Relativization: a Revisionistic Retrospective*

Juris Hartmanis Richard Chang† Suresh Chari
Desh Ranjan‡ Pankaj Rohatgi

Department of Computer Science, Cornell University
Ithaca, New York 14853, USA

In this column we examine the role of relativization in complexity theory in light of recent non-relativizing results involving interactive protocols. We begin with the twice-told tale of the relativization principle and ponder upon its possible demise. Then, we discuss whether usual assumptions are historically accurate.

1 The Twice-Told Tale

For almost two decades, contradictory relativization has been a central theme in complexity theory. This concept was first introduced by Baker, Gill and Solovay [BGS75] in their ground breaking paper where they exhibited oracles A and B such that $\mathrm{P}^A = \mathrm{NP}^A$ and $\mathrm{P}^B \neq \mathrm{NP}^B$. This result was startling because almost all results in recursion theory remain true in the presence of oracles and most techniques used in complexity theory had been resource bounded versions of those used in recursion theory.

Baker, Gill and Solovay offered the following explanation of their results [BGS75]:

> We feel that this is further evidence of the difficulty of the $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ question. ...It seems unlikely that ordinary diagonalization methods are adequate for producing an example of a language in NP but not in P; such diagonalizations, we would expect, would apply equally well to the relativized classes. ...On the other hand, we do not feel that one can give a general method for simulating nondeterministic machines by deterministic machines in polynomial time, since such a method should apply as well to relativized machines.

*This work was supported in part by NSF Research Grant CCR 88-23053.

†Current Address: Department of Computer Science, University of Maryland, Baltimore County Campus, Baltimore, MD 21228, USA.

‡Current Address: Max Plank Institut für Informatik, Im Stadtwald, W 6600 Saarbrücken, Germany. On leave from Department of Computer Science, New Mexico State University, Las Cruces, NM 88003, USA.

Following this work, much effort was expended to find contradictory relativizations for other open problems in complexity theory. A major part of this effort revolved around obtaining relativized results about the polynomial time hierarchy and its relationship to other classes. In a seminal work, Furst, Saxe and Sipser [FSS84] linked these questions to the power of bounded depth boolean circuits. This led to the development of an entire subfield of circuit complexity devoted to understanding into the the power of such circuits. As a result of advances made in this area [Yao85, Ko89, Hås89] contradictory relativizations for various unsolved problems regarding the Polynomial Hierarchy were found. These and other results led to a strong belief that problems with contradictory relativization are very hard to solve and are not amenable to current proof techniques. This sentiment was best summarized by Hopcroft in the following "meta-theorem" [Hop84] :

> 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.

Some researchers were able to exhibit cases where diagonalization arguments could overcome the relativization barrier [Koz80, Cha90]. Others were able to overcome this barrier through double relativizations [Kur82, Har85]. However, these examples were considered highly artificial and did not refute the basic import of the meta-theorem above. Indeed, some effort was even made to identify open problems which cannot have contradictory relativizations, in the the hope that they were easier to solve [HCKM88]. Thus, as a cursory glance at the proceedings of any STRUCTURES conference will confirm, proving contradictory relativizations became a pervasive paradigm for showing the optimality of results up to "known techniques".

Many researchers also explored what relativized separations of complexity classes might imply about the relationship between these classes in the unrelativized case. From the beginning, it was observed that relativized separations usually exploit differences in the oracle access mechanisms. For instance, a polynomial time bounded oracle machine can only make polynomially queries, whereas an NP oracle machine can make exponentially many queries in its computation tree. The oracle constructed by Baker, Gill and Solovay exploits precisely this difference in the access mechanisms. Thus it is not clear whether relativized separation truly reflects the different computational power of the complexity classes or is just an artifact of the different oracle access mechanisms.

Motivated by this, Book, Long and Selman [BLS84, Boo87] proposed the theory of *positive relativizations*. They showed that once the oracle access mechanisms for different classes were made roughly comparable then a relativized separation would also separate the nonrelativized classes. For example, if nondeterministic oracle machines are restricted to asking only polynomially many questions in the entire computation tree, then a relativized separation of P and NP would imply P $\neq$ NP. Therefore, in this model, contradictory relativizations of P and NP cannot be achieved without also showing that P $\neq$ NP. This was perhaps the first real blow to the general relativization principle.

