The Polynomial Method in Circuit Complexity

Richard Beigel*
Yale University

January 4, 1995

Abstract

The representation of functions as low-degree polynomials over various rings has provided many insights in the theory of small-depth circuits. We survey some of the closure properties, upper bounds, and lower bounds obtained via this approach.

1. Introduction

There is a long history of using polynomials in order to prove complexity bounds. Minsky and Papert [39] used polynomials to prove early lower bounds on the order of perceptrons. Razborov [46] and Smolensky [49] used them to prove lower bounds on the size of AND-OR circuits. Other lower bounds via polynomials are due to [50, 4, 10, 51, 9, 55]. Paturi and Saks [44] discovered that rational functions could be used for lower bounds on the size of threshold circuits.

Toda [53] used polynomials to prove upper bounds on the power of the polynomial hierarchy. This led to a series of upper bounds on the power of the polynomial hierarchy [54, 52], AC^0 [2, 3, 52, 18], and ACC [58, 20, 30, 37], and related classes [21, 42].

Beigel and Gill [15] pushed the use of polynomials in order to prove closure properties of counting classes. This was extended by Hertrampf [34] (see also [16]). Beigel, Reingold, and Spielman [19] discovered that rational functions could be used as well for closure properties of PP. This work was then extended in [26, 27, 14].

In this paper we will attempt to expose the reader to the basic techniques for using polynomials in complexity theory. We will emphasize intuition rather than formality.

We will concentrate on the connections to constant-depth circuits, at the expense of polynomial-time Turing machines. Since nondeterministic polynomial-time oracle

*Yale University, Dept. of Computer Science, P.O. Box 208285, Yale Station, New Haven, CT 06520-8285. Email: `beigel-richard@cs.yale.edu`. This paper was prepared while the author was visiting the University of Maryland at College Park during a sabbatical from Yale. Support from Yale, the University of Maryland, UMIACS, and NSF grant CCR-8958528 is gratefully acknowledged.

Turing machines are equivalent to exponential-size depth-2 uniform circuits (see, for example, [36]), Turing machines fans will not really miss anything in this way. Conversely, circuit fans should not object to examples stated in terms of Turing machine complexity classes.

This is not a complete survey. Much history and some state-of-the-art results are omitted. The purpose of this paper is to expose the Structures community to some lovely techniques that do not seem to be well known. We recommend the surveys by Boppana and Sipser [23] and by Barrington [8] as well.

2. Notation

We write *polylog* to denote $(\log n)^{O(1)}$, *quasipolynomial* to denote 2^{polylog} (which is equal to n^{polylog}), and *subexponential* to denote $2^{n^{o(1)}}$.

3. Representing Boolean Functions as Polynomials

In order to represent Boolean functions by polynomials, we must represent the inputs and outputs by numbers. Several representations are useful. The choice of representation will usually depend on the application; furthermore it is sometimes convenient to represent inputs and outputs differently. There is no consensus on names for the different representations.

Standard representation: *false* $= 0$, *true* $= 1$. In this representation logical AND is equivalent to multiplication.

Dual representation: *false* $= 1$, *true* $= 0$. In this representation logical OR is equivalent to multiplication.

Fourier representation: *false* $= 1$, *true* $= -1$. In this representation XOR or parity is equivalent to multiplication.

Sign representation: *false* $= -1$, *true* $= 1$. In this representation EQUALS is equivalent to multiplication.

Ad hoc representation: *false* $= a$, *true* $= b$, for distinct numbers a and b in $R - \{0\}$. No special properties.

A useful property of the first four representations is that x^2 is identically equal to x in the first two and is identically equal to 1 in the next two. This means that polynomials are automatically multilinear.

The standard and the Fourier representations are the most commonly used for inputs. The standard, Fourier, and sign representations are the most commonly used for outputs. The dual representation appears above only for the sake of completeness. Note that $\overline{x} = 1 - x$ in the standard representation, and $\overline{x} = -x$ in the Fourier representation. To convert from the standard representation to the Fourier representation map $x \to 1 - 2x$. To convert from the Fourier representation to the standard representation map $x \to (1-x)/2$; this is possible only in rings where 2 has an inverse, e.g., the reals, the rationals, and Z_m for odd m.

There are three common ways to represent functions from $\{\mathit{true}, \mathit{false}\}^n$ into a ring R as polynomials.

Table lookup representation: Use the standard representation for the Boolean inputs. For each $S \subseteq \{0,1\}^n$, there is a term $\prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i)$. Every function from $\{0,1\}^n$ into R can be written as a linear combination of these terms. The coefficients of the linear combination are just the values of f. These 2^n terms are a basis for the vector space of functions from $\{0,1\}^n$ into R, because that space has dimension 2^n.

For example, take $n = 3$ and $f(x_1, x_2, x_3) = x_1 \vee x_2$. Then

$$\begin{aligned} f(x_1, x_2, x_3) &= x_1 x_2 x_3 + x_1 x_2 (1 - x_3) + \\ &\quad x_1 (1 - x_2) x_3 + x_1 (1 - x_2)(1 - x_3) + \\ &\quad (1 - x_1) x_2 x_3 + (1 - x_1) x_2 (1 - x_3). \end{aligned}$$

Let $c_S^{\mathrm{table}}(f)$ denote the coefficient of the Sth term in the table-lookup representation of f.

Standard polynomial representation: Use the standard representation for the Boolean inputs. For each $S \subseteq \{0, \ldots, n\}$, there is a term $\prod_{i \in S} x_i$. Every function from $\{0,1\}^n$ into R can be written as a linear combination of these terms. The coefficients of the linear combination can be obtained by multiplying out the table lookup representation via the distributive law. These 2^n terms are a basis for the vector space of functions from $\{0,1\}^n$ into R, because that space has dimension 2^n.

For example, take $n = 3$ and $f(x_1, x_2, x_3) = x_1 \vee x_2$. Then

$$f(x_1, x_2, x_3) = x_1 + x_2 - x_1 x_2.$$

Let $c_S^{\mathrm{standard}}(f)$ denote the coefficient of the Sth term in the standard polynomial representation of f.

Fourier polynomial representation: Use the Fourier representation for the Boolean inputs. For each $S \subseteq \{0, \ldots, n\}$, there is a term $\prod_{i \in S} x_i$. Provided that 2 has an inverse in R, every function from $\{0,1\}^n$ into R can be written as a linear combination of these terms. The coefficients of the linear combination can be obtained by replacing each x_i with $(1 - x_i)/2$ in one of the first two representations and then multiplying out via the distributive law. These 2^n terms are a basis for the vector space of functions from $\{0,1\}^n$ into R, because that space has dimension 2^n.

For example, take $n = 3$ and $f(x_1, x_2, x_3) = x_1 \vee x_2$. Then

$$f(x_1, x_2, x_3) = \frac{3}{4} + \frac{1}{4}x_1 + \frac{1}{4}x_2 - \frac{1}{4}x_1 x_2.$$

Let $c_S^{\text{Fourier}}(f)$ denote the coefficient of the Sth term in the Fourier polynomial representation of f.

Ad hoc polynomial representation: Use the ad hoc representation for the Boolean inputs. For each $S \subseteq \{0, \ldots, n\}$, there is a term $\prod_{i \in S} x_i$. Provided that F is a field, every function from $\{0,1\}^n$ into R can be written as a linear combination of these terms. The coefficients of the linear combination can be obtained by replacing each x_i with $(x_i - a)/(b - a)$ in one of the first two representations and then multiplying out via the distributive law. These 2^n terms are a basis for the vector space of functions from $\{0,1\}^n$ into R, because that space has dimension 2^n.

Since the coefficients in the table-lookup representation are the values of the function, it is trivial to convert to the table-lookup representation. We give two important formulas to convert from the table-lookup representation.

