Explicit Constructions of Depth-2 Majority Circuits for Comparison and Addition

Noga Alon *
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099
and Sackler Faculty of Exact Sciences
Tel Aviv University, Tel Aviv, Israel

Jehoshua Bruck
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

Abstract

All Boolean variables here range over the two element set $\{-1, 1\}$. Given n Boolean variables $x_1, \ldots, x_n$, a non-monotone MAJORITY gate (in the variables x_i) is a Boolean function whose value is the sign of $\sum_{i=1}^{n} \epsilon_i x_i$, where each ϵ_i is either 1 or -1. The $COMPARISON$ function is the Boolean function of two n-bits integers X and Y whose value is -1 iff $X \geq Y$. We construct an explicit sparse polynomial whose sign computes this function. Similar polynomials are constructed for computing all the bits of the summation of the two numbers X and Y. This supplies explicit constructions of depth-2 polynomial-size circuits computing these functions, which use only non-monotone MAJORITY gates. These constructions are optimal in terms of the depth and can be used to obtain the best known explicit constructions of MAJORITY circuits for other functions like the product of two n-bit numbers and the maximum of n n-bit numbers. A crucial ingredient is the construction of a discrete version of a sparse "delta polynomial"—one that has a large absolute value for a single assignment and extremely small absolute values for all other assignments.

*Research supported in part by the Fund for Basic Research administered by the Israel Academy of Sciences

1 Introduction

In this paper we address the problem of computing the $COMPARISON$ and $ADDITION$ functions of two n-bit numbers using circuits of (non-monotone) $MAJORITY$ gates. Throughout this paper, a Boolean function will be defined as $f : \{1, -1\}^n \rightarrow \{1, -1\}$; namely, logical 0 and logical 1 are represented by 1 and -1, respectively.

We first define a few concepts.

1.1 Definitions

Definition 1 A linear threshold function $f(X)$ is a Boolean function such that

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

where

$$F(X) = w_0 + \sum_{i=1}^{n} w_i x_i.$$

The coefficients w_i are called the *weights* of the threshold function. We denote the class of linear threshold functions by LT_1. Notice that the weights can be arbitrary real numbers. It is more interesting to consider the subclass of LT_1, which we call $\hat{LT}_1$ of functions that can be written with "small" weights. Each function

$$f(X) = sgn(w_0 + \sum_{i=1}^{n} w_i x_i)$$

in $\hat{LT}_1$ is characterized by the property that the weights, w_i, are integers bounded by a polynomial in n. Namely, $|w_i| \leq n^c$ for some constant $c > 0$.

Notice that when we say that a Boolean function belongs to a certain complexity class (like LT_1) we actually mean that the family of Boolean functions (as defined for all n) belong to that class.

In this paper we will be mostly interested in linear threshold functions in which the weights are either 1 or -1. Clearly, the elements that compute those functions are non-monotone analogues of usual $MAJORITY$ gates, which we call here , for short, MAJ gates.

Definition 2 A MAJ gate *computes a linear threshold function with weights which are either 1 or -1.*

We will be interested in circuits that consist of MAJ gates. Define MAJ_k to be the class of Boolean functions that can be computed by a polynomial size depth-k circuit of MAJ gates, where the depth of the circuit is the number of gates on the longest path from the input to the output. Note that MAJ_k is equivalent to the class $\hat{LT}_k$, which is the class of Boolean functions that can be computed by a depth-k polynomial size circuits of linear threshold elements with polynomial weights.

After presenting the computational model, let us introduce the functions that we would like to compute.

Let $X = (x_n, x_{n-1}, \ldots, x_1)$ and $Y = (y_n, y_{n-1}, \ldots, y_1)$ be two vectors in $\{1, -1\}^n$. Let a and b be the integers that correspond to X and Y, respectively. Since our convention is that a logical 0 is represented by 1 and a logical 1 is represented by -1 this means that $a = \sum_{i=1}^{n} \frac{1-x_i}{2} 2^{i-1}$ and $b = \sum_{i=1}^{n} \frac{1-y_i}{2} 2^{i-1}$.

