comput complexity 4 (1994), 301–313 1016-3328/94/040301-13 $1.50+0.20

ON THE DEGREE OF BOOLEAN FUNCTIONS AS REAL POLYNOMIALS

NOAM NISAN AND MARIO SZEGEDY

Abstract. Every Boolean function may be represented as a real polynomial. In this paper, we characterize the degree of this polynomial in terms of certain combinatorial properties of the Boolean function.

Our first result is a tight lower bound of $\Omega(\log n)$ on the degree needed to represent any Boolean function that depends on n variables.

Our second result states that for every Boolean function f, the following measures are all polynomially related:

- The decision tree complexity of f.
- The degree of the polynomial representing f.
- The smallest degree of a polynomial *approximating* f in the L_{max} norm.

Key words. Approximation; block sensitivity; Boolean functions; Fourier degree.

Subject classifications. 68Q05, 68Q99.

1. Introduction

1.1. Boolean functions as real polynomials. Boolean functions may be represented in various forms. Some of the simplest and most natural of these forms are representations as polynomials over various fields, and in particular over the real numbers. Let $f : \{F, T\}^n \to \{F, T\}$ be a Boolean function. If we encode "true" as the real number 1, and "false" as the real number 0, then f becomes a function from a subset of R^n to R. We say a real multivariate polynomial $p : R^n \to R$ *represents* f if, for every $x \in \{0,1\}^n$, $f(x) = p(x)$.

It is well known, and not difficult to see, that every Boolean function can be represented as a polynomial. Moreover, since for all $x \in \{0,1\}$ and integer $k \geq 1$ we have that $x^k = x$, there is no need to ever raise a variable x_i to a power greater than 1, and thus we can limit ourselves to polynomials p which are *multilinear* (i.e., each variable x_i appears with degree at most 1). The multilinear polynomials over any field form a 2^n dimensional space with respect to the addition and scalar multiplication over the field in question. Since the generalized AND functions (those Boolean functions that take value one at one particular point of the $\{0,1\}^n$ hypercube and take the value zero at any other point) naturally form a basis for this space, we also get that there is a unique multi-linear polynomial representing any given Boolean function.

The choice of representing "true" as 1 and "false" as 0 is of course somewhat arbitrary, and is a matter of convenience. Another convenient choice is to represent "true" as -1 and "false" as 1. The representation of a function as a polynomial under these conventions is sometimes called the Fourier transform of the function (see, e.g., [8,9]). The degree of this polynomial is called the Fourier degree of the polynomial. The polynomials that arise when representing the same Boolean function using different encodings of the inputs "true" and "false" may look very different. For instance, one polynomial can be very sparse (having only a few non-zero coefficients), while the other is dense. The degree, however, remains invariant under any choice of two different real numbers to represent "true" and "false".

1.2. Previous work. In their book "Perceptrons" [11], Minsky and Papert initiated the study of the computational properties of Boolean functions using their representation by polynomials. Recently, there have been many more studies that use these representations (or approximations) in order to study various complexity measures of the Boolean functions.

In [8], a relation between the *influence* of variables of Boolean functions and their Fourier coefficients was used that we will apply in Section 2. In [9], Fourier transforms were used to study AC^0 functions (functions computed by constant depth, polynomial size circuits). In [9,10], the same tool was used to devise learning algorithms. In [3], it was used to characterize "polynomial threshold" functions.

In [5], a tight lower bound for the time required to compute a Boolean function on a CREW PRAM is given in terms of the degree of the function as a real polynomial. In [1,2], lower bounds for constant depth circuits are obtained using approximations by real polynomials. Earlier, [13,15] obtained similiar lower bounds using polynomials over finite fields.

Our paper is self-contained, but the cited papers, especially the introductory chapters of [8] and [9], provide more background information on the methods we use.

1.3. New results. In this paper, we study the most basic parameter of the representation of a Boolean function as a real polynomial, its *degree.*

DEFINITION 1.1. *For a Boolean function f, the* degree *of f, denoted by* $\deg(f)$, *is the degree of the unique multilinear real polynomial that represents f (exactly).*

1.3.1. Minimum possible degree. Our first theorem answers the question of what is the smallest degree of a Boolean function that depends on n variables.

