anomalies were first described in [LL-76, L-78], and it was suggested that the proof technique used to prove NLOG $\subseteq$ P fails to relativize because the proof does not proceed via a "step-by-step" simulation. However, an alternative view is that NLOG $\subseteq$ P fails to relativize simply because the wrong notion of relativization is used. Thus a new notion of relativization for space-bounded Turing machines (sometimes called RST relativization) was proposed in [RST-84]; RST relativization avoids at least the most obvious anomalies associated with the other methods of relativization studied in [LL-76, L-78, Sa-83].

Space-bounded AuxPDAs provide an alternative characterization of deterministic time classes, and these characterizations also fail to hold relative to oracles [An-80]. Again, the problem may be blamed on ambiguities about what is the "correct" way to equip a space-bounded machine with access to an oracle.

Time-bounded alternating Turing machines and depth-bounded circuit complexity classes provide useful characterizations of space-bounded computation, and it is proved in [Or-83, Wi-87] that all of the natural methods of relativizing these modes of computation result in oracles relative to which classes that are equal in the unrelativized case are unequal relative to the oracles. It may be argued that these characterizations thus do not relativize; but many researchers are of the opinion that the real issue centers on the question of what is the "correct" way to provide machines such as alternating Turing machines with access to the oracle. Alternative oracle access mechanisms have been presented for space-bounded Turing machines [Wi-88] and alternating Turing machines

[Bu-88] that have the virtue of allowing most of the standard characterizations to hold relative to all oracles. A more general mechanism along the same lines was presented in [Bu-87].

The virtues and drawbacks of the various oracle access mechanisms have been debated in several different settings in a number of papers. In addition to the work mentioned above, the interested reader will find these issues discussed in [Si-77, Ha-88, Ha-88a, KL-87, AW-90a]. Let it suffice to say that it is far from clear that the results discussed so far in this section really represent "non-relativizing" proof techniques. It is simply not clear what "relativization" should really mean in this setting.

For time-bounded computation by deterministic and nondeterministic Turing machines, the picture is much less ambiguous. Yet even in this setting there are theorems that have been proved true in the unrelativized case, yet are false relative to some oracles. The simplest example is the linear speed-up theorem; $\mathrm{DTIME}(2n) = \mathrm{DTIME}(4n)$, but an oracle Turing machine running for $4n$ steps can ask queries that a machine running in time $2n$ cannot ask, and this intuition can be turned into a proof that the linear speed-up theorem is false relative to some oracle [RS-81, Mo-81]. A less trivial example is provided by the proof that $\mathrm{DTIME}(n \log^* n) \subseteq \Sigma_2\mathrm{Time}(n)$ [PPST-83]. It is shown in [Ga-87] that there are oracles relative to which this inclusion fails to hold.

However these two examples fail to be very compelling, since both the linear speed-up theorem and the result of [PPST-83] are in some sense automata-theoretic results about the Turing machine model. Certainly the linear speed-up theorem is false in many reasonable models of computation, and the result of [PPST-83] is known only for models of computation that are very similar to the Turing machine model. The questions of complexity theory that are of the most interest are those questions that are independent of the details of the particular models of computation that have been chosen for study; the complexity classes that are of interest are the complexity classes that are invariant under minor changes in the definition. The complexity classes $\mathrm{DTIME}(4n)$ and $\mathrm{DTIME}(n \log^* n)$ are very sensitive to small changes to the underlying model of computation, and thus the nonrelativizing proofs dealing with these complexity classes seem to be of little help in formulating attacks on the more important questions.

Hartmanis wrote an interesting article [Ha-85] in which a number of nonrelativizing proofs are presented. Some of these results have the flavor of the results cited at the start of this section, in which space-bounded machines are given access to an oracle. However, other results in [Ha-85] involve a novel "double-relativization" trick. For example, take an oracle A such that $\mathrm{P}^A = \mathrm{NP}^A$. Then there is an oracle B such that $\mathrm{P}^{A,B} \neq \mathrm{NP}^{A,B}$;

that is, the "real-world" result that $P^A = NP^A$ does not relativize. Although this is thought-provoking, it is hard to see how theorems of this sort can lead to the solution of any of the long-standing problems of complexity theory.

Other nonrelativizing proofs are discussed in [DGHM-89] and [IR-89,I-88]. However, the results mentioned in [IR-89,I-88] involve the concept of zero-knowledge, which conceivably should be defined differently in relativized models of computation, and the results of [DGHM-89] involve nonrecursive sets, and it is not clear how to apply those techniques to prove nonrelativizing results about familiar complexity classes.

