Polynomial Threshold Functions, AC^0 Functions and Spectral Norms

Extended Abstract

Jehoshua Bruck *   Roman Smolensky †

Abstract

We study the class of polynomial threshold functions using harmonic analysis and apply the results to derive lower bounds related to AC^0 functions. A Boolean function is polynomial threshold if it can be represented as a sign function of a sparse polynomial (one that consists of a polynomial number of terms). Our main result is that the class of polynomial threshold functions can be characterized using their spectral representation. In particular we prove that a Boolean function whose L_1 spectral norm is bounded by a polynomial in n is a polynomial threshold function. While a Boolean function which its L_∞^{-1} spectral norm is not bounded by a polynomial in n is not a polynomial threshold function [5].

When we consider AC^0 functions, we obtain the following: (i) There exists an AC^0 function whose L_1 spectral norm is exponentially large. (ii) There exists an AC^0 function whose L_∞^{-1} spectral norm is $\Omega(n^{poly\log(n)})$. Hence, applying our characterization results, we derive an $\Omega(n^{poly\log(n)})$ lower bound on the number of terms needed to compute *exactly* a Boolean AC^0 function as a sign function of a polynomial. This result complements the results of [12]. (iii) A lower bound of $\Omega(n^{poly\log(n)})$ on the size of a depth-2 circuit of MAJORITY gates that computes an AC^0 function. This is the first known lower bound for the result of [1].

*IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099.

†Computer Science Department, University of Toronto, Toronto, Canada M5S 1A4.

1 Introduction

Polynomial Threshold Functions: A Boolean function $f(X)$ is a *threshold function* if

$$f(X) = sgn\,(F(X)) = \begin{cases} 1 & \text{if } F(X) > 0 \\ -1 & \text{if } F(X) < 0 \end{cases}$$

where

$$F(X) = \sum_{\alpha \in \{0,1\}^n} w_\alpha X^\alpha$$

and

$$X^\alpha \stackrel{\text{def}}{=} \prod_{i=1}^{n} x_i^{\alpha_i}$$

Throughout this paper a *Boolean function* will be defined as $f : \{1,-1\}^n \longrightarrow \{1,-1\}$; namely, 0 and 1 are represented by 1 and -1, respectively. It is also assumed, without loss of generality, that $F(X) \neq 0$ for all X.

A *threshold gate* is a gate that computes a threshold function. Obviously, any Boolean function can be computed by a single threshold gate if we allow the number of monomials in $F(X)$ to be as large as 2^n. Although this fact is not interesting by itself, it stimulates the following natural question: What happens when the number of monomials (terms) in $F(X)$ is bounded by a polynomial in n?

The question can be formulated by defining a new complexity class of Boolean functions. This class, called PT_1 for *Polynomial Threshold* functions, consists of all the Boolean functions that can be computed by a single threshold gate where the number of monomials is bounded by a polynomial in n. The main goal of this paper is to characterize this class of functions using the spectral representation of Boolean functions and to understand its relationship with the class of AC^0 Boolean functions.

Some motivation: Recently, there has been a considerable interest in study of the computational model of bounded depth unbounded fan-in polynomial size circuits that consist of linear threshold gates [7, 13, 15, 18]. This interest follows from recent results in complexity of circuits [9, 14, 17] which indicate that MAJORITY (hence, linear threshold functions) can not be computed by a bounded depth unbounded fan-in polynomial size circuit that consists of $\vee$, $\wedge$, NOT and PARITY gates. Thus, the next natural step in the analysis is adding MAJORITY as a possible gate in the computational model. Another motivation for this work comes from the area of neural networks [4, 8], where a linear threshold element is the basic processing element.

Spectral representation of Boolean functions: The idea of representing Boolean functions as polynomials over the field of rational numbers was first used in the context of counting the number of equivalent Boolean functions [11]. Every Boolean function can be computed as a polynomial over the reals as follows,

$$f(X) = \sum_{\alpha \in \{0,1\}^n} a_\alpha X^\alpha.$$

This representation will be called the *polynomial representation of f*. This representation is unique and the coefficients of the polynomial, $\{a_\alpha | \alpha \in \{0,1\}^n\}$, are called the spectral coefficients of f or the *spectrum* of f. We are interested in the L_1 and L_∞ norms associated with the spectrum. Namely,

$$L_1 = \sum_{\alpha \in \{0,1\}^n} |a_\alpha|$$

and

$$L_\infty = \max_{\alpha \in \{0,1\}^n} |a_\alpha|.$$

Our main result is revealing the connection between the complexity of a Boolean function and its spectral norms.