THEOREM 1.2. *Let f be a Boolean function that depends on n variables. Then,* $\deg(f) \geq \log_2 n - O(\log\log n)$.

The proof of this theorem makes use of the relation between "influences" and the Fourier transform due to [8].

This result is tight up to the $O(\log\log n)$ term, as can be seen by the "address" function.

1.3.2. Degree and decision trees. We next relate the degree of a Boolean function to several combinatorial and complexity measures of the function.

The Boolean decision model is perhaps the simplest computational model for Boolean functions. In this model, the algorithm repeatedly queries input variables until it can determine the value of the function. The algorithm is adaptive, choosing which variable to query next based on the answers to the previous queries. The only cost in this model is the number of variables queried, and the cost of an algorithm is the number of queries made for the worst case input. The decision tree complexity of f, $D(f)$, is defined to be the cost of the best algorithm for f.

The decison tree complexity is well studied in the literature in many contexts. In particular, it is known that it is closely related to several other combinatorial and complexity measures. The decision tree complexity is known to be polynomially related to the *certificate* complexity [17] and to the *block sensitivity* [12]. Furthermore, up to a constant factor, $\log D(f)$ is equal to the time needed to compute f on a CREW PRAM [12]. We show that the degree of f is also polynomially related to all these measures.

THEOREM 1.3. *For every Boolean function, we have*

$$\deg(f) \leq D(f) \leq 16 \deg(f)^8.$$

The proof of this result requires results from real approximation theory [4,6,14].

We strongly suspect that the exponent 8 is not optimal. The strongest separation we can obtain is a function for which $D(f) = \deg(f)^{1.58\ldots}$.

1.3.3. Approximation in L_{max} norm. Our techniques are strong enough to allow us to give strong bounds on the degree needed even to *approximate* Boolean functions in the L_{max} norm.

DEFINITION 1.4. *Let f be a Boolean function, and let p be a real polynomial. We say that p* approximates *f if, for every $x \in \{0,1\}^n$, we have that $|p(x) - f(x)| \leq 1/3$. The approximate degree of f, $\widetilde{\deg}(f)$, is defined to be the minimum degree of p, over all polynomials p that approximate f.*

Note that the constant 1/3 is arbitrary and can be replace by any constant $0 < \epsilon < 1/2$ without affecting our results.

Perhaps surprisingly, we show that approximation is not much easier than exact representation.

THEOREM 1.5. *There exists a constant c such that for every Boolean function f, we have*

$$\widetilde{\deg}(f) \leq \deg(f) \leq D(f) \leq c\, \widetilde{\deg}(f)^8.$$

This theorem has been recently used in [5] to show that randomization does not give extra power to CREW PRAMs.

The best separation results that we know of between $\deg(f)$ and $\widetilde{\deg}(f)$ are quadratic, and we conjecture that this is indeed the worst case.

2. Minimum possible degree

In this section, we prove the following theorem:

THEOREM 2.1. *Every Boolean function f that depends on n variables has degree $\deg(f) \geq \log_2 n - O(\log\log n)$.*

For the proof of this theorem, it will be convenient to use the Fourier transform representation, i.e., -1 for *true* and 1 for *false*. (This is used in this section only.) Thus, a Boolean function will be viewed as a real function $f : \{-1,1\}^n \to \{-1,1\}$. For a subset $S \subseteq \{1,\ldots,n\}$, we will denote $X_S = \prod_{i\in S} x_i$.

The next two subsections provide some necessary lemmas, and the proof of the theorem appears in the third subsection.

