

HARMONIC ANALYSIS OF POLYNOMIAL THRESHOLD FUNCTIONS*

JEHOSHUA BRUCK†

Abstract. The analysis of linear threshold Boolean functions has recently attracted the attention of those interested in circuit complexity as well as of those interested in neural networks. Here a generalization of linear threshold functions is defined, namely, polynomial threshold functions, and its relation to the class of linear threshold functions is investigated.

A Boolean function is polynomial threshold if it can be represented as a sign function of a polynomial that consists of a polynomial (in the number of variables) number of terms. The main result of this paper is showing that the class of polynomial threshold functions (which is called PT_1) is strictly contained in the class of Boolean functions that can be computed by a depth 2, unbounded fan-in polynomial size circuit of linear threshold gates (which is called LT_2).

Harmonic analysis of Boolean functions is used to derive a necessary and sufficient condition for a function to be an S -threshold function for a given set S of monomials. This condition is used to show that the number of different S -threshold functions, for a given S , is at most $2^{(n+1)|S|}$. Based on the necessary and sufficient condition, a lower bound is derived on the number of terms in a threshold function. The lower bound is expressed in terms of the spectral representation of a Boolean function. It is found that Boolean functions having an exponentially small spectrum are not polynomial threshold. A family of functions is exhibited that has an exponentially small spectrum; they are called “semibent” functions. A function is constructed that is both semibent and symmetric to prove that PT_1 is properly contained in LT_2 .

Key words. bent functions, Boolean functions, circuit complexity, harmonic analysis, lower bounds, threshold functions

1. Introduction. A Boolean function $f(X)$ is a *threshold function* if

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

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 $f: \{1, -1\}^n \rightarrow \{1, -1\}$; 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. It can be shown that 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 such a result 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, is made of all the Boolean functions that can be computed by a single threshold gate in which the number

* Received by the editors July 18, 1988; accepted for publication (in revised form) May 19, 1989. This work was supported by the Air Force Office of Scientific Research.

† IBM Almaden Research Center, 650 Harry Road, K54/802, San Jose, California 95120-6099.

of monomials is bounded by a polynomial in n . The main goal of this paper is the study of this complexity class and its relations with other known complexity classes of Boolean functions.

More precisely, let $S \subseteq \{0, 1\}^n$; a Boolean function f is an S -threshold function if there exist integers that we call weights (the w_α 's) such that $f(X) = \text{sgn}(\sum_{\alpha \in S} w_\alpha X^\alpha)$. Hence, a Boolean function $f(X)$ is in PT_1 if there exists a set S , with $|S|$ bounded by a polynomial in n , such that $f(X)$ is S -threshold. Notice that there is no restriction on the size of the weights.

A related class of functions is the class of *linear threshold* functions [7], [11]. A Boolean function is *linear threshold* if it is S -threshold with S corresponding to the constant and linear terms. We define LT_1 to be the class of functions that are computable by a single linear threshold gate. The next step is to define complexity classes that relate to circuits. Define $LT_d(PT_d)$ to be the class of Boolean functions that can be computed by an unbounded fan-in polynomial size circuit of depth at most d which consists of linear (polynomial) threshold gates.

Some motivation: recently, there has been a considerable interest in the study of the computational model of bounded depth unbounded fan-in polynomial size circuits that consist of linear threshold gates [5], [12], [14], [17]. This interest follows from recent results in complexity of circuits [8], [13], [16] which indicate that MAJORITY (hence, linear threshold functions) cannot 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. Notice that in the results in [5] the weights are bounded by a polynomial in n . To make the distinction from the case in which the weights are not bounded we put "hats." Namely, $\widehat{LT}_d$ and $\widehat{PT}_d$ correspond to circuits with bounded weights. Using this notation, a related result in [5] is:

$$\widehat{LT}_1 \subset \widehat{LT}_2 \subset \widehat{LT}_3.$$

In this context, the study of circuits of polynomial threshold functions can be viewed as study of a model in which a single gate is rather powerful. Namely, there is no "trivial" function that cannot be computed by a single gate. For example, PARITY, EXACT $_k$ (output -1 if and only if k of its inputs are -1) and the characteristic function of a linear subspace (code) [1], [2], [10] are all in PT_1 but none are in LT_1 . This fact suggests that the separation between the classes PT_1 and PT_2 is not going to be an easy problem. The other motivation for this work comes from the area of neural networks [1], [6], where a linear threshold element is the basic processing element. One of the reasons for the limited computational power of neural networks is the fact that every neuron (node) is not very powerful. One solution to that is to consider generalized networks in which every node is computing a polynomial threshold function. In this context, our result is a characterization of generalized neural networks with respect to ordinary neural networks.

The main result in the paper is a characterization of the power of PT_1 with respect to the hierarchy of circuits of linear threshold functions. We have:

$$LT_1 \subset PT_1 \subset LT_2,$$

which also implies that $PT_1 \subset PT_2$.

Clearly, $LT_1 \subset PT_1$ follows from the fact that PARITY is not in LT_1 . But, PARITY(X) = $\text{sgn}(x_1 x_2 \cdots x_n)$, hence it is in PT_1 . Proving that $PT_1 \subseteq LT_2$ is based on the observation that PARITY does not require the full strength of a depth 2 circuit of linear threshold elements and is described in § 2. To prove that this containment is proper, we

developed a technique for deriving lower bounds for the number of monomials in a threshold function. This technique is based on the spectral representation of a Boolean function. Most of the paper is devoted to the development of this technique and its applications.

In § 3 we review the subject of harmonic analysis of Boolean functions [9] and show that every Boolean function has a representation as a polynomial over the rationals and hence as a threshold function.

In § 4 we use the spectral representation of Boolean functions and derive a necessary and sufficient condition for a function to be an S -threshold function for a given S . We use this condition to show that the number of different S -threshold functions, for a given S , is at most $2^{(n+1)|S|}$. These results turn out to be a generalization of known results for linear threshold functions [3], [7], [11].

In § 5 we use the necessary and sufficient condition to derive a lower bound on the number of monomials in a threshold function. The lower bound is expressed in terms of the spectral representation of a Boolean function. We find that Boolean functions that have an exponentially small spectrum are not polynomial threshold.

In § 6 we exhibit a family of functions that has an exponentially small spectrum; we call them “semibent” functions. We construct a function that is both semibent and symmetric to prove that PT_1 is properly contained in LT_2 . Finally, we address some open problems.

2. Simulation of polynomial threshold functions. It is a well-known result that PARITY (as well as other symmetric functions) is in LT_2 [5], [12]. From this fact it follows that a polynomial threshold function can be simulated by a depth 3 circuit of linear threshold gates. The idea is to compute the monomials using depth 2 circuits and combine the monomials in the gate in the third layer. What we will show here is that depth 2 is enough.

THEOREM 2.1. $PT_1 \subseteq LT_2$.

Proof. The idea is to notice that PARITY does not require the full power of a depth 2 circuit of linear threshold gates. Actually, PARITY can be realized by a set of linear threshold elements in the first layer while, in the second layer, we need only to sum and add a constant to get the desired result. Namely, we do not have to use the threshold operation in the second layer.

Example. Let $f(X) = x_1 \oplus x_2$. Let

$$F_1(X) = -1 - x_1 - x_2 \quad \text{and} \quad F_2(X) = -1 + x_1 + x_2.$$

It can be verified that:

$$f(X) = 1 + \text{sgn}(F_1(X)) + \text{sgn}(F_2(X)).$$

Note that we are using the $\{1, -1\}$ representation instead of $\{0, 1\}$, respectively. The above is true in general for PARITY of n variables. In the general case we need up to $n + 1$ linear threshold gates in the first layer and again only summation and addition of a constant in the second layer. Using this observation a polynomial threshold function can be simulated by depth 2 linear threshold circuit in a way similar to that done with depth 3. $\square$

Proving containment of polynomial threshold functions in LT_2 turns out to be a very easy problem compared to the problem of proving that this containment is proper; the latter requires proving lower bounds. The rest of the paper is devoted to the development of a technique for getting lower bounds for polynomial threshold functions and to the application of this technique to getting separation results.