Möbius inversion formula:

$$c_S^{\text{standard}}(f) = \sum_{T \subseteq S} (-1)^{|S|-|T|} c_T^{\text{table}}(f).$$

Fourier transform: If $S \neq \emptyset$, then

$$c_S^{\text{fourier}}(f) = -2^{-n-1} \sum_{T \subseteq \{1,\ldots,n\}} (1 - 2c_T^{\text{table}}(f))(1 - 2(|T \cap S| \bmod 2)).$$

$$c_\emptyset^{\text{fourier}}(f) = \frac{1}{2} - 2^{-n-1} \sum_{T \subseteq \{1,\ldots,n\}} (1 - 2c_T^{\text{table}}(f)).$$

4. Symmetric Functions

A function $f(x_1, \ldots, x_n)$ is *symmetric* if, for all permutations π, $f(x_{\pi(1)}, \ldots, x_{\pi(n)}) = f(x_1, \ldots, x_n)$. If the inputs $x_1, \ldots, x_n$ are Boolean and f is symmetric, then f can only depend on how many of the inputs are true. Thus f depends only on the sum of the inputs, no matter how the values true and false are represented.

Lemma 1 below says that if we are working over a field, then f is a degree-d polynomial and is a symmetric function iff f is a degree-d polynomial in the sum of the inputs. The kth *elementary symmetric* function is

$$\sum_{S \subseteq \{1,\ldots,n\}, |S|=k} \prod_{i \in S} x_i.$$

When the x_i's are 0, 1-valued, exactly $\binom{x_1+\cdots+x_n}{k}$ of those products evaluate to 1, so the kth elementary symmetric function is equal to $\binom{x_1+\cdots+x_n}{k}$. A folklore theorem [39] says that if f is symmetric and has degree-d, then f is a linear combination of the 0th through dth elementary symmetric functions.

Lemma 1 (Folklore). *Let p be a degree-d polynomial. Assume furthermore that p restricted to $\{0,1\}^n$ is a symmetric function. Then there exist integers $c_0, \ldots, c_d$ such that for all inputs in $\{0,1\}^n$*

$$p(x_1, \ldots, x_n) = \sum_{0 \le k \le d} c_k \binom{x_1 + \cdots + x_n}{k}.$$

Proof: Because $x_i^2 = x_i$, we may rewrite p so that every monomial in p is of the form $x_{i_1} x_{i_2} \cdots x_{i_k}$ for some set $S = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$. Because p is a symmetric function, an easy induction on k shows that the coefficient of such a monomial is a constant that depends only on k, rather than on the exact choice of elements in S; call this constant c_k. Thus p is equal to the summation of c_k times the kth elementary symmetric function. Since $c_k = 0$ for $k > d$,

$$p(x_1, \ldots, x_n) = \sum_{0 \le k \le d} c_k \binom{x_1 + \cdots + x_n}{k}. \qquad \blacksquare$$

5. Circuits

We review some definitions concerning circuits. For complete coverage, we recommend [7]. The *size* of a circuit is the number of gates it contains (others have defined the size as the number of wires, which is at most the square of our notion of size). The *depth* of a circuit is the length of the longest path from the output to an input. A gate that computes a symmetric function is called a *symmetric gate*; typically its fanin is unbounded.

Most of the circuits we consider will have depth 2. They will be stratified, so no input is connected directly to the output gate. The output gate will be a symmetric gate. The input-level gates will compute arbitrary functions of a bounded number of inputs; however this bound is a parameter, not a constant. We call the maximum fanin of an input-level gate the *order* of the circuit.

As shown in Section 3, any function of k Boolean inputs may be written as a linear combination of ANDs. Thus we may restrict the input-level gates to be AND-gates; this does not affect the order of the circuit. (Alternatively, we could restrict the input-level gates to be parity gates.)

If w of the input level-gates compute the same function, then we may replace them by a single gate that is connected by w wires to the output gate. This is usually represented by a single wire with weight w. We extend this idea by allowing arbitrary integer weights. The *weight* of a circuit is the maximum absolute value of its weights. By convention, all input-level gates must be distinct; the output gate computes a function of the weighted sum of its inputs.

A threshold gate tests whether a weighted sum of its inputs is greater than some number t.

We have the following straightforward connection between circuits and polynomials:

Theorem 2 (Folklore). *Use the standard representation of inputs. The following are equivalent:*

- *f is computed by a depth-2 stratified circuit with a symmetric gate G at the root, order d, size s, and weight w, where G computes g of the weighted sum of its inputs.*
- *There exists a polynomial p with degree d, $s-1$ monomials, and coefficients bounded in absolute value by w such that*

$$f(x_1, \ldots, x_n) = g(p(x_1, \ldots, x_n)).$$

See, for example, [19] for the explicit connection between Turing machine complexity classes and polynomials.

6. Sign of polynomials over the integers (early lower bounds)

Because AND-OR circuits can be approximated by perceptrons (Section 9), lower bounds on perceptrons tell us about AC^0. In this section we define perceptrons and present some of the early lower bounds.

Minsky and Papert [39] developed much of the mathematical theory of perceptrons. A perceptron is a depth-2 circuit with a threshold gate at the output. They considered various restrictions on the gates at the input level. The simplest restriction is on the fanin of the input-level gates. In a perceptron of *order* k the input-level gates are required to have fanin k or less. Because every function can be expressed as a linear combination of AND gates, we lose no generality by requiring the input-level gates be AND gates of fanin k or less. Because the inputs to the threshold gate are weighted, there is no need for duplicate input-level gates, so the size of an order-d perceptron must be at most $1 + \sum_{0 \le k \le d} \binom{n}{k}$, which is $O(n^d)$. If we use 0 to represent false, and 1 to represent true, then an order-k perceptron is simply determining whether a degree-k multilinear polynomial is positive. In fact order-k perceptrons are equivalent to degree-k polynomials:

Corollary 3. *Use the standard representation of inputs. Use the Fourier representation of outputs. The following are equivalent:*

- *f is computed by a perceptron of order d, size s, and weight w.*
- *There exists a polynomial p with degree d, $s-1$ monomials, and coefficients bounded in absolute value by w such that*

$$f(x_1, \ldots, x_n) = \operatorname{sign}(p(x_1, \ldots, x_n)).$$

Proof: Apply Theorem 2 to $-f$. ▮

Let f be a Boolean function and p be a polynomial over the integers. We say that f is represented by the sign of p if $f(x_1, \ldots, x_n) = \mathit{true}$ if and only if $p(x_1, \ldots, x_n) > 0$. Define $d_{\text{sign}}(f)$ to be the least d such that f is represented by the sign of a degree-d polynomial. By replacing p by $2p-1$ if necessary, we may guarantee in addition that $f(x_1, \ldots, x_n) = \mathit{false}$ if and only if $p(x_1, \ldots, x_n) < 0$. Now, by considering $-p$, it follows that $d_{\text{sign}}(f) = d_{\text{sign}}(\neg f)$.

It is often easier to bound the power of symmetric polynomials than of general polynomials. Minsky and Papert developed techniques for exploiting symmetry in the target function, in order to reduce the general case to the symmetric case.

As an easy but important example, they proved that parity has the maximum possible degree.

Theorem 4 (Minsky and Papert). *Define the parity function by*

$$\operatorname{parity}(x_1, \ldots, x_n) = \begin{cases} \mathit{true} & \text{if } x_1 + \cdots + x_n \equiv 1 \pmod 2, \\ \mathit{false} & \text{otherwise.} \end{cases}$$

Then $d_{\text{sign}}(\text{parity}) = n$.

Proof: For every Boolean function f, $d_{\text{sign}}(f) \leq n$.

For the lower bound let p be a degree-d polynomial such that $\operatorname{parity}(x_1, \ldots, x_n) = \mathit{true}$ if and only if $p(x_1, \ldots, x_n) > 0$. We will prove that $d \geq n$. For every permutation π of $x_1, \ldots, x_n$ the function $p \circ \pi$ is a degree-d polynomial such that $\operatorname{parity}(x_1, \ldots, x_n) = \mathit{true}$ if and only if $p \circ \pi(x_1, \ldots, x_n) > 0$. Let $P(x_1, \ldots, x_n)$ be the sum over all permutations π of $p \circ \pi$. Then P is a degree-d polynomial such that $\operatorname{parity}(x_1, \ldots, x_n) = \mathit{true}$ if and only if $P(x_1, \ldots, x_n) > 0$. P is also symmetric. Because P is symmetric there is a degree-d polynomial Q in one integer variable such that $P(x_1, \ldots, x_n) = Q(x_1 + \cdots + x_n)$. Thus

$$\operatorname{parity}(x_1, \ldots, x_n) = \mathit{true} \iff Q(x_1 + \cdots + x_n) > 0.$$

(Observe that so far the only fact we have used about parity is that it is a symmetric Boolean function.) Therefore

$$x_1 + \cdots + x_n \equiv 1 \pmod 2 \iff Q(x_1 + \cdots + x_n) > 0,$$

so $Q(0) \leq 0$, $Q(1) > 0$, $Q(2) \leq 0$, $Q(3) > 0$, ... Therefore the polynomial Q has n zeroes, counting multiplicity. Since Q is not identically 0, the degree of Q is at least n. Therefore $d \geq n$. ▮

Next, they proved that an AND-of-ORs predicate requires degree $\sqrt[3]{n/4}$.