2.1. Degree and influences. We will require the following definition and lemmas from [8].

LEMMA 2.2 (PARSEVAL'S EQUALITY). *If we represent a Boolean funcion f as $f = \sum_S \alpha_S X_S$, then*

$$\sum_S \alpha_S^2 = 1.$$

DEFINITION 2.3. *For a Boolean function f and a variable x_i, the* influence *of x_i on f, $\mathrm{Inf}_i(f)$, is defined to be the following probability:*

$$\mathbf{Pr}[f(x_1,\ldots,x_{i-1},\mathit{true},x_{i+1},\ldots,x_n) \neq f(x_1,\ldots,x_{i-1},\mathit{false},x_{i+1},\ldots,x_n)],$$

where $x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n$ are chosen at random in $\{\mathit{false},\mathit{true}\}$.

LEMMA 2.4 ([8]). *For any Boolean function f on n variables, if we represent $f = \sum_S \alpha_S X_S$, then*

$$\sum_{i=1}^n \mathrm{Inf}_i(f) = \sum_S |S|\alpha_S^2.$$

From these lemmas, we easily deduce the following result:

COROLLARY 2.5. *For any Boolean function f,*

$$\sum_{i=1}^n \mathrm{Inf}_i(f) \leq \deg(f).$$

2.2. Zeroes of a multilinear polynomial. The following simple lemma gives an upper bound for the number of zeroes of any multilinear polynomial over $\{-1,1\}^n$. It is known as the Lemma of Schwartz [16], but we prove it below for completeness.

LEMMA 2.6 (SCHWARTZ). *Let $p(x_1, ..., x_n)$ be a non-zero multilinear polynomial of degree d. If we choose $x_1, \ldots, x_n$ independently at random in $\{-1, 1\}$, then the following inequality holds:*

$$\mathbf{Pr}[p(x_1, ..., x_n) \neq 0] \geq 2^{-d}.$$

PROOF. The proof is by induction on n. For $n = 1$, we just have a linear function of one variable which may have only one zero.

For the induction step, write

$$p(x_1, \ldots, x_n) = x_n g(x_1, \ldots, x_{n-1}) + h(x_1, \ldots, x_{n-1}).$$

We can see that the non-zeroes of p over $\{-1, 1\}^n$ yield non-zeroes of $h + g$ or of $h - g$ over $\{-1, 1\}^{n-1}$: if $p(x_1, \ldots, x_{n-1}, 1) \neq 0$ then $h(x_1, \ldots, x_{n-1}) + g(x_1, \ldots, x_{n-1}) \neq 0$, and if $p(x_1, \ldots, x_{n-1}, -1) \neq 0$ then $h(x_1, \ldots, x_{n-1}) - g(x_1, \ldots, x_{n-1}) \neq 0$. We now distinguish between three cases.

1. $h + g$ is identically equal to zero. In this case, $p = (x_n - 1)g$, where $\deg(g) = d - 1$ and we use the induction hypothesis on g for the x's satisfying $x_n = -1$.

2. $h - g$ is identically equal to zero. In this case, $f = (1 + x_n)g$, where $\deg(g) = d - 1$, and again we use the induction hypothesis on g for the x's satisfying $x_n = 1$.

3. Both $h + g$ and $h - g$ are not identically equal to zero. The degrees of $h + g$ and of $h - g$ are both bounded by d and thus we use the induction hypothesis on $h + g$ for the x's satisfying $x_n = 1$ and on $h - g$ for the x's satisfying $x_n = -1$.

2.3. Proof of Theorem 2.1. We have now assembled all that we need in order to prove the theorem.

For each i, $1 \leq i \leq n$, define a function f^i on $n - 1$ variables as follows:

$$f^i(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) =$$

$$f(x_1, \ldots, x_{i-1}, -1, x_{i+1}, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n).$$

Under this notation, it is clear that

$$\mathrm{Inf}_i(f) = \mathbf{Pr}[f^i(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \neq 0],$$

where $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ are chosen at random in $\{-1, 1\}$.

Since f depends on all of its variables, we have that for every i, f^i is not identically zero, and thus, we can apply Lemma 2.6 and conclude that for all i, $\mathrm{Inf}_i(f) \geq 2^{-d}$.