Even though the proof techniques discussed in this section are all "nonrelativizing" in one way or another, the techniques all are "traditional" in some vague sense. In contrast, there is something fundamentally new about the proofs presented in [Sh-89, LFKN-89, BFL-90]. The next section will focus on one aspect of these proofs that is particularly novel.

3 New Nonrelativizing Proof Techniques

This section begins with a review of the assumptions underlying the use of relativizations as a tool for indicating that certain complexity-theoretic problems are "difficult" in some sense. Then we will show how the proofs of [Sh-89, LFKN-89, BFL-90] violate these assumptions.

Many of the basic results of recursive function theory and complexity theory are based on the simple intuition that it is impossible to say anything about the behavior of a machine other than by simulating the machine. For example, one cannot determine if a machine will halt except by running it until it halts; one cannot determine if a $t(n)$ time-bounded machine will accept or reject its input except by running it for $t(n)$ steps, etc. Of course, the same intuition is also correct when one is trying to predict the behavior of a nondeterministic Turing machine (NTM), using a deterministic Turing machine: it is impossible to determine anything about the acceptance behavior of the NTM without carrying out some sort of simulation. The question of course is: what kinds of simulation are possible?

The P=NP question is really the question of whether or not it is possible, given a configuration of an NTM, to determine anything about the number of accepting paths in the tree, without doing a brute-force search of all the paths in the tree. The conventional intuition is that it is *not* possible to do significantly better than this, and relativization

is one way of providing a setting in which this intuition is correct.

A configuration of a NTM is a small object that completely describes a much larger object: the computation tree rooted at that configuration. The results of [BGS-75] were the first to capitalize on the fact that relative to some oracles, the computation tree *cannot* be described compactly, and thus no deterministic strategy for inferring information about the tree can succeed, unless it involves evaluating essentially every branch in the tree. That is, the computation tree is no longer described by its root, as in the unrelativized case; instead it can only be described by the values at each of its leaves.

The proof of [FS-88] is a conventional oracle construction in this regard. In [FS-88], it is shown that, relative to some oracle, coNP is not in IP. (That is, there is a set in coNP for which membership cannot be proved via an interactive proof system.) The coNP language considered is $L(A) = \{1^n : \text{every string of length } n \text{ is in } A\}$, where A is the oracle. The intuition used in [FS-88] is quite simple: if a probabilistic machine communicating with a prover *cannot* accept 1^n with acceptably high probability relative to *any* oracle, then the machine is clearly not usable in an interactive protocol for $L(A)$. Otherwise, if a machine *can* accept 1^n with high enough probability relative to some prover and some oracle, then there has to be some string of length n that is queried on a minority of the paths. Removing that string from the oracle cannot change the probability of acceptance very much; thus in this case it cannot reject 1^n with high enough probability, and thus in this case also it does not define an interactive protocol for $L(A)$.

This relativization argument of [FS-88] is entirely conventional; there is nothing controversial about the method in which access to the oracle is provided. It is difficult to see how any of the objections raised to the "anomalous" relativizations surveyed in Section 2 could be raised here.

And yet, in the unrelativized case, coNP *is* contained in IP. The original proof of this fact, in [LFKN-89], in some sense provides a method of inferring information about the structure of a computation tree *without* doing a brute-force search through the tree. Instead, the configuration describing the tree is manipulated as an object with arithmetic properties that can be efficiently verified in a particular sense.[3]

Let us provide a few more details. (The brief sketch provided here is based on one of

[3]Just as the oracle of [FS-88] shows that the results of [Sh-89, LFKN-89] do not relativize, there is an oracle construction in [FRS-88] that shows that the characterization of NEXP in terms of two-prover interactive proof systems [BFL-90] does not relativize.

the proofs in [BF-90].) The standard reduction given by Cook's theorem [Co-71] shows that any computation tree on inputs of length n can be described with a polynomial-sized instance of 3SAT. Replace each of the disjunctions in this 3SAT instance by an equivalent disjunction of 7 conjunctions on the same variables. Now treat this resulting formula as an arithmetic polynomial in m variables, where AND and OR are treated as multiplication and addition, respectively, and $\overline{x}$ is replaced by $1 - x$. As is observed in [BF-90], when this expression is summed over all of the 2^m possible assignments of the variables to $\{0, 1\}$, the resulting number is simply the number of accepting paths in the original computation tree.

