DIMACS Technical Report 97-49

Circuit Complexity before the Dawn of the New Millennium[1]

by

Eric Allender[2,3]
Department of Computer Science
Rutgers University
Piscataway, NJ 08855
USA
`allender@cs.rutgers.edu`
`http://www.cs.rutgers.edu/ allender`

[1]An earlier version of this paper appeared in Proc. 16th Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS '96), Lecture Notes in Computer Science 1180, pp. 1–18.
[2]Permanent Member
[3]Supported in part by NSF grant CCR-9509603.

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs, Bellcore, and Bell Labs.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

The 1980's saw rapid and exciting development of techniques for proving lower bounds in circuit complexity. This pace has slowed recently, and there has even been work indicating that quite different proof techniques must be employed to advance beyond the current frontier of circuit lower bounds. Although this has engendered pessimism in some quarters, there have in fact been many positive developments in the past few years showing that significant progress is possible on many fronts. This paper is a (necessarily incomplete) survey of the state of circuit complexity as we await the dawn of the new millennium.

1 Superpolynomial Size Lower Bounds

Complexity theory long ago achieved its goal of presenting interesting and important computational problems that, although computable, nonetheless require such huge circuits to compute that they are computationally intractable. In fact, in Stockmeyer's thesis, the unusual step was taken of translating an *asymptotic* result into *concrete* terms:

Theorem 1.1 *[Sto74] Any circuit that takes as input a formula (in the language of WS1S) with up to 616 symbols and produces as output a correct answer saying whether the formula is valid or not, requires at least 10^{123} gates.*

To quote from [Sto87]:

> Even if gates were the size of a proton and were connected by infinitely thin wires, the network would densely fill the known universe.

In the intervening years complexity theory has made some progress proving that other problems A require circuits of superpolynomial size (in symbols: $A \notin \mathrm{P/poly}$), but no such A has been shown to exist in nondeterministic exponential time ($\mathrm{NTIME}(2^{n^{O(1)}})$) or even in the potentially larger class $\mathrm{DTIME}(2^{n^{O(1)}})^{\mathrm{NP}}$. Where can we find sets that are not in P/poly? A straightforward diagonalization shows that for any superpolynomial time-bound T, there is a problem in $\mathrm{DSPACE}(T(n))-$ P/poly. Recall that deterministic space complexity is roughly the same as alternating time complexity [CKS81]. It turns out that the full power of alternation is not needed to obtain sets outside of P/poly – two alternations suffice, as can be shown using techniques of [Kan82] (see also [BH92]). Combined with Toda's theorem [Tod91] we obtain the following.

Theorem 1.2 *[Kan82, BH92, Tod91] Let T be a time-constructible superpolynomial function. Then*

- $\mathrm{NTIME}(T(n))^{\mathrm{NP}} \not\subseteq \mathrm{P/poly}$.
- $\mathrm{DTIME}(T(n))^{\mathrm{PP}} \not\subseteq \mathrm{P/poly}$.

A further improvement was reported by Köbler and Watanabe, who showed that even $\mathrm{ZPTIME}(T(n))^{\mathrm{NP}}$ is not contained in P/poly [KW]. (Here, $\mathrm{ZPTIME}(T(n))$ is zero-error probabilistic time $T(n)$.)

Is this the best that we can do? To the best of my knowledge, it is not known if the classes $\mathrm{PrTIME}(2^{\log^{O(1)} n})$ (unbounded error probabilistic quasipolynomial time) and $\mathrm{DTIME}(2^{n^{O(1)}})^{\mathrm{C_{=}P}}$ are contained in P/poly (even relative to an oracle). There are oracles relative to which $\mathrm{DTIME}(2^{n^{O(1)}})^{\mathrm{NP}}$ *has* polynomial-size circuits [Hel86, Wil85], thus showing that relativizable techniques cannot be used to present superpolynomial circuit size bounds for $\mathrm{NTIME}(2^{n^{O(1)}})$. Note, however that nonrelativizing techniques have been used on closely-related problems [BFNW93]. More to the point, as reported in [KW], Buhrman and Fortnow and also Thierauf have shown that the exponential-time version of the complexity class MA contains problems outside of P/poly, although this is *false relative to some oracles.* (In particular, this shows that PrTIME($2^{n^{O(1)}}$) is not contained P/poly.) One can hope that further insights will lead to more progress on this front.

In the mean time, it has turned out to be very worthwhile to consider some important subclasses of P/poly.

2 Smaller Circuit Classes

We will focus our attention on five important circuit complexity classes:[1]

1. AC^0 is the class of problems solvable by polynomial-size, constant-depth circuits of AND, OR, and NOT gates of unbounded fan-in. AC^0 corresponds to $O(1)$-time computation on a parallel computer, and it also consists exactly of the languages that can be specified in first-order logic [Imm89, BIS90]. AC^0 circuits are powerful enough to add and subtract n-bit numbers.

2. NC^1 is the class of problems solvable by circuits of AND, OR, and NOT gates of fan-in two and depth $O(\log n)$. NC^1 circuits capture exactly the circuit complexity required to evaluate a Boolean formula [Bus93], and to recognize a regular set [Bar89]. There are deep connections between circuit complexity and algebra, and NC^1 corresponds to computation over any *non-solvable* algebra [Bar89].

3. ACC^0 is the class of problems solvable by polynomial-size, constant-depth circuits of unbounded fan-in AND, OR, NOT, and MODm gates. (A MODm gate takes inputs $x_1, \ldots, x_n$ and determines if the number of 1's among these inputs is a multiple of m.) To be more precise, $\mathrm{AC}^0(m)$ is the class of problems solvable by polynomial-size, constant-depth circuits of unbounded fan-in AND, OR, NOT, and MODm gates, and $\mathrm{ACC}^0 = \bigcup_m \mathrm{AC}^0(m)$. In the algebraic theory mentioned above, ACC^0 corresponds to computation over any *solvable* algebra [BT88]. Thus in the algebraic theory, ACC^0 is the most natural subclass of NC^1.