On the other hand, it follows from Corollary 2.5 that $\sum_i \mathrm{Inf}_i(f) \leq d$. Combining these two bounds, we get:

$$n/2^d \leq \sum_i \mathrm{Inf}_i(f) \leq d.$$

Thus, $d2^d \geq n$, and the theorem follows. $\square$

3. Degree and decision trees

We remind the reader that at this point, we return to the representation of *true* $= 1$ and *false* $= 0$.

3.1. The Method of Symmetrization. We will use the method of symmetrization, first used by Minsky and Papert [11].

DEFINITION 3.1. *If $p : R^n \to R$ is a multivariate polynomial, then the* symmetrization *of p is defined as follows:*

$$p^{sym}(x_1, \ldots, x_n) = \frac{\sum_{\pi \in S_n} p(x_{\pi(1)}, \ldots, x_{\pi(n)})}{n!}.$$

The important point is that if we are only interested in inputs $x \in \{0,1\}^n$, then p^{sym} turns out to depend only upon $x_1 + \cdots + x_n$. We can thus represent it as a univariate polynomial of $x_1 + \cdots + x_n$.

LEMMA 3.2 ([11]). *If $p : R^n \to R$ is a multivariate polynomial, then there exists a unique univariate polynomial $\tilde{p} : R \to R$ of degree at most n such that for all $x_1, \ldots, x_n \in \{0,1\}^n$, we have*

$$p^{sym}(x_1, \ldots, x_n) = \tilde{p}(x_1 + \cdots + x_n).$$

Moreover, $\deg(\tilde{p}) \leq \deg(p)$.

3.2. A theorem from approximation theory. We will need the following result of H. Ehlich and K. Zeller [6] and T. J. Rivlin and E. W. Cheney [14]:

THEOREM 3.3 (EHLICH, ZELLER; RIVLIN, CHENEY). *Let p be a polynomial with the following properties:*

1. *For any integer $0 \le i \le n$, we have $b_1 \le p(i) \le b_2$.*
2. *For some real $0 \le x \le n$, the derivative of p satifies $|p'(x)| \ge c$.*

Then, $\deg(p) \ge \sqrt{cn/(c + b_2 - b_1)}$.

Again, for completeness, we prove this theorem. The proof is based on the following well known theorem of Markov [4]:

THEOREM 3.4 (MARKOV). *Let $p : R \to R$ be a univariate polynomial of degree d such that any real number $a_1 \le x \le a_2$ satisfies $b_1 \le p(x) \le b_2$. Then for all $a_1 \le x \le a_2$, the derivative of p satisfies $|p'(x)| \le d^2(b_2 - b_1)/(a_2 - a_1)$.*

The two theorems are similar, but in the former, we do not have the information on the value of $p(n)$ for all real x in the range but rather only for integer x. Thus, the theorem can be perceived as a generalization of that of Markov. There is a surprisingly simple proof of it, however, by Markov's theorem:

PROOF OF THEOREM 3.3. Let $c' = \max_{0 \le x \le n} |p'(x)| \ge c$. It is clear that for all real $0 \le x \le n$,

$$b_1 - c'/2 \le p(x) \le b_2 + c'/2.$$

Using the Markov inequalty, we therefore have the following inequalities:

$$\begin{aligned} c' &\le \frac{\deg(p)^2(c' + b_2 - b_1)}{n}, \\ \deg(p)^2 &\ge \frac{c'n}{c' + b_2 - b_1} \ge \frac{cn}{c + b_2 - b_1}. \quad \square \end{aligned}$$

3.3. Main lemma.

LEMMA 3.5. *Let f be a Boolean function such that $f(0, 0, \dots, 0) = 0$ and for every Boolean vector $\vec{x}$ of Hamming weight 1, $f(\vec{x}) = 1$. Then, the following inequalities hold:*

$$\begin{aligned} \deg(f) &\ge \sqrt{n/2}, \\ \widetilde{\deg}(f) &\ge \sqrt{n/6}. \end{aligned}$$

PROOF. We will prove the bound for $\widetilde{\deg}(f)$. The sharper bound for $\deg(f)$ follows exactly the same lines. Let p be a polynomial approximating f, and consider $\tilde{p}$ the univariate polynomial giving its symmetrization. $\tilde{p}$ satisfies the following properties:

1. $\deg(\tilde{p}) \leq \deg(p)$. (By Lemma 3.2.)
2. For every integer $0 \leq i \leq n$, $-1/3 \leq \tilde{p}(i) \leq 4/3$. (Since for every Boolean vector $\vec{x}$, $p(\vec{x})$ is within $1/3$ of a Boolean value.)
3. $\tilde{p}(0) \leq 1/3$. (Since $f(0,0,\ldots,0) = 0$.)
4. $\tilde{p}(1) \geq 2/3$. (Since for all Boolean vectors $\vec{x}$ of Hamming weight 1, $f(\vec{x}) = 1$.)

Properties (3) and (4) together imply that for some real $0 \leq z \leq 1$, the derivative $\tilde{p}'(z) \geq 1/3$. We can now apply Theorem 3.3 to obtain the lower bound for $\deg(\tilde{p})$, and thus also for $\deg(p)$. We remark that the bound for $\deg(f)$ is proven exactly the same way, except that the inequalities that correspond to (2)–(4) contain different constants. □

The examples given below in Section 3.5 show that the bound for $\widetilde{\deg}(f)$ is tight (up to a constant factor). We do not know whether the bound for $\deg(f)$ is tight.

3.4. General Boolean functions. Although the main Lemma concerns very special types of Boolean functions, it turns out that it is enough to give good bounds for *all* Boolean function. This is done by relating the degree to other combinatorial properties of Boolean functions. But first, let us introduce a new notation: For a string $x \in \{0,1\}^n$ and a set $S \subseteq \{1,\ldots,n\}$, we define $x^{(S)}$ to be the Boolean string which differs from x on exactly the bits in S.

DEFINITION 3.6 ([12]). *For a Boolean function f the* block sensitivity *of f, $\mathrm{bs}(f)$, is defined to be the maximum number t for which there exists an input $x \in \{0,1\}^n$ and t disjoint subsets $S_1,\ldots,S_t \subset \{1,\ldots,n\}$ such that for all $1 \leq i \leq t$, $f(x) \neq f(x^{(S_i)})$.*

The block sensitivity of a function is known to be related to its decision tree complexity, $D(f)$.

LEMMA 3.7 ([12]). *For every Boolean function f, $\mathrm{bs}(f) \leq D(f) \leq \mathrm{bs}^4(f)$.*

We can easily get lower bounds for the degree in terms of the block sensitivity.

LEMMA 3.8. *For every Boolean function f, the following inequalities hold:*

$$\begin{aligned} \deg(f) &\geq \sqrt{\mathrm{bs}(f)/2}, \\ \widetilde{\deg}(f) &\geq \sqrt{\mathrm{bs}(f)/6}. \end{aligned}$$

PROOF. Let f be a Boolean function, and let $\vec{x}$ and $S_1, \ldots, S_t$ be the input and sets achieving the block sensitivity. Let us assume without loss of generality that $f(\vec{x}) = 0$. We define a function $f'(y_1, \ldots, y_t)$ as follows:

$$f'(y_1, \ldots, y_t) = f(\vec{x} \oplus y_1 S_1 \oplus \cdots \oplus y_t S_t),$$

i.e., the j'th bit fed to f is $x_j \oplus y_i$ if $j \in S_i$, and is x_j if j is not in any of the S_i's (the $\oplus$ operator adds bits or vectors of bits modulo 2). The following facts can easily be verified:

1. $\deg(f') \leq \deg(f)$. (The bits x_j are constants in the definition of f'.)

2. f' satisfies the conditions of Lemma 3.5.

Our lemma thus follows from Lemma 3.5. □

Combining Lemmas 3.7 and 3.8 we get the following result:

THEOREM 3.9. *For every Boolean function f we have*

$$\begin{aligned} \deg(f) \leq D(f) &\leq 16 \deg(f)^8, \\ \widetilde{\deg}(f) \leq \deg(f) \leq D(f) &\leq 1296\, \widetilde{\deg}(f)^8. \end{aligned}$$

3.5. Separations. The best separation results we know of between $D(f)$, $\deg(f)$ and $\widetilde{\deg}(f)$ are given by the following examples.

