COUNTING HIERARCHIES: POLYNOMIAL TIME AND CONSTANT DEPTH CIRCUITS

ERIC W. ALLENDER[1]
Department of Computer Science, Rutgers University
New Brunswick, NJ 08903, USA

and

KLAUS W. WAGNER
Institut für Informatik, Universität Würzburg
D-8700 Würzburg, FRG

Abstract

In the spring of 1989, Seinosuke Toda of the University of Electro-Communications in Tokyo, Japan, proved that the polynomial hierarchy is contained in P^{PP} [To-89]. In this Structural Complexity Column, we will briefly review Toda's result, and explore how it relates to other topics of interest in computer science. In particular, we will introduce the reader to

The Counting Hierarchy: a hierarchy of complexity classes contained in PSPACE and containing the Polynomial Hierarchy.

Threshold Circuits: circuits constructed of MAJORITY gates; this notion of circuit is being studied not only by complexity theoreticians, but also by researchers in an active subfield of AI studying "neural networks".

Along the way, we'll review the important notion of an *operator* on a complexity class.

1. The Counting Hierarchy, and Operators on Complexity Classes

The counting hierarchy was defined in [Wa-86] and independently by Parberry and Schnitger in [PS-88]. (The motivation for [Wa-86] was the desire to classify precisely the complexity of certain combinatorial problems on graphs with succinct descriptions. Parberry and Schnitger were studying "threshold Turing machines" in connection with parallel computation.) One way to define the counting hierarchy is to take the usual definition of the polynomial hierarchy:

[1]Supported in part by National Science Foundation Grants CCR-8810467 and CCR-9204874

- $\Sigma_0^p = \mathrm{P}$
- $\Sigma_{k+1}^p = \mathrm{NP}^{\Sigma_k^p}$ for $k \geq 0$.

and replace "NP" with "PP". (Recall that PP is "probabilistic polynomial time" as defined by Gill [Gi-77] and Simon [Si-77]; L is in PP^A if there is a polynomial-time nondeterministic oracle Turing machine M such that $x \in L$ iff more than half of the computation paths of M on input x with oracle A are accepting.) That gives us the definition:

- $\mathbf{C}_0^p = \mathrm{P}$
- $\mathbf{C}_{k+1}^p = \mathrm{PP}^{\mathbf{C}_k^p}$ for $k \geq 0$.

Thus $\mathbf{C}_1^p = \mathrm{PP}$, $\mathbf{C}_2^p = \mathrm{PP}^{\mathrm{PP}}$, and so on.

This characterization of the counting hierarchy is due to Jacobo Torán, who has studied the counting hierarchy in depth [Tor-88, Tor-91]. Although this is perhaps the simplest way to define the counting hierarchy, it is not the original definition, and the proof that this characterization (in terms of oracle Turing machines) is equivalent to the one given below (in terms of generalized quantifiers) is not obvious.

Recall that an alternative characterization of the polynomial hierarchy is given by applying bounded existential and universal quantifiers to polynomial-time predicates. One way to formalize this is to define *operators* acting on classes of sets. That is:

Definition: Let $\mathcal{C}$ be a class of languages. Then define $\exists \cdot \mathcal{C}$ to be the set of all languages L such that there is some polynomial p and some set $A \in \mathcal{C}$ such that $x \in L \Longleftrightarrow \exists y\ (|y| \leq p(|x|)$ and $\langle x, y\rangle \in A)$. In a similar way, one can define $\forall \cdot \mathcal{C}$.

It is a familiar fact [St-77, Wr-77] that $\mathrm{NP} = \exists{\cdot}\mathrm{P}$, $\Sigma_2^p = \exists \cdot \forall{\cdot}\mathrm{P}$, and so on. It has turned out to be useful to consider other operators, in addition to $\exists$ and $\forall$. In particular, the operators $\mathbf{C}$, BP, and $\oplus$ will be important for this survey.

