EVALUATING SPECTRAL NORMS FOR CONSTANT DEPTH CIRCUITS WITH SYMMETRIC GATES

Meera Sitharam

Abstract.

- The Fourier spectrum and its norms are given as explicit arithmetic expressions and evaluated, for Boolean functions computed by classes of constant depth, read-once circuits consisting of an arbitrary set of symmetric gates. Previous results of this nature estimate the spectral L_1 norm of functions computed by certain types of decision trees [20] [7], and in some cases, give randomized *procedures* that evaluate the spectrum by clever rounding [20]. One corollary of our results provides a large class of AC^0 functions whose spectral L_1 norm is exponential, thus generalizing the single example of such a function given in [9]. This shows that almost every read-once AC^0 function does not belong in the class PL_1 of functions with polynomially bounded spectral norms.

- Implications of our results and technique are discussed, for estimating the spectral norms of *any* function in a constant depth circuit class, using the coding theoretic concept of weight distributions. Evaluating the spectral norms for any such function reduces to estimating certain non-trivial weight distributions of simple, linear codes.

Key words. Circuit complexity; Lower bounds; Fourier transforms.

Subject classifications. 68Q15, 68Q99.

1. Introduction

Complexity bounds for classes of constant depth circuits consisting of specific sets of gates have been the subject of extensive study. The gates are often chosen to be a particular set of symmetric Boolean functions that form a complete basis. For example the set $\{\wedge, \vee, \neg\}$ yielding AC^0 is studied in ([13], [30], [36], [17], [23], [12], survey [8]); the sets $\{\wedge, mod_p\}$, $\{\vee, \neg, mod_p\}$, $\{\vee, \neg, mod_q\}$, for p prime, q composite are studied in ([2], [31], [33]), [4]. In some cases, the chosen gates are non-symmetric and less constrained, for example, threshold gates, the most powerful of which is $sign(p(x))$, for a sparse multilinear real polynomial p ([18], [1], [37], [6], [38], [9], [15], [29], [34], [21], [22], [?], [3]).

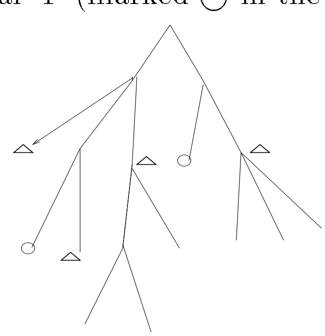
Spectral analysis is one of the techniques employed recently to study some of these classes, and sometimes Boolean functions in general. The Fourier spectrum of a Boolean function f is so named since it is obtained by viewing f as a function from the group $\mathbb{Z}_2^n$ to the field of complex numbers. However, there are several natural transformations - one of which is the Hadamard transform - of a Boolean function that result in the Fourier spectrum. Another example: if a Boolean function f is viewed as a function from $\{1,-1\}^n \rightarrow \{0,1\}$, the Fourier coefficients of f are exactly coefficients of the unique multilinear polynomial $\tilde{f}$ over $\mathbb{R}^n$ that represents f, i.e, interpolates the values of f on its domain. This versatility of the Fourier spectrum of the Boolean function enables its wide applicability, and furthermore, transports some of the classical spectral analytic techniques to the study of Boolean function complexity.

Estimating spectral norms of a Boolean function f ($L_\infty(\hat{f})$, $L_1(\hat{f})$, $L_2(\hat{f})$, etc.) is essential for answering the basic questions of approximability and non-approximability of Boolean functions. Therefore such estimations have predictably found several applications. For example, the L_1 and the inverse of the L_∞ norm of $\hat{f}$ respectively provide upper and lower bounds on the number of terms in a polynomial whose sign represents f, and is used to determine the threshold complexity of f [9]. Estimations of the size of the support of $\hat{f}$ has also found applications. This corresponds to the degree and sparsity of the multilinear polynomial $\tilde{f}$ that interpolates f on $\{1,-1\}^n$. The sparsity of the function that approximates $\hat{f}$ with respect to the L_1 or L_2 norms is related to the time complexity of learning f as shown in and [20], [23], and [32]. In addition, the multiparty communication complexity of Boolean functions can also be bounded via their spectral L_1 norm [16]. (See [9] for a brief survey of some of these applications). Although some of these spectral analytic studies have concerned relatively general Boolean functions [19], [20], [7], [27], [28], [12], [4], many others have been specific to Boolean functions computed either by (constant depth) circuits consisting of very specific sets of (symmetric) gates [23], [5], [3], or by depth 1 or depth 2 circuits consisting of a less constrained set of gates, [6], [9], [34] [?].

We consider Boolean functions computed by constant depth circuits consisting of arbitrary sets of symmetric gates. However, we restrict ourselves primarily to functions computed by read-once circuits, and give explicit arithmetic expressions that evaluate to the spectral coefficients of such functions, and thereby also to their norms. In addition, we directly evaluate the spectral L_1 norm, without evaluating the individual spectral coefficients. Previous results of this nature give upper bounds on the spectral L_1 norm of functions computed by certain types of decision trees [20] [7], and in some cases, give randomized *procedures* that evaluate the spectrum by clever rounding [20]. One corollary of our results provides a large class of read-once AC^0 functions whose spectral L_1 norm is exponential, thus generalizing the single example of such a function given in [9]. We show that almost every read-once AC^0 function is not contained in the class PL_1 of functions with polynomially bounded spectral L_1 norms. This shows the limitations of methods that bound complexity in terms of the spectral L_1 norm, such as the method in [20] which establishes that the (randomized) complexity of learning Boolean functions is polynomial in their spectral L_1 norms.

While functions computed by read-once circuits are simple and easy to deal with in a certain sense, any non-read-once Boolean functions f of n variables - computed by any constant depth, polynomial size circuit of a set of symmetric gates - can be expressed in terms of read-once functions f_r of $m \leq poly(n)$ variables, in the same class, using projections. It therefore follows that a Fourier coefficient of f can be expressed as a sum of our explicitly evaluated Fourier coefficients of f_r over a translate of a simple, linear subspace of $\mathbb{F}_2^m$. This observation, the properties of f_r that follow from our evaluation, and an application of the coding theory concept of weight distributions, altogether imply a new method for the estimation of the spectral norms of f. The usefulness of this method is yet to be investigated and depends on whether for simple, linear codes, the weight distribution with respect to certain non-trivial weights can be estimated with reasonable accuracy.

Our technique for evaluating spectral norms is based on the simple idea that for a function f computed by *any* read-once circuit C the Fourier coefficient $\hat{f}(x)$ can be expressed in terms of the expected values of the functions computed by the maximal subcircuits of C (marked $\triangle$ below) whose inputs do not include any of the bits x_i that equal '1' (marked $\bigcirc$ in the figure below).



For the case of homogeneous circuits with symmetric gates, this idea yields a technique for the explicit expression of $\hat{f}(x)$ in terms of d natural parameters, which we call the *type* parameters of x, where d is the depth of the homogeneous, read-once circuit computing f. By the same idea, $L_1(\hat{f})$ can also be expressed recursively using the expected values of the functions computed by the subircuits of C, thereby permitting the evaluation of $L_1(\hat{f})$ by evaluating at most as many Fourier coefficients of f as the number of subcircuits of C.

The paper is organized as follows. Section 2 clarifies notational conventions, and provides some basic background. Section 3 defines the *type* parameters, and gives explicit arithmetic expressions, in terms of the *type* parameters, for evaluating the spectra and their L_1 norm, for functions computed by homogeneous read-once circuits with arbitrary sets of symmetric gates. The specific example of read-once AC^0 functions is worked through, their spectral L_1 norm is evaluated, and shown to be exponentially large. Section 4 gives a method for the estimation of the spectral norms of arbitrary functions computed by constant depth circuits of symmetric gates, by observing some properties of *type* parameters, and applying the concept of weight distributions.