Let $E_{12}(x_1, x_2, x_3)$ be the symmetric function taking value *true* if exactly one or two of the input bits are true. It is not difficult to see that $\deg(E_{12}) = 2$ (we can write $E_{12} = x_1 + x_2 + x_3 - x_1x_2 - x_2x_3 - x_1x_3$). On the other hand, $D(E_{12}) = 3$. For every integer k, we define the function E_{12}^k on 3^k variables as a composition of E_{12} on three disjoint copies (on separate inputs) of E_{12}^{k-1}. We now have the following result:

EXAMPLE 3.10. *The function E_{12}^k on $n = 3^k$ variables satisfies the following relations:*

$$\begin{aligned} D(E_{12}^k) &= 3^k = n, \\ \deg(E_{12}^k) &= 2^k = n^{\log_3 2} = n^{0.63\ldots}. \end{aligned}$$

PROOF. The value of the degree simply follows by induction from the definition of E_{12}^k. The lower bound on the decision tree complexity follows from the fact that on input $(0, 0, \ldots, 0)$, the decision tree must look at every bit because if even one bit is changed to 1, the value of E_{12}^k changes from *false* to *true*. □

Consider the function OR_n on n variable returning *true* if at least one of the inputs is true. Using Chebyshev polynomials, we can approximate OR_n by a rather low degree polynomial.

EXAMPLE 3.11. *The function OR_n satisfies the following equalities:*

$$\begin{aligned} \deg(OR_n) &= n, \\ \widetilde{\deg}(OR_n) &= O(\sqrt{n}). \end{aligned}$$

PROOF. We will use Chebyshev polynomials. The k'th Chebyshev polynomial, $T_k(x)$, is a real polynomial of degree k having the following properties (see [4]):

1. For every $-1 \leq x \leq 1$, $|T_k(x)| \leq 1$.
2. For all $x \geq 1$, the derivative satisfies $T_k'(x) \geq k^2$.

Now choose $k = 2\sqrt{n}$ and $c = 1/T_k \times n/(n-1)$, and define the polynomial $p(x) = 1 - cT_k(x/(n-1))$. Property (2) insures that $c \leq 1/4$. By property (1), we have that $|p(x) - 1| \leq 1/3$ for all $0 \leq x \leq n-1$, and $p(n) = 0$. It follows that $p(x_1 + \cdots + x_n)$ approximates OR_n. □

4. Open problems

Besides the intriguing questions that remain open about the exact relation between $\widetilde{\deg}(f)$, $\deg(f)$, and $D(f)$, we would like to mention three related open problems.

The first question is known as the question of "sensitivity versus block sensitivity." The sensitivity of a Boolean function f, $S(f)$, is the maximum of $S_x(f)$ over all inputs x, as defined below: Let x^i be the input that we obtain from x by negating its i^{th} bit but leaving all the other bits intact; $S_x(f)$ is the number of i's such that $f(x) \neq f(x^i)$. For example, the sensitivity of the "OR" function is n, because for its "most sensitive input", $(0, 0, \ldots, 0)$, the value of OR changes from 0 to 1 if we exchange any of the input zeros by 1. Clearly, $\mathrm{bs}(f) \geq S(f)$. Is it true that there is a polynomial relation between the sensitivity and the block sensitivity (say, $S^2(f) \geq \mathrm{bs}(f)$)?

The second question was asked recently by Lance Fortnow: What if we express f as a rational function $(p(x)/q(x))$ rather than as a polynomial (p and q are multivariate polynomials and we can take them to be multilinear). Can $\max(\deg(p), \deg(q))$ be much less than $\deg(f)$? Again, does a polynomial relation hold between $\deg(f)$ and $\max(\deg(p), \deg(q))$? An answer would have interesting applications in structural complexity theory.