Definition: Let $\mathcal{C}$ be a class of languages. Define

- $\mathbf{C}{\cdot}\mathcal{C}$ to be the set of all languages L such that there is some polynomial p and some set $A \in \mathcal{C}$ such that $x \in L \Longleftrightarrow$

$$\|\{y : |y| \leq p(|x|) \text{ and } \langle x, y\rangle \in A\}\| \,/ 2^{p(|x|)} > 1/2.$$

- $BP \cdot \mathcal{C}$ to be the set of all languages L such that there is some polynomial p and some set $A \in \mathcal{C}$ such that

$$\|\{y : |y| \leq p(|x|) \text{ and } [\langle x,y\rangle \in A \iff x \in L]\}\| / 2^{p(|x|)} > 3/4.$$

- $\oplus \cdot \mathcal{C}$ to be the set of all languages L such that there is some polynomial p and some set $A \in \mathcal{C}$ such that $x \in L \iff$

$$\|\{y : |y| \leq p(|x|) \text{ and } \langle x,y\rangle \in A\}\| \text{ is odd.}$$

(The original definition of [Wa-86, Wa-86a] actually used a more general definition of the operator **C**; however it turns out that the complexity classes defined in this way are quite robust to small changes to the definitions, and for any reasonable complexity class $\mathcal{C}$, the definition given above is equivalent to the definition given in [Wa-86].) The original definition of the counting hierarchy is thus:

- $\mathbf{C}_0^p = \mathrm{P}$
- $\mathbf{C}_{k+1}^p = \mathbf{C}\cdot\mathbf{C}_k^p$.

Using the operators $\forall, \exists, \oplus$, **C**, and BP in different combinations, one obtains other interesting complexity classes. (This list of operators is by no means exclusive. For example, Torán has shown that an "exact counting" version of the **C** operator (which he denotes by $\mathbf{C}_=$) is very useful tool for proving results about other classes in the counting hierarchy [Tor-88, Tor-91].)

The BP operator is obviously a way to generalize the notion of probabilistic computation to other complexity classes. For example, $BP \cdot \exists \cdot \mathrm{P}$ is the complexity class AM [BM-89,Sc-87]. The $\oplus$ operator is a way to generalize the complexity class Parity-P of [PZ-83,GP-86].

We should mention that Stathis Zachos has an alternative logic-based formulation for defining complexity classes of this sort. His approach has also been quite useful in proving results about these classes; see [HZ-84, Za-88, ZH-86, BHZ-87, ZF-87].

We would now like to briefly catalogue some of the known relationships among the complexity classes defined using these operators. Some of these inclusions are known to hold for *all* classes of sets $\mathcal{C}$, but some of the other inclusions only hold if $\mathcal{C}$ is sufficiently "nice". For the sake of simplicity, we will assume here that $\mathcal{C}$ is a complexity class defined by applying the operators from the set $\{\exists, \forall, \oplus, BP, \mathbf{C}\}$ to the class P. All such classes satisfy the appropriate "niceness" conditions.[2]

[2]Note, however, that one must be careful what one assumes about the classes $\mathcal{C}$. Some subclasses of the counting hierarchy (such as PP) seem not to be closed under union or intersection* while other classes (such as NP) seem not to be closed under complementation. Information about the

Theorem: *A catalog of facts about subclasses of the Counting Hierarchy.*

1. $\exists \cdot \mathcal{C} \cup \forall \cdot \mathcal{C} \cup BP \cdot \mathcal{C} \subseteq \mathbf{C}{\cdot}\mathcal{C}$. (This is a generalization of Gill's result [Gi-77] that NP and coNP are contained in PP, and also of the trivial inclusion BPP $\subseteq$ PP.)
2. $\oplus \cdot \mathcal{C} \subseteq \mathrm{P}^{\mathbf{C}\cdot\mathcal{C}}$
3. $\exists \cdot \mathcal{C} = \mathrm{NP}^{\mathcal{C}}$
4. $\mathbf{C}{\cdot}\mathcal{C} = \mathrm{PP}^{\mathcal{C}}$ [Tor-91]
5. $BP \cdot \mathcal{C} \subseteq \mathrm{BPP}^{\mathcal{C}}$
6. $\oplus \cdot \mathcal{C} \subseteq \oplus{\cdot}\mathrm{P}^{\mathcal{C}}$
7. NP $\subseteq BP \cdot \oplus{\cdot}\mathrm{P}$. (This follows from the proof of [VV-86], showing that SAT is reducible via probabilistic reductions to the "unique satisfiability problem".)
8. $\oplus \cdot \oplus{\cdot}\, \mathcal{C} = \oplus{\cdot}\, \mathcal{C}$.
9. $\oplus{\cdot}\mathrm{P}^{\oplus\cdot\mathrm{P}} = \oplus{\cdot}\mathrm{P}$. [PZ-83] (Thus $\oplus{\cdot}$P is closed under $\leq_T^p$.)
10. $\mathrm{PP}^{\mathrm{BPP}} = \mathrm{PP}$. [KSTT-89]

If the underlying class $\mathcal{C}$ is closed under "positive reducibility" (see [Sc-87]) then the usual techniques of "amplification" can be used to exponentially reduce the error probability for sets in the class $BP{\cdot}\mathcal{C}$. Thus for such classes $\mathcal{C}$, the following inclusions and equalities hold. (It should be noted that classes defined using the **C** operator are not known to be closed under this sort of "positive reducibility," and thus the following inclusions may not hold for such classes.)