Example: Consider the function $f(x_1, x_2) = x_1 \wedge x_2$. Then

$$f(x_1, x_2) = \frac{1}{2}(1 + x_1 + x_2 - x_1 x_2).$$

Notice that $L_1 = 2$ and $L_\infty = 1/2$

Spectral representation and circuit complexity: Recently, the spectral approach turned out to be a useful tool in the study of Boolean functions. In [3] this approach was used to derive lower bounds for the size of decision trees and AND/OR circuits of depth two. In [10] this approach was used to obtain results on the influence of Boolean functions. Also a characterization of AC^0 functions using the spectrum was obtained in [12], where it was proved that an AC^0 function has almost all of its "power spectrum" on the low order coefficients and that it can be *approximated* by a sign of a polynomial with $O(n^{poly \log(n)})$ terms. The spectral approach was also used in [5] to obtain results on threshold functions. In particular, it was proved in [5] that the class of polynomial threshold functions is strictly contained in the class of functions that can be computed by a depth-2 circuit of linear threshold elements. The main tool in obtaining the results in [5] is a necessary condition for a function to be polynomial threshold which is based on the spectral approach.

Main result: The main result in this paper is a characterization of the functions in PT_1 using spectral norms. The characterization result we obtained here can be perceived as an extension of the result in [5] where it was proved that the number of terms in a threshold function that computes a Boolean function is $\Omega(L_\infty^{-1})$. In particular, we obtain a dual result—if a function has an L_1 norm that is bounded by a polynomial in n (number of variables) then it is polynomial threshold. Namely, we have a characterization of polynomial threshold functions using their spectral norms. We also prove that those conditions are strict, i.e. they are only necessary/sufficient conditions. Formally, let PL_1 be the class of Boolean functions for which the spectral norm L_1 is bounded by a polynomial in n. And let PL_∞ be the class of Boolean functions for which L_∞^{-1} is bounded by a polynomial in n. Then our main result is

Theorem 1:

$$PL_1 \subset PT_1 \subset PL_\infty.$$

Applications: There are two possible applications to our characterization results. The sufficient condition can be used to obtain upper bounds on the depth/size of a circuit that computes a certain function. For example, our result was used in [16] to prove the existence of depth-2, polynomial size MAJORITY circuits for comparison and for addition of two n-bit numbers (recently, constructions for both functions were obtained [2]). These results also led to a construction of a depth-4, polynomial size, MAJORITY circuit that computes the product of two n-bit numbers [16].

Necessary conditions can be used to obtain lower bounds. For example, in [5] it was proved that there are functions that can be computed by a depth-2 MAJORITY circuit but are not polynomial threshold functions.

In this paper we are mainly interested in using this approach for the analysis of AC^0 functions. In particular, it is interesting to find out whether $AC^0 \subset PL_1$? a result like this would imply that any AC^0 function can be computed with 2 layers of MAJORITY. We prove that this is not true, namely, that there exists an AC^0 function that has an exponential L_1 norm. Actually, we are able to prove a much stronger result. We exhibit a Boolean function such that $L_\infty^{-1} = \Omega(n^{poly \log(n)})$. Namely, there are AC^0 functions that can not be computed as a sign of a sparse polynomial. This result complements the result in [12] (about approximation of AC^0 functions).

From [5] we know that the class of polynomial threshold functions is strictly contained in the class of Boolean functions that can be computed by a depth-2, polynomial size, circuit of MAJORITY gates. Hence, in view of the results we obtained, it is natural to ask whether there are AC^0 functions that cannot be computed by a depth-2, polynomial size circuit of MAJORITY gates? We find a $\Omega(n^{poly \log(n)})$ lower bound on the size of a depth-2 circuit of MAJORITY gates that computes a certain AC^0 function (which is a degenerate version of the Inner Product Mod-2 function). This result provides a lower bound to the fact that three layers are sufficient to compute functions in AC^0 with MAJORITY gates [1].

The paper is organized as follows, in the next section we prove the characterization result, in Section 3 we describe the application to AC^0 functions and finally we address some open problems.