An interactive proof protocol is presented in [LFKN-89] that allows a prover to convince a verifier of the value of such an expression. In an oversimplified form, the protocol is as follows. Let the original polynomial in m variables be $p(x_1, \ldots, x_m)$. In the first round, the prover sends the verifier a number v_1 and claims that this number is the value of

$$\sum_{x_1=0}^{1} x_1 \ldots \sum_{x_m=0}^{1} x_m \; p(x_1, \ldots, x_m).$$

The prover also sends the verifier the coefficients of a polynomial q', claiming that this represents the polynomial

$$q(x_1) = \sum_{x_2=0}^{1} x_2 \ldots \sum_{x_m=0}^{1} x_m \; p(x_1, \ldots, x_m).$$

Up to this point, the proof is not so different from several other "relativizing" proofs; all of these polynomials have a straightfoward interpretation in terms of more standard machine-based proofs. However what happens next is a significant break with the past.

The verifier can easily check to see if $q'(0) + q'(1) = v_1$, but the verifier must also have some way to check that the prover is not lying about the claim that $q' = q$. The significant insight of [LFKN-89] is that if the prover is lying, then for a randomly chosen integer r in a given range, $q'(r) \neq q(r)$. Thus the verifier picks r, and asks the prover to prove that

$$q'(r) = \sum_{x_2=0}^{1} x_2 \ldots \sum_{x_m=0}^{1} x_m \; p(r, x_2 \ldots, x_m).$$

This polynomial has one less variable. After m rounds, the verifier will, with high probability, be able to see if the prover has been lying or not. (The reader is referred to [LFKN-89] to see the complete proof.)

The reason that this represents a break with more traditional methods is that the polynomial $p(r, x_2 \ldots, x_m)$ now bears no obvious relation to the original Turing machine

whose tree we were trying to evaluate. In fact, the relativization result of [FS-88] implies that, in some sense, there *is* no obvious way to relate $p(r, x_2 \ldots, x_m)$ to this machine; if there were, it would allow us an avenue for making this proof hold relative to all oracles, and this is impossible by [FS-88].

The polynomial p is an arithmetic object that encodes, in a succinct way, all of the information needed to completely evaluate the computation tree of our original NTM. The arithmetic properties of this polynomial can be exploited in the framework of interactive proof systems. This contrasts with the relativization result of [FS-88], which seems to indicate that any evaluation of the computation tree that considers entire paths (e.g. to look at a sequence of queries) cannot be accomplished in polynomial time by interactive proof protocols.

It may be argued that this nonrelativizing proof simply reinforces the intuition of [LL-76] that proofs that proceed via a step-by-step simulation relativize, while proofs that do not proceed in this way do not relativize. However, all earlier examples of such proofs failed to be convincing examples of nonrelativization, because of ambiguity about what is the "correct" way to relativize those classes. (Indeed, as the results of [Wi-88, Bu-88, Bu-87] indicate, it *is* possible to define notions of relativization that defeat those early attempts at nonrelativizing proofs.) The oracle construction of [FS-88] does not seem to suffer from this ambiguity.

The [LFKN-89] technique of evaluating computation trees does not seem to translate to any mode of computation other than interactive proofs. It will be interesting to see if other techniques for deriving information about computation trees shed any light on the DTIME versus NTIME question.

4 The Role of Oracle Results

What place do oracle constructions have in a world with nonrelativizing proof techniques? "None" is perhaps the first answer that springs to mind, but there are in fact a number of situations in which oracle constructions provide a great deal of useful information. In this section we shall consider a few of these situations.

4.1 Properties of Complexity Classes

Many complexity classes are *defined* in terms of oracle computations; thus it is sometimes important to know what properties hold relative to all oracles, and which properties do

not.

For example, in [AG-90], it is shown that there are sets in P^{PP} that are immune to uniform AC^0. (An infinite set is *immune* to a class $\mathcal{C}$ if it has no infinite subset in $\mathcal{C}$.) The proof proceeds by first showing that there is a set $L \in \mathrm{PP}$ and a constant k such that $\mathrm{AC}^0 \subseteq \mathrm{DTIME}(n^k)^L$; then we note that there is a set in $\mathrm{DTIME}(n^{k+1})^L$ that is immune to $\mathrm{DTIME}(n^k)^L$, and hence to AC^0. Thus in this instance, it was important that the immunity properties of DTIME hold relative to all oracles.