Definition 3 The COMPARISON function, $C(X, Y)$, is the Boolean function which is -1 iff $a \geq b$.

Definition 4 Let $c = a + b$ and let $Z = (z_{n+1}, z_n, \ldots, z_1)$ be the binary representation of c. Then the ADDITION function is $ADD(X, Y) = Z$.

1.2 Motivation and known results

Why is it interesting to consider these two functions?

1. It was proved in [15] that the $PRODUCT$ of two n-bit numbers is in MAJ_4. However, the proof is non-constructive. Our construction for the $ADDITION$ function can be used to describe explicitly a depth-4 MAJ circuit for $PRODUCT$. A different way of obtaining such an explicit depth-4 circuit has been recently found, independently, in [10].

2. It was proved in [15] that any LT_1 function (one that can have large weights) is in MAJ_3. This proof is also non-constructive. Our construction for $COMPARISON$ can be used to construct explicitly a depth-3 MAJ circuit for any LT_1 function.

3. The construction for $COMPARISON$ can be used also as a building block for a depth-3 circuit that sorts n n-bits numbers (see [8, 16, 17]).

It is known [8, 17] that $COMPARISON \in MAJ_3$ and $ADDITION \in MAJ_4$. It was also observed in [15] that $COMPARISON \in LT_1$, namely, the $COMPARISON$ function can be computed by a single linear threshold element. However, this linear threshold element has exponentially big weights. As shown in [15] $COMPARISON \notin \hat{LT}_1$. On the other hand, using the results in [7], it was proved in [15] that both $COMPARISON$ and $ADDITION$ are in MAJ_2. The proofs in [15] are existence proofs, while finding explicit constructions was left as an open problem, which we solve here.

1.3 The main contribution

Our main contributions in this paper are explicit constructions of depth-2 polynomial size circuits of MAJ gates that compute the $COMPARISON$ and $ADDITION$ functions. Actually, we show that the $COMPARISON$ and $ADDITION$ functions can be computed as sign functions of explicit sparse polynomials (i.e., polynomials with $n^{O(1)}$ monomials and with $1, -1$-coefficients). In [6] it is proved that any function that can be computed as a sign of such a polynomial is also in MAJ_2. Hence, $COMPARISON$ and $ADDITION$ are in MAJ_2. The key to the construction is the idea that we can construct sparse polynomials that have the property of a "discrete delta function" in the sense that the value of the polynomial is very large for X being the all-1 vector and extremely small for all other values. The construction of these polynomials, that we call delta polynomials, is presented in the next section. In Section 3 we use the delta polynomials as a building block in the construction of depth-2 MAJ circuits for $COMPARISON$ and $ADDITION$. These constructions can be practical, as they may be used in the actual design of small depth circuits for addition and multiplication based on MAJ gates. The final Section 4 contains some concluding remarks, extensions and open problems.

2 Character Sums, Linear Codes and Delta Polynomials

Let $x_1, \ldots, x_n$ be n variables, where each x_i ranges over the two-element set $\{-1, 1\}$. Since $x_i^2 = 1$ for all i, every polynomial in the variables x_i can be represented as a multilinear polynomial. We thus define a *monomial* in the variables x_i to be a product of a subset of the set of variables with a coefficient $+1$ or -1, i.e., a product of the form $\epsilon_j \prod_{i \in A} x_i$, where $\epsilon_j \in \{-1, 1\}$ and $A \subseteq \{1, \ldots, n\}$.

A polynomial in the variables x_i above is called *t-sparse* if it is the sum of at most t monomials. We are mainly interested in the case that t is at most $n^{O(1)}$.