3. Polynomial representation of Boolean functions. In this section the representation of Boolean functions as polynomials over the field of rational numbers is presented.

DEFINITION. A *Hadamard matrix* of order m , denoted by H_m , is an $m \times m$ matrix of $+1$'s and -1 's such that:

$$(1) \quad H_m H_m^T = m I_m$$

where I_m is the $m \times m$ identity matrix. The above definition is equivalent to saying that any two rows of H are orthogonal.

Hadamard matrices of order 2^k exist for all $k \geq 0$. The so-called Sylvester construction is as follows [10]:

$$(2) \quad \begin{aligned} H_1 &= [1] \\ H_2 &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ H_{2^{n+1}} &= \begin{bmatrix} H_{2^n} & H_{2^n} \\ H_{2^n} & -H_{2^n} \end{bmatrix}. \end{aligned}$$

DEFINITION. Given a Boolean function f of order n , p is a polynomial (with coefficients over the field of rational numbers) *equivalent* to f if and only if for all $X \in \{1, -1\}^n$:

$$f(X) = p(X).$$

As an example, let $f = x_1 \oplus x_2$; that is, f is the XOR function of two variables. It is easy to check that in the $\{1, -1\}$ representation $p(x_1, x_2) = x_1 x_2$. Notice that for every Boolean function f , the polynomial p is linear in each of its variables because $x^2 = 1$ for $x \in \{-1, 1\}$. It is known that every Boolean function has a unique representation as a polynomial [2], [9]. This representation is derived by using the Hadamard matrix, as described by the following theorem.

THEOREM 3.1. *Let f be a Boolean function of order n . Let p be a polynomial equivalent to f . Let A_{2^n} denote the vector of coefficients of p . Let P_{2^n} denote the vector of the 2^n values of p (and f). Then:*

1. *The polynomial p always exists and is unique.*
2. *The coefficients of p are computed as follows:*

$$A_{2^n} = \frac{1}{2^n} H_{2^n} P_{2^n}.$$

Proof (sketch). The proof is constructive. The idea is to compute A_{2^n} by solving a system of linear equations. $\square$

Example. Let f be the AND function of two variables,

$$(3) \quad f(x_1, x_2) = x_1 \wedge x_2.$$

By Theorem 3.1,

$$(4) \quad p(x_1, x_2) = \frac{1}{2} (1 + x_1 + x_2 - x_1 x_2).$$

The entries of the vector A are denoted by $\{a_\alpha \mid \alpha \in \{0, 1\}^n\}$ and called the spectral representation of a function. Note that a_α is the coefficient of X^α in the polynomial representation where $X^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$. Hence, every Boolean function can be written as:

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

The above method is applicable not only to Boolean functions but also to any function of the form $f : \{1, -1\}^n \rightarrow \mathbb{R}$.

4. Necessary and sufficient conditions and their applications. We use the polynomial representation of Boolean functions presented in the previous section to derive a necessary and sufficient condition for a function to be an S -threshold function for arbitrary S . This result turned out to be a generalization of a known result for linear threshold functions [3], [7], [11].

Let $f(X) = \text{sgn}(F(X))$ be a threshold function. Without loss of generality, assume that $F(X) \neq 0$ for all $X \in \{1, -1\}^n$. The following simple lemma enables us to express the relation between $f(X)$ and $F(X)$ in a global way. That is, instead of having 2^n conditions we have only one.