Theorem: For classes $\mathcal{C}$ closed under positive reducibility,

1. $\mathrm{P}^{BP\cdot\mathcal{C}} \subseteq BP{\cdot}\mathrm{P}^{\mathcal{C}}$.
2. $BP \cdot BP \cdot \mathcal{C} = BP \cdot \mathcal{C}$.

closure properties of classes in the Counting Hierarchy may be found in [Tor-88].

*In the brief period that has elapsed between the time when this article was first written for the EATCS Bulletin (January, 1990) and the time of the compilation of the present volume (September, 1992), much has been learned about the counting hierarchy. It was shown in [BRS-91] that PP *is* closed under union and intersection. This beautiful and surprising result has been generalized to the entire counting hierarchy [Gu-90].

3. $BP \cdot \mathcal{C} \subseteq \exists \cdot \forall \cdot \mathcal{C} \cap \forall \cdot \exists \cdot \mathcal{C}$. [Sc-87] (This is a generalization of the result of [Si-83a, La-83], showing that BPP is in the polynomial hierarchy.)

4. $\oplus \cdot BP \cdot \mathcal{C} \subseteq BP \cdot \oplus \cdot \mathcal{C}$. (This is the "operator swapping" lemma of [To-89].)

Some of the inclusions and equalities listed above are quite easy to prove, and others are not at all obvious. It is important to note that all of the facts listed above *relativize*, so that the same relationships hold relative to any oracle.

Armed with this list of facts, we can now give a short proof of the first part of Toda's proof.

Theorem: [To-89] The polynomial hierarchy is contained in $BP \cdot \oplus \cdot \mathrm{P}$.

Proof: We have already observed that $\Sigma_1^p \subseteq BP \cdot \oplus \cdot \mathrm{P}$. Thus assume inductively that $\Sigma_k^p \subseteq BP \cdot \oplus \cdot \mathrm{P}$, and note that

$$\begin{aligned}
\Sigma_{k+1}^p &= \mathrm{NP}^{\Sigma_k^p} \\
&\subseteq BP \cdot \oplus \cdot \mathrm{P}^{BP \cdot \oplus \cdot \mathrm{P}} \\
&\subseteq BP \cdot \oplus \cdot BP \cdot \mathrm{P}^{\oplus \cdot \mathrm{P}} \\
&= BP \cdot \oplus \cdot BP \cdot \oplus \cdot \mathrm{P} \\
&\subseteq BP \cdot BP \cdot \oplus \cdot \oplus \cdot \mathrm{P} \\
&= BP \cdot \oplus \cdot \mathrm{P}
\end{aligned}$$

Having now shown that the polynomial hierarchy is contained in $BP \cdot \oplus \cdot \mathrm{P} \subseteq \mathbf{C} \cdot \oplus \cdot \mathrm{P}$, Toda's theorem follows from the following inclusion (the proof of which involves a detailed manipulation of the computation tree of a machine, which we do not have sufficient space to present here).

Theorem: [To-89] $\mathbf{C} \cdot \oplus \cdot \mathrm{P} \subseteq \mathrm{P}^{\mathrm{PP}}$.

Toda in fact shows that P^{PP} contains a seemingly larger class. Let PH denote the polynomial hierarchy.

Theorem: [To-89] $\mathrm{PP}^{\mathrm{PH}} \subseteq \mathrm{P}^{\mathrm{PP}}$.