2. Background and conventions

Unless otherwise specified, all n-tuples are elements of either $\mathbb{R}^n$ or the finite vector space $\mathbb{F}_2^n$. The number of non-zero entries in x is denoted $|x|$, the n-tuple $(a, \ldots, a)$ is denoted (a^n), and for any n, the n-tuple of all zeroes is denoted $\vec{0}$. When $x \in \mathbb{F}_2^n$, x is identified with the set of co-ordinates $1 \leq i \leq n$ where $x_i = 1$. Thus, given vectors x and y, we will refer to the vectors $x \cup y$, $x \cap y$, $x \setminus y$, $\bar{x}$ (for the bitwise complement of x), and expressions such as $i \in x$ (meaning $x_i = 1$). The inner product $< x, y >$ for $x, y \in \mathbb{F}_2^n$ is '1' if the parity of $|x \cap y|$ is odd.

Boolean functions of n variables map from $\mathbb{F}_2^n$ to $\{0, 1\}$ or $\{1, -1\}$ to facilitate the use of the Fourier transform (see [10]) for functions from the group $\mathbb{Z}_2^n$ to the field of complex numbers. The Fourier transform of f is denoted $\hat{f}$ and is given by

$$\hat{f}(x) = 1/2^n \sum_{u \in \mathbf{F}_2^n} f(u)(-1)^{<x,u>};$$

thereby $f(x)$ can be written as $\sum_{u \in \mathbb{F}_2^n} \hat{f}(u)(-1)^{<x,u>}$. The domains of Boolean functions f and their transforms $\hat{f}$ are always considered to be $\mathbb{F}_2^n$. All other functions of n variables are multilinear polynomials that map from $\mathbb{R}^n$ to $\mathbb{R}$. For a Boolean function f, $\tilde{f}$ will be used to denote the unique multilinear polynomial over $\mathbb{R}^n$ that interpolates f at $\{0,1\}^n$ or at $\{1,-1\}^n$, depending on what the real domain of f, denoted $\mathbb{R}domain(f)$, and the range of f are chosen to be. In other words, a real representation of $\mathbb{F}_2^n$ is chosen, either with 0 and 1 (in $\mathbb{F}_2^n$) mapping to the real values 0 and 1 or to 1 and -1 respectively. While a Boolean function f and its Fourier transform are independent of this choice of $\mathbb{R}domain(f)$, the polynomial $\tilde{f}$ does depend on this choice.

Note that when $\mathbb{R}domain(f) = \{1,-1\}^n$, and $range(f) = \{1,-1\}$ then for $x \in \{1,-1\}^n$, $f(x) = \tilde{f}(x) = \sum_{y\in\mathbb{F}_2^n} \hat{f}(y) \prod_{i\in y} x_i$. In other words, the coefficient of $\prod_{i\in y} x_i$ in the multilinear polynomial $\tilde{f}$ over $\mathbb{R}^n$ that represents f on the $\mathbb{R}domain$ $\{1,-1\}^n$ is nothing but the y^{th} Fourier coefficient of f. Notice that given $\tilde{f}$ that represents f on the $\mathbb{R}domain$ $\{0,1\}^n$, the Fourier coefficients of f can thus be obtained by applying the change of variable $x_i \to \frac{1-x_i}{2}$, and $\bar{x}_i = (1-x_i) \to \frac{1+x_i}{2}$, to $\tilde{f}$; and finding the coefficients of the resulting polynomial in standard power form. Finally, for $y \in \mathbb{F}_2^n$ we denote the y^{th} partial derivative of order $|y|$ over $\mathbb{R}^n$, i.e, $\prod_{i\in y} D_{x_i}$, by D_y. Finally, the norms used are the following: $L_1(f) =_{def} \sum_{x\in\mathbb{F}_2^n} |f(x)|$; $L_2(f) =_{def} \sum_{x\in\mathbb{F}_2^n} f^2(x)$; and $L_\infty(f) =_{def} max_{x\in\mathbb{F}_2^n} |f(x)|$.

The following are basic properties of the Fourier spectra of Boolean functions.

FACT 2.1. *For functions f and g over $\mathbb{F}_2^n$ the following hold.*

(i) Parseval's identity:

$$(1/2^n)L_2(f) = (1/2^n) \sum_{x\in\mathbb{F}_2^n} f^2(x) \;=\; \sum_{x\in\mathbb{F}_2^n} \hat{f}^2(x) =_{def} L_2(\hat{f}).$$

Notice that if $range(f) = \{0,1\}$, then

$$L_1(f) =_{def} \sum_{x\in\mathbb{F}_2^n} |f(x)| = \sum_{x\in\mathbb{F}_2^n} f^2(x) =_{def} L_2(f).$$

Thus, for Boolean f, bounds on the L_2 norm of $\hat{f}$ provide bounds on the L_1 norm of f, and furthermore, the L_1 norm of $\hat{f}$ provides an upper bound on the L_2 norm of $\hat{f}$, since $|\hat{f}(x)| \leq 1$ for all x. In addition, the L_1 norm of $\hat{f}$ provides a lower bound on the size of the support of $\hat{f}$, and an upper bound on the sparsity of the polynomial approximating $\tilde{f}$ when $\mathbb{R}domain = \{1,-1\}^n$. Moreover, the L_1 norm of $\hat{f}$ gives a lower bound on the L_∞ norm of $\hat{f}$. These facts are crucial in the development of several of the learning algorithms [23], [20], [12], [7], [32], and also used in some of the results about threshold circuits, for example [6], [9].

(ii) *The value of the transform at* $\vec{0}$ *is the expected value of the function:*

$$\hat{f}(\vec{0}) = (1/2^n) \sum_u f(u).$$

(iii) *If* $range(f) = \{0,1\}$, *then* $\hat{f}(\vec{0}) = (1/2^n)|support(f)|$, *and if* $range(f) = \{1,-1\}$, *and* $\mathbb{R}\,domain(f) = \{1,-1\}^n$, *then* $\hat{f}(\vec{0}) = \tilde{f}(\vec{0})$.

(iv) *If* $range(f) = \{0,1\}$, $range(g) = \{1,-1\}$, *and* $f = (1-g)/2$, *then*

$$\hat{f}(\vec{0}) = (1/2)\ (1 - \hat{g}(\vec{0})),$$

and

$$\forall u : |u| > 0 \quad \hat{f}(u) = -(1/2)\ \hat{g}(u).$$

The following fact gives a simple property of the multilinear polynomial that represents a Boolean function.

FACT 2.2. *Let* f *be a Boolean function of* n *variables.*

(i) *For any* $y \in \mathbb{F}_2^n$, *and* $y \neq \vec{0}$, *the multilinear polynomial* $\tilde{f}$ *over* $\mathbb{R}^n$ *can be expressed as follows:*

$$\tilde{f}(x) = f'_y(x \setminus y) + \tilde{f}_y(x \setminus y) \prod_{i \in y} x_i,$$

where f'_y *and* $\tilde{f}_y$ *are unique multilinear polynomials over* $\mathbb{R}^{|\bar{y}|}$. *Moreover,* $\tilde{f}_y = D_y \tilde{f}$, *where* D_y *denotes the* y^{th} *partial derivative (see the first paragraph of this section).*

(ii) *If* $\mathbb{R}\,domain(f) = \{0,1\}^n$, *then* $\hat{f}(y) = (1/2^{|y|})\tilde{f}_y(\frac{1}{2}^{n-|y|})$,
and if $\mathbb{R}\,domain(f) = \{1,-1\}^n$, *then* $\hat{f}(y) = \tilde{f}_y(\vec{0})$.

PROOF. The proof of (i) is straightforward using basic properties of multilinear polynomials. (Note that $\tilde{f}_{\vec{0}} = \tilde{f}$). For (ii), if $\mathbb{R}\,domain(f) = \{1,-1\}^n$, then by the definition of $\hat{f}$, $\hat{f}(y)$ is simply the coefficient of $\prod_{i \in y} x_i$ in the standard power form of $\tilde{f}$, which is clearly $\tilde{f}_y(\vec{0})$. If $\mathbb{R}\,domain(f) = \{0,1\}^n$, then $\hat{f}(y)$ is given by the coefficient of $\prod_{i \in y} x_i$ in the standard power form of $\tilde{f}$, *after* the change of variables: $x_i \to \frac{1-x_i}{2}$, and $(1-x_i) \to \frac{1+x_i}{2}$, which is clearly $(1/2^{|y|})\tilde{f}_y(\frac{1}{2}^{n-|y|})$. $\square$

3. Spectral norms for functions computed by read once circuits.

The main result of this section, Theorem 3.5, gives explicit expressions for the spectral values and their L_1 norm for functions f computed by (resp. expressible as) constant depth, homogeneous, read-once circuits (resp. formulae) of an arbitrary pair of symmetric gates. These expressions are given in terms of d natural parameters, called the *type* parameters, of the argument to f. We begin by setting up the machinery to state and prove the main theorem. The first theorem we prove is a direct application of Fact 2.2 to functions computed by any read-once circuit C consisting of possibly nonsymmetric gates. The theorem gives recursive expressions for the spectrum of f and its L_1 norm, in terms of the spectra of the functions computed by the subcircuits of C.