For a vector $\epsilon = \{\epsilon_1, \ldots, \epsilon_n\}$, where $\epsilon_i \in \{-1, 1\}$, and for a positive real c, we call a polynomial $P(x_1, \ldots, x_n)$ a *delta polynomial* for ϵ and c if there are two positive constants a and b satisfying $\frac{a}{b} \geq c$ such that:

(i) $P(\epsilon_1, \ldots, \epsilon_n) = a$ and

(ii) For all $(x_1, \ldots, x_n) \in \{-1, 1\}^n$ which satisfies $(x_1, \ldots, x_n) \neq \epsilon$, $|P(x_1, \ldots, x_n)| \leq b$.

Therefore, P is a delta polynomial for ϵ and c if it attains a positive value at ϵ and the absolute value of P on any other point in $\{-1, 1\}^n$ is smaller by at least a factor of c.

Observe that the polynomial $\prod_{i=1}^{n}(1+x_i)$ is a delta polynomial for $(1, 1, \ldots, 1)$ and any positive c. However, this polynomial is a sum of exponentially many monomials. Our objective in this section is to construct explicitly relatively sparse delta polynomials. A probabilistic construction follows from the techniques presented in [7, 15]; however, their explicit construction seems to be more difficult.

One can easily check that if $P(x_1, \ldots, x_n)$ is a delta polynomial for $(1, 1, \ldots, 1)$ and c then for any vector $(\epsilon_1, \ldots, \epsilon_n) \in \{-1, 1\}^n$, $P(\epsilon_1 x_1, \ldots, \epsilon_n x_n)$ is a delta polynomial for ϵ and c, which has exactly the same number of monomials as P. Thus we may restrict our attention to the construction of sparse delta polynomials for $(1, 1, \ldots, 1)$.

Our construction can be obtained by using linear error-correcting codes over $GF(2)$ that have certain distance properties. We discuss this general approach at the end of the section. Now we present in more detail one such construction which is based on the properties of the quadratic residue character which are proved using Weil's famous theorem known as the Riemann hypothesis for curves over finite fields [18]. These properties have been used before to derive the pseudo-random properties of Paley graphs and quadratic tournaments ([9], see also [5],[2]), and have also been used in the analysis of certain randomized algorithms for various number-theoretic problems, (see [4], [13]). Other constructions can be given based on some of the ideas of [1] and [12] together with the known constructions of expander-graphs, or based on the results of [3]. For our purposes here the quadratic residue construction suffices and we thus describe only this construction in detail, and only comment briefly in the end of the section on the ways to obtain additional similar constructions.

Let q be an odd prime power and let χ be the quadratic residue character defined on the elements of the finite field $GF(q)$ by $\chi(y) = y^{(q-1)/2}$. Equivalently, $\chi(y)$ is 1 if y is a nonzero square, 0 if y is 0 and -1 otherwise. Suppose $q \geq n$ and let $B = \{b_1, \ldots b_n\}$ be an arbitrary subset of cardinality n of $GF(q)$. Consider the following polynomial in the n variables $x_1, \ldots, x_n$;

$$P_B(x_1, \ldots, x_n) = \sum_{y \in GF(q) \setminus B} \prod_{i=1}^{n} \frac{1 + \chi(y - b_i) + x_i(1 - \chi(y - b_i))}{2}.$$

Observe that P_B is a sum of exactly $q - n$ monomials, since for each fixed y in $GF(q) \setminus B$, the quantity $\frac{1+\chi(y-b_i)+x_i(1-\chi(y-b_i))}{2}$ is either 1 or x_i.

Theorem 1 *For every odd prime power q and for every subset B of cardinality n of $GF(q)$ the polynomial P_B defined above satisfies:*
(i) $P_B(1, 1, \ldots, 1) = q - n$, and
(ii) For every $(x_1, \ldots x_n) \in \{-1, 1\}^n$ which is not $(1, 1, \ldots, 1)$,
$|P_B(x_1, \ldots, x_n)| \leq (n-1)q^{1/2}$.
Therefore, P_B is a $(q-n)$-sparse delta polynomial for $(1, 1, \ldots, 1)$ and $c = \frac{q-n}{(n-1)q^{1/2}}$.