Theorem 5 (Minsky and Papert). *Let $n = 4m^3$. Partition the inputs into m sets $A_1, \ldots, A_m$, each having size $4m^2$. Let $f(x_1, \ldots, x_n)$ be true if and only if each of the m sets contains at least one input that is true. Then $d_{\text{sign}}(f) \geq m$.*

Proof: Let p be a degree-d polynomial such that $f(x_1, \ldots, x_n)$ is true if and only if $p(x_1, \ldots, x_n) > 0$. We will prove that $d \geq m$. If π_1 is a permutation on A_1, ..., π_m is a permutation on A_m, then $f(x_1, \ldots, x_n)$ is true if and only if $p \circ \pi_1 \circ \cdots \circ \pi_m(x_1, \ldots, x_m) > 0$. Let P be the sum over all permutations π_1 on A_1, ..., all permutations π_m on A_m of $p \circ \pi_1 \circ \cdots \circ \pi_m$. Then P is a degree-d polynomial such that $f(x_1, \ldots, x_n) = \mathit{true}$ if and only if $P(x_1, \ldots, x_n) > 0$. Furthermore, for each A_i, P is symmetric in the elements of A_i. A simple induction shows that there is a polynomial Q in m integer variables such that $P(x_1, \ldots, x_n) = Q(\sum_{x \in A_1} x, \ldots, \sum_{x \in A_m} x)$. Thus

$$f(x_1, \ldots, x_n) = \mathit{true} \iff Q(\sum_{x \in A_1} x, \ldots, \sum_{x \in A_m} x) > 0.$$

Therefore, if $0 \leq a_1, \ldots, a_m \leq 4m^2$, then

$$Q(a_1, \ldots, a_m) > 0 \iff a_1 \neq 0 \wedge \cdots \wedge a_m \neq 0.$$

Let $Q'(b_1, \ldots, b_m) = Q(b_1^2, \ldots, b_m^2)$. Then the degree of Q' is $2d$ and, if $-2m \leq b_1, \ldots, b_m \leq 2m$ then

$$Q'(b_1, \ldots, b_m) > 0 \iff b_1 \neq 0 \wedge \cdots \wedge b_m \neq 0.$$

But the predicate $b_i \neq 0$ can mimic the logical negation of an exact threshold gate on $2m$ inputs, i.e., let $b_i = x_1 + \cdots + x_{2m} - t$ where $0 \leq t \leq 2m$. Then $-2m \leq b_i \leq 2m$, and $b_i \neq 0 \iff x_1 + \cdots + x_{2m} \neq t$. Thus $Q'(b_1, \ldots, b_m) > 0$ if and only if m exact threshold tests on $2m$ inputs are false. However, it is well known that we can test evenness on $2m$ inputs by feeding all $2m$ inputs into m exact threshold gates and using the thresholds $1, 3, \ldots, 2m-1$. Evenness requires degree $2m$, because parity does. Therefore, the degree of Q' is at least $2m$, so $d \geq m$. ∎

Minsky and Papert's proof contains a remarkable translation lemma: If an AND-of-ORs is represented by the sign of a low degree polynomial, then an AND-of-NOT-exact-thresholds is also represented by the sign of a low degree polynomial. The latter is easily seen to be false; hence the former must be false as well. Furthermore, the translational technique is quite general; it is not limited to circuit models, and it can be used to obtain upper bounds as well as lower bounds. For example, it could be used to prove the following two statements:

i. If $\mathrm{P}^{\mathrm{NP[log]}} \subseteq \mathrm{PP}$ relative to all oracles, then $\mathrm{P}^{\mathrm{C_=P[log]}} \subseteq \mathrm{PP}$ relative to all oracles.

ii. If $\mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{PP}$ relative to all oracles, then $\mathrm{P}^{\mathrm{C_=P}} \subseteq \mathrm{PP}$ relative to all oracles.

The first statement is of the form *true* $\Rightarrow$ *true* [17, 31]. However, we would not have expected to prove $\mathrm{P}^{\mathrm{C_=P[log]}} \subseteq \mathrm{PP}$ from the mere fact that a smaller class is contained in PP. The second statement is of the form *false* $\Rightarrow$ *false* [13]. However the construction of an oracle where $\mathrm{P}^{\mathrm{NP}} \not\subseteq \mathrm{PP}$ is difficult, and this is the only way known to get an oracle where $\mathrm{P}^{\mathrm{C_=P[log]}} \subseteq \mathrm{PP}$; we hope that an easier construction for the latter can be found.

Fu [28] has pointed out that Theorem 5 implies that there is an oracle such that $\Sigma_2^p \not\subseteq \mathrm{PP}$.

What's more, Theorem 5 gives an alternative proof that there is an oracle that separates Σ_2^p from Δ_2^p. Why? By the hierarchical voting technique of Hemachandra and Wechsung [33] it is easy to see that Δ_2^p-predicates are represented by the sign of a low degree polynomial with large coefficients. Of course, Baker and Selman's [6] oracle that separates Σ_2^p from Π_2^p does the job as well, but its construction is relatively difficult.

That's not all. Theorem 5 gives an alternative to Tarui's [51] proof that there is an oracle such that $\Sigma_2^p \not\subseteq \mathrm{P}^{\mathrm{C_=P}}$. Why? By the translational technique above, we know that $\mathrm{P}^{\mathrm{C_=P}}$-predicates are also represented by the sign of a low degree polynomial with large coefficients. Actually, a stronger result is already known; Tarui [51] has constructed an oracle such that $\mathrm{BPP} \not\subseteq \mathrm{P}^{\mathrm{C_=P}}$. Tarui's proof technique makes a lovely and different use of polynomials.

*****

Minsky and Papert [39] also proved that the AND of two MAJORITYs is not represented by the sign of a degree-$O(1)$ polynomial. We will not go into the proof here.

*****

Paturi and Saks [44] used rational approximations to $|x|$ to prove that depth-2 threshold circuits require $\Omega(n/\log^2 n)$ gates and $\Omega(n^2/\log^2 n)$ wires to compute parity. This was extended by [48, 35].

7. Sign of polynomials over the integers (recent lower bounds)

There has been a flurry of recent work on functions that are represented by the sign of a low-degree polynomial. We will present the negative results in this section.

Beigel [13] considered the ODDMAXBIT predicate which asks whether the last 1 in the sequence $x_1, \ldots, x_n$ is in an odd-numbered position. Using the hierarchical voting idea developed by Hemachandra and Wechsung [33], it is easy to see that this predicate is represented by the sign of a linear function, namely

$$\sum_i (-2)^i x_i.$$

However, Beigel showed that ODDMAXBIT is not represented by the sign of a low degree polynomial with small coefficients. This implies that there is an oracle for which $\mathrm{P}^{\mathrm{NP}} \not\subseteq \mathrm{PP}$.

*****

Aspnes, Beigel, Furst, and Rudich proved that parity is not represented by a low-degree polynomial, even on a large fraction of the inputs.

Theorem 6 (Aspnes, Beigel, Furst, Rudich). *Let us use* 1 *to denote the result true and* -1 *to denote the result false. The following two statements are equivalent:*

i. There is a degree-d polynomial p such that $\mathrm{parity}(x_1, \ldots, x_n) = \mathrm{sign}(p(x_1, \ldots, x_n))$ *for at least m inputs* $x_1, \ldots, x_n$.

ii. $m \leq \sum_{0 \leq k < \frac{1}{2}(n+d+1)} \binom{n}{k}$.