4. TC^0 is the class of problems solvable by polynomial-size, constant-depth threshold circuits. TC^0 captures exactly the complexity of integer multiplication and division, and sorting [CSV84]. Also, TC^0 is a good complexity-theoretic model for "neural net" computation [PS88, PS89].

5. NC^0 is the class of problems solvable by circuits of AND, OR, and NOT gates of fan-in two and depth $O(1)$. Note that each output bit can only depend on $O(1)$ input bits in such a circuit. Thus any function in NC^0 is computed by depth two AC^0 circuits, merely using DNF or CNF expansion.

[1]Thus this survey will ignore the large body of beautiful work on the circuit complexity of larger subclasses of P and NC.

NC^0 is obviously extremely limited; such circuits cannot even compute the logical OR of n input bits. One of the surprises of circuit complexity is that, in spite of its severe limitations, NC^0 is in some sense quite "close" to AC^0 in computational power.

Quite a few powerful techniques are known for proving lower bounds for AC^0 circuits; it is known that AC^0 is properly contained in ACC^0. It is not hard to see that $ACC^0 \subseteq TC^0 \subseteq NC^1$. As we shall see below, weak lower bounds have been proven for ACC^0 and TC^0, whereas almost nothing is known for NC^1.

3 $\mathbf{AC^0}$

A dramatic series of papers in the 1980's [Ajt83, FSS84, Cai89, Yao85, Hås87] gave us a proof that AC^0 circuits require exponential size even to determine if the number of 1's in the input is odd or even. (See also the excellent tutorial [BS90].) The main tool in proving this and other lower bounds for AC^0 is Håstad's Switching Lemma, one version of which states that most of the "sub-functions" of any AC^0 function f are in NC^0. (A sub-function of f is obtained by setting most of the n input bits to 0 or 1, leaving a function of the n^ϵ remaining unset bits. Such a sub-function is called a restriction of f.) An interesting new proof of the Switching Lemma was presented by [Raz95] (see also [FL95, AAR]), and further extensions were presented by [Bea], the latter motivated in particular by the usefulness of the Switching Lemma as a tool in proving bounds on the length of propositional proofs.

Although the switching lemma is the most powerful tool we have for proving lower bounds for AC^0, it is not the only one. Lower bound arguments were presented in [Rad94, HJP93] for depth three circuits, and a notion of deterministic restriction was presented in [CR96] that is useful for proving nonlinear size bounds.

It is important to note that, although the Switching Lemma tells us that any function f in AC^0 is "close to" functions computed by depth two circuits (since most restrictions of f are computed in depth two), it also provides the tools to show that for all k, there are depth $k+1$ circuits of linear size that require exponential size to simulate with depth k circuits [Hås87]. This is in sharp contrast to the class of circuits considered in the next section, where efficient depth reduction is possible.

The Switching Lemma also provides extremely strong bounds on the difficulty of approximating the parity function (in the sense of giving the correct answer more than half of the time). This enabled Nisan and Wigderson [NW94] to construct, for any k, a pseudorandom generator that is (a) computable in AC^0, and (b) takes $\log^{O(1)} n$ bits of input and produces n bits of output, and (c) is secure against statistical tests computed by depth k AC^0 circuits. (That is, any depth k circuit has essentially the same probability of accepting when the input is the "pseudorandom" output of f, as when the input is a random string of length n.) This has many applications in derandomization. For instance, given a depth k circuit C_n, the Switching Lemma tells us that a randomly-chosen restriction ρ will simplify C_n to a depth two circuit. Can such a ρ be found quickly *deterministically*? The Nisan-Wigderson generator easily provides an algorithm running in time $2^{\log^{O(1)} n}$: Note that the set $\{(C, \rho) : C \text{ is a depth } k \text{ circuit and } C_\rho \text{ is a depth two circuit}\}$ is in AC^0. Set C to C_n and letting ρ

be random; with high probability the AC^0 circuit accepts. Thus with high probability the circuit also accepts when ρ is pseudorandom. Since there are only $2^{\log^{O(1)} n}$ pseudorandom strings, this set can be searched exhaustively.

It is important to note that, although strong "non-approximability" bounds are known for some other classes of circuits (as we will see below), as of yet the AC^0 lower bounds are the only ones that are strong enough to allow use of the Nisan-Wigderson construction. Pseudorandomness for AC^0 was further studied by Sitharam [Sit95], who related pseudorandomness to polylog(n)-wise independence.

Although AC^0 circuits can produce output that looks pseudorandom to other AC^0 circuits, AC^0 *lacks* the ability to compute pseudorandom function generators for general polynomial-time computations; this was proved in [LMN93] as a corollary to their main results analyzing the Fourier spectrum of AC^0 functions. (This relates to one of the many ways of representing functions by polynomials; for a survey, see [Bei93].)

4 Toward ACC^0 and TC^0

Algbraic considerations are what led to ACC^0 being identified as an object of study, and algebraic tools are what led to the lower bounds for the classes $AC^0(p)$ that form the basic building blocks of ACC^0. Smolensky [Smo87] (building on the work of Razborov [Raz87]) showed that if m is not a power of prime p, then the MODm function is is not in $AC^0(p)$. This can be proved by combining two arguments:

- Any depth k $AC^0(p)$ circuit is equivalent to a depth two probabilistic circuit with a single MODp output gate and $n^{\log^{O(1)} n}$ AND gates with polylogarithmic fan in on the bottom level.[2] This sort of depth-reduction stands in sharp contrast to the fact that AC^0 circuits of depth k cannot in general be simulated by depth $k-1$ circuits without an exponential blow-up in size. This sort of depth reduction is studied in more detail in [AH94, KVVY93, Tar93, ABFR94]. Since this probabilistic depth two circuit is equivalent to the original circuit, there is some setting of the probabilistic bits that can be used to obtain a deterministic circuit (equivalently, a polynomial over GFp having degree $\log^{O(1)} n$) that agrees with the original circuit on a large number of inputs.

- The MODm function cannot agree with any low-degree polynomial over GFp^ℓ on very many inputs (for any constant ℓ).

(Also see the presentation of Smolensky's proof in [BS90]. A very different proof was later published, again by Smolensky [Smo93].)