Notice that when q is a prime and B is simply the set $\{1, 2, \ldots n\}$ the expression for the polynomial P_B is relatively simple.

In order to prove theorem 1 we need the following known estimate for character sums, due to Weil [18], (see also [14], page 43, Theorem 2C; the lemma stated below is a special case).

Lemma 1 *Let q be an odd prime power and let F be the field $GF(q)$. Let $f(y)$ be a non-constant polynomial over F which decomposes into the product of m distinct linear factors. Then, for the quadratic character χ,*

$$|\sum_{y \in F} \chi(f(y))| \leq (m-1)q^{1/2}.$$

Proof of Theorem 1 Since P_B is a sum of $q - n$ monomials it is $(q - n)$-sparse. Moreover, since the coefficient of every monomial is 1 it follows that $P_B(1, 1, \ldots, 1) = q - n$. Suppose, now, that $(x_1, \ldots, x_n) \neq (1, 1, \ldots, 1)$ is a vector in $\{-1, 1\}^n$. Put $I = \{i : 1 \leq i \leq n, x_i = -1\}$, $J = \{b_i : i \in I\}$. By substituting the values of the variables x_i and by the fact that the quadratic character is multiplicative we obtain:

$$\prod_{i=1}^{n} \frac{1 + \chi(y - b_i) + x_i(1 - \chi(y - b_i))}{2} = \prod_{i \in I} \chi(y - b_i) = \chi(\prod_{i \in I}(y - b_i)).$$

Define $f(y) = \prod_{i \in I}(y - b_i)$. Observe that for the quadratic character χ, $\chi(f(y)) = 0$ whenever y is equal to one of the elements b_i for $i \in I$. Therefore:

$$\begin{aligned} P_B(x_1, \ldots, x_n) &= \sum_{y \in GF(q) \setminus B} \prod_{i=1}^{n} \frac{1 + \chi(y - b_i) + x_i(1 - \chi(y - b_i))}{2} \\ &= \sum_{y \in GF(q) \setminus B} \chi(f(y)) \\ &= \sum_{y \in GF(q) \setminus (B \setminus J)} \chi(f(y)) \\ &= \sum_{y \in GF(q)} \chi(f(y)) - \sum_{y \in B \setminus J} \chi(f(y)). \end{aligned}$$

Observe that since I is not empty and since the elements b_i are distinct we can apply Lemma 1 to $f(y)$ and obtain, by the triangle inequality:

$$|P_B(x_1, \ldots, x_n)| \leq |\sum_{y \in GF(q)} \chi(f(y))| + |\sum_{y \in B \setminus J} \chi(f(y))| \leq (|I| - 1)q^{1/2} + n - |I|.$$

The quantity $(|I| - 1)q^{1/2} + n - |I|$ is clearly an increasing function of $|I|$, and since $|I| \leq n$ this quantity is at most $(n - 1)q^{1/2}$. This completes the proof. □

Linear Codes and Delta Polynomials

The argument above can be modified to obtain a similar construction of a delta polynomial from any linear error-correcting code over $GF(2)$ with length which is polynomial in the dimension and with the property that the Hamming weight of any non-zero codeword is sufficiently close to half the length. Here is a sketch of the argument. Let $A = (a_{ij})_{1 \leq i \leq n, 1 \leq j \leq t}$ be the generating $0, 1$-matrix of a linear error-correcting code of length t and dimension n, and suppose that the Hamming weight of each non-zero codeword is between $(1 - \epsilon)\frac{t}{2}$ and $(1 + \epsilon)\frac{t}{2}$. Let $P_A = P_A(x_1, \ldots, x_n)$ be the polynomial defined by

$$P_A(x_1, \ldots, x_n) = \sum_{j=1}^{t} \prod_{i; a_{ij} = 1} x_i.$$

Clearly $P_A(1, \ldots, 1) = t$, and it is not difficult to check that for every $(x_1, \ldots, x_n) \in \{-1, 1\}^n$ which is not $(1, \ldots, 1)$,

$$|P_A(x_1, \ldots, x_n)| \leq \epsilon t,$$

since $P_A(x_1, \ldots, x_n)$ is precisely the difference between the number of 0's and the number of 1's in the codeword defined by the sum (in $GF(2)$) of all rows i of A such that $x_i = -1$.