Proof sketch: We use the standard polynomial representation.

(ii) $\Rightarrow$ (i): For integers k, let $\mathrm{odd}(k)$ denote the predicate "k is odd." By LaGrange's interpolation formula, we can pass a degree-d polynomial through any $d+1$ points. Construct a degree-d polynomial P such that $P(k) = \mathrm{odd}(k)$ for $\frac{1}{2}(n-d) \leq k < \frac{1}{2}(n+d+1)$. Then the sign of P is constant for $k \leq \frac{1}{2}(n-d)$ and it is also constant for $\frac{1}{2}(n+d) \leq k$. Thus, $P(k) = \mathrm{odd}(k)$ for all k between $\frac{1}{2}(n-d)$ and $\frac{1}{2}(n+d+1)$ and for every second value of k outside that half-open interval. Let $p(x_1, \ldots, x_n) = P(x_1 + \cdots + x_n)$. Then $\mathrm{sign}(p(x_1, \ldots, x_n)) = \mathrm{parity}(x_1, \ldots, x_n)$ whenever $P(x_1 + \cdots + x_n) = \mathrm{odd}(x_1 + \cdots + x_n)$. The total number of such points is obtained by summing the binomial coefficients $\binom{n}{k}$ for k such that $P(k) = \mathrm{odd}(k)$. Because $\binom{n}{k} = \binom{n}{n-k}$, this sum is equal to $\sum_{0 \leq k < \frac{1}{2}(n+d+1)} \binom{n}{k}$.

(i) $\Rightarrow$ (ii): Let p be a degree-d polynomial. Let E be the set of points such that

$$\mathrm{sign}(p(x_1, \ldots, x_n)) \neq \mathrm{parity}(x_1, \ldots, x_n).$$

Let $e = \frac{1}{2}(n-d-1)$ and

$$B = \sum_{0 \leq k \leq e} \binom{n}{k}.$$

We will show that $|E| \geq B$, so by the symmetry of binomial coefficients

$$m = 2^n - |E| \leq 2^n - B = \sum_{1 \leq k < n-e} \binom{n}{k}.$$

Suppose that $|E| < B$. A degree-e polynomial in n variables has exactly B coefficients. By Gaussian elimination we may construct a degree e polynomial z that is 0 on all the points in E but is not identically 0. Let $q = pz^2$. The degree of q is $d+2e$, which is less than n, and q is not identically 0. Furthermore, $\mathrm{sign}(q(x_1, \ldots, x_n)) = \mathrm{parity}(x_1, \ldots, x_n)$ whenever $q(x_1, \ldots, x_n) \neq 0$.

We finish by using the Minsky-Papert symmetrization technique. Let

$$Q(x_1,\ldots,x_n)=\sum_{\pi} q\circ\pi(x_1,\ldots,x_n),$$

where the sum ranges over all permutations of $x_1,\ldots,x_n$. Then Q is not identically 0 and the degree of Q equals the degree of q, which is less than n. Furthermore, for $k=0,\ldots,n$, $Q(k)=0$ or $\mathrm{sign}(Q(k))=\mathrm{odd}(k)$. Therefore Q has at least n zeroes, counting multiplicity. But Q is not identically 0, so the degree of Q is at least n. This contradiction completes the proof. ∎

In particular if $d=o(\sqrt{n})$ and $\epsilon>0$, there is no degree-d polynomial p such that $\mathrm{sign}(p(x_1,\ldots,x_n))=\mathrm{parity}(x_1,\ldots,x_n)$ on a $\frac{1}{2}+\epsilon$ fraction of all inputs. This implies [4] that relative to a random oracle PARITYP $\not\subseteq$ PP and PSPACE $\not\subseteq$ PP, theorems that were conjectured by Bennett and Gill [22].

Barrington and Straubing [10] proved a nontrivial extension of Theorem 6. Let $\mathrm{MOD}_{i,k}(x_1,\ldots,x_n)=0$ if the number of x_i's that are true is congruent to i modulo q; $\mathrm{MOD}_{i,k}(x_1,\ldots,x_n)=1$, otherwise.

Define the MOD_k function from Z_k^n into $\{\mathit{true},\mathit{false}\}$ by

$$\mathrm{MOD}_k(x_1,\ldots,x_n)=\begin{cases}\mathit{true} & \text{if } x_1+\cdots+x_n\not\equiv 0 \pmod{k},\\ \mathit{false} & \text{otherwise.}\end{cases}$$

Here we must be careful in representing the inputs. We represent the number $i\in Z_k$ by the k-bit vector $(v_0,\ldots,v_{k-1})$ that has a 1 in the ith position and 0s elsewhere.

As above, use -1 as output to represent *true* and 1 as output to represent *false*. It is easy to find a low-degree polynomial p such that $\mathrm{sign}(p(x_1,\ldots,x_n))=\mathrm{MOD}_k(x_1,\ldots,x_n)$ for a $1-1/k$ fraction of all inputs: let p be identically 1. However, Barrington and Straubing proved that the sign of every low-degree polynomial disagrees with the MOD_k function on some constant fraction of all inputs.

Theorem 7 (Barrington and Straubing). *For every k there exists c such that if p is a degree-$o(\sqrt{n})$ polynomial, then*

$$\mathrm{sign}(p(x_1,\ldots,x_n))\neq\mathrm{MOD}_k(x_1,\ldots,x_n)$$

for at least a c-fraction of all inputs.

As a corollary, they obtain a similar result for $\mathrm{MOD}_{0,k}$, but with a smaller c. In neither case do they give c explicitly as a function of k. It is an open problem to see how close c can be made to $1/k$.

8. Sign of polynomials over the integers (recent upper bounds)

In this section we use the Fourier representation of inputs and outputs. Although the Fourier transform results in non-integral coefficients, we can clear denominators by multiplying by 2^n, which does not affect the sign.

Let

$$\mathrm{maj}(x_1, \ldots, x_n) = \begin{cases} \textit{true} & \text{if } x_1 + \cdots + x_n < 0, \\ \textit{false} & \text{otherwise.} \end{cases}$$

Obviously maj is represented by the sign of a degree-1 polynomial.

It is easy to see that the XOR of two MAJORITYs is represented by the sign of a degree-2 polynomial.

Proposition 8. *Let*

$$f(x_1, \ldots, x_{2n}) = \mathrm{maj}(x_1, \ldots, x_n) \oplus \mathrm{maj}(x_{n+1}, \ldots, x_{2n}).$$

Then f is represented by the sign of a degree-2 polynomial.

Proof: Let

$$p(x_1, \ldots, x_{2n}) = (x_1 + \cdots + x_n)(x_{n+1} + \cdots + x_{2n}).$$

∎

This gives Russo's [47] theorem that PP is closed under symmetric difference. This has been extended to closure under polynomial-time parity reductions [7, 17].

*****

In Section 6 we mentioned that the AND of two MAJORITYs is not represented by the sign of a degree-$O(1)$ polynomial. However, Beigel, Reingold, and Spielman [19] proved that it is represented by the sign of a degree-$O(\log n)$ polynomial.

Their proof uses rational approximations to the sign() function, which were originally developed by Newman [40]. A rational function is the quotient of two polynomials; its degree is the larger of its numerator's degree and its denominator's degree.

Lemma 9 (Beigel, Reingold, Spielman). *There exists a family of rational functions with the following properties:*

- *The degree of s_n is $O(\log n)$,*
- *The coefficients of s_n are bounded in absolute value by $n^{O(\log n)}$.*
- *For all x such that $-n \le x \le -1$ or $1 \le x \le n$,*

$$|\mathrm{sign}(x) - s_n(x)| \le \frac{1}{3}.$$

Theorem 10 (Beigel, Reingold, Spielman). *Let*

$$f(x_1, \ldots, x_{2n}) = \mathrm{maj}(x_1, \ldots, x_n) \wedge \mathrm{maj}(x_{n+1}, \ldots, x_{2n}).$$

Then f is represented by the sign of a degree-$O(\log n)$ polynomial with coefficients bounded in absolute value by $n^{O(\log n)}$.

