Electronic Colloquium on Computational Complexity, Report No. 21 (2013)

Polynomial threshold functions and Boolean threshold circuits

Kristoffer Arnsfelt Hansen[1] and Vladimir V. Podolskii[2]

[1]Aarhus University, arnsfelt@cs.au.dk
[2]Steklov Mathematical Institute, podolskii@mi.ras.ru

February 5, 2013

Abstract

We study the complexity of computing Boolean functions on general Boolean domains by polynomial threshold functions (PTFs). A typical example of a general Boolean domain is $\{1, 2\}^n$. We are mainly interested in the length (the number of monomials) of PTFs, with their degree and weight being of secondary interest. We show that PTFs on general Boolean domains are tightly connected to depth two threshold circuits. Our main results in regard to this connection are:

- PTFs of polynomial length and polynomial degree compute exactly the functions computed by $\mathsf{THR} \circ \mathsf{MAJ}$ circuits.
- An exponential length lower bound for PTFs that holds regardless of degree, thereby extending known lower bounds for $\mathsf{THR} \circ \mathsf{MAJ}$ circuits.
- We generalize two-party unbounded error communication complexity to the multi-party number-on-the-forehead setting, and show that communication lower bounds for 3-player protocols would yield size lower bounds for $\mathsf{THR} \circ \mathsf{THR}$ circuits.

We obtain several other results about PTFs. These include relationships between weight and degree of PTFs, and a degree lower bound for PTFs of constant length. We also consider a variant of PTFs over the max-plus algebra. We show that they are connected to PTFs over general domains and to $\mathsf{AC}^0 \circ \mathsf{THR}$ circuits.

ISSN 1433-8092

1 Introduction

Let $f : X^n \to \{-1, 1\}$ be a Boolean function on a domain $X \subseteq \mathbb{R}^n$. We say that a real n-variate polynomial P is a *polynomial threshold function* (PTF) computing f if for all $x \in X$ it holds that

$$f(x) = \operatorname{sgn}(P(x)) \ .$$

Other terminology has been used in the literature for the same notion. We may alternatively say that P is a *sign representation* of f, or that P is a *voting polynomial* computing f. Polynomial threshold function have been studied intensively for decades. Much of this work was motivated by questions in computer science [27], and PTFs are now an important object of study in areas such as Boolean circuit complexity [4, 11, 2, 19], learning theory [16, 17], and communication complexity [28]. The main motivation of this paper is Boolean circuit complexity. A major and long-standing open problem is to obtain an explicit super-polynomial lower bound for depth two threshold circuits. A long line of research have established lower bounds for several subclasses of depth two threshold circuits. The largest subclass for which super-polynomial lower bounds are known is the class $\mathsf{THR} \circ \mathsf{MAJ}$ of depth two threshold circuits, where all gates except the output gate is required to compute threshold functions with polynomially bounded weights [9]. We shall see that PTFs on general Boolean domains are tightly connected to both these classes of circuits.

For a PTF P we will be interested in the several measures of complexity. The *length* of P, denoted by $\operatorname{len}(P)$, is the number of monomials of P. The *degree* of P, denoted by $\deg(P)$, is the usual total degree of P. Finally, note that in the case that X is a finite domain, without loss of generality one may assume that the coefficients of P are integers, and can thus speak of the *weight* of P, meaning the largest magnitude of a coefficient of P.

We restrict our focus to the case of computing Boolean functions with Boolean inputs. More precisely we only consider the case when the domain X is a Boolean n-cube, $X = \{a, b\}^n$, for distinct $a, b \in \mathbb{R}$. Such sign representations of Boolean functions have been studied intensively due to their fundamental nature and vast number of applications. This research has almost exclusively focused on the two Boolean n-cubes, $\{0, 1\}^n$ and $\{-1, 1\}^n$, sometimes denoted as the "standard basis" and the "Fourier basis", respectively. Indeed, most often the notion of PTFs is defined specifically for the case of the domain $\{-1, 1\}^n$. This choice is, however, of little consequence when one disregards the length as a parameter and focuses on the degree, as is the case in many applications of PTFs. Note also that for these two domains any PTF can without loss of generality be assumed to be *multilinear*, meaning that all variables have individual degree at most 1.

Focusing on the length of a PTF rather than the degree, the choice of domain becomes crucial already for the case of the two domains $\{0, 1\}^n$ and $\{-1, 1\}^n$. This was studied in depth by Krause and Pudlák [19]. Minksy and Papert [20] has shown that the parity function requires exponential length over the domain $\{0, 1\}^n$ (cf. [10, 1]), whereas it can be computed by a PTF of length 1 over the domain $\{-1, 1\}^n$. Conversely, Krause and Pudlák construct a PTF on domain $\{0, 1\}^n$ of length $\sqrt{n}$ that require length $2^{n^{\Omega(1)}}$ on domain $\{-1, 1\}^n$. For this construction, large weight is crucial. Indeed, Krause and Pudlák also show that any function computed by a polynomial length and polynomial weight PTF on the domain $\{0, 1\}^n$ can also be computed by a polynomial length and polynomial weight PTF on the domain $\{-1, 1\}^n$.