Proof:

$$\begin{aligned}
\mathrm{PP}^{\mathrm{PH}} &\subseteq \mathrm{PP}^{BP \cdot \oplus \cdot \mathrm{P}} \\
&\subseteq \mathrm{PP}^{\mathrm{BPP}^{\oplus \cdot \mathrm{P}}} \\
&= \mathrm{PP}^{\oplus \cdot \mathrm{P}} \\
&= \mathbf{C} \cdot \oplus \cdot \mathrm{P} \\
&\subseteq \mathrm{P}^{\mathrm{PP}}
\end{aligned}$$

An obvious open question is whether or not PH is contained in PP itself. It is not even known if P^{NP} is contained in PP. The largest subclass of PH known to be contained in PP is the class $\mathrm{P}^{\mathrm{NP}[\log]}$ (the class of languages that can be recognized with $O(\log n)$ queries to SAT) [BHW-91].[3]

2. Circuits (and Neural Nets)

Toda's theorem has some very interesting consequences for circuit complexity. Before we can present these consequences, it will be instructive to review some basic notions from circuit complexity.

A *circuit family* is a set $\{C_n : n \in \mathbf{N}\}$ of circuits, where each circuit C_n takes (binary) inputs of length n, and produces a single output. Each circuit family thus defines a language. A circuit family is *uniform* if the function $n \mapsto C_n$ is easily computable in some sense. For the very small circuit complexity classes we discuss here, a very strong notion of uniformity is appropriate. This is discussed in detail in [BIS-90]; we will not give detailed definitions concerning uniformity here.

Two important circuit complexity classes are NC^1 and AC^0; NC^1 is the class of languages accepted by (uniform) circuits of AND and OR gates of fan-in 2, of polynomial size and logarithmic depth. AC^0 is the class of languages accepted by (uniform) circuits of AND and OR gates of unbounded fan-in, of polynomial size and $O(1)$ depth. (AC^0_k denotes the languages in AC^0 accepted by circuits of depth k.) Clearly, $\mathrm{AC}^0 \subseteq \mathrm{NC}^1$, and both classes are contained in P.

In some sense, AC^0 and NC^1 represent the extremes of our knowledge about complexity classes. On the one hand, many combinatorial methods have been developed that enable us to show that many problems in NC^1 are not in AC^0 [FSS-84, Aj-83, Ya-85, Hå-86]. On the other hand, it is still an open problem whether or not NP $=$ NC^1. Thus we know a great deal about AC^0, but really quite little about NC^1.

Thus a good place to try to make progress is in the range between AC^0 and NC^1. Among the various complexity classes that have been studied in this range, TC^0 has attracted a great deal of attention.

Definition: A *threshold circuit* is a circuit composed entirely of majority gates. (A majority gate outputs 1 iff the majority of its inputs have value 1.) TC^0 is the class of languages accepted by threshold circuits of polynomial size and depth $O(1)$. TC^0_k denotes the class of languages accepted by threshold circuits of depth k.

[3] In the meantime, this question has been addressed by Richard Beigel. In [Be-92], Beigel presents an oracle relative to which P^{NP} is not contained in PP. In fact, he shows the stronger result that the inclusion presented in [BHW-91] can not be improved using relativizable proof techniques.

The following points explain in part why TC^0 has been the focus of attention.

- The majority gate is of essentially the same power as a gate for integer multiplication [CSV-84]. Thus TC^0 characterizes the power of certain arithmetic circuits.
- TC^0 exactly characterizes the complexity of symmetric functions [FKPS-85].
- Division is in P-uniform TC^0 [Re-87,RT-90]. This is one of the few cases in which uniformity considerations come into play when classifying the complexity of natural problems.
- Many computer scientists have been studying "connectionist" models of the brain, also known as "neural nets". It turns out that one of the most popular models studied by connectionists is computationally equivalent to the threshold circuit model [PS-88, PS-89]. Ian Parberry has written quite eloquently on the relationship between computational complexity theory and the study of neural networks [Pa-90,PS-89]. Any complexity-theoretic work on threshold circuits thus is of some interest to the neural network community, and conversely, some theoretical work of interest to complexity theoreticians was motivated primarily by the study of neural networks [Go-89, Bru-90, BS-92].
- A large body of work on threshold logic and threshold circuits already exists (e.g., [Mu-71]). Much of this work was motivated by interest in building computers with threshold logic components.
- As we shall see, TC^0 is intimately connected with the counting hierarchy.

Now that we have established that there is sufficient interest for studying TC^0, it is unfortunate that we must report that we know abysmally little about threshold circuits. With some effort, Hajnal et. al. have shown the following result:

Theorem: [HM-87] $\mathrm{TC}^0_1 \subset \mathrm{TC}^0_2 \subset \mathrm{TC}^0_3$.