Proof sketch: Let

$$r(x_1, \ldots, x_{2n}) = s_{2n+1}(2(x_1 + \cdots + x_n) - 1) + s_{2n+1}(2(x_{n+1} + \cdots + x_{2n}) - 1) + 1.$$

Then $|r(x_1, \ldots, x_{2n}) - \text{sign}(x_1 + \cdots + x_n) - \text{sign}(x_{n+1} + \cdots + x_{2n}) - 1| \leq \frac{2}{3}$, so $\text{sign}(r(x_1, \ldots, x_{2n})) = f(x_1, \ldots, x_{2n})$. Let $r = p/q$. Then pq is the desired polynomial. ∎

They used this to prove that PP is closed under intersection. This was extended to show that PP is closed under polynomial-time truth-table reductions [27]. The techniques were further extended in [14]:

Theorem 11. *Every function computed by a constant-depth subexponential-size* AND/OR *circuit with $n^{o(1)}$ many majority gates is in fact represented by the sign of a degree-$n^{o(1)}$ polynomial whose inputs are subexponential-size* AND/OR *circuits.*

9. Probabilistic polynomials over the integers

In this section we use the standard polynomial representation.

A probabilistic polynomial p has, in addition to its ordinary inputs $x_1, \ldots, x_n$, some random inputs $r_1, \ldots, r_m$. We say that p represents a function f with error ϵ if for all inputs $x_1, \ldots, x_n$ for at least a $1 - \epsilon$ fraction of the random inputs $r_1, \ldots, r_m$

$$p(x_1, \ldots, x_n, r_1, \ldots, r_m) = f(x_1, \ldots, x_m).$$

Toda and Ogiwara [54], Tarui [52], and Beigel, Reingold, and Spielman [18] extended techniques of Valiant and Vazirani [56] to construct probablistic polynomials that represent the OR function.

Lemma 12. (Toda, Ogiwara / Tarui / Beigel, Reingold, Spielman) *There exists a probabilistic polynomial of degree $O\left(\log(1/\epsilon)\log^2 n\right)$ with $O\left(\log(1/\epsilon)\log^2 n\right)$ random inputs that represents $\text{OR}(x_1, \ldots, x_n)$ with error ϵ.*

Since

$$\text{AND}(x_1, \ldots, x_n) = 1 - \text{OR}(1 - x_1, \ldots, 1 - x_n),$$

there exist similar probabilistic polynomials that represent AND. The class AC^0 consists of all functions computed by constant-depth polynomial-size AND/OR circuits. By composing the polynomials for AND and OR, they [52, 18] obtain low-degree probabilistic polynomials that represent any function in AC^0.

Theorem 13. (Tarui / Beigel, Reingold, Spielman) *Every function in AC^0 is represented by a degree-*polylog *probabilistic polynomial with error $1/n$.*

Similarly,

Theorem 14. (Tarui / Beigel, Reingold, Spielman) *Every function computed by subexponential-size* AND/OR *circuits is represented by a degree-$n^{o(1)}$ probabilistic polynomial with error $1/n$.*

This yields an alternate proof [18, 4] that parity is not in AC^0 [29, 57, 32, 49].

Corollary 15. (Yao / Hastad / Smolensky) parity *is not computed by subexponential-size* AND/OR *circuits.*

Proof: Let f be computed by subexponential-size AND/OR circuits. Use the Fourier representation for outputs. By fixing an optimal choice of the random inputs in Theorem 14, there exists an ordinary polynomial p of degree $n^{o(1)}$ such that for a $1-1/n$ fraction of all inputs

$$f(x_1,\ldots,x_n) = p(x_1,\ldots,x_n) = \mathrm{sign}(p(x_1,\ldots,x_n)).$$

By Theorem 6, f must not be the parity function. ∎

In fact, combining Theorem 11, Theorem 13, and Theorem 6, we [14] obtain the following:

Theorem 16 (Beigel). *Let f be computed by a constant-depth subexponential-size* AND/OR *circuit with $n^{o(1)}$ many majority gates. Then $f(x_1,\ldots,x_n) \neq$* parity$(x_1,\ldots,x_n)$ *on at least a $\frac{1}{2}-o(1)$ fraction of all inputs.*

10. Real-valued approximation by polynomials

As a direct consequence of the Fourier transform formula, Linial, Mansour, and Nisan [38] observed that $c_S^{\mathrm{Fourier}}(f)$ measures the correlation of f with the parity of the inputs in S. Then they used Hastad's [32] switching lemma in a nontrivial way to prove that every AC^0 function has extremely low correlation with parity. This means that the coefficients of the high degree terms in f's Fourier polynomial representation are extremely small. Because a single coefficient may be approximated by random sampling, one can efficiently determine a good approximation to f's Fourier polynomial representation. Thus they showed that AC^0 functions can be approximately learned, for a uniform distribution of inputs, in quasipolynomial time.

*****

Consider a real-valued polynomial p that approximates a symmetric function $g(x_1+\cdots+x_n)$ with error $1/3$. Using Minsky and Papert's symmetrization technique, we obtain a polynomial P in one integer variable that approximates g with error $1/3$. Using Markov's theorem, Ehlich and Zeller [25] proved that if the value of a degree-d polynomial is bounded by a constant on $\{0,\ldots,n\}$, then its slope is $O(d^2/n)$ on that interval. Thus, if the degree of P is $o(\sqrt{n})$, then $P(x+1)-P(x)=o(1)$ so

p can approximate only constant functions. In particular, OR, AND, and MAJORITY are not approximated with error 1/3 by degree-$o(\sqrt{n})$ polynomials. Paturi [43] developed new techniques and obtained tight bounds, e.g., MAJORITY is not approximated with error 1/3 by a degree-$o(n)$ polynomial.

Ehlich and Zeller's lemma was used as well by Nisan and Szegedy [41] to prove that if a Boolean function depends on all of its inputs, then it cannot be approximated with error ϵ by a degree-$o(\log n)$ polynomial, and by Beigel [13] to prove that low-degree polynomials whose sign represents ODDMAXBIT must have large coefficients.

11. Polynomials modulo a prime

Consider polynomials over a ring R. A non-zero output represents *true*, and a zero output represents *false*, i.e., a polynomial g over R represents a function f if and only if

$$f(x) = true \iff g(x) \neq 0.$$

In particular, a polynomial g over Z_m represents a function f if and only if

$$f(x) = true \iff g(x) \not\equiv 0 \pmod{m}.$$

Polynomials over a finite field F are an important special case, where $x \neq 0 \iff x^{|F|-1} = 1$. Henceforth use the standard representation for outputs. Thus a polynomial g over F represents a function f if and only if

$$f(x) = g(x)^{|F|-1}.$$

Henceforth let p and q denote distinct primes. A polynomial g over Z_p represents a function f if and only if

$$f(x) = g(x)^{p-1}.$$

It is well known [24, 49, 15, 12] that the class of Boolean functions represented by low-degree polynomials over Z_p is closed under Boolean operations.

Theorem 17 (Folklore). *Let g and h be represented by degree-d polynomials over Z_p. Then*

i. $\neg g$ is represented by a degree-$((p-1)d)$ polynomial over Z_p.

ii. $g \wedge h$ is represented by a degree-$2d$ polynomial over Z_p.

iii. $g \vee h$ is represented by a degree-$(2(p-1)d)$ polynomial over Z_p.

Proof: Let G represent g and H represent h.

i. $1 - G^{p-1}$ represents $\neg g$.

ii. GH represents $g \wedge h$.

iii. $G^{p-1} + H^{p-1} - (GH)^{p-1}$ represents $g \vee h$. ∎

Another folk theorem [24, 49, 15, 12] says that for every constant k the class of Boolean functions represented by low-degree polynomials over Z_p is the same as the class of Boolean functions represented by low-degree polynomials over Z_{p^k}.

Theorem 18 (Folklore). *Let m and k be positive integers.*

i. If g is represented by a degree-d polynomial over Z_p, then g is represented by a degree-d polynomial over Z_{mp}.

ii. If g is represented by a degree-d polynomial over Z_{p^k}, then g is represented by a degree-$(d(p^k-1)/(p-1))$ polynomial over Z_p.