On the other hand, Bennett and Gill [BG81] proposed the *random oracle* approach to avoid oracles which are highly structured to exploit the differences in the access mechanism. A random oracle is formed by considering every possible string and including it in the oracle with probability $\frac{1}{2}$. Bennett and Gill conjectured that since a random oracle is intuitively unbiased and unstructured, relativized results which hold with probability 1 for a random oracle should also be true in the absence of an oracle. This conjecture was then formalized as the *Random Oracle Hypothesis*. Most of the results obtained along these lines, e.g., P $\neq$ NP $\neq$ co-NP, agreed with commonly held beliefs, an exception being BPP = P. The susceptibility of even this hypothesis to restricted oracle access mechanisms was demonstrated by Kurtz [Kur82] who provided some simple counterexamples. However, these results were not considered compelling enough to reject the hypothesis outright. In fact, Book [Boo90] showed that once the access mechanisms were made comparable then random oracle separation results would also separate the base classes.

Thus, for many years the relativization principle was a dominant research paradigm in complexity theory. Once contradictory relativizations for an open problem was found it was generally considered impossible to solve. Researchers learned to cope with this obviously unsatisfactory state of affairs until December 1989, when the famous results on the power of interactive protocols were announced by electronic mail[Bab90].

2 The Current State of Mathematics

In their ground breaking work, Lund, Karloff, Fortnow, and Nisan [LFKN90] showed that the Polynomial Hierarchy is contained in the class IP. This was immediately followed by the Shamir's result that IP = PSPACE [Sha90]. One of the remarkable aspects of these results was that a few years earlier Fortnow and Sipser [FS88] had constructed an oracle A such that co-NP$^A \not\subseteq$ IP^A. Thus it seemed that the contradictory relativization barrier was finally breached. Subsequently, it was shown that with probability 1, IP$^A \neq$ PSPACE^A for a random oracle A, and thus the IP = PSPACE result also convincingly refutes the random oracle hypothesis [HCRR90, CGH90].

Due to recent dramatic advances by Arora and Safra [AS92] and Arora *et al* [ALM$^+$92] we now have a natural example of two unequal classes which can be collapsed by an oracle. These new results are a culmination of a sequence of insights into probabilistic proof checking [BFL90, BFLS91, FGL$^+$91]. The concept of probabilistic proof checking arose from a characterization of Multi-Prover Interactive Proofs (MIP) due to Fortnow, Rompel and Sipser [FRS88, FRS90].

Intuitively, probabilistic proof checking is a generalization of the concept of NP. A language L is in NP if there is a deterministic polynomial time machine, M, such that if $x \in L$ then there is a proof of membership which M accepts, and if $x \notin L$, no "proof" will make M accept. One might relax this requirement by allowing proofs to be checked by a probabilistic machine which is correct with high probability. The real gain could be that the proof checker need not look at the entire proof and membership may be decidable with high probability by looking only at a small portion of the proof. Using this notion Arora and Safra define classes with probabilistically checkable proofs as follows:

Definition 1 [AS92, FGL+91] *A* verifier *M is a random polynomial time machine having two inputs x and Π. Π is a proof of membership of x in a language L and M has* random access *to any bit of Π. M is an $r(n)$ restricted verifier if on input x of size n it is allowed to use at most $r(n)$ random bits and can query at most $r(n)$ bits of the proof Π.*

Definition 2 [AS92] *A language L is in* PCP$(r(n))$ *if there exists an $r(n)$ restricted verifier M such that for every input x:*

1. *If $x \in L$ there exists a proof $\Pi(x)$ which causes M to accept with probability 1.*
2. *If $x \notin L$ then for all proofs Π,* $\Pr[M(\Pi, x) \text{ accepts}] < \frac{1}{2}$.

In this terminology the famous MIP = NEXP result [BFL90] can be rephrased as NEXP = PCP(poly). Thus, the exponentially long proof of membership of a NEXP language can be checked randomly by a polynomial time machine which looks only at polynomial number of bits of the proof. The new result of Arora and Safra shows that this can be scaled down to NP!