However it is still an open question if $\mathrm{TC}^0_3 = \mathrm{NP}$. (Even worse, it is not known if NP is equal to the class of languages accepted by uniform circuits of polynomial size and depth three, composed of AND, OR and MOD 6 gates.)

It is fairly natural to conjecture that the TC^0 hierarchy is infinite (i.e., that $\mathrm{TC}^0 \neq \mathrm{TC}^0_k$ for any k), and indeed this conjecture is mentioned in [BIS-90,Ya-89]. However, it has also been conjectured that $\mathrm{TC}^0 = \mathrm{NC}^1$ [IL-89], which would imply that the TC^0 hierarchy collapses.

Yao has shown that when one considers *monotone* circuits, the corresponding TC^0 hierarchy *is* infinite. In fact, he shows the stronger result that for every k, there is a set in AC^0_{k+1} that requires *exponential* size on threshold circuits of size k [Ya-89]. As we shall see, one of the surprising consequences of Toda's theorem is that the corresponding result is *not* true when one considers non-monotone circuits.

2.1. Relating Circuits to the Counting Hierarchy

When the alternating Turing machine (ATM) was introduced in [CKS-81], one of the motivations was to have a Turing machine model suitable for studying parallelism. Results relating circuit size and depth to alternating Turing machine space and time, respectively, can be found in [Ru-81]. In order to model circuits of sublinear depth, it is necessary to have Turing machines with sublinear running times; thus a mechanism is provided for "random access" to any bit of the input. (Details may be found in [Ru-81].) NC^1 is alternating log time, in this model of computation.

Sipser studied the subclass of NC^1 defined by log-time ATMs that make only $O(1)$ alternations; this is called the log-time hierarchy [Si-83]. The levels of this hierarchy are denoted Σ_k-logtime. An analogous extension, called the logarithmic-time counting hierarchy (LCH) was defined in [Tor-88]; the levels of the LCH are defined using logarithmic versions of the $\exists, \forall$ and $\mathbf{C}$ operators. We will denote the k-th level of the LCH by $\mathbf{C}_k$-logtime.

Levels of the LCH correspond roughly to circuit depth. It was shown in [BIS-90] that $\bigcup_k \Sigma_k$-logtime $=$ (uniform) AC^0, and the same techniques can be used to show that LCH $=$ (uniform) TC^0. However, the correspondence is not exact. Depending somewhat on the precise way that the uniformity condition is defined, one can show that $\mathbf{C}_k$-logtime $\subseteq \mathrm{TC}^0_{k+1}$, and $\mathrm{TC}^0_k \subseteq \mathbf{C}_{k+1}$-logtime, but no tighter correspondence is known.

On the other hand, levels of the LCH correspond exactly to levels of the counting hierarchy. Generalizing a result of [Si-83], Torán showed in [Tor-88] that if there is an oracle separating two levels of the counting hierarchy, then the corresponding two levels of LCH are distinct. (The intuition here is that there is essentially no difference between an oracle Turing machine writing an oracle query on its query tape, and an ATM writing an address on its address tape giving it "random access" to the input. That is, the characteristic sequence of an oracle to a polynomial-time Turing machine can be viewed as *input* to a log-time ATM.) Thus an oracle result about the counting hierarchy implies a real separation in LCH.

There is a partial converse, as well. It was pointed out [FSS-84, Si-83] that if there is a language in Σ_k-logtime that requires more than size $2^{\log^{O(1)} n}$ to recognize on depth $k-1$ circuits of AND and OR gates, then there is an oracle relative to which

$\Sigma_{k-1}^p \neq \Sigma_k^p$. Circuit lower bounds of this sort were achieved first by Yao [Ya-85], and further developments may be found in [Hå-86 , Ko-89]. Along the same lines, if one can show that for every k there is a set in $\mathbf{C}_k$ that requires more than size $2^{\log^{O(1)} n}$ to recognize on depth $k-1$ threshold circuits, then the counting hierarchy is infinite.

Circuit lower bounds (or equivalently, oracles separating levels of the counting hierarchy) have been quite difficult to construct. Torán reviews the separations that are currently known in [Tor-91]. (See also [Be-92, Gr-90].)

Seen in this setting, it is clear that Toda's result that PH $\subseteq$ P$^{\text{PP}}$ says something about the threshold circuit complexity of AC^0. However, because as mentioned above, the mapping between the levels of LCH and circuit depth is only approximate; thus a direct proof is necessary if one wants to achieve the sharpest possible result.