THEOREM 3.1. *Let f be a Boolean function computable by a read-once circuit. Without loss of generality, let f be defined as*

$$f(x) = g\Big(h_1(1(x)), \ldots, h_k(k(x))\Big),$$

where the tuples of arguments, $i(x)$ to the h_i's form a partition of the arguments to f, i.e, $i(x) \cap j(x) = \vec{0}$ when $i \neq j$, and $\bigcup_{i=1}^{k} i(1^n) = (1^n)$. The functions g and h_i are computed by read-once Boolean circuits over $\mathbb{F}_2^k$ and $\mathbb{F}_2^{|i(1^n)|}$ respectively. For any $y \in \mathbb{F}_2^k$, let $\tilde{g}$ be expressed as in Fact 2.2, as follows:

$$\tilde{g}(z) = g'_y(z \setminus y) + \tilde{g}_y(z \setminus y) \prod_{i \in y} z_i.$$

Furthermore, for any $w \in \mathbb{F}_2^n$, let $y_w \in \mathbb{F}_2^k$ be the characteristic vector of the set $\{i : i(w) \neq \vec{0}\}$. Then

(i)

$$\hat{f}(w) = \Big(\prod_{i \in y_w} \hat{h}_i(i(w))\Big) \;\; \tilde{g}_{y_w}\Big(\hat{h}_1(\vec{0}), \ldots, \hat{h}_k(\vec{0})) \;\setminus\; y_w\Big);$$

and

(ii)

$$L_1(\hat{f}) = \sum_{y \in \mathbb{F}_2^k} \Big(\;\; \prod_{i \in y} \Big(L_1(\hat{h}_i) - \hat{h}_i(\vec{0})\Big) \;\; \Big) |\tilde{g}_y\Big((\hat{h}_1(\vec{0}), \ldots, \hat{h}_k(\vec{0})) \;\setminus\; y\Big)|.$$

PROOF. By applying the chain rule and noticing that the h_i's are representable by multilinear polynomials $\tilde{h}_i$ with disjoint sets of variables, it follows that

$$D_w \tilde{f} = \prod_{i \in y_w} \Big(D_{i(w)} \tilde{h}_i D_{y_w} \tilde{g} \Big).$$

Now, if $\mathbb{R} domain(f) = \{0,1\}^n$, it follows from Fact 2.2 that

$$\hat{f}(w) = (1/2^{|w|}) \tilde{f}_w (\frac{1}{2}^{n-|w|}) = (1/2^{|w|}) D_w \tilde{f} (\frac{1}{2}^{n-|w|})$$

and substituting the above expression for $D_w \tilde{f}$, we get

$$\hat{f}(w) = \Big(\prod_{i \in y_w} (1/2^{|i(w)|}) D_{i(w)} \tilde{h}_i (\frac{1}{2}^{|i(1^n)|-|i(w)|}) \Big) *$$

$$\tilde{g}_{y_w} \Big((\tilde{h}_1 (\frac{1}{2}^{|1(1^n)|}), \ldots, \tilde{h}_n (\frac{1}{2}^{|n(1^n)|})) \quad \setminus y_w \Big)$$

and applying Fact 2.2 again to the $\tilde{h}_i$'s, the above quantity

$$= \prod_{i \in y_w} \Big(\hat{h}_i (i(w)) \Big) \quad \tilde{g}_{y_w} \Big((\hat{h}_1(\vec{0}), \ldots, \hat{h}_k(\vec{0})) \ \setminus \ y_w \Big).$$

On the other hand, if $\mathbb{R} domain(f) = \{1,-1\}^n$, it follows from Fact 2.2 that

$$\hat{f}(w) = \tilde{f}_w(\vec{0}) = D_w \tilde{f}(\vec{0}),$$

and substituting the above expression for $D_w \tilde{f}$, we get

$$\hat{f}(w) = \Big(\prod_{i \in y_w} D_{i(w)} \tilde{h}_i(\vec{0}) \Big) \quad \tilde{g}_{y_w} \Big((\tilde{h}_1(\vec{0}), \ldots \tilde{h}_n(\vec{0})) \quad \setminus y_w \Big),$$

and just as in the earlier case, applying Fact 2.2 to the $\tilde{h}_i$'s, the above quantity becomes

$$= \Big(\prod_{i \in y_w} \hat{h}_i (i(w)) \Big) \quad \tilde{g}_{y_w} \Big((\hat{h}_1(\vec{0}), \ldots, \hat{h}_k(\vec{0})) \ \setminus \ y_w \Big).$$

To show (ii), we observe

$$L_1(\hat{f}) = \sum_{w \in \mathbb{F}_2^n} |\tilde{g}_{y_w} \Big((\hat{h}_1(\vec{0}), \ldots, \hat{h}_k(\vec{0})) \ \setminus \ y_w \Big)| \quad \Big(| \prod_{i \in y_w} \hat{h}_i (i(w)) | \Big)$$

which can be written as

$$\sum_{y\in \mathbb{F}_2^k} |\tilde{g}_y\Big((\hat{h}_1(\vec{0}),\ldots,\hat{h}_k(\vec{0}))\ \setminus\ y\Big)|\Big(\sum_{\substack{\{w:i(w)\neq\vec{0}\\ \iff i\in y\}}} |\prod_{i\in y}\ \hat{h}_i(i(w))|\ \Big).$$

But

$$\sum_{\substack{\{w:i(w)\neq\vec{0}\\ \iff i\in y\}}} |\prod_{i\in y}\ \hat{h}_i(i(w))|$$

can be rewritten as

$$\prod_{i\in y}\ \sum_{\substack{\{i(w):w\in\mathbb{F}_2^n\\ \&\ i(w)\neq\vec{0}\}}} |\hat{h}_i(i(w))|$$

from which the result follows since

$$\sum_{\substack{\{i(w):w\in\mathbb{F}_2^n\\ \&\ i(w)\neq\vec{0}\}}} |\ \hat{h}_i(i(w))\ | = L_1(\hat{h}_i) - \hat{h}_i(\vec{0}).$$

□

Next we give a uniform definition of symmetric Boolean functions.

DEFINITION 3.2. *For $u\subseteq\{0,1,\ldots,k\}$ the symmetric Boolean function s_u is defined as $s_u(x)=1$ if and only if $|x|\in u$. When $\mathbb{R}domain(s_u)=\{0,1\}^k$ and $range(s_u)=\{0,1\}$, for example, it is clear that*

$$\tilde{s}_u(x)=\sum_{i\in u}\sum_{\substack{y\in\mathbb{F}_2^k\\ |y|=i}}\Big(\prod_{j\in y}x_j\Big)\Big(\prod_{j\notin y}(1-x_j)\Big).$$

The following is a direct application of Fact 2.2 to symmetric Boolean functions: for any $y\in\mathbb{F}_2^k$, with $|y|=j$, the multilinear polynomial $\tilde{s}_u$ can be expressed as

$$\tilde{s}_u(x)=s'_{u,j}(x\setminus y)+\tilde{s}_{u,j}(x\setminus y)\prod_{i\in y}x_i$$

where $s'_{u,j}$ and $\tilde{s}_{u,j}$ are unique multilinear polynomials over $\mathbb{R}^{k-j}$.

Next, we formally define the class of homogeneous, read-once circuits of pairs of symmetric gates.

Definition 3.3. *The class* $RO[k,d,u,v]$ *is the class of Boolean functions* f *over* $\mathbb{F}_2^{k^d}$ *computable by homogeneous read-once circuits of depth* d *and uniform fan-in* k*, consisting of alternating levels of* s_u *and* s_v *gates of* k *variables. The arguments to* f *are the* k^d *inputs to the circuit, which may be individually negated. For convenience, we will often assume that these are the only negations that appear in the circuit, and that when* d *is even, the topmost gate computes* s_u *and when* d *is odd, the topmost gate computes* s_v*.*

Next we formally define the *type* parameters and other quantities that are used in the statement of the main theorem.