Theorem 1 [AS92] NP = PCP(log).

This theorem was further improved by Arora, Lund, Motwani, Sudan and Szegedy [ALM+92], who showed that all NP languages have PCP proofs where the verifier uses $\log(n)$ random bits but queries only a *constant* number of bits of the proof. As with the original MIP = NEXP result, both of these new theorems can be used to extend results which show that certain NP optimization problems are difficult to approximate [FGL+91]. In particular, Arora and Safra's results show that if the maximum clique size of a graph can be approximated in polynomial time to a constant factor, then P = NP. The results of Arora, Lund *et al* show that if there exist polynomial time approximation schemes for any MAXSNP [PY88] hard problem, then P = NP. Also, as in the case of the IP = PSPACE result, these results provide compelling counterexamples to the relativization principle and the random oracle hypothesis.

The PCP classes can be relativized by allowing the verifier access to an oracle. It is easy to construct a relativized world where Theorem 1 is not true. In fact, with some more work it can be shown that

Theorem 2 *With probability* 1, $\mathrm{NP}^A \neq \mathrm{PCP}^A(\log)$ *for a random oracle A.*

We can obtain another interesting relativization results using the PCP characterization. It is well known that NP $\neq$ NEXP [SFM78]. Thus PCP(log) $\neq$ PCP(poly). However, the following shows that this separation result does not relativize and gives an example where relativization collapses two classes known to be different in the unrelativized case.

Theorem 3
Let A be any language complete for EXPSPACE. *Then* $\mathrm{PCP}^A(\log) = \mathrm{PCP}^A(\mathrm{poly})$.

Proof: A polynomial time verifier which uses $O(\mathrm{poly}(n))$ random bits can make at most a polynomial number of polynomial sized queries to the oracle A on each path. In addition it examines at most a polynomial number of bits of an exponentially long proof Π. In exponential space one can easily try out all possible proofs Π and check whether the verifier accepts with the required probability. Thus one query to the exponential space complete oracle A, suffices to decide whether the input is in the language. Hence we obtain $\mathrm{PCP}^A(O(\log(n)) = \mathrm{PCP}^A(O(\mathrm{poly}(n)) = \mathrm{EXPSPACE}$.
∎

It is worth mentioning that the relativization principle, which began with the exploitation of the difference in the oracle access mechanisms of different Turing machines, is now jeopardized by this very same technique — live by the sword, die by the sword. The relativized separations and collapses mentioned above all exploit the fact that the complexity classes such as NP and PSPACE can be represented either by time and space bounded Turing machines or by interactive protocols and proof systems. These two different kinds of representations have very different oracle access mechanisms. For example, a PSPACE machine can exhaustively query exponentially many strings, but each computation path of an IP verifier can only ask about polynomially many strings. This difference in the oracle access mechanism can be used to separate IP from PSPACE in a relativized world. This example demonstrates the danger in thinking of oracle worlds as a way of relativizing *complexity classes* — oracles do not relativize complexity classes, they only relativize the machines.

3 A Revisionist's Tale

The discovery of non-relativizing techniques involving interactive proofs has had a dramatic impact on complexity theory. It seems that problems which have contradictory relativizations are not as intimidating as we once believed and perhaps we can finally shed ourselves of this relativization principle.

However, many complexity theorists are still reluctant to declare the outright demise of the relativization principle. It is debatable whether techniques such as arithmetization and the algebra of polynomials, which were successful in solving questions about interactive proofs, are applicable to other problems. So, researchers in many areas of complexity theory would find a contradictory relativization just as intimidating as before. Still, these new results do force us to examine the assumptions that we usually make in complexity theory. It is often assumed that any proof which uses only direct simulation and diagonalization must relativize. Do all of the "standard techniques" in complexity theory relativize? If we would shed our inhibitions and take advantage of the clarity of hindsight, we just might find another way to tell the tale of the relativization principle.

It turns out that the PH $\subseteq$ IP and IP = PSPACE results were *not* the first theorems in complexity theory that do not relativize — they are simply the first ones that were so compelling that they could not be ignored. In fact, there have been non-relativizing theorems since 1977 (only two years after Baker, Gill and Solovay!), yet these rather significant theorems have not been brought up as serious challengers of the relativization principle.