It is encouraging that something very like the depth-reduction to depth two circuits holds also for all of ACC^0. The results of [Yao90, BT94] show that every set in ACC^0 is recognized

[2] Since so many theorems about constant-depth circuits provide simulating circuits of "quasipolynomial" size (that is, size $2^{\log^{O(1)} n}$), Barrington has given a framework where quasipolynomial circuit size is studied, instead of polynomial size [Bar92].

by a depth two deterministic circuit with $n^{\log^{O(1)} n}$ AND gates at the input level, and a single symmetric gate at the root. Circuits of this sort, called SYM^+ circuits because they are in some sense only "a bit" more powerful than a single symmetric gate, were shown to be able to simulate an even larger class of circuits in [BTT92]. Later work by [GKR+95] shows that the symmetric gate can be chosen to be the "middle bit" function (that outputs the middle bit of the number r, where r is the number of inputs to the gate that evaluate to 1).

Unfortunately, no analog to the second part of Smolensky's argument is known to hold when p is replaced by a composite number. Although initially it seems that a MOD6 gate should not be significantly more useful than a MOD7 gate for computing functions such as the MAJORITY function (or SAT), this has not been established. Indeed, as we shall see in the next section, there are certain settings where composite moduli are provably more powerful than prime moduli.

It remains unknown if there is any problem in $\mathrm{DTIME}(2^{n^{O(1)}})^{\mathrm{NP}}$ that is not in ACC^0. Even worse, it is not known if any problem in $\mathrm{DTIME}(2^{n^{O(1)}})^{\mathrm{NP}}$ requires more than polynomial size to compute on depth three circuits consisting only of MOD6 gates!

5 Special Cases: Depth Two and Depth Three

In this section we try to survey the recent work attacking special cases of ACC^0 and TC^0 circuits. Although there are many incomparable results (and some have probably been overlooked) there a few main streams of work that have developed.

5.1 Low-Degree Polynomials

A great deal of the work on ACC^0 and its subclasses deals with simulating circuits (in one of several ways) by polynomials of low degree. We refer the reader to the survey by Beigel [Bei93] for better coverage of this topic. Here, we will pick out only a few ideas and recent developments.

Most work on simulating circuits by polynomials concentrates on the *degree* of the polynomial as the relevant measure of complexity. It was shown in [NS94] that this measure corresponds to Boolean decision tree complexity (and remains roughly the same regardless of whether the function is being computed exactly by the polynomial or only "approximately" for one notion of "approximation"). The degree required to compute a function is fairly robust to changes in representation (for example, should {YES,NO} correspond to {1,0} or to {-1,1}, ...). Recently the size of a polynomial (i.e., the number of terms) has also been studied, and it has been shown that this is more sensitive to the choice of representation [KP96].

Low-degree polynomials over the reals can be simulated by circuits consisting of a single MAJORITY gate with small-fan-in AND gates. (These are so-called MAJ^+ circuits, also called (generalized) perceptrons.) A sequence of papers including [ABFR94, Bei94] led to the result that an AND, OR, NOT circuit with $n^{o(1)}$ MAJORITY gates can be "efficiently" simulated by depth two circuits with a single MAJORITY gate at the output, with small fan-in AND gates at the input level. This bound is shown to be optimal in [ZBT93], where a characterization of the symmetric functions computed by MAJ^+ circuits is given.

This simulation, combined with arguments about the degree required to compute the MOD2 function (even approximately), shows that any AND, OR, NOT circuit with $n^{o(1)}$ MAJORITY gates requires exponential size to compute MOD2. Barrington and Straubing [BS94] generalize this to MODm for any m.

5.2 The Surprising Power of Composite Moduli

Several papers have shown senses in which MODm gates for composite m (m not a prime power) have more computational power than MODp gates for prime p. For instance, using communication complexity, Grolmusz [Gro95a] presented a function computable by depth two MODm circuits than cannot be computed by depth two MODp circuits.

Continuing the line of work simulating circuits by polynomials, Barrington, Beigel, and Rudich defined the MODm degree of a function as the minimal degree required to represent the function over the ring of integers MODm [BBR94]. Although the OR function has degree $n/(p-1)$ for prime p, it is shown in [BBR94] that for composite m the degree MODm is $O(\sqrt[r]{n})$, where r is the number of prime factors of m.

The definition of MODm degree is rather delicate, which led the authors of [BBR94] to define a related notion called "weak degree MODm" that is more robust to slight changes. Tardos and Barrington gave the first lower bounds for weak degree [TB95]. Later these results were extended by Grolmusz [Gro95b], who also studied the size (number of monomials) required to represent a function mod m.

[BBR94] also presents a lower bound for the MODm degree of the MODm' function. This bound is improved in [Tsa96] and again by Green in [Gre]. Green uses this bound to partially extend the lower bounds of [BS94], showing that the MODq function requires exponential size to compute on depth three circuits with an exact threshold gate at the output, MODp gates on the middle level, and small-fan-in AND gates at the inputs.

5.3 Solvability versus Nonsolvability

Another body of lower bounds comes directly from the algebraic characterization of ACC^0. (For more background about this approach to circuit complexity, see [MPT91, Lem96].) Recall that few lower bounds are known even for circuits consisting *only* of MOD6 gates. A natural conjecture is that these circuits cannot compute the AND function (just as AC^0 circuits cannot compute MOD6). The first lower bounds in this direction appear in [BST90], where the authors show that programs over a particular class of groups need exponential size in order to compute the AND function. This is translated into a lower bound for a certain kind of depth two circuits of MOD gates by Caussinus [Cau96, Cau]. A related lower bound is provided by [YP94], showing that a class of restricted depth three circuits also cannot compute the AND.

Finally, a different sort of bound on the complexity of computing the AND with MOD gates is given by [Thé], who shows that such circuits must have at least a linear number of gates on the input level.

(It is appropriate to also mention [BS95], which does not provide a circuit lower bound *per se*, but does provide a nonlinear bound on the ACC^0 *formula* size, using algebraic techniques.)