The polynomial P_B described in Theorem 1 is a special case of the above construction, which corresponds to a linear error-correcting code of dimension n and length $q-n$, in which the Hamming weight of any non-zero codeword is between $\frac{q-n}{2} - \frac{(n-1)q^{1/2}}{2}$ and $\frac{q-n}{2} + \frac{(n-1)q^{1/2}}{2}$. Three additional simple constructions of linear codes whose parameters are asymptotically comparable to this one (up to some polylogarithmic factors) are given in [3], and any of these can be used for constructing a sparse delta polynomial in the manner described above.

3 The Constructions

In this section we prove that the COMPARISON and ADDITION functions can be computed as sign functions of (explicit) sparse polynomials. From a (simple) result in [6] this implies that both functions can be computed by an explicit depth-2, polynomial size circuit of MAJ elements. Both constructions apply the delta polynomials described in the previous section.

First we note that the following is an equivalent description of the COMPARISON function: For $X, Y \in \{1, -1\}^n$, $C(X, Y) = -1$ iff either $X = Y$ or there exists an i, $1 \leq i \leq n$ such that $x_i = -1$ and $y_i = 1$ and also $x_j = y_j$ for all j, such that $i < j \leq n$. The following theorem gives the construction for COMPARISON.

Theorem 2 *Let $m_k(X, Y) = P(x_n y_n, x_{n-1} y_{n-1}, \cdots, x_{k+1} y_{k+1})$ and let $m_n(X, Y) = q - n$, where $P(\cdot)$ is the delta polynomial described in Theorem 1 with $q \geq n^4$ an odd prime power. Define*

$$\hat{C}(X, Y) = m_0(X, Y) + \sum_{i=1}^{n} (y_i - x_i) m_i(X, Y).$$

Then $C(X, Y) = sgn(-\hat{C}(X, Y))$.

Proof: We consider the two cases ($X \geq Y$ or $X < Y$) and prove that $C(X, Y) = sgn(-\hat{C}(X, Y))$ in both cases.

First assume that X is strictly greater than Y. Hence, there is an i such that $x_i = -1$ and $y_i = 1$ and also $x_j = y_j$ for all j, $i < j \leq n$. Hence, $(y_i - x_i) m_i \geq 2(q - n)$ and

$$\hat{C}(X, Y) \geq 2(q - n) - 2n(n - 1)\sqrt{q} > 0.$$

If $X = Y$ then clearly $\hat{C}(X, Y) = q - n > 0$. Hence, if $X \geq Y$ then $-1 = C(X, Y) = sgn(-\hat{C}(X, Y))$.

Similarly, if $X < Y$ then $\hat{C}(X,Y) \leq -2(q-n) + 2n(n-1)\sqrt{q} < 0$. Hence, $C(X,Y) = sgn(-\hat{C}(X,Y))$ in this case as well, completing the proof. □

Next we consider the $ADDITION$ function. In order to compute the bits of the sum of the two n-bit numbers X and Y as signs of sparse polynomials it suffices to construct a sparse polynomial for each of the carry bits. This is because the i^{th} bit in the result of the addition is $x_i y_i c_i$ where c_i is the corresponding carry bit. If we can compute c_i as a sign of a sparse polynomial, say $c_i = sgn(p(X,Y))$, then we can also compute $x_i y_i c_i = sgn(x_i y_i p(X,Y))$ as a sign function of a sparse polynomial. From now on we will concentrate, without loss of generality, on proving that the carry to the last bit (*i.e.* c_n) can be computed as a sign of a sparse polynomial. We denote the carry function to the last bit as $CAR(X,Y)$ and prove that it can be computed as a sign function of a sparse polynomial.