We are talking about the theorem due to Hopcroft, Paul and Valiant which showed that space bounded computations are strictly more powerful than time bounded computations.

Theorem 4 [HPV77]
For all space constructible $f(n)$, $\mathrm{TIME}[f(n)] \subsetneq \mathrm{SPACE}[f(n)]$.

The techniques in this proof eventually led to the following result by Paul, Pippenger, Szemerédi and Trotter, which is one of the few known separations between nondeterminism and determinism. (Actually, it is one of the few known separations, period.)

Theorem 5 [PPST83] $\mathrm{TIME}[n] \subsetneq \mathrm{NTIME}[n]$.

As we have mentioned above, these results do not relativize and it is not clear why they have not been championed as counterexamples to the relativization principle. Perhaps the fact that Theorem 5 depends heavily on the linearity of the time bounds dissipated some of the enthusiasm for this result. However, Theorem 4 holds for all space constructible $f(n)$. Moreover, the result holds for multi-tape Turing machines. Furthermore the theorem confirms our intuition that space is a more powerful resource than time. So, this theorem cannot be casually brushed aside as contrived or model dependent.

Theorem 6 *There exists an oracle* A *such that for all* $f(n)$,

$$\mathrm{TIME}^A[f(n)] = \mathrm{SPACE}^A[f(n)].$$

Proof: Let $D_1, D_2, D_3, \ldots$ be a standard enumeration of SPACE$[2^n]$ machines. Consider the language A_0 defined by

$$A_0 = \{ \langle M, D_i, 1^c, x, 1^r \rangle \mid M^{L(D_i)}(x) \text{ accepts using } r \text{ tape cells and } c > |\langle M, D_i \rangle|^2 \}.$$

Since M is restricted to querying $L(D_i)$ on strings of length at most r, 1^c provides enough padding for A_0 to be recognized in SPACE$[2^n]$. A direct simulation of the

$M^{L(D_i)}(x)$ computation would require at most $c_1|x| + c_2 2^r$ tape cells, where c_1 and c_2 are constants that depend only on the machine description of D_i and M (e.g. size of the tape alphabet and number of work tapes). So, some fixed SPACE$[2^n]$ machine D_{i_0} recognizes A_0.

Now, suppose that $f(n)$ is time constructible. Let L be a language accepted by some $f(n)$-space bounded TM M with oracle A_0. Then, M^{A_0} accepts an input string x if and only if $\langle M, D_{i_0}, 1^c, x, 1^{f(n)}\rangle \in A_0$ where $c > |\langle M, D_{i_0}\rangle|^2$. Since, $f(n)$ is time constructible, a TIME$^{A_0}[f(n)]$ machine can write down the query

$$\text{“}\langle M, D_{i_0}, 1^c, x, 1^{f(n)}\rangle \in A_0?\text{”}.$$

Hence, $L \in \text{TIME}^{A_0}[f(n)]$.

In the case where $f(n)$ is not time constructible, the time bounded machine will first compute the exact amount of space used by an $f(n)$-space bounded machine M with the help of the oracle A_1,

$$A_1 = \{\ \langle M, D_i, 1^c, x, 1^r\rangle \mid M^{L(D_i)}(x) \text{ uses at least } r \text{ tape cells and } c > |\langle M, D_i\rangle|^2\ \}.$$

As before, $A = A_0 \oplus A_1$ can be recognized in SPACE$[2^n]$ by some fixed TM D_j. Now, let L be the language recognized by M^A an $f(n)$-space bounded oracle TM and let $s(x)$ be the number of tape cells that M^A uses on input x. Hence, $s(x) \leq f(|x|)$. Using $s(x)$ queries to the oracle A_1, a time-bounded oracle TM N^A can compute $s(x)$ in $O(s(x))$ time steps. Essentially, N^A asks the oracle A_1 questions of the form

$$\text{“}\langle M, D_j, 1^c, x, 1^r\rangle \in A_1?\text{”}.$$

If the answer is yes then, N repeats the question with r replaced by $r+1$. This takes one only time step, assuming the oracle query model which does not erase the query tape immediately after the query. If the answer is no, then $s(x) = r - 1$ and with one final query

$$\text{“}\langle M, D_j, 1^c, x, 1^{s(x)}\rangle \in A_0?\text{”},$$

N^A can determine if M^A accepts x. Since $s(x) \leq f(|x|)$, $L \in \text{TIME}^A[f(n)]$. ∎