5.4 Low Levels of the TC^0 Hierarchy

The first important lower bound for threshold circuits is still one of the best. Using the techniques of communication complexity, [HMP+93] shows that depth three MAJORITY circuits are exponentially more powerful than depth two circuits. (Extensions may be found in [Kra91, KW95].)

It is important to note that there are many different decompositions of TC^0 that are useful, depending on whether the basic gates are MAJORITY, exact threshold, or weighted threshold, etc., or alternatively if AC^0 circuitry is considered "cheap" and only applications of MAJORITY are considered expensive [MT93, Mac95]. Many separations are known among the various low levels; a good survey of these separations and inclusions is found in [Raz92]. See also [GHR92, GK93, Hof96] and the articles in [RSO94].

The state of the art in this direction still only yields superpolynomial bounds for restricted classes of depth two or depth three circuits:

- Threshold-of-MODm [KP94], extending [Gol95]. An alternate proof is presented in [ES].
- Depth Three MAJORITY circuits where the middle level is AND [HG91, RW93].
- Depth Three MAJORITY circuits where the bottom level is AND [Gro94].

Sitharam presents a unified framework in which many of these bounds can be obtained [Sit]. Limitations of some of these techniques are discussed in [RSOK95]. A different technique (yielding weaker lower bounds) is presented in [Juk95].

Finally, it is appropriate to mention two other streams of work that may be viewed as initial steps for proving circuit lower bounds (although they do not explicitly yield circuit bounds in their current forms). TC^0 can be characterized in terms of first-order logic with counting quantifiers [BIS90]. Work such as that of Etessami on non-expressibility [Ete] can be viewed as providing a limited circuit lower bound. As another example, [HNW93] proves a result for read-once formulae, whose extension to general formulae would provide lower bounds for TC^0.

6 A Large Obstacle to Progress

The question of whether ACC^0 circuits can compute MAJORITY has now been considered for a decade, and has withstood all attacks so far. Similarly, the question of whether $\mathrm{TC}^0 = \mathrm{NC}^1$ remains open in spite of considerable attention. Although it is at least conceivable that some variation on a known proof technique will suffice to prove lower bounds for ACC^0, there is strong evidence that a radically different approach will be necessary to prove lower bounds for TC^0. This evidence comes from the work on "Natural Proofs" by Razborov and Rudich [RR94].

Razborov and Rudich formulate a notion of lower bound proof that is general enough to include all of the papers dealing with constant-depth circuits cited in the preceding sections. They show that if there is a proof of this sort (which they call a "Natural Proof") proving that NP is not contained in TC^0, then there are no cryptographically-secure functions computed in TC^0. But cryptographers believe that there are cryptographically secure functions computable in TC^0 [IN89].[3] If they are right, then [RR94] shows that any proof showing $\mathrm{TC}^0 \neq \mathrm{NP}$ must look quite unlike any circuit lower bound proof that has been seen yet: an "unnatural" proof.

On the other hand, there is no strong evidence for the existence of cryptographically-secure functions in ACC^0. Thus the results of [RR94] do not seem to indicate any obstacles to answering whether $\mathrm{ACC}^0 = \mathrm{TC}^0$.

7 An Old and Unnatural Proof Technique

Razborov and Rudich were careful to argue that [RR94] should not be taken as a cause for pessimism. Rather, it should serve as a guide indicating which approaches to rule out. Certainly, many complexity theoreticians are trying to formulate arguments that avoid the problems faced by Natural Proofs.

It should be noted that one of the oldest and most powerful weapons in the arsenal of complexity theory does, in fact, yield "unnatural" proofs: Diagonalization. Unfortunately, diagonalization is not well-suited for arguments about circuit complexity, since diagonalization proceeds by satisfying a countable number of requirements (such as: Requirement i: language A is not accepted by machine i), and there are uncountably many circuit families. This problem can be side-stepped by considering only *uniform* families of circuits; circuit family $\{C_n\}$ is *uniform* if there is an efficient algorithm for the mapping $n \mapsto C_n$. We will follow the lead of [BIS90] and use "Dlogtime" uniformity. (For the purposes of this survey it will not be necessary to deal with the details of the definition of the uniformity condition.)

Since uniform TC^0 is contained in $\mathrm{DSPACE}(\log n) \neq \mathrm{PSPACE}$, it is easy to show that the standard PSPACE-complete sets require exponential-size uniform TC^0 circuits. (Similarly, although we don't know if $\mathrm{DTIME}(2^n)$ has polynomial-size circuits, it is easy to show that it requires exponential-size *uniform* circuits.) Can we improve on these trivial bounds?

The first paper to make explicit use of uniformity in proving a circuit lower bound was [AG94]. There, we showed that computing the permanent of a matrix requires size at least 2^{n^ϵ} on ACC^0 circuits. The proof combines the circuit simulations of [Yao90, BT94] with

[3] This is perhaps the appropriate place to mention that it has been conjectured that TC^0 is in fact equal to NC^1 [IL95]. Certainly it seems possible to do much more significant computation in TC^0 than in ACC^0, although [BC91] does present some natural computational problems that are in ACC^0. It has also been conjectured that TC^0 and NC^1 are *not* equal [BC89].

diagonalizations. We were able to prove lower bounds showing that complete sets for PP and $C_=P$ are hard for ACC^0 circuits to compute, too – but our size bounds are weaker there than for the permanent. If $T(T(n)) < 2^n$, then a complete set for PP requires more than size $T(n)$ to compute on uniform ACC^0 circuits. (In [AG94] this is called a *sub*-subexponential size bound. Note that this is still much larger than, say, $n^{\log n}$.)

No lower bounds for uniform TC^0 were presented in [AG94]. The first bounds of that sort were presented in [CMTV96]; there it was proved that there is a set in the counting hierarchy that requires superpolynomial size to compute on uniform TC^0 circuits. (The counting hierarchy is the union of the sequence PP, PP^{PP}, $PP^{PP^{PP}}$, ... However, the proof in [CMTV96] did not give a clue as to *which* set in the counting hierarchy would be hard, and the size bound was only superpolynomial, and not even, say, $n^{\log^* n}$. In [All] I build on [CMTV96] to show lower bounds for the permanent and for the standard complete sets for PP: these problems all require more than size $T(n)$ to compute on uniform TC^0 circuits, if $T(T(\ldots(T(n))\ldots)) = o(2^n)$ (for any constant number of compositions). Note that, although this size bound is smaller than the bound in [AG94], it is for a more powerful class of circuits.