Definition 3.4. *(i) For* $x \in \mathbb{F}_2^{k^d}$*, represented either as* $\{0,1\}^{k^d}$ *or as* $\{1,-1\}^{k^d}$*, the parameters* $tier_{k,i}(x) \in \{0,\ldots,k\}^{k^{d-i}}$*, for* $0 \le i \le d$*, are defined recursively. Let* $tier_{k,i,j}(x)$ *denote the* j^{th} *entry of the vector* $tier_{k,i}(x)$*.*

- $tier_{k,0,j}(x) =_{def} 1 \iff x_j = 1.$
- $tier_{k,i+1,j}(x) =_{def} \quad |\{(j-1)k \le l \le jk : tier_{k,i,l}(x) \ne 0\}|.$

For example, for $x = (0,1,1,0,1,0,1,1)$*,* $tier_{2,0}(x) = x$*,* $tier_{2,1}(x) = (1,1,1,2)$*,* $tier_{2,2}(x) = (2,2)$*, and* $tier_{2,3}(x) = (2)$*.*

(ii) For $x \in \mathbb{F}_2^{k^d}$*, the parameters* $type_{k,i}(x) \in \{0,\ldots,k^{d-i}\}^k$*, for* $0 \le i \le d$ *are defined as follows.*

- $type_{k,i,j}(x) =_{def} \quad |\{l : tier_{k,i,l}(x) = j\}|$;
  i.e, the j^{th} *entry in the vector* $type_{k,i}(x)$ *is the number of entries in* $tier_{k,i}(x)$ *that equal* j*. Notice that* $\sum_{j=1}^{k} j\, type_{k,i+1,j}(x) = \sum_{j=1}^{k} type_{k,i,j}(x)$*;* $\sum_{j=1}^{k^{d-i}} tier_{k,i,j}(x) = \sum_{j=1}^{k} type_{k,i-1,j}(x)$*; and* $\sum_{j=1}^{k} type_{k,d,j}(x) = 1$ *if* $x \ne \vec{0}$ *and 0 otherwise. Furthermore,* $type_{k,0,1}(x) = |x|$ *and for* $2 \le j \le k$*,* $type_{k,0,j} = 0$*.*

For example, for $x = (0,1,1,0,1,0,1,1)$*,* $type_{2,0}(x) = (5,0)$*,* $type_{2,1}(x) = (3,1)$*,* $type_{2,2}(x) = (0,2)$*, and* $type_{2,3}(x) = (0,1)$*.*

(iii) The following quantities, depending on $k,d \in \mathbb{N}$ *and* $u,v \subseteq \{0,\ldots,k\}$*, are also used extensively in the next theorem. We will omit the subscripts* u *and* v *on these quantities when the context is clear.*

- *For $2 \le i \le d$, and $0 \le j \le k$, the quantities $T_{k,i,j} \in \mathbb{R}$ are defined as follows.*
$$T_{k,i+1,j} =_{def} \tilde{s}_{u,j}(T_{k,i,0}^{k-j})$$
when i is odd, and when i is even, u is replaced by v. Recall the definition (3.2) of the symmetric Boolean functions s_u and s_v and the related multilinear polynomials $\tilde{s}_{u,j}$ and $\tilde{s}_{v,j}$ over $\mathbb{R}^{k-j}$.
- *The definition of $T_{k,1,j}$ depends on whether the $\mathbb{R}$domain of s_u and s_v is viewed as $\{0,1\}^k$ or as $\{1,-1\}^k$. In the latter case, $T_{k,1,j} =_{def} \tilde{s}_{v,j}(\vec{0})$; $T_{k,0,0} = 0$; and in the former, $T_{k,1,j} =_{def} \tilde{s}_{v,j}(\frac{1}{2}^{k-j})$; $T_{k,0,0} = 1/2$.*

We have now set up all the machinery required to state and and prove the main theorem.

THEOREM 3.5. *Let f be computed by an $RO[k,d,u,v]$ circuit, and let $\mu \in \mathbb{F}_2^{k^d}$ be the characteristic vector of those arguments to f that are negated inputs to the circuit.*

(i) If $\mathbb{R}$domain(f) is $\{0,1\}^{k^d}$, then if $x \neq \vec{0}$,
$$\hat{f}(x) = (-1)^{|x \cap \mu|}(1/2)^{|x|} \prod_{i=1}^{d} \prod_{j=1}^{k} T_{k,i,j}^{type_{k,i,j}(x)};$$
if $\mathbb{R}$domain(f) is $\{1,-1\}^{k^d}$, then if $x \neq \vec{1}$,
$$\hat{f}(x) = (-1)^{|x \cap \mu|} \prod_{i=1}^{d} \prod_{j=1}^{k} T_{k,i,j}^{type_{k,i,j}(x)},$$
and in both cases, $\hat{f}(\vec{0}) = T_{k,d,0}$. Recall from the Definition 3.4 that the quantities $T_{k,i,j}$ depend on the choice of s_u, s_v, and $\mathbb{R}$domain(f).

(ii) $L_1(\hat{f}) = L_{k,d}$, where $L_{k,0} = 1$ and in general,
$$L_{k,i} =_{def} \sum_{j=0}^{k} (L_{k,i-1} - T_{k,i-1,0})^j \; |T_{k,i,j}| \binom{k}{j}.$$

PROOF. The proofs are by induction on d. For (i), the induction basis is straightforward: when $d = 1$, then f is computed by a single s_v gate, thus, from Fact 2.2,
$$\hat{f}(x) = (-1)^{|x \cap \mu|}(1/2^{|x|})\tilde{s}_{v,|x|}(\frac{1}{2}^{k-|x|}) \quad (3.1)$$

If $x \in \mathbb{F}_2^k$ and $x = \vec{0}$, then $\hat{f}(\vec{0}) = \tilde{s}_{v,0}(\frac{1}{2}^k)$, which is defined in 3.4 as $T_{k,1,0}$. From the general definition of $T_{k,i,j}$ in 3.4, and since, when $d = 1$, $x \in \mathbb{F}_2^k$, and $x \neq \vec{0}$, then there is exactly one j, namely $|x|$, where $type_{k,1,j}(x)$ is non-zero, it follows that

$$\tilde{s}_{v,|x|}(\frac{1}{2}^{k-|x|}) = T_{k,1,|x|} = \prod_{j=1}^{k} T_{k,1,j}^{type_{k,1,j}(x)}.$$

Substituting the above in (3.1) completes the proof of the induction basis. For the induction step, we apply Theorem 3.1 with $g = s_u$, and $h_1, \ldots, h_k \in RO[k, d-1, u, v]$, with $|l(1^{k^d})|$, (the size of the set of arguments to h_l) being k^{d-1} for all $1 \leq l \leq k$. This yields

$$\hat{f}(w) = \prod_{l \in y_w} \left(\hat{h}_l(l(w))\right) \;\; \tilde{g}_{y_w}\left(\hat{h}_1(\vec{0}), \ldots, \hat{h}_k(\vec{0})) \;\setminus\; y_w\right).$$

Assuming the induction hypothesis that the theorem holds for the functions $h_1, \ldots, h_k \in RO[k, d-1, u, v]$, the above equation becomes

$$\hat{f}(w) = \tilde{s}_{u,|y_w|}\left(T_{k,d-1,0}^{k-|y_w|}\right) \prod_{l \in y_w} (-1)^{|l(w) \cap \mu|}(1/2^{|l(w)|})\Big(\prod_{i=1}^{d-1} \prod_{j=1}^{k} T_{k,i,j}^{type_{k,i,j}(l(w))} \quad \Big). \quad (3.2)$$