In [Al-89], a very simple proof was presented of the fact that (non-uniform) AC$_k^0$ can be recognized by (non-uniform) depth three threshold circuits[4] of size $n^{\log^k n}$. The proof in [Al-89] was inspired by Toda's theorem, but it does not follow directly from [To-89]. In fact the circuit complexity result of [Al-89] appears to be much easier to prove than Toda's theorem; the proof of [Al-89] makes use of some important but elementary observations of Razborov and Smolensky [Ra-87, Sm-87].

It is also true that (uniform) AC$_k^0$ can be recognized by (uniform) depth three threshold circuits of size $n^{\log^k n}$, although the proof is slightly more complex than that of [Al-89]. (The proof of the uniform version still does not appeal to Toda's theorem.) The proof of the theorem for uniform circuits is found in [AH-90].

3. Conclusion

We hope that this will serve as a useful survey of an area of research we find very exciting. There are a great many open problems, and we feel confident that much progress on these problems will be made in the next few years. To close this survey, we mention a few problems that seem worthy of study:

- What inclusions can be shown among the complexity classes defined by the operators $\exists, \forall, \oplus, BP$, $\mathbf{C}$? Many of the relationships have yet to be worked out.
- Are there other operators that would be more useful for study? (Note that before Toda's work it was not widely suspected that $BP \cdot \oplus \cdot$P would turn out to be so interesting.)
- Is IP (or IP(log)) contained in the counting hierarchy? (IP (IP(log)) is the class of languages accepted by interactive proof systems with $n^{O(1)}$ ($\log n$) rounds of

[4]In the meantime, Yao has shown stronger inclusions of this sort [Ya-90].

communication.) Is there an interesting circuit complexity class corresponding to IP(log)?[5]

- Can the circuit lower bounds of [HM-87] be extended to circuits of greater depth?
- Is $AC^0 \subseteq TC^0_3$, or is the simulation of AC^0 presented in [Al-89] optimal? (A first step in this direction has been taken by Bruck and Smolensky [BS-92]. They show that AC^0 requires size at least $2^{\mathrm{polylog} n}$ on depth 2 threshold circuits.)
- And last but not least, we leave the big question: is $TC^0 = NC^1$?

References

[Aj-83] M. Ajtai, *Σ^1_1 formulae on finite structures*, Annals of Pure and Applied Logic 24, 1-48.

[Al-89] E. Allender, *A note on the power of threshold circuits*, Proc. 30th IEEE Symposium on Foundations of Computer Science, pp. 580–584. See also [AH-92].

[Al-90] E. Allender, *Oracles vs proof techniques that do not relativize*, Proc. International Symposium SIGAL '90, Lecture Notes in Computer Science 450, pp. 39–52.

[AH-90] E. Allender and U. Hertrampf, *On the power of uniform families of constant depth threshold circuits*, Proc. 15th International Symposium on Mathematical Foundations of Computer Science, 1990, Lecture Notes in Computer Science 452, pp. 158–164. See also [AH-92].

[AH-92] E. Allender and U. Hertrampf, *Depth reduction for circuits of unbounded fan-in*, to appear in Information and Computation.

[Ba-90] L. Babai, *E-mail and the unexpected power of interaction*, Proc. 5th IEEE Structure in Complexity Theory Conference, pp. 30–44.

[5] Just before this paper was ready to be sent off to EATCS Bulletin, we learned of a very important result of Adi Shamir, building on earlier work of N. Nisan, and of C. Lund, L. Fortnow, and H. Karloff [LFKN-90]. Shamir has shown that IP=PSPACE [Sh-90]. One reason this is an especially interesting result is the fact that there is an oracle relative to which IP does not contain coNP [FS-88]. Thus this is the most important example of a result concerning "robust" complexity classes that does not relativize; this is sure to have wide-ranging implications. (In the meantime, a number of papers have appeared in which these implications are discussed: [Br-90, Br-90a, HCR-90, HCRR-90, Al-90, Ba-90, CGH-90].)

[BM-89] L. Babai and S. Moran, *Arthur-Merlin games: a randomized proof system and a hierarchy of complexity classes*, J. Comput. and System Sciences 36, 254–276.

[BIS-90] D. A. Mix Barrington, N. Immerman, and H. Straubing, *On uniformity within* NC^1, Journal of Computer and System Sciences 41, 274–306.