2 Characterizing Polynomial Threshold Functions

In this section we present a characterization of polynomial threshold functions using spectral norms. We will use the L_1 and L_∞ norms. Let PL_1 be the class of Boolean functions for which the spectral norm L_1 is bounded from above by a polynomial in n. And let PL_∞ be the class of Boolean functions for which L_∞^{-1} is bounded from above by a polynomial in n. Then our main result is that the class of polynomial threshold function is strictly between those two classes.

Theorem 1

$$PL_1 \subset PT_1 \subset PL_\infty.$$

Proof: In [5] it was proved that $PT_1 \subseteq PL_\infty$. Hence, to prove the theorem we need to prove the following three results.

1. Prove that $PL_1 \subseteq PT_1$. We do that in Lemma 1 below by using probabilistic arguments.
2. Prove that this is a proper inclusion, i.e. $PL_1 \subset PT_1$, by exhibiting a function f such that $f \in PT_1$ but $f \notin PL_1$. We prove that the EXACT function (outputs -1 if exactly half of its inputs are -1) is in PT_1 (Lemma 2) but not in PL_1 (Lemma 3).
3. Prove that the inclusion $PT_1 \subseteq PL_\infty$ is proper by exhibiting a function f, such that $f \in PL_\infty$ but $f \notin PT_1$ (Lemma 4).

Lemma 1

$$PL_1 \subseteq PT_1.$$

Proof: Let $f(X) \in PL_1$. We need to show that there exists a polynomial $F(X) = \sum_{\alpha \in S} w_\alpha X^\alpha$ such that $f(X) = sgn(F(X))$. Where $S \subset \{0,1\}^n$ and the size of S is bounded by some polynomial in n. The proof is by using the probabilistic method [6].

Let $\{a_\alpha | \alpha \in \{0,1\}^n\}$ be the spectral coefficients of a Boolean function $f(X) \in PL_1$. That is,

$$f(X) = \sum_{\alpha \in \{0,1\}^n} a_\alpha X^\alpha.$$

By the definition of PL_1, we have that $L_1 = \sum_{\alpha \in \{0,1\}^n} |a_\alpha|$ is bounded by some polynomial in n. We will prove that a sparse polynomial $F(X)$, such that $f(X) = sgn(F(X))$, exists; by constructing it from the polynomial representation of $f(X)$.

Define a probability distribution over the α's, $\alpha \in \{0,1\}^n$, as follows:

$$p_\alpha = \frac{|a_\alpha|}{L_1}.$$

We choose the terms to be included in $F(X)$ according to the foregoing probability distribution. A term $sgn(a_\alpha)X^\alpha$ is included in $F(X)$ with probability p_α. Formally, define N i.i.d. random variables Z_i, where $1 \leq i \leq N$, as follows,

$$Z_i = sgn(a_\alpha)X^\alpha \text{ with probability } p_\alpha.$$

And let

$$F(X) = \sum_{i=1}^{N} Z_i.$$

Hence, $F(X)$ is a polynomial constructed by selecting the terms at random. For any given X, $F(X)$ is a random variable. Now calculate the mean and variance of this random variable. Namely, consider a specific assignment to X and calculate the expected value and variance of $F(X)$. Assume, without loss of generality, that X is the all-1's vector (the absolute value of the mean and variance are the same for all assignments and the sign of the mean is $f(X)$). Hence, $F(1,1,\ldots,1)$ is the sum of N i.i.d. random variables, the Z_i's, each distributed as follows,

$$Z_i = \begin{cases} 1 & \text{with probability } \sum_{a_\alpha > 0} p_\alpha \\ -1 & \text{with probability } \sum_{a_\alpha < 0} p_\alpha. \end{cases}$$

Hence, the expected value of Z_i is,

$$\begin{aligned} E(Z_i) &= (\sum_{a_\alpha > 0} p_\alpha - \sum_{a_\alpha < 0} p_\alpha) \\ &= \frac{f(1,1,\ldots,1)}{L_1}. \end{aligned}$$

And the variance of Z_i is,

$$\begin{aligned} Var(Z_i) &= E(Z_i^2) - \frac{f^2(1,1,\ldots,1)}{L_1^2} \\ &= 1 - \frac{1}{L_1^2}. \end{aligned}$$