It seems unlikely, though possible, that no one has noticed that Theorem 4 does not relativize. It must have been immediately obvious that the proof technique does not relativize, since Hopcroft, Paul and Valiant showed by a graph pebbling argument that a SPACE$[f(n)/\log(f(n))]$ machine can directly simulate a TIME$[f(n)]$ machine. I.e., they showed that

$$\text{TIME}[f(n)] \subseteq \text{SPACE}[f(n)/\log(f(n))].$$

Since $f(n)$ is space constructible[1], by invoking the Space Hierarchy Theorem [SHL65], they conclude that

$$\text{TIME}[f(n)] \subsetneq \text{SPACE}[f(n)].$$

Clearly, a SPACE[$f(n)/\log(f(n))$] machine cannot query strings that a TIME[$f(n)$] machine can, so the first step of the proof does not relativize. However, the theorem itself does not relativize, so any proof of the theorem must use non-relativizing techniques. So it is surprising to find that the technique which Hopcroft, Paul and Valiant employed is essentially simulation and diagonalization. Thus, it should have been obvious in 1977, that the relativization principle does not hold water, because standard techniques can produce non-relativizing results.

4 A Turning Point

What is really at stake here is not just the relativization principle, but also our concept of computation. The way that the computation of a Turing machine is generally viewed is as a sequence of discrete steps regulated by the finite control of the machine. In this model, the steps taken at one end of the computation do not seem to have much effect on the rest of the computation. Papadimitriou and Yannakakis complained about this apparent defect of machine models when they defined the approximation classes MAXNP and MAXSNP [PY88] as sets of logical sentences:

> The intuitive reason is that computation is an inherently unstable, non-robust mathematical object, in the sense that it can be turned from non-accepting to accepting by changes that would be insignificant in any reasonable metric—say, by flipping a single state to accepting.

Another consequence of this view of computation is that the size of a single step of the computation does not seem to matter. Whether the machine changes the tape symbol based upon the entries of the transition table or upon the recommendation of an oracle, the overall structure of the computation remains the same. The relativization principle is born out of this view of computation. Adding an oracle merely increases the size of each step of the computation, much like adding axioms to a proof system. The steps involved in the reasoning are the same, we are just allowed to make more assumptions. This view of computation is very appealing because it allows us to deal with computation at a very high level. However by doing this we ignore a crucial difference between unrelativized and relativized computations. The step size of an unrelativized computation is much smaller because transitions are made only on the basis of a finite transition table. Moreover, the change from one configuration to the next is based completely on local information near the machine heads. Thus, in some sense, the structure of an unrelativized computation is "coherent". Adding an oracle immediately destroys this coherence because successive steps now depend on entire queries and not just on finite local information.

[1] If this space constructibility assumption were not necessary, then we would have a corollary that P $\subsetneq$ PSPACE, because by the Union Theorem [MM69], there exists a recursive function $S(n)$ such that P = TIME[$S(n)$] and PSPACE = SPACE[$S(n)$]. However, $S(n)$ cannot be space constructible, so we cannot conclude that TIME[$S(n)$] $\subsetneq$ SPACE[$S(n)$].

Both the non-relativizing techniques mentioned earlier exploit this coherence property in a essential way. The techniques used in results about interactive proofs and probabilistic proof checking rely on the fact that coherence enables us to obtain small descriptions for entire machine computations. For example, an entire PSPACE computation can be encoded as a QBF and a nondeterministic computation is representable as a boolean formula. Such descriptions make these problems amenable to the algebra of low degree polynomials. This insight played a key role in the recent results about interactive proofs and has opened a fresh avenue for exploring computations. Even the pebbling approach of Hopcroft, Paul and Valiant uses the fact that coherence ensures that a computation can be simulated without having to store entire configurations of the machine at the same time. Only those parts that change are required at any one time and others can be recomputed when required without using too much space.