Theorem 3 *Let* $l_k(X,Y) = P(-x_{n-1}y_{n-1}, -x_{n-2}y_{n-2}, \cdots, -x_{k+1}y_{k+1})$ *and let* $l_{n-1}(X,Y) = q-n$, *where* $P(\cdot)$ *is the delta polynomial described in Theorem 1 with* $q \geq 4n^4$ *an odd prime power. Let* $f_1(w_1,w_2) = (1 - w_1 - w_2 + w_1 w_2)$. *Let*

$$\widehat{CAR}(X,Y) = \sum_{i=1}^{n-1} f_1(x_i,y_i) l_i(X,Y).$$

Then $CAR(X,Y) = sgn(2q - \widehat{CAR}(X,Y))$.

Proof: Note that $f_1(-1,-1) = 4$ and $f_1(1,1) = f_1(1,-1) = f_1(-1,1) = 0$.

First assume that there is carry to bit n in the addition of X and Y, namely that $CAR(X,Y) = -1$. In such a case we have carry generation and propagation. Namely, there is an $i < n$ such that $x_i = -1$ and $y_i = -1$ in which the carry is generated, and in addition $x_j \neq y_j$ for all j, $i < j < n$ (so that the carry will propagate). Note that the carry will propagate also in the case $x_j = y_j = -1$. However, without loss of generality we can consider the leftmost place i in which the carry was generated. Since $f_1(x_i,y_i) l_i \geq 4(q-n)$ then, by the properties of the delta polynomials,

$$\widehat{CAR}(X,Y) \geq 4(q-n) - 4(n-2)(n-1)\sqrt{q} > 2q$$

Hence, if there is carry then $CAR(X,Y) = sgn(2q - \widehat{CAR}(X,Y))$.

Next we consider the case in which there is no carry. The reason for not having a carry is that for each index i either there is no carry generation (and then $f_1(x_i, y_i) = 0$) or there is a carry generation but there is no carry propagation. In the latter case $|l_i(X,Y)| \leq (n-1)\sqrt{q}$. Hence, for this case

$$\widehat{CAR}(X,Y) \leq 4(n-1)^2\sqrt{q} < 2q.$$

Hence, if there is no carry then $CAR(X,Y) = sgn(2q - \widehat{CAR}(X,Y))$, completing the proof. □

4 Concluding remarks and extensions

- A family of vectors F in $\{-1,1\}^n$ is a *linear subspace* if for every $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ in F, the vector $x * y = (x_1 y_1, \ldots, x_n y_n)$ is also in F. (This is the usual definition of a subspace together with our mapping that replaces 0 and 1 by 1 and -1 respectively.) Similarly, A is an *affine subspace* if it is the set of all vectors of the form $x * y$ for some fixed vector x as y ranges over all vectors of a linear subspace. Generalizing the notion of a delta polynomial we can construct, for every affine subspace a sparse polynomial whose value on the members of the subspace is much larger than whose value on vectors outside the subspace. (The delta polynomials correspond to the case that the subspace contains only one point). This enables us, among other things, to express explicitly every function which is the characteristic function of a union of polynomially many affine subspaces as a sign of a sparse polynomial. In order to construct the generalized delta polynomials we observe first that it suffices to construct those for linear subspaces. For every linear subspace of co-dimension k in $\{-1,1\}^n$ there are k monomials in $x_1, \ldots, x_n$ such that a vector $(x_1, \ldots, x_n)$ is in the subspace iff all these monomials evaluated in the coordinates of the above vector are 1. We can thus simply substitute these monomials in the delta polynomial of Section 2 and obtain the desired generalized sparse polynomial. We omit the details.