Hence,

$$E(F(1,1,\ldots,1)) = \frac{Nf(1,1,\ldots,1)}{L_1}.$$

And

$$Var(F(1,1,\ldots,1)) = N(1 - \frac{1}{L_1^2}).$$

Hence, choosing $N \geq 2nL_1^2$ and applying the Chernoff bound (or the Central Limit Theorem)[6] we have that for any given X,

$$\text{Prob}[f(X) \neq sgn(F(X)] \leq e^{-n} < 2^{-n}$$

And by the union bound we get that

$$\text{Prob}[f(X) \neq sgn(F(X)), \text{ for some } X] < 1.$$

Hence,

$$\text{Prob}[f(X) = sgn(F(X)), \text{ for all } X] > 0.$$

Thus, for any Boolean function f with a "small" spectral L_1 norm there exists a sparse $F(X)$ (the number of terms is $O(nL_1^2)$), such that $f(X) = sgn(F(X))$ for all $X \in \{1, -1\}^n$. □

A couple of important remarks with regard to the above lemma:

1. By the same proof technique we get that, in general, a Boolean function can be computed as a sign of a polynomial with $O(nL_1^2)$ terms.

2. Using the same proof technique we can prove that a Boolean function with a "small" L_1 norm can be approximated by a sparse polynomial (without a sign).

Before we prove the next three lemmas we define the following two useful functions.

Definition 1 The EXACT_n function is a Boolean function which is defined for even $n = 2k$ variables.

$$\text{EXACT}_n(X) = \begin{cases} -1 & \text{exactly } k \text{ -1's in } X \\ 1 & \text{otherwise.} \end{cases}$$

Definition 2 The Complete Quadratic (CQ_n) function is a Boolean function of n variables such that

$$\text{CQ}_n(X) = (x_1 \wedge x_2) \oplus (x_1 \wedge x_3) \oplus \cdots \oplus (x_{n-1} \wedge x_n)$$

That is, CQ_n consists of the sum mode 2 of all the $\binom{n}{2}$ AND's between pairs of variables.

Lemma 2

$$EXACT_n \in PT_1.$$

Proof (sketch): We prove it by constructing a polynomial $F(X)$ such that $\text{EXACT}_n(X) = sgn(F(X))$. Recall that $n = 2k$. Then

$$F(X) = (k-1) + x_1x_2 + x_1x_3 + \cdots + x_{2k-1}x_{2k}.$$

□

Lemma 3 *Let $\hat{L}^n$ be the L_1 norm of the $EXACT_n$ function. Let $n = 2k$; then $\hat{L}^n \geq \frac{2^k}{k}$. That is, $EXACT_n \notin PL_1$.*

Proof: omitted.

Lemma 4

$$PT_1 \subset PL_\infty.$$

Proof (sketch): We prove that the inclusion is proper by constructing a function which is not in PT_1 but is in PL_∞. Consider the Complete Quadratic function ($CQ_n(X)$) which is defined in Definition 2 above. From [5] we know that $CQ_n(X)$ is not in PT_1 (and also not in PL_∞). Consider the function, $f_{n+1}(X)$, of $n+1$ variables which is constructed from $CQ_n(X)$ as follows:

$$f_{n+1}(X) = CQ_n(X) \wedge x_{n+1}.$$

One can prove that $f_{n+1}(X) \in PL_\infty$ and $f_{n+1}(X) \notin PT_1$. □

3 AC^0 Functions and Spectral Norms

One of the possible applications of our results is to obtain bounds on the complexity of Boolean functions. In particular, the complexity of computing Boolean functions with circuits of MAJORITY gates. In this section we address a few questions related to computing AC^0 functions using MAJORITY gates. See [16] for results related to the complexity of computing arithmetic functions using MAJORITY gates.

Definition 3 Let MAJ_k be the class of functions that can be computed by a depth-2, polynomial size circuit of MAJORITY gates assuming that at every gate it is possible to negate any of the inputs. This model of computation is equivalent to a polynomial size circuit of linear threshold elements of depth k, such that the weights at every gate are bounded by a polynomial (in the number of variables).

The following theorem is a summary of our results related to AC^0 functions.

Theorem 2 *The following is true:*

1. $AC^0 \not\subset PL_1$.
2. $AC^0 \not\subset PL_\infty$.
3. $AC^0 \not\subset \mathrm{MAJ}_2$.

In what follows we describe the proofs for the three parts of the theorem. Clearly, the above claims are related. We give the details of the proofs for all the three claims since we use a different technique for every one of them.