The above examples demonstrate that if we are to resolve the major open questions in computational complexity, we must take into account the coherence of unrelativized computations. A fascinating question is whether further insights into unrelativized computation can be obtained by representing entire computations by other "continuous" mathematical objects. In any case, with the emergence of new non-relativizing techniques, we can be sure that the old problems will be attacked with renewed vigor. The next few years promise to be exciting times for complexity theory.

References

[ALM+92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. Unpublished manuscript, 1992.

[AS92] S. Arora and S. Safra. Approximating clique is NP-complete. Unpublished manuscript, 1992.

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

[BFL90] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 16–25, 1990.

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

[BG81] C. Bennett and J. Gill. Relative to a random oracle A, $P^A \neq NP^A \neq \text{co-}NP^A$ with probability 1. *SIAM Journal on Computing*, 10(1):96–113, February 1981.

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

[BLS84] R. V. Book, T. J. Long, and A. L. Selman. Quantitative relativizations of complexity classes. *SIAM Journal on Computing*, 13:461–487, 1984.

[Boo87] R. V. Book. Towards a theory of relativizations: Positive relativizations. In *Proceedings of the 4th Symposium on Theoretical Aspects of Computer Science*, volume 247 of *Lecture Notes in Computer Science*, pages 1–21. Springer-Verlag, 1987.

[Boo90] R. V. Book. On separating complexity classes. In *Proceedings of the 5th Structure in Complexity Theory Conference*, pages 299–304, July 1990.

[CGH90] B. Chor, O. Goldreich, and J. Håstad. The random oracle hypothesis is false. Technical Report 631, Department of Computer Science, Technion, 1990. Revised paper to appear in the *Journal of Computer and System Sciences*.

[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.

[FGL+91] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 2–12, 1991.

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

[FRS90] L. Fortnow, J. Rompel, and M. Sipser. Errata for on the power of multi-prover interactive protocols. In *Proceedings of the 5th Structure in Complexity Theory Conference*, pages 318–319, July 1990.

[FS88] L. Fortnow and M. Sipser. Are there interactive protocols for co-NP languages? *Information Processing Letters*, 28(5):249–251, August 1988.

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

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

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

[HCKM88] J. Hartmanis, R. Chang, J. Kadin, and S. Mitchell. Some observations about relativization 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 *Proceedings of the 2nd Scandinavian Workshop on Algorithm Theory*, volume 447 of *Lecture Notes in Computer Science*, pages 1–12. Springer-Verlag, 1990. Revised paper to appear in the *Journal of Computer and System Sciences*.

[Hop84] J. E. Hopcroft. Turing machines. *Scientific American*, pages 86–98, May 1984.

[HPV77] J. Hopcroft, W. Paul, and L. Valiant. On time versus space. *Journal of the ACM*, 24(2):332–337, April 1977.

[Ko89] K. Ko. Relativized polynomial time hierarchies having exactly k levels. *SIAM Journal on Computing*, 18(2):392, 1989.

[Koz80] D. Kozen. Indexings of subrecursive classes. *Theoretical Computer Science*, 11:277–301, 1980.

[Kur82] S. A. Kurtz. On the random oracle hypothesis. In *ACM Symposium on Theory of Computing*, pages 224–230, 1982.

[LFKN90] 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.

[MM69] E. M. McCreight and A. R. Meyer. Classes of computable functions defined by bounds on computation. In *ACM Symposium on Theory of Computing*, pages 79–88, 1969.

[PPST83] W. J. Paul, N. Pippenger, E. Szemerédi, and W. T. Trotter. On determinism versus non-determinism. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 429–438, 1983.

[PY88] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. In *ACM Symposium on Theory of Computing*, pages 229–234, 1988.

[SFM78] J. I. Seiferas, M. J. Fischer, and A. R. Meyer. Separating nondeterministic time complexity classes. *Journal of the ACM*, 25:146–167, 1978.

[Sha90] A. Shamir. IP = PSPACE. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 11–15, 1990.

[SHL65] R. E. Stearns, J. Hartmanis, and P. M. Lewis II. Hierarchies of memory limited computations. In *Proceedings of the Sixth Annual Symposium on Switching Circuit Theory and Logical Design*, pages 179–190, 1965.

[Yao85] 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.