An obvious question is whether these bounds can be improved. Can the hard sets for PP really be that much easier than the permanent? Do the hard sets for PP require exponential size on uniform TC^0 circuits? Is it possible to apply these techniques to show that smaller complexity classes require superpolynomial size TC^0 circuits? (Other applications of diagonalization in uniform circuit complexity have been presented recently by [II96].)

8 Stronger Separations from AC^0

In order to define a framework where the probabilistic method might conceivably be applied to questions about countable classes in complexity theory, Lutz defined a notion of resource-bounded measure [Lut92]. With this notion it is possible to talk in a meaningful way about whether NP is a "large" or "small" subset of DTIME($2^{n^{O(1)}}$). Several papers have been written (e.g., [LM94]) considering the hypothesis "NP is not a measure-zero subset of DTIME($2^{n^{O(1)}}$)" as a likely complexity-theoretic hypothesis, in the same way that "the polynomial hierarchy does not collapse" and "P $\neq$ NP" are used as likely complexity-theoretic hypotheses.

In order to investigate this hypothesis, one step would be to consider the analogous question "scaled down" to polynomial time. That is, is NTIME($\log n$) a measure zero subset of P? An initial obstacle to overcome is that the definition of measure provided by Lutz does not extend in any obvious way to classes smaller than P. Nonetheless, a notion of measure on P that generalizes Lutz's notion was defined in [AS94], thus successfully overcoming this first obstacle.

Using many techniques developed for proving lower bounds for AC^0, Cai, Sivakumar, and Strauss [CSS, Siv96] succeeded in showing that not only is NTIME($\log n$) a measure zero subset of P, but in fact all of AC^0 has measure zero in P (using a notion of measure that differs only slightly from that of [AS94]). This is the most exciting application of resource-bounded measure on P thus far, and it also gives cause to reconsider how likely it is that NP is a measure zero subset of DTIME($2^{n^{O(1)}}$).

There are also results (using a slightly different notion of measure) showing that $\mathrm{AC}^0(2)$ does *not* have P-measure zero [Siv96]. Since the Nisan-Wigderson pseudorandom generator is an important tool in proving the measure zero result of [CSS], it is tempting to speculate that there is a connection between these contrasting measure results for AC^0 and $\mathrm{AC}^0(2)$, and our inability thus far to construct pseudorandom generators for $\mathrm{AC}^0(2)$.

Finally, it is interesting to note that Lutz's notion of measure is actually quite closely related to notion of Natural Proof presented by [RR94]; it was shown in [RSC95] that the existence of a natural proof showing that a problem is not in some class $\mathcal{C}$ corresponds (roughly) to an argument that the class $\mathcal{C}$ is a "small" complexity class in the sense of resource-bounded measure. These connections are still not understood as well as they should be.

9 Constant-Depth Reducibility

Thus far in this survey, I have concentrated on that aspect where progress in complexity theory has been most modest: proving lower bounds. Complexity theory has been incredibly successful on another front, however. For the overwhelming majority of natural computational problems that arise in practice, there is a natural complexity class for which that problem is complete. Thus complexity theory has been very successful at classifying and characterizing the complexity of problems in terms of reducibility, completeness, and complexity classes. In this section I will discuss how to use the techniques of circuit complexity to build on this strength.

There are many important and natural problems that are in NC^1 and are in no strictly smaller complexity class; there are also many important and natural problems that are in DSPACE($\log n$) and are in no strictly smaller complexity class. These problems are "complete" for NC^1 and DSPACE($\log n$), respectively – but in order to make this precise we need a notion of completeness. Arguably the most natural notion of completeness for classes such as these is the notion given by AC^0 many-one reductions. In fact, as was pointed out by [AG91], the first time AC^0 was studied in complexity theory was precisely for this purpose [Jon75]. Also, the first-order translations of Immerman [Imm87] (which provide a notion of completeness defined entirely in terms of logic) correspond to AC^0 reductions.

Finally, it is an empirical fact that the NP-complete problems that one encounters in practice are all complete under AC^0 reductions. In fact, it is not known if there is any complexity class larger than P for which there is a set complete under polynomial-time reductions but not under AC^0 reductions. Thus we lose nothing of practical importance if we re-define all notions of NP-completeness and completeness for other classes solely in terms of AC^0 reductions.

Here are two rather startling facts about complete sets under AC^0 reducibility.

Theorem 9.1 *[AAR] Let $\mathcal{C}$ be any complexity class closed under TC^0 reductions. (Thus $\mathcal{C}$ can be P or NP or NC^1 or PP, etc.)*

1. *All sets complete for $\mathcal{C}$ under AC^0 reductions are complete under NC^0 reductions.*
2. *All sets complete for $\mathcal{C}$ under AC^0 reductions are isomorphic under isomorphisms computable and invertible by depth three AC^0 circuits.*

The first theorem is a sort of "Gap" theorem. It says that, although NC^0 is much weaker than AC^0, AC^0 reductions do not yield any more NP-complete sets than NC^0 reductions do. A natural and important open question asks how large this "gap" is. Are all sets complete under polynomial-time reductions also complete under NC^0 reductions? If so, then $P \neq NP$ (because this would imply that all NP-complete sets are P/poly isomorphic,[4] which implies that no finite set is NP-complete, and hence $P \neq NP$). If the "gap" is not that large, then how far does it extend?

The second theorem is an analog of the Berman-Hartmanis conjecture. It says that there are unexpected similarities among the NP-complete sets. It is particularly striking when one considers a function f computable in AC^0 that produces output that is pseudorandom to depth three AC^0 circuits. Theorem 9.1 says that $f(SAT)$ is isomorphic to SAT via an isomorphism that is provably too weak to distinguish meaningful inputs from noise.