LEMMA 4.1. *Let $f(X)$ be a Boolean function and let $F(X) \neq 0$ for all $X \in \{1, -1\}^n$; then*

$$(5) \quad f(X) = \text{sgn}(F(X)) \quad \forall X \in \{1, -1\}^n$$

if and only if

$$(6) \quad \sum_{X \in \{1, -1\}^n} |F(X)| = \sum_{X \in \{1, -1\}^n} f(X)F(X).$$

Proof. Suppose there exists an X_1 that violates (5); this implies $(F(X) \neq 0 \text{ for all } X)$ that

$$|F(X_1)| > f(X_1)F(X_1).$$

Hence, (6) is also not true because violation in (5) can only decrease the value on the right-hand side of (6). Clearly, if (5) is true so is (6). $\square$

LEMMA 4.2.

$$(7) \quad \sum_{X \in \{1, -1\}^n} X^\alpha = \begin{cases} 2^n & \text{if } \alpha = \text{all-0 vector} \\ 0 & \text{else.} \end{cases}$$

Proof. The proof follows from the fact that X^α corresponds to a row of a Sylvester type Hadamard matrix (see Theorem 3.1). $\square$

The necessary and sufficient condition follows from (6) by using the polynomial representation of a Boolean function.

THEOREM 4.1. *Fix $S \subseteq \{0, 1\}^n$. Let $F(X) = \sum_{\alpha \in S} w_\alpha X^\alpha$. Let $f(X)$, $X \in \{-1, 1\}^n$, be a Boolean function with spectral representation $\{a_\alpha | \alpha \in \{0, 1\}^n\}$. Then:*

$$(8) \quad f(X) = \text{sgn}(F(X)) \quad \forall X \in \{1, -1\}^n$$

if and only if

$$\sum_{X \in \{1, -1\}^n} |F(X)| = 2^n \sum_{\alpha \in S} w_\alpha a_\alpha.$$

Proof. By the Lemma 4.1 it is enough to show that

$$\sum_{X \in \{1, -1\}^n} f(X)F(X) = 2^n \sum_{\alpha \in S} w_\alpha a_\alpha.$$

We write $f(X)$ as a polynomial:

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

and find that the constant term of $f(X)F(X)$ is $\sum_{\alpha \in S} w_\alpha a_\alpha$. Hence, the result follows from Lemma 4.2. $\square$

Theorem 4.1 is interesting because it suggests that an S -threshold function is fully characterized by the set of spectral coefficients that correspond to S . To see the power of the condition we present two applications. The first application (Theorem 4.2) is a generalization of the known result that PARITY is not in LT_1 and the second application (Theorem 4.3) is an upper bound on the number of S -threshold functions.

THEOREM 4.2. *Let*

$$\text{PARITY}(X) = \begin{cases} -1 & \text{odd number of } -1\text{'s in } X \\ 1 & \text{otherwise.} \end{cases}$$

Then PARITY is not an S -threshold function for any S such that the all-1 vector is not in S , i.e., $(1, 1, \dots, 1) \notin S$.

Proof. Note that $\text{PARITY}(X) = x_1 x_2 \cdots x_n$; hence, for a set S that does not include the all-1 vector we have

$$\sum_{X \in \{1, -1\}^n} |F(X)| \neq 2^n \sum_{\alpha \in S} w_\alpha a_\alpha = 0.$$

Thus, the condition cannot be satisfied ($F(X) \neq 0$). $\square$

A nice interpretation of this result is as follows: the X^α form a basis for the space of Boolean functions. Hence, by the definition of a basis, an element of the basis cannot be represented as a linear combination of the other elements in the basis. What Theorem 4.2 is saying is that it is also impossible to represent an element of the basis as the *sign* of a linear combination of the other elements in the basis.