On the other hand, what if the class under consideration were $\mathrm{NP}^{\mathrm{PP}}$ instead of P^{PP}? Would similar observations apply? Note that a great many complexity classes of interest are *defined* in terms of relativized computations. (The classes of the polynomial hierarchy are the most obvious examples.) When investigating the structure of these classes, it is thus useful to know, for example, if $\mathrm{NTIME}(n^{k+1})^L$ will always contain a set immune to $\mathrm{NTIME}(n^k)^L$.

Questions such as these were investigated in [ABHH-90]. There it was shown that, unless $T(n)$ is almost exponentially larger than $t(n)$, then relative to some oracles L, $\mathrm{NTIME}(T(n))^L$ does *not* contain any sets immune to $\mathrm{NTIME}(t(n))^L$. Other immunity properties of NTIME classes were also investigated in [ABHH-90] in connection with the almost-everywhere hierarchy for nondeterministic time. As illustrated above, such results are often useful when studying complexity classes that are defined in terms of relativized computation.

4.2 Oracles as Circuit Lower Bounds

Some of the most impressive lower bound results in complexity theory are lower bounds for constant depth circuits. This line of research was begun by [FSS-84], and at least some of the motivation for these results came from the connection between constant depth circuits and the polynomial time hierarchy; it was shown in [FSS-84, Si-83] that circuit lower bounds for constant depth circuits translate directly into oracle constructions for the polynomial hierarchy, and vice-versa. This connection was generalized by Torán [Tor-89].

Thus in some sense there is little difference between a circuit lower bound and an oracle construction. For instance, Torán has studied the counting hierarchy (consisting of classes such as PP, $\mathrm{PP}^{\mathrm{PP}}$, etc.) and has constructed oracles separating some sublevels of this hierarchy [Tor-89]. He then defines a number of subclasses of NC^1 and shows that, via his oracle constructions, the corresponding subclasses of NC^1 are also distinct.

If an oracle could be found such that all of the levels PP, $\mathrm{PP}^{\mathrm{PP}}$, ... of the counting hierarchy are distinct, then this would answer some important open questions about threshold circuits. This connection between circuits and the counting hierarchy is discussed further in the survey [AW-90].

4.3 Oracles in Cryptography

The security of a cryptographic protocol is usually predicated on some complexity-theoretic assumption, such as "one-way functions exist" or "trap-door functions exist". Since it is always desirable to have as weak an assumption as possible, an ongoing theme in cryptographic research is to show new implications and relationships among cryptographic assumptions.

Oracle constructions have been used as a tool to investigate when certain assumptions are *not* sufficient for certain tasks. One of the first examples of this sort of work is [IR-89]. Since many cryptographic constructions use a one-way function as a sort of "black box," an oracle-based approach has obvious applications here.

4.4 Oracles as Indicators

Oracles can also be used as a tool to find questions where progress is still likely to be made. For example, the relationship between PP and the polynomial hierarchy has been a wide-open question for years, and one reason for this is that there are so few oracle results concerning possible relationships between these classes. In the wake of Toda's breakthrough [To-89] in showing that the polynomial hierarchy is contained in P^{PP}, there is even more interest in knowing, for example, whether P^{NP} is contained in PP. Certainly, construction of an oracle relative to which P^{NP} is not contained in PP would be viewed as a very interesting result.

Another example is provided by the Berman-Hartmanis conjecture concerning whether or not all NP-complete sets are p-isomorphic. In spite of all of the work that has been done on this question it still not known if it is "possible" for the conjecture to be true, in the sense that no oracle is known relative to which all NP-complete sets are p-isomorphic. (An oracle relative to which all complete sets for Σ_2^p are p-isomorphic is presented in [HS-89].) In the absence of such an oracle, there is good reason to devote some energy to trying to prove the existence, in the unrelativized case, of non-p-isomorphic NP-complete sets. On the other hand, if someone does construct an oracle relative to which the Berman-Hartmanis conjecture holds, then it would have a substantial impact upon all subsequent

approaches to this problem; such an oracle construction would imply that an entire repertory of proof techniques would have to be abandoned in order for further progress to be made.