10 Conclusions

The fundamental questions of complexity theory are important, and they won't go away. Let us never forget that all cryptosystems in existence today are based on conjecture and wishful thinking. Before we can have confidence that a cryptosystem is secure, it will be necessary to have non-asymptotic bounds on the average case complexity of problems. Before such bounds can be achieved, the fundamental and basic questions (such as $NP \not\subseteq TC^0$) will need to be resolved. For some of those questions, we need to have "unnatural" proof techniques.

Until these "unnatural" proofs are developed and usher in the new millennium, there are still significant and interesting advances in our understanding that are possible and amenable to the tools of circuit complexity. I have mentioned three areas that are close to my own research: (a) obtaining lower bounds for uniform circuits, (b) obtaining measure-based separation of circuit complexity classes, and (c) studying reducibilities defined in terms of circuit classes.

References

[AA96] M. Agrawal and E. Allender. An isomorphism theorem for circuit complexity. In *IEEE Conference on Computational Complexity*, pages 2–11, 1996.

[4] The isomorphism is P/poly instead of P-isomorphism, because Theorem 9.1 actually holds only for non-uniform AC^0 and NC^0 reductions. It is natural to ask if Theorem 9.1 holds also for uniform circuits. [AAR] partially answers this by showing that the "gap" theorem does *not* hold for uniform circuits.

[AAR] M. Agrawal, E. Allender, , and S. Rudich. Reductions in circuit complexity: An isomorphism theorem and a gap theorem. submitted; preliminary version appeared as [AA96].

[ABFR94] J. Aspnes, R. Beigel, M. Furst, and S. Rudich. The expressive power of voting polynomials. *Combinatorica*, 14:135–148, 1994.

[AG91] E. Allender and V. Gore. Rudimentary reductions revisited. *Information Processing Letters*, 40:89–95, 1991.

[AG94] E. Allender and V. Gore. A uniform circuit lower bound for the permanent. *SIAM Journal on Computing*, 23:1026–1049, 1994.

[AH94] E. Allender and U. Hertrampf. Depth reductions for circuits of unbounded fan-in. *Information and Computation*, 112:217–238, 1994.

[Ajt83] M. Ajtai. Σ^1_1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24:1–48, 1983.

[All] E. Allender. The permanent requires large uniform threshold circuits. Submitted. A preliminary version of this paper appeared as [All96].

[All96] E. Allender. A note on uniform circuit lower bounds for the counting hierarchy. In *International Conference on Computing and Combinatorics Conference (COCOON)*, volume 1090 of *Lecture Notes in Computer Science*, pages 127–135. Springer-Verlag, 1996.

[AS94] E. Allender and M. Strauss. Measure on small complexity classes, with applications for BPP. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 807–818, 1994.

[Bar89] D. A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences*, 38:150–164, 1989.

[Bar92] D. A. Mix Barrington. Quasipolynomial size circuit classes. In *IEEE Structure in Complexity Theory Conference*, pages 86–93, 1992.

[BBR94] D. A. Mix Barrington, R. Beigel, and S. Rudich. Representing Boolean functions as polynomials modulo composite numbers. *Computational Complexity*, 4:367–382, 1994.

[BC89] D. A. Mix Barrington and J. Corbett. On the relative complexity of some languages in NC^1. *Information Processing Letters*, 32:251–256, 1989.

[BC91] D. A. Mix Barrington and J. Corbett. A note on some languages in uniform ACC^0. *Theoretical Computer Science*, 78:357–362, 1991.

[Bea] P. Beame. A switching lemma primer. manuscript, available from `http://www.cs.washington.edu/homes/beame/papers.html`.

[Bei93] R. Beigel. The polynomial method in circuit complexity. In *IEEE Structure in Complexity Theory Conference*, pages 82–95, 1993.

[Bei94] R. Beigel. When do extra majority gates help? polylog(n) majority gates are equivalent to one. *Computational Complexity*, 4:314–324, 1994.

[BFNW93] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[BH92] H. Buhrman and S. Homer. Superpolynomial circuits, almost sparse oracles and the exponential hierarchy. In *Foundations of Software Technology and Theoretical Computer Science (FST&TCS)*, volume 652 of *Lecture Notes in Computer Science*, pages 116–127. Springer-Verlag, 1992.

[BIS90] D. A. Mix Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41:274–306, 1990.

[BS90] R. Boppana and M. Sipser. The complexity of finite functions. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science (Vol. A: Algorithms and Complexity)*. Elsevier and MIT Press, 1990.

[BS94] D. A. Mix Barrington and H. Straubing. Complex polynomials and circuit lower bounds for modular counting. *Computational Complexity*, 4:325–338, 1994.

[BS95] D. A. Mix Barrington and H. Straubing. Superlinear lower bounds for bounded-width branching programs. *Journal of Computer and System Sciences*, 50:374–381, 1995.

[BST90] D. A. Mix Barrington, H. Straubing, and D. Thérien. Non-uniform automata over groups. *Information and Computation*, 89:109–132, 1990.

[BT88] D. A. Mix Barrington and D. Thérien. Finite monoids and the fine structure of NC^1. *Journal of the ACM*, 35:941–952, 1988.

[BT94] R. Beigel and J. Tarui. On ACC. *Computational Complexity*, 4:350–367, 1994.

[BTT92] R. Beigel, J. Tarui, and S. Toda. On probabilistic ACC circuits with an exact-threshold output gate. In *Proceedings of the 3rd ACM-SIGSAM International Symposium on Symbolic and Algebraic Computation (ISAAC)*, volume 650 of *Lecture Notes in Computer Science*, pages 420–429. Springer-Verlag, 1992.

[Bus93] S. Buss. Algorithm for Boolean formula evaluation and for tree contraction. In P. Clote and J. Krajíček, editors, *Arithmetic, Proof Theory, and Computational Complexity*, volume 25 of *Oxford Logic Guides*, pages 96–115. Clarendon Press, 1993.

[Cai89] J. Cai. With probability 1, a random oracle separates PSPACE from the polynomial-time hierarchy. *J. Computer and System Science*, 38:68–85, 1989.