LEMMA 4.3. *Let f_1 and f_2 be Boolean functions with $\{a_\alpha^i \mid \alpha \in \{0, 1\}^n\}$, $i = 1, 2$, as their spectral representation, respectively. Assume that at least one of the functions is an S -threshold function. Then $f_1(X) = f_2(X)$ for all $X \in \{1, -1\}^n$ if and only if $a_\alpha^1 = a_\alpha^2$ for all $\alpha \in S$.*

Proof. Suppose $f_1 = f_2$. By the uniqueness of the spectral representation (Theorem 3.1) we get the “only if.” Assume without loss of generality that f_1 is an S -threshold function. Now suppose $a_\alpha^1 = a_\alpha^2$ for all $\alpha \in S$. By Theorem 4.1 and the assumption that f_1 is S -threshold we get that there exists a set of weights that satisfies (8) for both f_1 and f_2 . Hence, f_2 is also S -threshold and $F_1(X) = F_2(X)$ for all $X \in \{1, -1\}^n$. Thus, $f_1 = f_2$. $\square$

COROLLARY 4.1. *Consider the set $S \subseteq \{0, 1\}^n$. Let f_1 and f_2 be Boolean functions of n variables. If $a_\alpha^1 = a_\alpha^2$ for all $\alpha \in S$, then either both f_1 and f_2 are S -threshold or both are not S -threshold.*

One application of the above is counting the number of different S -threshold functions.

THEOREM 4.3. *Fix $S \subseteq \{0, 1\}^n$. There are at most $2^{(n+1)|S|}$ different S -threshold functions.*

Proof. It can be shown that for all $\alpha \in \{0, 1\}^n$, a_α can assume at most $(2^n + 1)$ different values. Hence, there are at most $2^{(n+1)|S|}$ different sets of $|S|$ spectral coefficients. Thus, the result follows from Corollary 4.1. $\square$

The above turns out to be a generalization of a known [7], [11] upper bound on the number of linear threshold functions for which $|S| = n + 1$.

5. Lower bounds. The necessary and sufficient condition that is derived above is used to derive lower bounds on the number of monomials in a threshold function, again, by using the spectral representation. Let $f(X) = \text{sgn}(F(X))$ be an S -threshold function, namely,

$$F(X) = \sum_{\alpha \in S} w_\alpha X^\alpha.$$

We want to find lower bounds for $|S|$ as a function of the spectral representation of $f(X)$.

LEMMA 5.1. *For all $\alpha \in S$:*

$$2^n |w_\alpha| \leq \sum_{X \in \{1, -1\}^n} |F(X)|.$$

Proof. First we prove the statement for α being the all-0 vector:

$$\begin{aligned} \sum_{X \in \{1, -1\}^n} |F(X)| &= \sum_{F(X) > 0} F(X) - \sum_{F(X) < 0} F(X) \\ &= \sum_{X \in \{1, -1\}^n} F(X) - 2 \sum_{F(X) < 0} F(X) \\ &\stackrel{(a)}{=} 2^n w_{00 \dots 00} - 2 \sum_{F(X) < 0} F(X) \\ &\geq 2^n w_{00 \dots 00}. \end{aligned}$$

Note that (a) follows from Lemma 4.2. The proof for arbitrary α follows from the fact that $|X^\alpha| = 1$; hence:

$$|F(X)| = |X^\alpha| F(X) = |X^\alpha F(X)|.$$

Hence, we can make any w_α be the constant term without changing the value of $|F(X)|$. If $w_\alpha < 0$ we take $-F(X)$ and get the result. $\square$

THEOREM 5.1. *Let $f(X) = \text{sgn}(\sum_{\alpha \in S} w_\alpha X^\alpha)$ be an S -threshold function and let $\{a_\alpha | \alpha \in \{0, 1\}^n\}$ be its spectral representation; then*

$$(i) \quad |S| \geq \left(\sum_{\alpha \in S} p_\alpha |a_\alpha| \right)^{-1}$$

$$(ii) \quad \geq \hat{a}^{-1}$$

where

$$p_\alpha = \frac{|w_\alpha|}{\sum_{\alpha \in S} |w_\alpha|}$$

and

$$\hat{a} = \max_{\alpha \in S} |a_\alpha|.$$

Proof. We first prove (i). By Theorem 4.1 and Lemma 5.1, for all $\alpha \in S$:

$$|w_\alpha| \leq \sum_{\alpha \in S} w_\alpha a_\alpha.$$

We sum the above inequality over all $\alpha \in S$ and get:

$$\begin{aligned} \sum_{\alpha \in S} |w_\alpha| &\leq |S| \sum_{\alpha \in S} w_\alpha a_\alpha \\ &\leq |S| \sum_{\alpha \in S} |w_\alpha| a_\alpha. \end{aligned}$$

Therefore, we get (i). For (ii), just note that $p_\alpha \geq 0$ and $\sum p_\alpha = 1$. $\square$

We summarize this section by the following corollary.