Now, by definition of the *type* parameters, it follows that

$$\prod_{l \in y_w} T_{k,i,j}^{type_{k,i,j}(l(w))} = T_{k,i,j}^{type_{k,i,j}(w)}, \quad (3.3)$$

and

$$\prod_{l \in y_w} (-1)^{|l(w) \cap \mu|}(1/2^{|l(w)|}) = (-1)^{|w \cap \mu|}(1/2^{|w|}), \quad (3.4)$$

for every $1 \leq i \leq d-1$ and $1 \leq j \leq k$. Furthermore, by the definition of $T_{k,i,j}$, we get $\tilde{s}_{u,|y_w|}\left(T_{k,d-1,0}^{k-|y_w|}\right) = T_{k,d,|y_w|}$, and since $w \in \mathbb{F}_2^{k^d}$, $w \neq \vec{0}$, by the definition of the *type* parameters, there is exactly one j, namely $j = |y_w|$, where $type_{k,d,j}(w)$ is non-zero, it follows that

$$\tilde{s}_{u,|y_w|}\left(T_{k,d-1,0}^{k-|y_w|}\right) = \prod_{j=1}^{k} T_{k,d,j}^{type_{k,d,j}(w)}. \quad (3.5)$$

Now substituting (3.3),(3.4),and (3.5) in (3.2), it follows that when $w \neq \vec{0}$,

$$\hat{f}(w) = (-1)^{|w \cap \mu|}(1/2^{|w|}) \prod_{i=1}^{d} \prod_{j=1}^{k} T_{k,i,j}^{type_{k,i,j}(w)}.$$

Furthermore, when $w = \vec{0}$, it follows that $y_w = \vec{0}$ as well, and thus (3.2) reduces to $\hat{f}(\vec{0}) = \tilde{s}_{u,0}\left(T^k_{k,d-1,0}\right) = T_{k,d,0}$. This proves (i).

We show (ii) again by induction on d. The induction basis, for $d = 0$ is direct. For the induction step, we apply Theorem 3.1 again with $g = s_u$, and $h_1, \ldots, h_k \in RO[k, d-1, u, v]$, to get

$$L_1(\hat{f}) = \sum_{y \in \mathbb{F}_2^k} \Big(\prod_{i \in y} \left(L_1(\hat{h}_i) - \hat{h}_i(\vec{0})\right) \Big) | \ \tilde{s}_{u,|y|}\left((\hat{h}_1(\vec{0}), \ldots, \hat{h}_k(\vec{0})) \ \setminus \ y\right) |.$$

Assuming the induction hypothesis that the theorem holds for the functions $h_1, \ldots, h_k \in RO[k, d-1, u, v]$, the above equation becomes

$$\begin{aligned} L_1(\hat{f}) &= \sum_{y \in \mathbb{F}_2^k} \left(L_{k,d-1} - T_{k,d-1,0}\right)^{|y|} \ | \ \tilde{s}_{u,|y|}\left(T^{k-|y|}_{k,d-1,0}\right) \ | \\ &= \sum_{y \in \mathbb{F}_2^k} \left(L_{k,d-1} - T_{k,d-1,0}\right)^{|y|} \ \ |T_{k,d,|y|}|, \end{aligned}$$

The summands of the last quantity depend only on $|y|$ allowing it to be rewritten as:

$$\sum_{j=1}^{k} \binom{k}{j} \left(L_{k,d-1} - T_{k,d-1,0}\right)^{j} \ \ |T_{k,d,j}|,$$

thus proving (ii). □

REMARK 3.6. *The above proof extends to read-once circuits of arbitrary structure, constructed from arbitrary, but fixed sets of symmetric gates.*

We apply Definition 3.4 and Theorem 3.5 to obtain explicit expressions for the Fourier coefficients and their L_1 norm for the special case of homogeneous, read-once AC^0 functions, f. In this case, we will show that the values $\hat{f}(x)$ do not depend on the vector-valued parameters $type_{k,i}(x)$ but rather on the d scalars: $\sum_j type_{k,i,j}(x)$ for $1 \le i \le d$. Noticing that $\vee = s_{\{1,\ldots,k\}}$, and $\wedge = s_{\{k\}}$, we study the class $RO[k, d, u = \{1, \ldots, k\}, v = \{k\}]$. Again, we assume that the topmost gate is an $\vee$ ($\wedge$) gate if the depth is even (odd).

COROLLARY 3.7. *Let f be computed by an $RO[k, d, u = \{1, \ldots, k\},\ v = \{k\}]$ circuit and let μ be the vector of its negated inputs. Let* $\mathbb{R}$ *$domain(f) = \{0,1\}^{k^d}$, and $range(f) = \{0,1\}$. Then* *(i)* $\hat{f}(\vec{0}) = T_{k,d,0}$, *which, for the chosen* u *and* v, *becomes:*

$$T_{k,2i+1,0} = (T_{k,2i,0})^k, \text{ and } T_{k,2i,0} = 1 - (1 - T_{k,2i-1,0})^k; \quad T_{k,1,0} = 1/2^k;$$

and $T_{k,0,0} = 1/2$.

(ii) If $x \neq \vec{0}$, *and* $b_i =_{def} \sum_{j=1}^{k} type_{k,i,j}(x)$ *for* $0 \leq i \leq d$, *then*

$$|\hat{f}(x)| = (1/2^{b_1 k})\Big(\prod_{\substack{i=1\\ i\ even}}^{d-1} (T_{k,i,0})^{kb_{i+1}-b_i}\Big)\Big(\prod_{\substack{i=1\\ i\ odd}}^{d-1} (1 - T_{k,i,0})^{kb_{i+1}-b_i}\Big),$$

and

$$sign(\hat{f}(x)) = (-1)^{|x\cap\mu| + \sum_{i=1}^{d} b_i} \text{ if } d \text{ is even and } (-1)^{|x\cap\mu| + \sum_{i=1}^{d-1} b_i} \text{ if } d \text{ is odd.}$$

(iii) $L_1(\hat{f}) = L_{k,d}$ *where*

$$L_{k,0} =_{def} 1; \quad L_{k,1} =_{def} 1; \quad L_{k,i} =_{def} (L_{k,i-1})^k \text{ if } i \text{ is odd and}$$
$$=_{def} (1 - 2T_{k,i-1,0} + L_{k,i-1})^k \text{ if } i \text{ is even.}$$

PROOF. We first define the quantities $T_{k,i,j}$ as in Definition 3.4, for the special case where $u = \{1, \ldots, k\}$ and $v = \{k\}$. Noticing that

$$\tilde{s}_u(x) = 1 - \prod_{i=1}^{k}(1 - x_i) \text{ and } \tilde{s}_v(x) = \prod_{i=1}^{k} x_i,$$

it is not hard to see that Definition 3.4, for $T_{k,i,0}$, $0 \leq i \leq d$ matches that given in statement (i) of the corollary. Thus (i) is straightforward, from Theorem 3.5. For general $1 \leq j \leq k$, Definition 3.4 gives

$$T_{k,1,j} =_{def} 1/2^{k-j}; \quad T_{k,2i+1,j} = (T_{k,2i,0})^{k-j},$$

and

$$T_{k,2i,j} =_{def} (-1)^{j+1}(1 - T_{k,2i-1,0})^{k-j}.$$

Now, applying Theorem 3.5, and substituting the new, specific expressions for $T_{k,i,j}$, we get

$$\hat{f}(x) = (-1)^{|x\cap\mu|}(1/2^{|x|})\Big(\prod_{\substack{i=0\\ i\ even}}^{d-1} \prod_{j=1}^{k} T_{k,i,0}^{(k-j)type_{k,i+1,j}(x)}\Big)*$$

$$\Big(\prod_{\substack{i=1 \\ i \ odd}}^{d-1} \prod_{j=1}^{k} (1 - T_{k,i,0})^{(k-j) type_{k,i+1,j}(x)} (-1)^{(j+1) type_{k,i+1,j}(x)} \Big). \qquad (3.6)$$

Assembling the exponents of the two products over j in (3.6), noticing - by definition of the *type* parameters - that $\sum_{j=1}^{k} j \ type_{k,i+1,j}(x) = \sum_{j=1}^{k} type_{k,i,j}(x)$; and by the definition of the symbols b_i, (3.6) reduces to

$$\hat{f}(x) = (-1)^{|x \cap \mu|} (1/2^{|x|}) (T_{k,0,0})^{kb_1 - b_0} \Big(\prod_{\substack{i=1 \\ i \ even}}^{d-1} (T_{k,i,0})^{kb_{i+1} - b_i} \Big) *$$

$$\Big(\prod_{\substack{i=1 \\ i \ odd}}^{d-1} (1 - T_{k,i,0})^{kb_{i+1} - b_i} (-1)^{b_i + b_{i+1}} \Big). \qquad (3.7)$$

Now, noticing that $T_{k,0,0} = 1/2$, and, by definition of the *type* parameters, that $b_0 = \sum_{j=1}^{k} type_{k,0,j}(x) = |x|$, (3.7) gives the values of $|\hat{f}(x)|$ and $sign(\hat{f}(x))$ required to prove (ii).