[Cau] H. Caussinus. A note on a theorem of Barrington, Straubing, and Thérien. To appear in *Information Processing Letters.*

[Cau96] H. Caussinus. *Contributions à l'Etude du Non-déterminisme Restreint.* PhD thesis, Université de Montréal, 1996.

[CKS81] A. Chandra, D. Kozen, and L. Stockmeyer. Alternation. *Journal of the ACM*, 28:114–133, 1981.

[CMTV96] H. Caussinus, P. McKenzie, D. Thérien, and H. Vollmer. Nondeterministic NC^1 computation. In *Proceedings, 11th Annual IEEE Conference on Computational Complexity*, pages 12–21, 1996.

[CR96] S. Chaudhuri and J. Radhakrishnan. Deterministic restrictions in circuit complexity. In *ACM Symposium on Theory of Computing (STOC)*, pages 30–36, 1996.

[CSS] J.-Y. Cai, D. Sivakumar, and M. Strauss. Constant-depth circuits and the Lutz hypothesis. Manuscript.

[CSV84] A. Chandra, L. Stockmeyer, and U. Vishkin. Constant depth reducibility. *SIAM Journal on Computing*, 13:423–439, 1984.

[ES] P. Enflo and M. Sitharam. Stable basis families and complexity lower bounds. Submitted.

[Ete] K. Etessami. Counting quantifiers, successor relations, and logarithmic space. To appear in Journal of Computer and System Sciences. Preliminary version appeared in IEEE Structure in Complexity Theory Conference, 1995, pp. 2-11.

[FL95] L. Fortnow and S. Laplante. Circuit lower bounds *à la* Kolmogorov. *Information and Computation*, 123:121–126, 1995.

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

[GHR92] M. Goldmann, J. Håstad, and A. A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2:277–300, 1992.

[GK93] M. Goldmann and M. Karpinski. Simulating threshold circuits by majority circuits. In *ACM Symposium on Theory of Computing (STOC)*, pages 551–560, 1993.

[GKR+95] F. Green, J. Köbler, K. Regan, T. Schwentick, and J. Torán. The power of the middle bit of a #P function. *Journal of Computer and System Sciences*, 50:456–467, 1995.

[Gol95] M. Goldmann. A note on the power of majority gates and modular gates. *Information Processing Letters*, 53:321–327, 1995.

[Gre] F. Green. Complex Fourier technique for lower bounds on the Mod-m degree. Submitted. An earlier version appeared as [Gre95].

[Gre95] F. Green. Lower bounds for depth-three circuits with equals and mod-gates. In *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 900 of *Lecture Notes in Computer Science*, pages 71–82. Springer-Verlag, 1995.

[Gro94] V. Grolmusz. A weight-size trade-off for circuits with MOD m gates. In *ACM Symposium on Theory of Computing (STOC)*, 1994.

[Gro95a] V. Grolmusz. Separating the communication complexities of MOD m and MOD p circuits. *Journal of Computer and System Sciences*, 51:307–313, 1995.

[Gro95b] Vince Grolmusz. On the weak mod m representation of Boolean functions. *Chicago Journal of Theoretical Computer Science*, 1995(2), July 1995.

[Hås87] J. Håstad. *Computational Limitations for Small Depth Circuits.* MIT Press, Cambridge, MA, 1987.

[Hel86] H. Heller. On relativized exponential and probabilistic complexity classes. *Information and Computation*, 71:231–243, 1986.

[HG91] J. Håstad and M. Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:113–129, 1991.

[HJP93] J. Håstad, S. Jukna, and P. Pudlák. Top-down lower bounds for depth 3 circuits. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 124–129, 1993.

[HMP+93] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46:129–154, 1993.

[HNW93] R. Heiman, I. Newman, and A. Wigderson. On read-once threshold formulae and their randomized decision tree complexity. *Theoretical Computer Science*, 107:63–76, 1993.

[Hof96] T. Hofmeister. A note on the simulation of exponential threshold weights. In *International Conference on Computing and Combinatorics (COCOON)*, volume 1090 of *Lecture Notes in Computer Science*, pages 136–141. Springer-Verlag, 1996.

[II96] K. Iwama and C. Iwamoto. Parallel complexity hierarchies based on PRAMs and DLOGTIME-uniform circuits. In *IEEE Conference on Computational Complexity*, pages 24–32, 1996.

[IL95] N. Immerman and S. Landau. The complexity of iterated multiplication. *Information and Computation*, 116:103–116, 1995.

[Imm87] N. Immerman. Languages which capture complexity classes. *SIAM J. Comput.*, 4:760–778, 1987.

[Imm89] N. Immerman. Expressibility and parallel complexity. *SIAM Journal on Computing*, 18:625–638, 1989.

[IN89] R. Impagliazzo and M. Naor. Efficient cryptographic schemes provably as secure as subset sum. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 236–243, 1989.

[Jon75] N. Jones. Space-bounded reducibility among combinatorial problems. *Journal of Computer and System Sciences*, 11:68–85, 1975. Corrigendum: *Journal of Computer and System Sciences* 15:241, 1977.

[Juk95] S. Jukna. Computing threshold functions by depth-3 threshold circuits with smaller thresholds of their gates. *Information Processing Letters*, 56:147–150, 1995.

[Kan82] R. Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55:40–56, 1982.

[KP94] M. Krause and P. Pudlák. On the computational power of depth 2 circuits with threshold and modulo gates. In *ACM Symposium on Theory of Computing (STOC)*, pages 48–57, 1994.

[KP95] M. Krause and P. Pudlák. On computing Boolean functions by sparse real polynomials. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 682–691, 1995.

[KP96] M. Krause and P. Pudlák. More on computing Boolean functions by sparse real polynomials and related types of threshold circuits. Technical Report 622, University of Dortmund, 1996. A preliminary version appeared as [KP95].

[Kra91] M. Krause. Geometric arguments yield better bounds for threshold circuits and distributed computing. In *Structure in Complexity Theory Conference*, pages 314–321, 1991.

[KVVY93] R. Kannan, H. Venkateswaran, V. Vinay, and A. Yao. A circuit-based proof of Toda's theorem. *Information and Computation*, 104:271–276, 1993.