COROLLARY 5.1. *Fix any $\varepsilon > 0$. Let $f(X)$ be a Boolean function of n variables. If $|a_\alpha| \leq 2^{-\varepsilon n}$ for all $\alpha \in \{0, 1\}^n$, then, for n sufficiently large, $f(X)$ is not a polynomial threshold function.*

6. Separating by semibent functions. We use Corollary 5.1 to get separation results by looking at functions that have an exponentially small spectrum.

DEFINITION. A Boolean function $f(X)$ is called “bent” [4], [10], [15] if and only if $|a_\alpha| = 2^{-n/2}$ for all $\alpha \in \{0, 1\}^n$. Notice that bent functions are defined for even n only.

PROPOSITION 6.1. *The Inner Product Mode 2 (IP2) function, i.e.,*

$$f(X) = (x_1 \wedge x_2) \oplus (x_3 \wedge x_4) \oplus \cdots \oplus (x_{2n-1} \wedge x_{2n})$$

is a bent function.

Proof. See [10]. A sketch of an alternative proof: it can be proven by induction on n that IP2, when written as a vector, is actually an eigenvector (with eigenvalue $= 2^n$) of a Sylvester type Hadamard matrix of order 2^{2n} . Hence, $|a_\alpha| = 2^{-n}$ for all $\alpha \in \{0, 1\}^n$. $\square$

THEOREM 6.1. $PT_1 \subset PT_2$.

Proof. By Corollary 5.1 and Proposition 6.1 the function IP2 is not in PT_1 . But it is in PT_2 : the AND’s are computed in the first level and the XOR is computed by the gate in the second layer. $\square$

DEFINITION. Let $\varepsilon > 0$ be fixed. A Boolean function $f(X)$ is called “semibent” if for all $\alpha \in \{0, 1\}^n$, $|a_\alpha| \leq 2^{-\varepsilon n}$. Clearly, a bent function is also a semibent function.

THEOREM 6.2. $PT_1 \subset LT_2$.

Proof. The fact that $PT_1 \subseteq LT_2$ is proved in Theorem 2.1. To show that it is a proper containment we must find a function that is in LT_2 but not in PT_1 . Every symmetric function is in LT_2 [5], [12] and every semibent function is not in PT_1 (Corollary 5.1). Hence, a natural candidate for such a function will be a symmetric semibent function. Indeed, a symmetric semibent function exists for all n as stated in Proposition 6.2 below. $\square$

Consider the function:

$$(9) \quad f(X) = (x_1 \wedge x_2) \oplus (x_1 \wedge x_3) \oplus \cdots \oplus (x_{n-1} \wedge x_n).$$

Hence, $f(X)$ consists of the sum mod 2 of all the $\binom{n}{2}$ possible AND’s between pairs of variables. We call this function the Complete Quadratic (CQ) function. Clearly, CQ is a symmetric function.

PROPOSITION 6.2. *CQ is bent for n even, and semibent for n odd.*

Proof. Actually, we can compute the exact spectral representation of CQ for every n ; see the Appendix.

7. Concluding remarks. We proved that the class of polynomial threshold functions is strictly contained in the class of functions computable by depth-2 linear threshold circuits. The technique we are using is based on harmonic analysis (polynomial representation) of Boolean functions. One interesting future direction is to try and prove other (even known) results in circuit complexity using this technique. For example, it is possible to prove that $\widehat{LT}_2 \subset \widehat{LT}_3$ (see [5]) using those techniques [1]. However, the main open problem is to find the exact relation between the class $LT^0 = \cup_{d \in \mathcal{A}} LT_d$ and the well known class NC^1 . In [17] Yao proved separation in the monotonic version of LT^0 . Here, we were concentrating on the relation between subclasses of PT^0 and subclasses of LT^0 , with the goal of getting separation in LT^0 . In particular we have the following conjecture.