Proof:

i. Let G represent g over Z_p. Then mG represents g over Z_{mp}.

ii. Let G represent g over Z_{p^k}. We assert without proof that for every natural number x,

$$x \equiv 0 \pmod{p^k} \iff (\forall i < k)\left[\binom{x}{p^i} \equiv 0 \pmod{p}\right].$$

Therefore

$$\prod_{0 \le i < k} \binom{G}{p^i}$$

represents g over Z_{p^k}. Write G as a sum of monomials with coefficient 1. The function $\binom{G}{p^i}$ is the p^ith elementary symmetric function of those monomials, so it has degree dp^i. ∎

Smolensky [49] proved that a low-degree polynomial over a field cannot represent $x_1 \cdots x_n$ on a $\frac{1}{2} + \epsilon$ fraction of the inputs; this is true for every ad hoc polynomial representation.

Lemma 19 (Smolensky). *Choose any ad hoc polynomial representation over a field F (see Section 3). Let g be a polynomial of degree $o(\sqrt{n})$. Then*

$$x_1 \cdots x_n \neq g(x_1, \ldots, x_n)$$

for at least a $\frac{1}{2} - o(1)$ fraction of all inputs.

Proof: Let S be the set of n-tuples for which $x_1 \cdots x_n = g(x_1, \ldots, x_n)$. The vector space of function from S to F has dimension $|S|$. We will represent these functions by polynomials of degree $\frac{1}{2}n + o(\sqrt{n})$. To do so, start with the ad hoc polynomial representation. Let $M(x_1, \ldots, x_n)$ be a monomial of high degree, i.e., for some set T with $|T| > \frac{1}{2}n$

$$\begin{aligned} M(x_1, \ldots, x_n) &= \prod_{i \in T} x_i \\ &= x_1 \cdots x_n \prod_{i \notin T} x_i^{-1} \\ &= g(x_1, \ldots, x_n) \prod_{i \notin T} x_i^{-1}. \end{aligned}$$

for points in S. Since $x^{-1} = b^{-1}(x-a)/(b-a) + a^{-1}(x-b)/(a-b)$, M is equal to a polynomial of degree $o(\sqrt{n}) + |T| < o(\sqrt{n}) + \frac{1}{2}n$, for all points in S. Thus every function from S to F can be represented by a degree-$(\frac{1}{2}n + o(\sqrt{n}))$ polynomial over F.

There are exactly $\binom{n}{d}$ monomials of degree exactly d. It is well known that the sum of the first $\frac{1}{2}n + o(\sqrt{n})$ binomial coefficients is $(\frac{1}{2} + o(1))2^n$. Therefore the vector space of polynomials over F of degree $\frac{1}{2}n + o(\sqrt{n})$ has dimension $(\frac{1}{2} + o(1))2^n$, so $|S| \leq (\frac{1}{2} + o(1))2^n$. ∎

In particular, using the Fourier representation, Smolensky showed that no degree-$o(\sqrt{n})$ polynomial over Z_p, for any odd p, can represent parity on a $\frac{1}{2} + \epsilon$ fraction of the inputs.

Smolensky proved a similar lower bound for the $\mathrm{MOD}_{i,q}$ functions.

Theorem 20 (Smolensky). *Let p and q be distinct primes. Let $g_1, \ldots, g_{q-1}$ be degree-$o(\sqrt{n})$ polynomials over Z_p. Then for at least one value of i,*

$$\mathrm{MOD}_{i,q}(x_1, \ldots, x_n) \neq g_i(x_1, \ldots, x_n)$$

for at least a $\frac{1}{2(q-1)} - o(1)$ fraction of all inputs.

(By taking g_i to be identically 1 we can compute $\mathrm{MOD}_{i,q}$ correctly on all but a $\frac{1}{q}$ fraction of the inputs. We conjecture a lower bound of $\frac{1}{q} - o(1)$.)

Proof: Since p and q are distinct primes, there exists an extension field F of Z_p that contains a nontrivial qth root of unity ω. We use an ad hoc polynomial representation over F where 1 represents the input *false* and ω represents the input *true*. Then

$$x_1 \cdots x_n = \sum_{0 \leq i \leq q-1} \omega^i \mathrm{MOD}_{i,q}(x_1, \ldots, x_n)$$

Since

$$\mathrm{MOD}_{0,q} = 1 - (\mathrm{MOD}_{1,q} + \cdots + \mathrm{MOD}_{q-1,q})$$

$x_1 \cdots x_n$ is a linear combination of $\mathrm{MOD}_{1,q}$, ..., $\mathrm{MOD}_{q-1,q}$. Assume, for the sake of contradiction, that there exist degree-$o(\sqrt{n})$ polynomials $g_1, \ldots, g_{q-1}$ such that $\mathrm{MOD}_{i,q} \neq g_i$ for at most a $\frac{1}{2(q-1)}(1-\epsilon)$ fraction of all inputs. Then there is a degree-$o(\sqrt{n})$ polynomial g such that

$$x_1 \cdots x_n = g(x_1, \ldots, x_n)$$

for at least $\frac{1}{2} + (q-1)\epsilon$ of all inputs. ∎

A MOD_p gate computes the $\mathrm{MOD}_{0,p}$ function. The class ACC[p] consists of all functions computable by constant-depth, polynomial-size circuits consisting of AND, OR, NOT, and MOD_p gates.

Corollary 21 (Smolensky). *The* $\mathrm{MOD}_{0,q}$ *function is not in* ACC[p].

Proof: For each i, an easy modification converts any ACC[p] circuit that computes $\mathrm{MOD}_{0,q}$ into one that computes $\mathrm{MOD}_{i,q}$.

Start with such an ACC[p] circuit. By Lemma 12 and the comment following it, AND and OR can be represented by probabilistic polynomials of polylog-degree over Z. The value of a MOD_p gate on inputs $x_1, \ldots, x_n$ is $(x_1 + \cdots + x_n)^{p-1}$, which is a constant-degree polynomial over Z_p. By composing these polynomials, we obtain a polylog-degree probabilistic polynomial over Z_p. By fixing an optimal choice of the random bits, we obtain a polynomial over Z_p that represents the ACC[p] function on at least a $1 - 1/n$ fraction of the inputs. But then this polynomial represents $\mathrm{MOD}_{i,q}$ on a $1 - 1/n$ fraction of all inputs, contradicting Theorem 20. ∎

Although Smolensky's bounds for representing $\mathrm{MOD}_{i,q}$ by a polynomial over Z_p are not tight, techniques of Babai [5] (originally used only for the case $q = 2$) push through tight bounds for representing MOD_q by ACC[p] functions.

Corollary 22 (Babai). *Let p and q be distinct primes, and let g be a function in* ACC[p]. *Then*

$$\mathrm{MOD}_q(x_1, \ldots, x_n) \neq g(x_1, \ldots, x_n)$$

for at least a $\frac{1}{q} - o(1)$ fraction of all inputs.

Proof: Suppose, for the sake of contradiction, that

$$\mathrm{MOD}_q(x_1, \ldots, x_n) = g(x_1, \ldots, x_n)$$

for at least a $1 - \frac{1}{q} + \epsilon$ fraction of all inputs, where ϵ is a positive constant. Let

$$\begin{aligned} a &= \Pr[g(x) = 0 \mid \mathrm{MOD}_q(x) = 0], \\ b &= \Pr[g(x) = 0 \mid \mathrm{MOD}_q(x) \neq 0]. \end{aligned}$$

Then

$$\frac{1}{q}a + (1 - \frac{1}{q})(1 - b) = \Pr[g(x) = \mathrm{MOD}_q(x)] \geq 1 - \frac{1}{q} + \epsilon.$$

By simple arithmetic, $a - b > \epsilon$. We randomly self-reduce $\mathrm{MOD}_q(x_1, \ldots, x_n)$ to convert the average-case behavior to worst-case probabilistic behavior. Choose a random vector $(r_1, \ldots, r_n)$ in Z_q^n and let $r_0 = r_n$. For each i, let

$$x_i' = x_i + r_i - r_{i-1}.$$

Then x' is a uniform random vector such that $\mathrm{MOD}_q(x) = \mathrm{MOD}_q(x')$. Compute $g(x')$. If $\mathrm{MOD}_q(x) = 0$, then $g(x) = 0$ with probability a. If $\mathrm{MOD}_q(x) \neq 0$, then $g(x) = 0$ with probability b. Thus we have a bounded-error probabilistic circuit for MOD_q. By a theorem of Ajtai and Ben-Or [1], MOD_q can be computed deterministically using 4 additional levels of AND- and OR-gates. Therefore, MOD_q is in ACC[p], contradicting Corollary 21. ∎

12. Polynomials modulo a composite

There are some striking upper bounds for circuits with MOD_m gates when m is composite. However, the lower bounds are unspectacular.