Notice the following facts,

1. $$PL_1 \subset MAJ_2.$$
   This follows from Theorem 1, $PL_1 \subset PT_1$, and the fact that $PT_1 \subset MAJ_2$ (see [5]).
2. By the same arguments as in Lemma 1, if the L_1 spectral norm is $O(n^{poly\log(n)})$ then the function can be computed by a depth-2, $O(n^{poly\log(n)})$ size circuit of MAJORITY gates.

Hence, it is natural to ask whether there are AC^0 functions that have an exponential L_1 norm? A negative answer to this question will result in upper bound on the complexity of computing AC^0 functions with MAJORITY gates. Unfortunately, we prove that indeed there is a function in AC^0 that has an exponential L_1 spectral norm. This result is another evidence that 3 layers of MAJORITY might be needed to compute a function in AC^0 [1].

Lemma 5

$$AC^0 \not\subset PL_1$$

Proof: We prove the lemma by exhibiting a function in AC^0 that has an exponential L_1 norm. First, consider the following Boolean function,

$$\hat{f}(X) = (x_1 \wedge x_2) \vee (x_3 \wedge x_4).$$

It can be checked that the L_1 spectral norm of $\hat{f}$ is 3.5. Now consider the function f which is the AND of n copies of $\hat{f}$ where each copy consists of 4 different variables. Namely, let

$$\hat{f}_i = (x_{4i-3} \wedge x_{4i-2}) \vee (x_{4i-1} \wedge x_{4i}).$$

Then

$$f = \bigwedge_{i=1}^{n} \hat{f}_i.$$

Let L be the L_1 spectral norm of f and $\hat{L}$ be the L_1 spectral norm of $\hat{f}_i$ then

$$L \geq -1 + \frac{1}{2^{n-1}}(\hat{L} - 1)^n \geq 1.25^n.$$

This follows from the fact that the polynomial representation of f can be reduced to the following form,

$$f = 1 - \frac{1}{2^{n-1}} \prod_{i=1}^{n} (1 - \hat{f}_i).$$

Hence, f has an exponential L_1 spectral norm. Since $f \in AC^0$ the result follows. □

The next natural question is whether $AC^0 \subset PL_\infty$? Namely, is there an AC^0 function which its L_∞^{-1} is not bounded by a polynomial in n. This will also show that there is an AC^0 function which is not in PT_1, and it will complement the result in [12] that AC^0 functions can be approximated by a sign of a polynomial with $O(n^{poly \log(n)})$ terms.

Lemma 6 *There exists a Boolean function f, such that $L_\infty^{-1} = \Omega(n^{poly \log(n)})$.*

Proof: Let f be a the AND function of 2 variables. Then

$$f(x_1, x_2) = \frac{1}{2}(1 + x_1 + x_2 - x_1 x_2).$$

The next step is just to take $\log(n)$ disjoint copies of f and compute the XOR of these functions. Clearly, we can do it in depth 2 with $O(n)$ gates (using the exhaustive approach). The resulting function will have $L_\infty^{-1} = n$. If we iterate this process k times we obtain a function with $L_\infty^{-1} = n^{\log^k(n)}$. Now, from [5] we have that the number of terms in the representation as a sign of a polynomial is $\Omega(L_\infty^{-1})$ and the result follows. □

If we look carefully at the function that was constructed in the previous lemma we find that this function is actually a degenerate version of the Inner Product Mod-2 (IP2) function. Where

$$IP2(X) = (x_1 \wedge x_2) \oplus (x_3 \wedge x_4) \cdots \oplus (x_{n-1} \wedge x_n)$$

is defined for even n and consists of the sum mod-2 of the AND's between the $n/2$ pairs. The function in the proof of the foregoing lemma is an inner product mod-2 of two vectors each with $\log^k(n)$ variables. We call this function DIP2 (Degenerate IP2). Now, recall that in [7] it was proved that the IP2 function is not in MAJ_2. We use the same technique as in [7] and get,

Lemma 7 *A depth-2 circuit of MAJORITY gates that computes the function DIP2 is of size $\Omega(n^{poly \log(n)})$.*

Since DIP2 $\in AC^0$, this result constitutes the first known lower bound to the result of Allender [1]. Namely, there are AC^0 functions that can not be computed by a depth-2 circuit of MAJORITY gates.

4 Open Problems

There are a few open problems related to the results in the paper:

1. We proved in Theorem 1 that any Boolean function with a "small" L_1 spectral norm

(PL_1 function) can be computed as a sign of a sparse polynomial (is a PT_1 function). Using this result it is proved in [16] that the COMPARISON and ADDITION functions are in PL_1, thus, are also in PT_1 and MAJ_2. Explicit constructions for those functions are presented in [2]. Given a Boolean function in PL_1, it will be nice to have a general method to construct a sparse polynomial whose sign equals to the function.