CONJECTURE. For all $d \in \mathcal{N}$: $LT_d \subset PT_d \subset LT_{2d}$.

Our main result is proving the above for $d = 1$. A possible way to prove the conjecture is by induction on d ; clearly, we did not succeed in pursuing this direction.

Appendix.

A. The spectrum of the Complete Quadratic function. The Complete Quadratic function is defined in § 6. Here we prove Proposition 6.2. We start by giving an equivalent definition of the function CQ.

PROPOSITION A.1.

$$\text{CQ}(X) = \begin{cases} 1 & \text{no. of } -1\text{'s in } X = 0 \text{ or } 1 \bmod 4 \\ -1 & \text{otherwise.} \end{cases}$$

Proof. Suppose there are m -1 's in X . Since a pair in (9) is -1 if and only if both variables are -1 , we have exactly $\binom{m}{2}$ pairs that are -1 . Hence, the value of $\text{CQ}(X)$ is determined by the evenness of $\binom{m}{2}$ and the result follows. $\square$

First we calculate the spectrum for the case when n is even.

PROPOSITION A.2. Let $\{a_\alpha \mid \alpha \in \{0, 1\}^n\}$ be the spectral representation of $\text{CQ}(X)$. Assume that n is even, then

$$|a_\alpha| = 2^{-n/2}, \quad \forall \alpha \in \{0, 1\}^n.$$

Proof. The proof is by induction on n . For $n = 2$ we have

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

Assume this is true for n and show that the statement is true for $(n + 2)$. We use the same notation as in § 3, namely, P_{2^n} represents the vector with the values of CQ and A_{2^n} represents the vector of the spectral coefficients of CQ. Using Proposition A.1 it can be shown that $P_{2^{n+2}}$ can be expressed as a function of P_{2^n} :

$$P_{2^{n+2}} = \begin{bmatrix} P_{2^n} \\ \hat{P}_{2^n} \\ \hat{P}_{2^n} \\ -P_{2^n} \end{bmatrix}$$

where

$$\hat{P}_{2^n} = \hat{X}_{2^n} \circ P_{2^n}.$$

$\hat{X}_{2^n}$ is the vector representation of $\text{PARITY}(X) = x_1 x_2 \cdots x_n$ and “ $\circ$ ” is bitwise multiplication.

Hence, by Theorem 3.1

$$\begin{aligned} A_{2^{n+2}} &= \frac{1}{2^{n+2}} H_{2^{n+2}} P_{2^{n+2}} \\ &= \frac{1}{2^{n+2}} \begin{bmatrix} H_{2^n} & H_{2^n} & H_{2^n} & H_{2^n} \\ H_{2^n} & -H_{2^n} & H_{2^n} & -H_{2^n} \\ H_{2^n} & H_{2^n} & -H_{2^n} & -H_{2^n} \\ H_{2^n} & -H_{2^n} & -H_{2^n} & H_{2^n} \end{bmatrix} \begin{bmatrix} P_{2^n} \\ \hat{P}_{2^n} \\ \hat{P}_{2^n} \\ -P_{2^n} \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} \hat{A}_{2^n} \\ A_{2^n} \\ A_{2^n} \\ -\hat{A}_{2^n} \end{bmatrix} \end{aligned}$$

where $\hat{A}_{2^n}$ is the reflection of A_{2^n} . Hence, if the result is true for n it is also true for $n + 2$. $\square$

Example. Using the above recursive description of the spectrum of $CQ(X)$ we can calculate A_{16} from A_4 :

$$A_4^T = \frac{1}{2}(1, 1, 1, -1),$$

and

$$A_{16}^T = \frac{1}{4}(-1, 1, 1, 1, \quad 1, 1, 1, -1, \quad 1, 1, 1, -1, \quad 1, -1, -1, -1).$$

The above is true for n even. For n odd we have the following proposition.