The third question is related to the degree of *symmetric* Boolean functions. Suppose that f is a symmetric Boolean function, but not identically zero or one. Give a lower bound on $\deg(f)$ in terms of the number of variables, n. It is easy to see that $n/2$ is always a lower bound, but apparently better bounds can be obtained. Recently, J. von zur Gathen and Jim Roche [7] showed that if the number of variables, n, is a prime minus 1, then $\deg(f) = n$. It is easy to construct symmetric functions for any odd n such that $\deg(f) = n - 1$. In [7] it is shown that there are infinitely many symmetric nontrivial f's such that $\deg(f) = n - 3$, and the authors conjecture that $n - 3$ is a general lower bound.

Acknowledgements

The authors wish to thank Ilan Newman for his insightful comments. Noam Nisan was supported by BSF 89–00126.

References

[1] J. Aspens, R. Beigel, M. Furst, and S. Rudich, On the expressive power of voting polynomials. *Proc. Twenty-third Ann. ACM Symp. Theor. Comput.*, 1991, 402–409.

[2] R. Beigel, N. Reingold, and D. Spielman, *The perceptron striles back.* Technical report YALEU/DCS/TR-813, Yale University, 1990.

[3] J. Brook and R. Smolenski, Polynomial threshold functions, AC^0 functions and spectral norms. *Proc. 31st Ann. IEEE Symp. Found. Comput. Sci.*, 1990, 632–642.

[4] E. W. Cheney, *Introduction to approximation theory.* McGraw-Hill Book Co., 1966.

[5] M. Dietzfelbinger, M. Kutylowski, R. Reischuk, Exact Time Bounds for Computing Boolean Functions on PRAMs Without Simultaneous Writes. *Symp. Parallel Algorithms and Architecture*, 1990, 125–137.

[6] H. Ehlich and K. Zeller, Schwankung von Polynomen zwischen Gitterpunkten. *Mathematische Zeitschrift* **86** (1964), 41–44.

[7] J. von zur Gathen and J. Roche, Polynomials with two values. *Preprint* (1993).

[8] J. Kahn, G. Kalai, and N. Linial, The influence of variables on Boolean functions. *Proc. 29th Ann. IEEE Symp. Found. Comput. Sci.*, 1988, 68–80.

[9] N. Linial, Y. Mansour, N. Nisan, Constant depth circuits, Fourier transform and Learnability. *Proc. 30th Ann. IEEE Symp. Found. Comput. Sci.*, 1989, 574–579.

[10] E. Kushilevitz and Y. Mansour, Learning decision trees using the Fourier transform. *Proc. Twenty-third Ann. ACM Symp. Theor. Comput.*, 1991, 455–464.

[11] M. Minsky and S. Papert, *Perceptrons.* MIT press, Cambridge, 1988 (Expanded edition). First edition appeared in 1968.

[12] N. Nisan, CREW PRAM's and decision trees. *Proc. Twenty-first Ann. ACM Symp. Theor. Comput.*, 1989, 327–335.

[13] A.A. Razborov, Lower bounds for the size of circuits of bounded depth with basis (and,xor). *Math. Notes of the Academy of Sciences of the USSR* **41**(4) (1987), 333–338.

[14] T. J. Rivlin and E. W. Cheney, A comparison of Uniform Approximations on an interval and a finite subset thereof. *SIAM J. Numer. Anal.* **3**(2) (1966), 311–320.

[15] R. Smolenski, Algebraic methods in the theory of lower bounds for Boolean circuit complexity. *Proc. Nineteenth Ann. ACM Symp. Theor. Comput.*, 1987, 77–82.

[16] J. T. Schwartz, Fast probabilistic Algorithms for verification of Polynomial identities. *J. Assoc. Comput. Mach.* **27** (1980), 701–717.

[17] I. Wegener, *The complexity of Boolean functions.* Wiley-Teubner Series in Comput. Sci., New York – Stuttgart, 1987.

Manuscript received 6 July 1992

Noam Nisan
The Hebrew University
Jerusalem 91904,
Israel
noam@cs.huji.ac.il

Mario Szegedy
AT& T Bell Laboratories,
600 Mountain Ave,
Murray Hill, NJ 07974, USA
ms@research.att.com