[Be-92] R. Beigel, *Perceptrons, PP, and the polynomial hierarchy*, Proc. 7th IEEE Structure in Complexity Theory Conference, pp. 14–19.

[BHW-91] R. Beigel, L. Hemachandra, and G. Wechsung, *Probabilistic polynomial time is closed under parity reductions*, Information Processing Letters 37, 91–94.

[BRS-91] R. Beigel, N. Reingold, and D. Spielman, *PP is closed under intersection*, Proc. 23rd ACM Symposium on Theory of Computing, pp. 1–9. To appear in J. Computer and System Science.

[BHZ-87] R. Boppana, J. Håstad, and S. Zachos, *Does co-NP have short interactive proofs?*, Information Processing Letters 25, 127–132.

[Br-90] G. Brassard, *Hot news on interactive protocols*, SIGACT News 21, 1, pp. 7–11.

[Br-90a] G. Brassard, *Hiding information from oracles*, SIGACT News 21, 2, pp. 5–11.

[Bru-90] J. Bruck, *Harmonic analysis of polynomial threshold functions*, SIAM J. Disc. Math. 3, 168–177.

[BS-92] J. Bruck and C. Smolensky, *Polynomial threshold functions,* AC^0 *functions, and spectral norms*, SIAM J. Comput. 21, 33–42.

[CKS-81] A. Chandra, D. Kozen, and L. Stockmeyer, *Alternation*, J. ACM 28, 114–133.

[CSV-84] A. Chandra, L. Stockmeyer, and U. Vishkin, *Constant depth reducibility*, SIAM J. Comput. 13, 423–439.

[CGH-90] B. Chor, O. Goldreich, and J. Håstad, *The random oracle hypothesis is false*, Technical Report 631, Department of Computer Science, Technion – Israel Institute of Technology.

[FKPS-85] R. Fagin, M. Klawe, N. Pippenger, and L. Stockmeyer, *Bounded-depth poly-size circuits for symmetric functions*, Theoretical Computer Science 36, 239–250.

[FS-88] L. Fortnow and M. Sipser, *Are there interactive protocols 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.

[Gi-77] J. Gill, *Computational complexity of probabilistic Turing machines*, SIAM J. Comput. 6, 675–695.

[GP-86] L. Goldschlager and I. Parberry, *On the construction of parallel computers from various bases of Boolean functions*, Theoretical Computer Science 43, 43–58.

[Go-89] C. Gotsman, *On Boolean functions, polynomials, and algebraic threshold functions*, Tech. Report TR-89-18, Hebrew University.

[Gr-90] F. Green, *An oracle separating $\oplus P$ from PP^{PH}*, Proc. 5th IEEE Structure in Complexity Theory Conference, pp. 295–298.

[Gu-90] S. Gupta, *A note on the counting hierarchy*, Technical Report OSU-CISRC-8/90-TR24, Computer and Information Science Research Center, Ohio State University.

[HM-87] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, G. Turán, *Threshold circuits of bounded depth*, Proc. 28th IEEE Symposium on Foundations of Computer Science, pp. 99–110.

[HCR-90] J. Hartmanis, R. Chang, D. Ranjan, and P. Rohatgi, *On IP=PSPACE and theorems with narrow proofs*, EATCS Bulletin 41, pp. 166–174.

[HCRR-90] J. Hartmanis, R. Chang, D. Ranjan, and P. Rohatgi, *Structural complexity theory: recent surprises*, Proc. 2nd Scandinavian Workshop on Algorithm Theory, Lecture Notes in Computer Science 447, pp. 1–12.

[HZ-84] P. Hinman and S. Zachos, *Probabilistic machines, oracles, and quantifiers*, Proc. Oberwolfach Recursion-Theoretic Week, Lecture Notes in Mathematics 1141, pp. 159–192.

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

[IL-89] N. Immerman and S. Landau, *The complexity of iterated multiplication*, Proc. 4th IEEE Structure in Complexity Theory Conference, pp. 104–111.

[Ko-89] K. Ko, *Relativized polynomial-time hierarchies having exactly K levels*, SIAM J. Comput. 18, 392–408.

[KSTT-89] J. Köbler, U. Schöning, S. Toda, and J. Torán, *Turing machines with few accepting computations and low sets for PP*, Proc. 4th IEEE Structure in Complexity Theory Conference, pp. 208–215.

[La-83] C. Lautemann, *BPP and the polynomial hierarchy*, Information Processing Letters 17, 215-217.