A folklore theorem [51, 11] says AND is not represented by a low-degree polynomial over Z_m.

Theorem 23 (Folklore). *If g is a polynomial over Z_m that represents* $\mathrm{AND}(x_1, \ldots, x_n)$, *then the degree of g is n.*

Proof: Let $g(1, \ldots, 1) = a$. Then

$$g(x_1, \ldots, x_n) = a \prod_{1 \leq i \leq n} x_i,$$

which has degree n. ∎

Barrington, Straubing, and Therien [11] also noted that OR can be represented by a degree-$n/(m-1)$ polynomial over Z_m. When m is prime, this is tight.

Theorem 24 (Barrington, Straubing, Therien).
There is a degree-$(n/(m-1))$ polynomial over Z_m that represents OR. *If m is prime, this degree is the best possible.*

Proof sketch: For simplicity assume that n is a multiple of $m - 1$. Let or() be a degree-$(n/(m-1))$ polynomial over Z that represents the OR of $n/(m-1)$ variables, and let $g(x_1, \ldots, x_n) =$

$$\mathrm{or}(x_1, \ldots, x_{n/(m-1)}) + \cdots + \mathrm{or}(x_{(m-2)n/(m-1)+1}, \ldots, x_n).$$

For the lower bound, use Theorem 23, DeMorgan's law, and Theorem 17(i). ∎

In fact, when m is composite, Barrington, Beigel, and Rudich [9] discovered a lower-degree representation of OR.

Theorem 25 (Barrington, Beigel, Rudich).
Suppose that m has k distinct prime factors. Then there is a degree-$O(n^{1/k})$ polynomial that represents $\mathrm{OR}(x_1, \ldots, x_n)$.

Proof sketch: We present the special case $m = pq$. Choose the least i, j such that $p^i > \sqrt{m}$ and $q^j > \sqrt{m}$. Let $N = p^i q^j$. By Theorem 18, there is a degree-$O(\sqrt{m})$ polynomial f_p over Z_p such that $f(x_1, \ldots, x_N) \equiv 0 \pmod{p}$ iff $x_1 + \cdots + x_N \equiv 0 \pmod{p^i}$. Construct f_q similarly. Let $f = qf_p + pf_q$. Then $f(x_1, \ldots, x_N) \equiv 0 \pmod{pq}$ iff $x_1 + \cdots + x_N \equiv 0 \pmod{p^i q^j}$. Then

$$f(1 - x_1, \ldots, 1 - x_n, \underbrace{1, \ldots, 1}_{p^i q^j - n \text{ 1's}})$$

represents $\mathrm{OR}(x_1, \ldots, x_n)$. ∎

*****

Let p be an odd prime. By Smolensky's [49] Theorem, a degree-$o(\sqrt{n})$ polynomial over Z_p can represent parity for only a $\frac{1}{2} + o(1)$ fraction of all inputs. How well can a polynomial over Z_{pq} do? We conjecture that the answer is also $\frac{1}{2} + o(1)$. However, our best bound is $\frac{3}{4} + o(1)$. This bound was obtained during conversations with Russell Impagliazzo about Szegedy's [50] dissertation; Szegedy obtained identical bounds for approximating MAJORITY with a degree-$o(\sqrt{n})$ polynomial over Z_m. To prove the bound for parity, observe that f is 0 in Z_{pq} if and only if f is 0 in Z_p and in Z_q. A simple logical argument shows that if f over Z_{pq} represents parity for a $\frac{3}{4} + \epsilon$ fraction of all inputs, then f over Z_p or Z_q must represent parity for a $\frac{1}{2} + \frac{1}{2}\epsilon$ of all inputs. It would be very interesting to obtain a better ratio than $\frac{3}{4}$. Similar arguments show that if p is not a divisor of m then there is a constant $c > 0$ such that no degree-$o(\sqrt{n})$ polynomial over Z_m can represent the $\mathrm{MOD}_{0,p}$ function for a $1 - c$ fraction of all inputs.

The approximation bounds above are for degree-$o(\sqrt{n})$ polynomials. Barrington, Beigel, and Rudich [9] and Tsai [55] proved analogous linear lower bounds for computing functions exactly over Z_m. In addition [9], $\mathrm{MOD}_{0,m}$ requires linear degree over Z_m, although degree 1 suffices for its negation.

13. ACC

The class ACC is $\bigcup_{k \geq 2} \mathrm{ACC}[k]$. Yao [58] and Beigel and Tarui [20] proved that for every function f in ACC there exist a function $g : Z \rightarrow \{\mathit{true}, \mathit{false}\}$ and a polylog-degree polynomial p such that $f = g \circ p$. First they use the fact that a MOD_m gate can be written as the AND of a constant number of MOD_p gates, where p ranges over all prime divisors of m. They use Lemma 12 in order to replace the AND- and OR-gates by PARITY gates. Then they use a version of Theorem 17 and some remarkable polynomials that were first constructed by Toda [53] and successively improved by [58, 20]. The *modulus-amplifying* polynomials have the following property: For every $m \geq 1$ and $N \geq 0$,

$$\begin{aligned} N \equiv 0 \pmod{m} &\Rightarrow P_k(N) \equiv 0 \pmod{m^k}, \\ N \equiv 1 \pmod{m} &\Rightarrow P_k(N) \equiv 1 \pmod{m^k}. \end{aligned}$$

The modulus-amplifying polynomials constructed by Beigel and Tarui [20] have degree exactly $2k-1$.

The results for ACC were extended by Green, Köbler, and Torán [30], who showed that the symmetric function could actually be made very simple: one can let $h(x)$ be the middle bit of the binary representation of x.

The polynomial p constructed in [58, 20, 30] has quasipolynomial many terms. The corresponding depth-2 circuit has quasipolynomial size. Many people wondered whether polynomial size and polylog degree might suffice. However, Razborov and Wigderson [45] have proved that, regardless of degree, size $n^{\Omega(\log n)}$ is necessary to represent a very simple ACC function, namely a PARITY-of-AND-of-PARITYs where the AND gates have fanin $\log n$.

Acknowledgments. I am very grateful for Bill Gasarch's invaluable assistance in the preparation of this survey. I also thank Will Hurwood for proofreading; Russell Impagliazzo, Steve Homer, David Barrington, Howard Straubing, and Shi-Chun Tsai for helpful discussions; and Pierre McKenzie and Denis Therien for organizing the 1992 McGill Complexity Workshop, which forced me to organize my thoughts on this topic. Finally, I thank Bill Pugh, Carl Smith, and Clyde Kruskal for assisting with my visit to the University of Maryland.

References

[1] M. Ajtai and M. Ben-Or. A theorem on probabilistic constant depth circuits. In *Proc. 16th STOC*, pp. 471–474, 1984.

[2] E. Allender. A note on the power of threshold circuits. In *Proc. 30th FOCS*, pp. 580–584, 1989.

[3] E. Allender and U. Hertrampf. Depth reduction for circuits of unbounded fan-in. *Inf. & Comp.*, 108, 1994. To appear.

[4] J. Aspnes, R. Beigel, M. Furst, and S. Rudich. The expressive power of voting polynomials. *Combinatorica*. To appear.

[5] L. Babai. A random oracle separates PSPACE from the polynomial hierarchy. *IPL*, 26:51–53, 1987.

[6] T. P. Baker and A. L. Selman. A second step toward the polynomial hierarchy. *Theoretical Computer Science*, 8:177–187, 1979. An earlier version appeard in FOCS 1976.