To prove (iii), we apply Theorem 3.5(ii), with the new definitions of $T_{k,i,j}$ and the fact that $L_{k,0} = 1$, to get $L_{k,1} = \sum_{j=0}^{k} (1 - \frac{1}{2})^j \frac{1}{2}^{k-j} \binom{k}{j} = 1$. In general, if i is odd, then $T_{k,i,j} = T_{k,i-1,0}^{k-j}$, thus

$$L_{k,i} = \sum_{j=0}^{k} (L_{k,i-1} - T_{k,i-1,0})^j |(T_{k,i-1,0})^{k-j}| \binom{k}{j} \quad = \quad (L_{k,i-1})^k;$$

and if i is even, then

$$L_{k,i} = \sum_{j=0}^{k} (L_{k,i-1} - T_{k,i-1,0})^j \quad |(-1)^{j+1} (1 - T_{k,i-1,0})^{k-j}| \quad \binom{k}{j}$$

$$= (1 - 2T_{k,i-1,0} + L_{k,i-1})^k,$$

thus proving (iii). □

We illustrate the above corollary with an example.

EXAMPLE 3.8. *Consider* $f \in RO[k = 2, d = 3, \{1, \ldots, k\}, \{k\}]$, *with* μ, *the vector of negated indices is* $(0,1,1,1,1,1,0,1)$. *If* $x = (1,0,1,0,0,0,0,0)$, *then* *defining* $b_i =_{def} \sum_{j=1}^{2} type_{2,i,j}(x)$, *we get* $b_0 = 2$, $b_1 = 2$, $b_2 = 1$, $b_3 = 1$, *and* $|x \cap \mu| = 1$. *In addition, it is clear that*

$$T_{2,1,0} = \frac{1}{2^2}; \quad T_{2,2,0} = 1 - (1 - \frac{1}{2^2})^2; \text{ and } T_{2,3,0} = (1 - (1 - \frac{1}{2^2})^2)^2.$$

In general,

$$T_{k,i,0} = (1 - (1 - (1 - \ldots (1 - \left(\frac{1}{2}\right) \overbrace{^{k})^{k} \ldots)^{k})^{k})^{k}}^{i \text{ times}}.$$

Thus

$$\begin{aligned} \hat{f}(x) &= (-1)^{b_1+b_2+|x,\cap\mu|}\frac{1}{2^{b_1 k}}(1-\frac{1}{2^k})^{b_2 k - b_1}(1-(1-\frac{1}{2^k})^k)^{b_3 k - b_2} \\ &= (-1)^4 \frac{1}{2^4}(1-(1-\frac{1}{2^2})^2). \end{aligned}$$

Furthermore, it is clear that

$$L_{2,1} = 1; \; L_{2,2} = 2^2(1 - \frac{1}{2^2})^2; \quad \text{and } L_1(\hat{f}) = L_{2,3} = 2^4(1 - \frac{1}{2^2})^4.$$

It follows from the recursive expression for $L_{k,i}$ and the fact that that the spectral L_1 norm of functions in $RO[k, d, \{1, \ldots, k\}, \{k\}]$ is roughly exponential in k^d, since the quantities $T_{k,i,0}$ are inverse exponentials in k. This generalizes the result in [9], and shows that $RO[k, d, \{1, \ldots, k\}, \{k\}] \subseteq AC^0 \setminus PL_1$.

4. General constant depth circuits with symmetric gates.

The main results of this section, Theorem 4.3, and Proposition 4.5 respectively give the weight distribution of $\mathbb{F}_2^{k^d}$ with respect to the $type_{k,i}$ parameters for $1 \le i \le d$ and show that the spectrum and its norms for arbitrary, non-read-once functions computed by constant depth circuits with symmetric gates can be estimated by determining the weight distributions of simple subspaces with respect to the *type* parameters.

The following basic fact expresses the spectrum of a non-read-once function in terms of the spectra of read-once functions. This fact is used to prove Proposition 4.5 from Theorem 4.3.

FACT 4.1. *Let a Boolean function f over $\mathbb{F}_2^n$ be computed by a depth d circuit of size M, with the symmetric gates s_u and s_v. Then there is a function f_r, computed by an $RO[k,d,u,v]$ circuit, for k no larger than a polynomial in M (but possibly exponential in d), such that*

(i)

$$f(x) = f_r(\sum_{i=1}^{n} x_i b_i),$$

where $b_i \in \mathbb{F}_2^{k^d}$ are fixed, mutually disjoint vectors;

(ii)

$$\hat{f}(x) = \sum_{y \in S_x} \hat{f}_r(y),$$

where S_x is a translate (coset) of a linear subspace of $\mathbb{F}_2^{k^d}$:

$$S_x = \{y \in \mathbb{F}_2^{k^d} :< y, b_i >= 1, \forall i : x_i = 1, \text{ and } < y, b_i >= 0, \text{ otherwise}\}$$

(recall <> is the inner product over $\mathbb{F}_2^{k^d}$: $< x, y >$ has been referred to earlier in our discussion as the parity of the number $|x \cap y|$); and

(iii) for any subspace $T \subseteq \mathbb{F}_2^n$,

$$\sum_{x \in T} \hat{f}(x) = \sum_{y \in S_T} \hat{f}_r(y),$$

where S_T is the linear subspace defined as $\bigcup_{x \in T} S_x$.

PROOF. (i) is straightforward. We show (ii), and (iii) follows directly from (ii). Since

$$\hat{f}(x) = \frac{1}{2^n} \sum_u (-1)^{\langle u,x \rangle} f_r(\sum_{i=1}^{n} u_i b_i),$$

expressing f_r in terms of its Fourier coefficients, $\hat{f}(x)$ becomes

$$\begin{aligned}
&= \frac{1}{2^n} \sum_{u \in \mathbb{F}_2^n} (-1)^{\langle u,x \rangle} \Big(\sum_{y \in \mathbb{F}_2^{k^d}} (-1)^{u_1 \langle y,b_1 \rangle} \ldots (-1)^{u_n \langle y,b_n \rangle} \hat{f}_r(y) \Big) \\
&= \frac{1}{2^n} \sum_{u \in \mathbb{F}_2^n} (-1)^{\langle u,x \rangle} \Big(\sum_{y \in \mathbb{F}_2^{k^d}} (-1)^{\langle u, (\langle y,b_1 \rangle, \ldots, \langle y,b_n \rangle) \rangle} \hat{f}_r(y) \Big) \\
&= \sum_{y \in \mathbb{F}_2^{k^d}} \frac{1}{2^n} \Big(\sum_{u \in \mathbb{F}_2^n} (-1)^{\langle u,x \rangle} (-1)^{\langle u, (\langle y,b_1 \rangle, \ldots \langle y,b_n \rangle) \rangle} \hat{f}_r(y) \Big).
\end{aligned}$$

Noticing that the inner sum is 0 whenever $x \neq (\langle y, b_1\rangle, \ldots \langle y, b_n\rangle)$, i.e, whenever $y \notin S_x$, and that it equals $\hat{f}_r(y)$ otherwise, we get

$$\hat{f}(x) = \sum_{y \in S_x} \hat{f}_r(y).$$

□

Before we state the main theorem of the section, we formally define weights and weight distributions.

DEFINITION 4.2. *For $x \in \mathbb{F}_2^n$, a set of* **weights** *or* **parameters** *of x, $p_1(x), \ldots p_d(x)$, where p_i typically take vector values in $\mathbb{N}^{l_i}$ for some $l_i < n$, must satisfy*

(i) the sets $\rho(a_1, \ldots, a_d)$ defined as

$$\{x \in \mathbb{F}_2^n \ : \ p_1(x) = a_1, \ldots, p_d(x) = a_d\}$$

for distinct $(a_1, \ldots, a_d)$ form a partition of $\mathbb{F}_2^n$, and

(ii) $|\rho(a_1, \ldots, a_d)|$ depends only on $a_1, \ldots, a_d$.