2. Note that the function DIP2 can actually be computed with $O(n^{poly \log(n)})$ terms/gates. Is it possible to compute *exactly* any AC^0 function by a sign of a polynomial with $O(n^{poly \log(n)})$ terms? A result like this implies that any AC^0 function is computable by a depth-2, $O(n^{poly \log(n)})$ size circuit of MAJORITY gates. We note here that in order to prove an exponential lower bound on the number of terms we will need a different technique than the one used here. The reason is that it is possible to prove (based on [12]) that for AC^0 functions $L_\infty^{-1} = O(n^{poly \log(n)})$.

3. A more interesting question will be to find an AC^0 function which cannot be computed by a depth-3, polynomial size circuit of MAJORITY gates. This will give a better lower bound for the result in [1].

4. Is there some fixed d, such that any AC^0 function can be computed by depth-d, polynomial size circuit of MAJORITY gates?

Acknowledgement: We thank Noga Alon for his useful observations that greatly improved this paper and to Chaim Gotsman and Sunny Siu for their comments on an early draft of this paper.

References

[1] E. Allender, *A Note on the Power of Threshold Circuits*, Proc. 30th IEEE Symposium on Foundations of Computer Science, pp. 580–584, 1989.

[2] N. Alon and J. Bruck, personal communication.

[3] Y. Brandman, A. Orlitsky and J. Hennessy, *A Spectral Lower Bound Technique for the size of Decision Trees and Two-Level AND/OR Circuits*, IEEE Trans. on Computers, Vol. 39, No. 2, pp. 282–287, February 1990.

[4] J. Bruck, *Computing with Networks of Threshold Elements*, Ph.D. Thesis, Stanford University, March 1989.

[5] J. Bruck, *Harmonic Analysis of Polynomial Threshold Functions*, SIAM Journal on Discrete Mathematics, Vol. 3, No. 2, pp. 168–177, May 1990.

[6] P. Erdös and J. Spencer, *Probabilistic Methods in Combinatorics*, Academic Press, 1974.

[7] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy and G. Turán, *Threshold Circuits*

of Bounded Depth, Proc. 28th IEEE Symposium on Foundations of Computer Science, pp. 99-110, 1987.

[8] J. J. Hopfield, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, Proceedings of the National Academy of Sciences USA, Vol. 79, pp. 2554-2558, 1982.

[9] D. S. Johnson, *The NP-Completeness Column: An Ongoing Guide*, J. of Algorithms, 7, pp. 289-305, 1986.

[10] J. Kahn, G. Kalai and N. Linial, *The Influence of Variables on Boolean Functions,* Proc. 29th IEEE Symposium on Foundations of Computer Science, pp. 68–80, 1988.

[11] R. J. Lechner, *Harmonic Analysis of Switching Functions*, in: A. Mukhopadhyay (ed.), *Recent Development in Switching Theory*, Academic Press, 1971.

[12] N. Linial, Y. Mansour and N. Nisan, *Constant Depth Circuits, Fourier Transforms, and Learnability* Proc. 30th IEEE Symposium on Foundations of Computer Science, pp. 574–579, 1989.

[13] I. Parberry and G. Schnitger, *Parallel Computation with Threshold Functions*, J. Comput. System Sci., vol. 36, no. 3, pp. 278–302, June 1988.

[14] A. A. Razborov, *Lower Bounds for the Size of Circuits of Bounded Depth with Basis* $\{\wedge, \oplus\}$, Math. notes of the Acad. of Science of the USSR, 41:4, pp. 333-338, Sept. 1987.

[15] J. H. Reif, *On Threshold Circuits and Polynomial Computation*, Proc. 2nd Structure in Complexity Theory Conf., pp. 118-123, 1987.

[16] K. Y. Siu and J. Bruck, *On the Dynamic Range of Linear Threshold Elements*, IBM Tech. Report, RJ7237, January 1990.

[17] R. Smolensky, *Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity*, 19th ACM Symposium on Theory of Computing, pp. 77-82, 1987.

[18] A. C. Yao, *Circuits and Local Computations*, 21st ACM Symposium on Theory of Computing, pp. 186–196, 1989.