[KW] J. Köbler and O. Watanabe. New collapse consequences of np having small circuits. To appear in SIAM Journal on Computing. Preliminary version appeared in Proc. International Colloquium on Automata, Languages and Programming (ICALP), Springer-Verlag, Lecture Notes in Computer Science 944, 1995, pp. 196-207.

[KW95] M. Krause and S. Waack. Variation ranks of communication matrices and lower bounds for depth two circuits having symmetric gates with unbounded fan-in. *Mathematical Systems Theory*, 28:553–564, 1995.

[Lem96] F. Lemieux. *Finite Groupoids and their Applications to Computational Complexity.* PhD thesis, McGill University, 1996.

[LM94] J. Lutz and E. Mayordomo. Cook versus Karp-Levin: Separating completeness notions if NP is not small. In *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 775 of *Lecture Notes in Computer Science*, pages 415–426. Springer-Verlag, 1994.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM*, 40:607–620, 1993.

[Lut92] J. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44:220–258, 1992.

[Mac95] A. Maciel. *Threshold Circuits of Small Majority-Depth.* PhD thesis, McGill University, 1995.

[MPT91] P. McKenzie, P. Péladeau, and D. Thérien. NC^1: The automata-theoretic viewpoint. *Computational Complexity*, 1:330–359, 1991.

[MT93] A. Maciel and D. Thérien. Threshold circuits for iterated multiplication: Using AC^0 for free. In *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 665 of *Lecture Notes in Computer Science*, pages 545–554. Springer-Verlag, 1993.

[NS94] N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4:301–313, 1994.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.

[PS88] I. Parberry and G. Schnitger. Parallel computation with threshold functions. *Journal of Computer and System Sciences*, 36:278–302, 1988.

[PS89] I. Parberry and G. Schnitger. Relating Boltzmann machines to conventional models of computation. *Neural Networks*, 2:59–67, 1989.

[Rad94] J. Radhakrishnan. $\Sigma\Pi\Sigma$ threshold formulas. *Combinatorica*, 14:345–374, 1994.

[Raz87] A. A. Razborov. Lower bounds on the size of bounded depth networks over a complete basis with logical addition. *Mathematicheskie Zametki*, 41:598–607, 1987. English translation in Mathematical Notes of the Academy of Sciences of the USSR 41:333-338, 1987.

[Raz92] A. A. Razborov. On small depth threshold circuits. In *Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 621 of *Lecture Notes in Computer Science*, pages 42–52. Springer-Verlag, 1992.

[Raz95] A. A. Razborov. Bounded arithmetic and lower bounds. In P. Clote and J. Remmel, editors, *Feasible Mathematics II,*, volume 13 of *Progress in Computer Science and Applied Logic*, pages 344–386. Birkhäuser, 1995.

[RR94] A. A. Razborov and S. Rudich. Natural proofs. In *Proceedings, 26th ACM Symposium on Theory of Computing*, pages 204–213, 1994.

[RSC95] K. Regan, D. Sivakumar, and J.-Y. Cai. Pseudorandom generators, measure theory, and natural proofs. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 26–35, 1995.

[RSO94] V. Roychowdhury, K.-Y. Siu, and A. Orlitsky, editors. *Theoretical Advances in Neural Computation and Learning.* Kluwer, 1994.

[RSOK95] V. Roychowdhury, K.-Y. Siu, A. Orlitsky, and T. Kailath. Vector analysis of threshold functions. *Information and Computation*, 120:22–31, 1995.

[RW93] A. A. Razborov and A. Wigderson. $n^{\Omega(\log n)}$ lower bounds on the size of depth-3 threshold circuits with AND gates at the bottom. *Information Processing Letters*, 45:303–307, 1993.

[Sit] M. Sitharam. Approximation from linear spaces, lower bounds, pseudorandomness, and learning. Submitted.

[Sit95] M. Sitharam. Pseudorandom generators and learning algorithms for AC^0. *Computational Complexity*, 5:248–266, 1995.

[Siv96] D. Sivakumar. *Probabilistic Techniques in Structural Complexity Theory.* PhD thesis, SUNY Buffalo, 1996.

[Smo87] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings, 19th ACM Symposium on Theory of Computing*, pages 77–82, 1987.

[Smo93] R. Smolensky. On representations by low-degree polynomials. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 1993.

[Sto74] L. Stockmeyer. *The complexity of decision problems in automata theory and logic*. PhD thesis, Mass. Inst. of Technology, 1974.

[Sto87] L. Stockmeyer. Classifying the computational complexity of problems. *Journal of Symbolic Logic*, 52:1–43, 1987.

[Tar93] J. Tarui. Probabilistic polynomials, AC^0 functions, and the polynomial-time hierarchy. *Theoretical Computer Science*, 113:167–183, 1993.

[TB95] G. Tardos and D. A. Mix Barrington. A lower bound on the MOD 6 degree of the OR function. In *Proc. 3rd Israel Symposium on the Theory of Computing and Systems*, pages 52–56. IEEE Press, 1995.

[Thé] D. Thérien. Circuits constructed with MOD_q gates cannot compute AND in sublinear size. *Computational Complexity*, 4:383–388, 1994.

[Tod91] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20:865–877, 1991.

[Tsa96] S.-C. Tsai. Lower bounds on representing Boolean functions as polynomials in z_m. *SIAM Journal on Discrete Mathematics*, 9:55–62, 1996.

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

[Yao85] A. Yao. Separating the polynomial-time hierarchy by oracles. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1–10, 1985.

[Yao90] A. Yao. On ACC and threshold circuits. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 619–627, 1990.

[YP94] P.Y. Yan and I. Parberry. Exponential size lower bounds for some depth three circuits. *Information and Computation (formerly Information and Control)*, 112:117–130, 1994.

[ZBT93] Z.-L. Zhang, D. A. Mix Barrington, and J. Tarui. Computing symmetric functions with AND/OR circuits and a single MAJORITY gate. In *10th Annual Symposium on Theoretical Aspects of Computer Science*, volume 665 of *Lecture Notes in Computer Science*, pages 535–544. Springer-Verlag, 1993.