4.5 Oracles as Guides to Nonrelativizing Techniques

As was mentioned above, there are still no nonrelativizing proof techniques that are suitable for attacking the NTIME versus DTIME question (given that the techniques of [PPST-83] do not seem to be able to provide very significant separations). On the other hand, significant lower bounds have been proved with respect to constant-depth circuits [Ya-85, Hå-86]. A recent oracle construction of [AG-90] seems to indicate that further progress with constant-depth circuits will provide nonrelativizing proof techniques that will be useful in settling the relationship between DTIME and NTIME classes.

More specifically, recall that it was mentioned above that there are sets in P^{PP} that are immune to uniform AC^0. Since P^{PP} is such a "large" complexity class, and AC^0 does not even contain simple sets such as Parity, this may seem like a very weak result. However, if it can be shown that there are sets in NP that are immune to AC^0, then there are statements concerning NE, E, and $\bigcup_k \Sigma_k$-time(n) that are true in the unrelativized world, but are false relative to some oracle.

At the very least, the oracle result of [AG-90] points to certain limitations of relativizing proof techniques when it comes to proving the existence of sets that are immune to AC^0. However, given that the program of proving lower bounds for AC^0 circuits has been so successful, this oracle construction could provide an indication of problems to work on, in the hope of developing an approach for finding out more about the NTIME vs DTIME question.

5 Conclusions

The development of nonrelativizing proof techniques will certainly have a profound impact upon research in complexity theory. Although certain types of nonrelativizing proof techniques have been known for quite some time, various considerations caused the significance of these earlier nonrelativizations to be called into question. Those considerations do not come into play with regard to the proofs of [Sh-89, LFKN-89, BFL-90].

In spite of the existence of nonrelativizing proof techniques, there are still a number of situations in which oracle constructions can serve as a useful tool in proving interesting

results.

Acknowledgments: I thank Ron Book, Jack Lutz, Pekka Orponen, and Seinosuke Toda for some interesting discussions that took place at the Workshop in Structural Complexity Theory at the University of California, Santa Barbara, in March, 1990. Thanks are especially due to Ron Book for organizing the Workshop. I also thank Osamu Watanabe for arranging support through the International Information Science Foundation, and for his many other efforts.

References

[An-80] D. Angluin, *On relativizing auxiliary pushdown machines*, Math. Systems Theory 13, 283–299.

[ABHH-90] E. Allender, R. Beigel, U. Hertrampf, and S. Homer, *A note on the almost-everywhere hierarchy for nondeterministic time*, Proc. 7th Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science 415, pp. 1–11.

[AG-90] E. Allender and V. Gore, *On strong separations from AC^0*, DIMACS Technical Report 90-32.

[AW-90] E. Allender and K. W. Wagner, *Counting hierarchies: polynomial time and constant depth circuits*, guest authors of The Structural Complexity Column (ed. Juris Hartmanis), EATCS Bulletin 40, pp. 182–194.

[AW-90a] E. Allender and C. Wilson, *Width-bounded reducibility and binary search over complexity classes*, Proc. 5th Annual IEEE Structure in Complexity Theory Conference, 1990.

[BF-90] L. Babai and L. Fortnow, *A characterization of P by straight line programs of polynomials, with applications to interactive proofs and Toda's theorem*, Technical Report 90-02, University of Chicago.

[BFL-90] L. Babai, L. Fortnow, and C. Lund, *Non-deterministic exponential time has two-prover interactive protocols*, manuscript.

[BGS-75] T. Baker, J. Gill, and R. Solovay, *Relativizations of the P=?NP question*, SIAM J. Comput. 4, 431–442.

[Bu-87] J. Buss, *A theory of oracle machines*, Proc. 2nd IEEE Structure in Complexity Theory Conference, pp. 175–181.

[Bu-88] J. Buss, *Relativized alternation and space-bounded computation*, J. Comput. and System Sci. 36, 351–378.

[Co-71] S. Cook, *The complexity of theorem proving procedures*, Proc. 3rd Annual ACM Symposium on Theory of Computing, pp. 151–158.

[DGHM-89] R. Downey, W. Gasarch, S. Homer, and M. Moses, *On honest polynomial reductions, relativizations, and P=NP*, Proc. 4th IEEE Structure in Complexity Theory Conference, pp. 196–207.

[FS-88] L. Fortnow and M. Sipser, *Are there interactive proofs for co-NP languages?* Information Processing Letters 28, 249–251.

[FSS-84] M. Furst, J. Saxe, M. Sipser, *Parity, circuits, and the polynomial-time hierarchy*, Mathematical Systems Theory 17, 13–27.