- The delta polynomials can be used to construct a sparse polynomial for the MAXIMUM function, that gets as input n integers (n bits each) and outputs -1 iff the first integer is the maximum. The construction for MAXIMUM is a simple generalization of the one for the

COMPARISON function.

- By a non-constructive argument one can prove that there is a $(q-n)$-sparse polynomial $P(x_1,\ldots,x_n)$ satisfying somewhat stronger properties than those given by Theorem 1; namely

  (i) $P(1,1,\ldots,1) = q-n$, and

  (ii) For every $(x_1,\ldots x_n) \in \{-1,1\}^n$ which is not $(1,1,\ldots,1)$,

  $|P(x_1,\ldots,x_n)| \leq O(n^{1/2}q^{1/2})$.

  It would be interesting to find an explicit construction of such polynomials. (This will supply, of course, smaller depth-2 MAJ-circuits for the functions considered in Section 3).

References

[1] M. Ajtai, J. Komlós and E. Szemerédi, *Deterministic simulation in LOGSPACE*, Proc. 19^{th} Annual ACM STOC, ACM Press, New York, 1987, 132-140.

[2] N. Alon, *Tools from higher algebra*, to appear in: Handbook of Combinatorics, R. L. Graham, M. Grotschel and L. Lovász, eds., North Holland.

[3] N. Alon, O. Goldreich, J. Hastad and R. Peralta, *Simple constructions of almost k-wise independent random variables*, Proc. 31^{st} IEEE FOCS, St. Louis, Missouri, IEEE (1990), 544-553. Also: Random Structures and Algorithms 3 (1992), 289-304.

[4] E. Bach, *Realistic analysis of some randomized algorithms*, Proc. 19^{th} Annual ACM STOC, ACM Press, New York, 1987, 453-461.

[5] B. Bollobás, *Random Graphs*, Academic Press, London 1985.

[6] J. Bruck, *Harmonic analysis of polynomial threshold functions*, SIAM J. on Disc. Math. 3 (1990), 168-177.

[7] J. Bruck and R. Smolensky, *Polynomial threshold functions, AC^0 functions and spectral norms*, SIAM J. on Comput. 21 (1992), 33-42.

[8] A. K. Chandra, L. Stockmeyer and U. Vishkin, *Constant depth reducibility*, SIAM J. on Comput. 13 (1984), 423-439.

[9] R. L. Graham and J. H. Spencer, *A constructive solution to a tournament problem*, Canad. Math. Bull. 14 (1971), 45-48.

[10] T. Hofmeister, T. Hohberg and S. Kohling, *Some notes on threshold circuits and multiplication in depth 4*, Information Processing Letters, 39 (year?), 219-225.

[11] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy and G. Turan, *Threshold circuits of bounded depth*, Proc. 28^{th} IEEE FOCS, 1987, 99–110.

[12] J. Naor and M. Naor, *Small-bias probability spaces: efficient constructions and applications*, Proc. 22^{nd} Annual ACM STOC, 1990, ACM Press, 213-223.

[13] R. Peralta, *On the randomness complexity of algorithms*, University of Wisconsin, Milwaukee, CS Research Report TR 90-1.

[14] W. M. Schmidt, *Equations Over Finite Fields, An Elementary Approach*, Springer Lecture Notes in Mathematics, vol. 536, Springer Verlag, Berlin 1976.

[15] K. Y. Siu and J. Bruck, *On the power of threshold circuits with small weights*, SIAM J. on Disc. Math. 4 (1991), 423-435.

[16] K. Y. Siu, J. Bruck, T. Kailath and T. Hofmeister, *Depth efficient Neural Networks for division and related problems*, IBM Research Report, RJ 7946, 1991.

[17] I. Wegener, *The Complexity of Boolean Functions*, John Wiley & Sons, page 322, 1987,

[18] A. Weil, *Sur les courbes algébriques et les variéstés qui sèn déduisent*, Actualités Sci. Ind. No. 1041 (1948).