[LFKN-90] C. Lund, L. Fortnow, H. Karloff, and N. Nisan, *Algebraic methods for interactive proof systems*, Proc. 31st IEEE Symposium on Foundations of Computer Science, pp. 2–10.

[Mu-71] S. Muroga, <u>Threshold Logic and Its Applications</u>, 1971, J. Wiley and Sons.

[PZ-83] C. Papadimitriou and S. Zachos, *Two remarks on the power of counting*, Proc. 6th GI Conference, Lecture Notes in Computer Science 145, pp. 269–275.

[Pa-90] I. Parberry, *A primer on the complexity theory of neural networks*, in *A Sourcebook on Formal Techniques in Artificial Intelligence*, ed. R. Banerji, North-Holland.

[PS-88] I. Parberry and G. Schnitger, *Parallel computation with threshold functions*, J. Computer and System Science 36, 278–302.

[PS-89] I. Parberry and G. Schnitger, *Relating Boltzmann machines to conventional models of computation*, Neural Networks 2, 59–67.

[Ra-87] A. A. Razborov, *Lower bounds on the size of bounded depth networks over a complete basis with logical addition*, Mathematicheskie Zametki 41(4), 598–607. English translation in Mathematical Notes of the Academy of Sciences of the USSR 41:4, 333-338.

[Re-87] J. Reif, *On threshold circuits and polynomial computation*, Proc. 2nd IEEE Structure in Complexity Theory Conference, pp. 118–123.

[RT-90] J. Reif and S. Tate, *On threshold circuits and polynomial computation*, submitted for publication.

[Ru-81] W. Ruzzo, *On Uniform Circuit Complexity*, J. Comput. and System Sci. 21, 365–383.

[Sc-87] U. Schöning, *Probabilistic complexity classes and lowness*, Proc. 2nd IEEE Structure in Complexity Theory Conference, pp. 2–8.

[Sh-90] A. Shamir, *IP = PSPACE*, Proc. 31st IEEE Symposium on Foundations of Computer Science, pp. 11–15.

[Si-77] J. Simon, *On the difference between one and many*, Proc. 4th ICALP, Lecture Notes in Computer Science 52, pp. 480–491.

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

[Si-83a] M. Sipser, *A complexity theoretic approach to randomness*, Proc. 15th Annual ACM Symposium on Theory of Computing, pp. 330–335.

[Sm-87] R. Smolensky, *Algebraic methods in the theory of lower bounds for Boolean circuit complexity*, Proc. 19th ACM Symposium on Theory of Computing, pp. 77–82.

[St-77] L. Stockmeyer, *The polynomial-time hierarchy*, Theoretical Computer Science 3, 1–22.

[To-89] S. Toda, *PP is $\leq_T^p$-hard for the polynomial-time hierarchy*, Proc. 30th IEEE Symposium on Foundations of Computer Science, pp. 514–519.

[Tor-88] J. Torán, *Structural properties of the counting hierarchies*, Doctoral dissertation, Universitat Politècnica de Catalunya, Barcelona.

[Tor-91] J. Torán, *Complexity classes defined by counting quantifiers*, J. ACM 38, 753–774.

[VV-86] L. Valiant and V. Vazirani, *NP is as easy as detecting unique solutions*, Theoretical Computer Science 47, 85–93.

[Wa-86] K. W. Wagner, *The complexity of combinatorial problems with succinct input representation*, Acta Informatica 23, 325–356.

[Wa-86a] K. W. Wagner, *Some observations on the connection between counting and recursion*, Theoretical Computer Science 47, 131–147.

[Wr-77] C. Wrathall, *Complete sets and the polynomial-time hierarchy*, Theoretical Computer Science 3, 23–33.

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

[Ya-89] A. C. Yao, *Circuits and local computation*, Proc. 21st ACM Symposium on Theory of Computing, pp. 186–196.

[Ya-90] A. C. Yao, *On ACC and threshold circuits*, Proc. 31st IEEE Symposium on Foundations of Computer Science.

[Za-88] S. Zachos, *Probabilistic Quantifiers and Games*, J. Comput. and System Sci. 36, 433–451.

[ZF-87] S. Zachos and M. Fürer, *Probabilistic quantifiers vs. distrustful adversaries*, Proc. 7th Conference on Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science 287, pp. 443–455.

[ZH-86] S. Zachos and H. Heller, *A decisive characterization of BPP*, Information and Control 69, 125–135.