The **weight distribution** *of a subset $S \subseteq \mathbb{F}_2^n$ with respect to the weights p_i is given by specifying the quantities $|\rho(a_1, \ldots, a_d) \cap S|/|S|$ for all relevant $a_1, \ldots, a_d$.*

THEOREM 4.3. *Let $a_i \in \{0, \ldots, k^{d-i}\}^k$, for $1 \leq i \leq d$, and let the set $\rho_{k,d}(a_1, \ldots, a_d)$ be defined as:*

$$\{x \in \mathbb{F}_2^{k^d} \ : \ type_{k,1}(x) = a_1, \ldots, type_{k,d}(x) = a_d\}.$$

In other words, the sets $\rho_{k,d}(a_1, \ldots, a_d)$ define a partition of $\mathbb{F}_2^{k^d}$. Denoting the j^{th} entry of a_i as $a_{i,j}$, observe that by the definition of the type parameters, $\rho_{k,d}(a_1, \ldots, a_d) = \emptyset$, if for some $1 \leq i \leq d$, $\sum_{j=1}^{k} j a_{i+1,j} \neq \sum_{j=1}^{k} a_{i,j}$. Otherwise,

(i) $|\rho_{k,d}(a_1, \ldots, a_d)| =$

$$\prod_{i=1}^{d} (\sum_{j=1}^{k} a_{i,j})! \prod_{j=1}^{k} \frac{\binom{k}{j}^{a_{i,j}}}{a_{i,j}!}.$$

(ii) If f is computed by a $RO[k, d, u, v]$ circuit, for some $u, v \subseteq \{0, \ldots, k\}$, with the vector $\mu \in \mathbb{F}_2^{k^d}$ representing the negated inputs, and if x, y both belong in the same set $\rho_{k,d}(a_1, \ldots, a_d)$, then $|\hat{f}(x)| = |\hat{f}(y)|$; and $sign(\hat{f}(y)) \neq sign(\hat{f}(x))$ exactly when $|x \cap \mu| \neq |y \cap \mu|$.

PROOF. For (i), we consider a k-ary tree of depth d, such that each $x \in \mathbb{F}_2^{k^d}$ marks a unique set of $|x|$ leaves. Now $\rho_{k,d}(a_1, \ldots, a_d)$ is the set of all x's such that at level i of the tree there are $\sum_{j=1}^{k} a_{i,j}$ "marked nodes" whose descendants have at least one marked leaf. Furthermore, for any $1 \leq j \leq k$, there are $a_{i,j}$ nodes that have exactly j "marked children." Therefore, assuming that the $a_{i,j}$ nodes, with j marked leaves, at level i, are indistinguishable, $(\sum_{j=1}^{k} a_{i,j})! \,/\, \prod_{j=1}^{k} a_{i,j}!$ is simply the number of distinct permutations of marked nodes at the i^{th} level. Furthermore, $\prod_{j=1}^{k} \binom{k}{j}^{a_{i,j}}$ is the number of ways of choosing the marked children of the marked nodes at the i^{th} level. It is not hard to see that the product of these two quantities with i ranging from 1 to d gives the size of the set $\rho_{k,d}(a_1, \ldots, a_d)$ thus showing (i). The proof of (ii) is a direct consequence of Theorem 3.5, and the definition of the sets $\rho_{k,d}(a_1, \ldots, a_d)$. $\square$

Notice that the above theorem, and Fact 4.1(iii) directly enable us to estimate the sum of the Fourier transform of functions f in any constant depth circuit class over certain subspaces T. This applies to those subspaces T for which the distribution of the corresponding subspace $S_T \subseteq \mathbb{F}_2^{k^d}$ (from Fact 4.1(iii)) with respect to the *type* parameters is close to that of $\mathbb{F}_2^{k^d}$ itself. In particular, if the distribution of S_T is identical to $\mathbb{F}_2^{k^d}$, then the required sum of Fourier coefficients is exactly $|S_T|/2^{k^d}$ or zero, depending on the value of the corresponding read-once function f_r at $\vec{0}$.

The following corollary applies specifically to read once AC^0 functions.

COROLLARY 4.4. *Let* $a_i \in \{0, \ldots, k^d\}^k$, *for* $0 \leq i \leq d-1$, *and let the set* $\sigma_{k,d}(a_1, \ldots, a_d)$ *be defined as:*

$$\{x \in \mathbb{F}_2^{k^d} \;:\; \sum_{j=1}^{k} type_{k,0,j}(x) = |x| = a_0, \ldots, \sum_{j=1}^{k} type_{k,d-1,j}(x) = a_{d-1}\}.$$

Then

(i)

$$|\sigma_{k,d}(a_0, \ldots, a_{d-1})| = \prod_{j=1}^{d} \xi_{k,j}(a_0, \ldots, a_{d-1}),$$

where for all $1 \le j < d$,

$$\xi_{k,j}(a_0,\ldots,a_{d-1}) =_{def} \sum_l (-1)^{(a_j-l)}\binom{a_j}{l}\binom{lk}{a_{j-1}};$$
$$\xi_{k,d}(a_0,\ldots,a_{d-1}) =_{def} \binom{k}{a_{d-1}}.$$

(ii) If f is computed by a $RO[k,d,\{1,\ldots,k\},\{k\}]$ circuit, with the vector $\mu \in \mathbb{F}_2^{k^d}$ representing the negated inputs, and if x, y both belong in the same set $\sigma_{k,d}(a_1,\ldots,a_d)$, then $|\hat{f}(x)| = |\hat{f}(y)|$; and $sign(\hat{f}(x))$ and $sign(\hat{f}(x))$ differ exactly when $|x \cap \mu| \neq |y \cap \mu|$.

PROOF. The proof of (i) could be derived from Theorem 4.3 (i), however, a direct proof is easier. As in Theorem 4.3 (i), if $x \in \mathbb{F}_2^{k^d}$ marks the leaves of a k-ary tree of depth d, then $\sigma_{k,d}(a_1,\ldots,a_d)$ is the set of all x's that have a_i marked nodes at the i^{th} level, i.e, nodes whose descendants include at least one marked leaf. Now the proof of (i) follows from observing that for $1 \le j < d$, $\xi_{k,j}(a_0,\ldots,a_{d-1})$ is the number of ways in which a_{j-1} distinct children at the $j-1^{st}$ level can be chosen from a_j distinct nodes at the j^{th} level, each of which contains k distinct children; and $\xi_{k,d}(a_0,\ldots,a_{d-1})$ is the number of ways in which a_{d-1} children can be chosen from a single root node with k children. The proof of (ii) is straightforward from the definition of the sets $\sigma_{k,d}$. □

The next proposition points out that it would be useful to develop a technique to determine the weight distribution of simple subspaces of $\mathbb{F}_2^{k^d}$ with respect to the *type* parameters.

PROPOSITION 4.5. *For a Boolean function f over $\mathbb{F}_2^n$ computed by a depth d circuit with symmetric gates s_u and s_v, let f_r be the corresponding function computed by an $RO[k,d,u,v]$ circuit, as in Fact 4.1, and furthermore let S_x be the translate such that*

$$\hat{f}(x) = \sum_{y \in S_x} \hat{f}_r(y),$$

and for any subspace $T \subseteq \mathbb{F}_2^n$, let S_T be the subspace $\bigcup_{x\in T} S_x$. Then the weight distributions of each S_x and S_T with respect to the weights $type_{k,1}(z)$, …, $type_{k,d}(z)$, and $|z \cap \mu|$ determine the quantities: $\hat{f}(x)$ for any $x \in \mathbb{F}_2^n$; $L_1(\hat{f})$; and $\sum_{x\in T} \hat{f}(x)$. In particular, if $f \in AC^0[d]$, then these quantities are determined by obtaining the distribution of S_x and S_T with respect to the d weights $\sum_j type_{k,i,j}(x)$ for $1 \le i \le d$.