[7] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*, volume 11 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, New York, 1988.

[8] D. A. M. Barrington. Quasipolynomial size circuit classes. In *Proc. 7th Structures*, pp. 86–93. IEEE, 1992.

[9] D. A. M. Barrington, R. Beigel, and S. Rudich. Representing Boolean functions as polynomials modulo composite integers. *Comp. Complexity*. Special issue devoted to the 4th Annual McGill Workshop on Complexity Theory. To appear.

[10] D. A. M. Barrington and H. Straubing. Complex polynomials and circuit lower bounds for modular counting. *Comp. Complexity*. Special issue devoted to the 4th Annual McGill Workshop on Complexity Theory. To appear.

[11] D. A. M. Barrington, H. Straubing, and D. Thérien. Non-uniform automata over groups. *Inf. & Comp.*, 89(2):109–132, 1990.

[12] D. M. Barrington. Some problems involving Razborov-Smolensky polynomials. In M. S. Paterson, editor, *London Mathematical Society Lecture Note Series 169: Boolean Function Complexity*, pp. 109–128. Cambridge University Press, 1992. Special volume devoted to the 1990 Durham Symposium. Also available as UMass COINS TR 90-59.

[13] R. Beigel. Perceptrons, PP, and the polynomial hierarchy. *Comp. Complexity*. Special issue devoted to the 4th Annual McGill Workshop on Complexity Theory. To appear.

[14] R. Beigel. When do extra majority gates help? polylog(n) majority gates are equivalent to one. *Comp. Complexity*. Special issue devoted to the 4th Annual McGill Workshop on Complexity Theory. To appear.

[15] R. Beigel and J. Gill. Counting classes: Thresholds, parity, mods, and fewness. *Theoretical Computer Science*, 103(1):3–23, 1992.

[16] R. Beigel, J. Gill, and U. Hertrampf. Counting classes: Thresholds, parity, mods, and fewness. In *Proc. 7th STACS*, pp. 49–57. Springer-Verlag, 1990. Vol. 415 of *LNCS*.

[17] R. Beigel, L. A. Hemachandra, and G. Wechsung. Probabilistic polynomial time is closed under parity reductions. *IPL*, 37(2):91–94, 1991.

[18] R. Beigel, N. Reingold, and D. Spielman. The perceptron strikes back. In *Proc. 6th Structures*, pp. 286–291, 1991.

[19] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. *JCSS*, 48, 1994. To appear.

[20] R. Beigel and J. Tarui. On ACC. *Comp. Complexity*, 1994. Special issue devoted to the 4th Annual McGill Workshop on Complexity Theory. To appear.

[21] R. Beigel, J. Tarui, and S. Toda. On probabilistic ACC circuits with an exact-threshold output gate. In *3rd Annual International Symposium on Algorithms and Computation*, pp. 420–429, 1992. LNCS 650.

[22] C. H. Bennett and J. Gill. Relative to a random oracle A, $\mathrm{P}^A \neq \mathrm{NP}^A \neq \text{co-NP}^A$ with probability 1. *SICOMP*, 10:96–112, 1981.

[23] R. Boppana and M. Sipser. The complexity of finite functions. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, pp. 757–804. MIT Press and Elsevier, The Netherlands, 1990.

[24] A. K. Chandra, L. Stockmeyer, and U. Vishkin. Constant depth reducibility. *SICOMP*, 13(2):423–438, 1984.

[25] H. Ehlich and K. Zeller. Schwankung von polynomen zwischen gitterpunkten. *Math. Zeitschift*, 86:41–44, 1964.

[26] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. In *Proc. 6th Structures*, pp. 30–42. IEEE Computer Society Press, 1991.

[27] L. Fortnow and N. Reingold. PP is closed under truth-table reductions. In *Proc. 6th Structures*, pp. 13–15, 1991.

[28] B. Fu. Separating PH from PP by relativization. *Acta Math. Sinica*, 8(3):329–336, 1992.

[29] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[30] F. Green, J. Köbler, and J. Torán. The power of the middle bit. In *Proc. 7th Structures*, pp. 111–117. IEEE, 1992. An extended version has been drafted by Green, Köbler, Regan, Schwentick, and Torán.

[31] T. Gundermann, N. Nasser, and G. Wechsung. A survey of counting classes. In *Proc. 5th Structures*, pp. 140–153. IEEE Computer Society Press, 1990.

[32] J. Hastad. Almost optimal lower bounds for small depth circuits. In S. Micali, editor, *Advances in Computing Research 5: Randomness and Computation*, pp. 143–170, Greenwich, CT, 1989. JAI Press.

[33] L. Hemachandra and G. Wechsung. On the power of probabilistic polynomial time: $\mathrm{P}^{\mathrm{NP[log]}} \subseteq \mathrm{PP}$. TR CUCS-372-88, Columbia Dept. of Computer Science, New York, NY, 1988.

[34] U. Hertrampf. Relations among MOD-classes. *Theoretical Computer Science*, 74(3):325–328, 1990.

[35] R. Impagliazzo, R. Paturi, and M. Saks. Size-depth trade-offs for threshold circuits. In *Proc. 25th STOC*, 1993. To appear.

[36] R. Kannan, H. Venkateswaran, V. Vinay, and A. C. Yao. A circuit-based proof of Toda's theorem. *Inf. & Comp.*, 1993. To appear.

[37] J. Köbler and S. Toda. On the power of generalized MOD-classes. In *Proc. 8th Structures*. IEEE, 1992. To appear.

[38] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. In *Proc. 30th FOCS*, pp. 574–579, 1989.

[39] M. L. Minsky and S. A. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1988. Expanded version of the original 1968 edition.

[40] D. J. Newman. Rational approximation to $|x|$. *Michigan Mathematical Journal*, 11:11–14, 1964.

[41] N. Nisan and M. Szegedy. On the degree of boolean functions as real polynomials. In *Proc. 24th STOC*, pp. 462–467, 1992.

[42] M. Ogiwara, T. Thierauf, S. Toda, and O. Watanabe. On closure properties of #P in the context of PF $\circ$#P. In *Proc. 8th Structures*. IEEE, 1993. To appear.

[43] R. Paturi. On the degree of polynomials that approximate symmetric Boolean functions. In *Proc. 24th STOC*, pp. 468–474, 1992.

[44] R. Paturi and M. E. Saks. Approximating threshold circuits by rational functions. *Inf. & Comp.*, 109, 1994. To appear.

[45] 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. *IPL*. To appear.

[46] A. A. Razborov. Lower bounds for the size of circuits of bounded depth with basis $\{\wedge, \oplus\}$. *Math. notes of the Academy of Science of the USSR*, 41(4):333–338, 1987.

[47] D. A. Russo. *Structural Properties of Complexity Classes*. PhD thesis, University of California at Santa Barbara, 1985.

[48] K.-Y. Siu, V. Roychowdhury, and T. Kailath. Computing with almost optimal size neural networks. In S. J. Hancock, J. Cowan, and L. Giles, editor, *Advances in Neural Information Processing Systems*, volume 5. Morgan Kaufman, 1993.

[49] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. 19th STOC*, pp. 77–82, 1987.

[50] M. Szegedy. *Algebraic Methods in Lower Bounds for Computational Models with Limited Communication*. PhD thesis, The University of Chicago, 1989.

[51] J. Tarui. Degree complexity of Boolean functions and its applications to relativized separations. In *Proc. 6th Structures*, pp. 382–390, 1991.

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

[53] S. Toda. PP is as hard as the polynomial-time hierarchy. *SICOMP*, 20(5):865–877, 1991.

[54] S. Toda and M. Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. *SICOMP*, 21(2):316–328, 1992.

[55] S.-C. Tsai. Lower bounds on representing boolean functions as polynomials in Z_m. In *Proc. 8th Structures*. IEEE, 1993. To appear.

[56] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[57] A. C. Yao. Separating the polynomial-time hierarchy by oracles. In *Proc. 26th FOCS*, pp. 1–10, 1985.

[58] A. C.-C. Yao. On ACC and threshold circuits. In *Proc. 31st FOCS*, pp. 619–627, 1990.