[FRS-88] L. Fortnow, J. Rompel, and M. Sipser, *On the power of multi-prover interactive protocols*, Proc. 3rd IEEE Structure in Complexity Theory Conference, pp. 156–161.

[Ga-87] W. Gasarch, *Oracles for deterministic versus alternating classes*, SIAM J. Comput. 16, 613–627.

[Hå-86] J. Håstad, *Almost optimal lower bounds for small depth circuits*, Proc. 18th ACM Symposium on Theory of Computing, pp. 6–20.

[Ha-85] J. Hartmanis, *Solvable problems with conflicting relativizations*, EATCS Bulletin 27, pp. 40–49.

[Ha-88] J. Hartmanis, *New developments in structural complexity theory*, Proc. 15th International Colloquium on Automata, Languages, and Programming, Lecture Notes in Computer Science 317, pp. 271–286.

[Ha-88a] J. Hartmanis, *Some observations about relativizations of space bounded computations*, the Structural Complexity Column, EATCS Bulletin 35, pp. 82–92.

[HS-89] S. Homer and A. Selman, *Oracles for structural properties: the isomorphism problem and public-key cryptography*, Proc. 4th IEEE Structure in Complexity Theory Conference, pp. 3–14.

[I-88] R. Impagliazzo, *Proofs that relativize, and proofs that do not*, manuscript.

[IR-89] R. Impagliazzo and S. Rudich, *Limits on the provable consequences of one-way permutations*, Proc. 21st Annual ACM Symposium on Theory of Computing, pp. 44–61.

[KL-87] B. Kirsig and K.-J. Lange, *Separation with the Ruzzo, Simon, and Tompa relativization implies DSPACE[log n]* $\neq$ *NSPACE[log n]*, Information Processing Letters 25, 13–15.

[LFKN-89] C. Lund, L. Fortnow, H. Karloff, and N. Nisan, *The polynomial-time hierarchy has interactive proofs*, manuscript.

[LL-76] R. Ladner and N. Lynch, *Relativization of questions about log space computability*, Math. Systems Theory 10, 19–32.

[L-78] N. Lynch, *Log space machines with multiple oracle tapes*, Theoretical Computer Science 6, 25–39.

[Mo-81] S. Moran, *Some results on relativized deterministic and nondeterministic time hierarchies*, Journal Comput. System Sci. 22, 1–8.

[Or-83] P. Orponen, *Complexity classes of alternating machines with oracles*, Proc. 10th International Colloquium on Automata, Languages, and Programming, Lecture Notes in Computer Science 154, pp. 573–584.

[PPST-83] W. Paul, N. Pippenger, E. Szemerédi, and W. Trotter, *On determinism versus non-determinism and related problems*, Proc. 24th Annual IEEE Symposium on Foundations of Computer Science, pp. 429–438.

[RS-81] C. Rackoff and J. Seiferas, *Limitations on separating nondeterministic complexity classes*, SIAM J. Comput. 10, 742–745.

[RST-84] Walter Ruzzo, Janos Simon, and M. Tompa, *Space-bounded hierarchies and probabilistic computation*, J. Comput. and System Sci. 28, 216–230.

[Sa-83] W. Savitch, *A note on relativized log space*, Math. Systems Theory 16, 229–235.

[Sh-89] A. Shamir, *IP = PSPACE*, manuscript.

[Si-77] I. Simon, *On some subrecursive reducibilities*, Ph.D. Dissertation, Stanford University.

[Si-83] M. Sipser, *Borel sets and circuit complexity*, Proc. 15th ACM Symposium on Theory of Computing, pp. 61–69.

[To-89] S. Toda, *On the computational power of PP and* $\oplus P$, Proc. 30th IEEE Symposium on Foundations of Computer Science, pp. 514–519.

[Tor-89] J. Torán, *A combinatorial technique for separating counting complexity classes*, Proc. 16th ICALP, Lecture Notes in Computer Science 372, pp. 732–744.

[Wi-87] C. B. Wilson, *Relativized NC*, Math. Systems Theory 20, 13–29.

[Wi-88] C. B. Wilson, *A measure of relativized space which is faithful with respect to depth*, J. Comput. and System Sciences 36, 303–312.

[Ya-85] A. C. Yao, *Separating the polynomial-time hierarchy by oracles*, Proc. 26th IEEE Symposium on Foundations of Computer Science, pp. 1–10.