PROOF. The proof follows directly from Fact 4.1, Definition 4.2, Theorem 3.5, Corollary 3.7 and Theorem 4.3(ii). □

Coding theory provides some tools - called the MacWilliams identities [25] - for determining weight distributions with respect to various weights. However, these weights usually satisfy properties that it is not clear if the $type_{k,i}(x)$ parameters satisfy. For instance, a "valid" parameter p for which a MacWilliams identity exists usually satisfies the property of linearity: there must exist "valid" parameters p_1 and p_2 of vectors over $\mathbb{F}_2^{n_1}$ and $\mathbb{F}_2^{n_2}$ respectively, such that if $x \in \mathbb{F}_2^n$ is expressed as the direct sum of two vectors $x_1 \in \mathbb{F}_2^{n_1}$ and $x_2 \in \mathbb{F}_2^{n_2}$, where $n_1 + n_2 = n$, i.e, $x = x_1 x_2$, then $p(x) = p_1(x_1) + p_2(x_2)$. The parameter $|x \cap \mu|$, however, does have a MacWilliams identity for any fixed vector μ.

Acknowledgements

I thank Jehoshua Bruck and Andrew Odlyzko for interesting discussions related to this paper.

References

[1] E. ALLENDER, A note on the power of threshold circuits. In *Proc.* 30^{th} *Ann. IEEE Symp. Foundations of Computer Science*, 1989, 580-584.

[2] D. A. BARRINGTON, Bounded width, polynomial size branching programs recognize exactly those languages in *NC*. In *Proc.* 18^{th} *Ann. ACM Symp. Theory of Computing*, 1986, 1-5.

[3] R. BEIGEL, "Why do extra majority gates help?" In *Proc.* 24^{th} *Ann. CM Symp. Theory of Computing*, 1992, 450-454.

[4] D.A. BARRINGTON, R. BEIGEL, S. RUDICH, Representing Boolean functions as polynomials modulo composite numbers. In *Proc.* 24^{th} *Ann. CM Symp. Theory of Computing*, 1992, 455-461.

[5] Y. BRANDMAN, A. ORLITSKY, J. HENNESSY, A spectral lower bound technique for the size of decision trees and two-level and-or circuits, In *IEEE Trans. on Computers*, **39** (2), 1990, 282-287.

[6] J. BRUCK, Harmonic analysis of polynomial threshold functions, In *SIAM Journal of Discrete Mathematics*, **3** (2), 1990, 168-177.

[7] M. Bellare, A technique for upper bounding the spectral norm with applications to learning. In *Proc.* 5^{th} *Ann. IEEE Symp. Computational Learning Theory (COLT),* 1992, 62-70.

[8] R. Boppana, M. Sipser, *The complexity of finite functions.* Technical Report, Massachussetts Institute of Technology, Laboratory for Computer Sciences, MIT/LCS/TM-405, 1989.

[9] J. Bruck, R. Smolensky, Polynomial Threshold Functions, AC^0 Functions and Spectral Norms. *SIAM J. Computing,* **21** (1), 1992, 33-42.

[10] H. Dym, H.P. McKean, *Fourier series and integrals.* Probability and Mathematical Statistics series, Academic Press, 1972.

[11] M. Dowd, M. Sitharam, *Shannon bounds for functions over* $\mathbb{Z}_n^2$. Technical Report 90-12-1, Kent State University, Department of Mathematics and Computer Science, 1990.

[12] M. Furst, J. Jackson, S. Smith, Improved learning of AC^0 functions. In *Proc.* 5^{th} *Ann. IEEE Symp. on Computational Learning Theory (COLT),* 1992, 317-325.

[13] M. Furst, J. Saxe, M. Sipser, Parity, circuits and the polynomial time hierarchy. *Mathematical Systems Theory,* **17**, 1984, 17-27.

[14] M. Goldman, J. Hastad, A.A. Razborov, Majority gates vs. general weighted threshold gates. In *J. Computational Complexity*, **2**, 1992, 277-300.

[15] M. Goldman, J. Hastad, On the power of small depth threshold circuits, In *Proc.* 31^{st} *Ann. IEEE Symp. Foundations of Computer Science*, 1990, 610-618.

[16] V. Grolmusz *Harmonic analysis, real approximation and communication complexity of Boolean functions*, Manuscript, 1994.

[17] J. Hastad, *Computational limitations of small depth circuits.* Ph. D thesis, Massachussetts Institute of Technology press, 1986.

[18] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, G. Turán, Threshold circuits of bounded depth. In *Proc.* 28^{th} *Ann. IEEE Symp. Foundations of Computer Science*, 1987, 99-110.

[19] J. Kahn, J. Kalai, N. Linial, The influence of variables on Boolean functions. In *Proc.* 29^{th} *Ann. IEEE Symp. Foundations of Computing (FOCS),* 1988, 68-80.

[20] E. Kushilevitz, Y. Mansour, Learning decision trees using the Fourier transform. *SIAM Journal on Computing,* **22** (6), 1993, 1331-1348.

[21] M. Krause, Geometric arguments yield better bounds for threshold circuits and distributed computing, In *Proc.* 6^{th} *Ann. IEEE Symp. in Structure in Complexity Theory*, 1991, 314-322.

[22] M. Krause, S. Waack, Variation ranks of communication matrices and lower bounds for depth two circuits having symmetric gates and unbounded fan-in, In *Proc.* 32^{nd} *Ann. IEEE Symp. in Foundations of Computer Science*, 1991, 777-782.

[23] N. Linial, Y. Mansour, N. Nisan, Constant depth circuits, Fourier transforms, and learnability. To appear in *JACM*; In *Proc.* 30^{th} *Ann. IEEE Symp. on Foundations of Computing (FOCS)*, 1989, 574-579.

[24] N. Linial, N. Nisan, Approximate inclusion-exclusion. *Combinatorica* **10** (4), 1990, 349-365.

[25] F.J. MacWiliams, N.J.A. Sloane, *The theory of error-correcting codes.* North Holland, 1977.

[26] N. Nisan, A.W. Widgerson, Hardness vs. randomness. To appear in *JCSS*; In *Proc.* 29^{th} *Ann. IEEE Symp. on Foundations of Computing*, 1988, 2-11.

[27] N. Nisan, M. Szegedy, On the degree of Boolean functions as real polynomials. In *Proc.* 24^{th} *Ann. ACM Symp. Theory of Computing*, 1992, 462-467.

[28] R. Paturi, On the degree of polynomials that approximate symmetric Boolean functions. In *Proc.* 24^{th} *Ann. ACM Symp. Theory of Computing*, 1992, 468-474.

[29] *R. Paturi, M. Saks,* Threshold circuits for parity. In *Proc.* 31^{st} *Ann. IEEE Symp. on Foundations of Computing*, 1990, 397-404.

[30] A.A. Razborov, Lower bounds on the monotone complexity of some Boolean functions. *Soviet Mathematics Doklady,* **31**, 1985, 354-357.

[31] A.A. Razborov, Lower bounds on the size of circuits of bounded depth with basis $\{\wedge, \oplus\}$. In *Math. notes of the Aca. of Science of the USSR,* **41** (4), 1987, 333-338.

[32] *M. Sitharam* Pseudorandom generators and learning algorithms for AC^0. In *Proc.* 26^{th} *Ann. ACM Symp. Theory of Computing*, 1994, 478-488; to appear in *Computational Complexity.*

[33] R. Smolensky, Algebraic methods in the theory of lower bounds for boolean circuit complexity, In *Proc.* 19^{th} *Ann. ACM Symp. Theory of Computing*, 1987, 77-82.

[34] K.I. Siu, J. Bruck, On the power of threshold circuits with small weights. *SIAM Journal of Discrete Mathematics,* 4 (3), 1991, 423-435.

[35] A.C. Yao, Lower bounds by probabilistic arguments. In *Proc.* 24^{th} *Ann. IEEE Symp. Foundations of Computing,* 1983, 420-428.

[36] A.C. Yao, Separating the polynomial time hierarchy by oracles. In *Proc.* 26^{th} *Ann. IEEE Symp. on Foundations of Computing* , 1985, 1-10.

[37] A.C. Yao, Circuits and local computation, In *Proc.* 21^{st} *Ann. ACM Symp. Theory of Computing,* 1989, 186-196.

[38] A.C. Yao, On ACC and threshold circuits, In *Proc.* 31^{st} *Ann. IEEE Symp. on Foundations of Computing*, 1990, 619-627.

Manuscript received 10 January 1994

Meera Sitharam
Department of Mathematics and Computer Sciences
Kent State University
Kent, OH 44240, USA
sitharam@mcs.kent.edu