PROPOSITION A.3. *Let $\{a_\alpha \mid \alpha \in \{0, 1\}^n\}$ be the spectral representation of $CQ(X)$. Assume that n is odd, then*

$$|a_\alpha| = 0 \text{ or } 2^{-(n-1)/2} \quad \forall \alpha \in \{0, 1\}^n.$$

The proof is similar to the even case. We use induction on n and can write the recursive description of the spectrum.

Example. Let $n = 3$ then

$$CQ(x_1, x_2, x_3) = \frac{1}{2}(x_1 + x_2 + x_3 - x_1 x_2 x_3).$$

Acknowledgment. I thank Professor J. W. Goodman for his helpful comments.

REFERENCES

- [1] J. BRUCK, *Computing with networks of threshold elements*, Ph.D. Thesis, Stanford University, Stanford, CA, March 1989.
- [2] J. BRUCK AND M. BLAUM, *Neural networks, error-correcting codes and polynomials over the binary n-cube*, IBM Research Report RJ 6003, 1987; IEEE Trans. Inform. Theory, September 1989.
- [3] C. K. CHOW, *On the characterization of threshold functions*, in Proc. Symp. on Switching Circuit Theory and Logical Design, New York, American Institute of Electrical Engineers, 1961, pp. 34–38.
- [4] J. F. DILLON, *Elementary Hadamard difference sets*, in Proc. 6th S-E Conf. Combinatorics, Graph Theory and Computing, (Utilitas Math. Winnipeg, 1975), pp. 237–249.
- [5] A. HAJNAL, W. MAASS, P. PUDLÁK, M. SZEGEDY, AND G. TURÁN, *Threshold circuits of bounded depth*, Proceeding of the 28th IEEE Symposium of Foundations of Computer Science, 1987, pp. 99–110.
- [6] J. J. HOPFIELD, *Neural networks and physical systems with emergent collective computational abilities*, in Proceedings of the National Academy of Sciences USA, Vol. 79, 1982, pp. 2554–2558.
- [7] S. HU, *Threshold Logic*, University of California Press, Berkeley and Los Angeles, California, 1965.
- [8] D. S. JOHNSON, *The NP-completeness column: An ongoing guide*, J. Algorithms, 7 (1986), pp. 289–305.
- [9] R. J. LECHNER, *Harmonic analysis of switching functions*, in Recent Development in Switching Theory, A. Mukhopadhyay, ed., Academic Press, New York, 1971.
- [10] F. J. MACWILLIAMS AND N. J. A. SLOANE, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, New York, 1977.
- [11] S. MUROGA, *Threshold Logic and its Application*, John Wiley, New York, 1971.
- [12] I. PARBERRY AND G. SCHNITGER, *Parallel computation with threshold functions*, in: Structure in Complexity Theory, Lecture Notes in Computer Science, vol. 223, 1986, pp. 272–290.
- [13] A. A. RAZBOROV, *Lower Bounds for the Size of Circuits of Bounded Depth with Basis $\{\wedge, \odot\}$* , Math. Notes, 41 (1987), pp. 333–338.
- [14] J. H. REIF, *On threshold circuits and polynomial computation*, in Proc. 2nd Structure in Complexity Theory Conf., 1987, pp. 118–123.
- [15] O. S. ROTHBAUS, *On “bent” functions*, J. Combin. Theory, Ser. A, 20 (1976), pp. 300–305.
- [16] R. SMOLENSKY, *Algebraic methods in the theory of lower bounds for Boolean circuit complexity*, in 19th Symposium on Theory of Computing, 1987, pp. 77–82.
- [17] A. C. YAO, *Circuits and local computations*, in 21st Symposium on Theory of Computing, 1989.