A notable exception to the focus on the domains $\{0, 1\}^n$ and $\{-1, 1\}^n$ is the work of Basu et al. [1] that consider representing the parity function (or rather, a natural generalization of the parity function) on domains of the form $X = A^n$, for a set $A \subseteq \mathbb{Z}$. They especially focus on the cases

$A = \{0, 1, \ldots, m\}$ and $A = \{1, 2, \ldots, m\}$, where $m \geqslant 2$. It is important to note that on most Boolean domains $\{a, b\}^n$, it is not without loss of generality to assume that polynomials are multilinear. One may easily convert a given PTF into a multilinear PTF computing the same function, but such a conversion may change both the length as well as the weight significantly. Indeed, Basu et al. show that the parity function provides such an example. Namely they show that on the domain $\{1, 2\}^n$ there is a PTF of length $n+1$ and degree n^2 computing the parity function, whereas any multilinear PTF computing the same function must have length 2^n. Thus evaluating PTFs on general Boolean domains has the effect that allowing high degree (meaning polynomial, exponential, or perhaps even higher), may help to greatly reduce the length needed to compute a given Boolean function.

In this paper our aim is to investigate in detail the computational power of PTFs of polynomial length over a general Boolean domain of the form $\{a, b\}^n$, $a \neq b$. Some of these domains essentially corresponds to the two usual domains $\{0, 1\}^n$ and $\{-1, 1\}^n$, namely those that are simple scalings $\{0, a\}^n$ and $\{a, -a\}^n$, and we shall hence not consider these further. In particular we shall by general Boolean domains, refer to any other Boolean domain $\{a, b\}$. For most of our results it turns out the precise choice of general domain does not matter (in fact all our results hold when $\mathrm{sgn}(a) = \mathrm{sgn}(b)$), and we shall henceforth develop our results in terms of the domain $\{1, 2\}^n$.

1.1 Our results

Over the usual Boolean domains $\{0, 1\}^n$ and $\{-1, 1\}^n$ PTFs are basic extensions of linear threshold functions that are still very limited in expressive power. Indeed, in these cases PTFs require exponential length to compute simple functions such as symmetric Boolean functions [4, 18]. Over a general Boolean domain the situation changes drastically. We show that in this case PTFs of just constant length can actually compute interesting classes of functions (see Proposition 1 and Proposition 4). More importantly, when moving to polynomial length PTFs obtain computational power right at the frontier of known circuit lower bounds for threshold circuits. Namely we show in Theorem 5 that PTFs of polynomial length and polynomial degree compute exactly the functions computed by polynomial size $\mathsf{THR} \circ \mathsf{MAJ}$ circuits. This circuit class is the largest depth two threshold circuit class for which superpolynomial lower bounds are known. These lower bounds were obtained by sign rank lower bounds of matrices, or equivalently lower bounds for unbounded error communication complexity [9], and this is still the only lower bound method known for this class of circuits. In Section 3.2 we show that this lower bound method applies to PTFs, even with no degree restriction. We tend to believe that allowing exponential or perhaps even larger degree allows for more Boolean functions to be computed by PTFs, and we relate this in Proposition 10 to a question about simulating large weights by small weights in threshold circuits in a very strong way. This in turn also gives an indication that the power of the sign rank lower bound method extends beyond $\mathsf{THR} \circ \mathsf{MAJ}$ circuits. Let us also note in the passing that polynomial length PTFs over domains $\{0, 1\}^n$ and $\{-1, 1\}^n$ also correspond to circuit classes, namely the subclass of $\mathsf{THR} \circ \mathsf{AND}$ circuits with no negations of inputs in the case of domain $\{0, 1\}^n$ and $\mathsf{THR} \circ \mathsf{XOR}$ circuits in the case of domain $\{-1, 1\}^n$. These are thus strictly less powerful than PTFs of polynomial length and polynomial degree over a general Boolean domain.

Our study of PTFs on general Boolean domains leads also to a possible way to approach the major open problem of proving lower bounds for $\mathsf{THR} \circ \mathsf{THR}$ circuits. Just as is the case of $\mathsf{THR} \circ \mathsf{MAJ}$ circuits, most lower bounds for classes of threshold circuits have been obtained using various models of communication complexity [12, 15, 11, 23, 9, 29, 26]. In Section 3.3 and Section 6 we generalize the notion of sign complexity to higher order tensors and unbounded error communication complexity to the multiparty number-on-the-forehead setting, and we show that sufficiently good lower bounds for these (for order 3 tensors, or equivalently, 3-party communication protocols) would yield circuit lower bounds for $\mathsf{THR} \circ \mathsf{THR}$ circuits. An important technical ingredient in this connection is a previous result showing that the threshold gates at the second level can be exchanged with *exact* threshold gates [14]. While we currently know *no* lower bounds for this communication model[1], we feel this relation is significant, given the previous successes of communication complexity for lower bounds for threshold circuit classes, and deserves further study. Multi-party communication complexity have been used earlier for threshold circuit classes, but in the bounded error setting. In particular, lower bounds have been obtained for depth 3 unweighted threshold circuits with small bottom fanin [15]. In the unbounded error setting we can additionally address depth 3 *weighted* threshold circuits with small bottom fanin.

The above relations between PTFs on general Boolean domains and threshold circuits further motivate an in-depth study of PTFs, besides them being a fundamental way to represent Boolean functions. For instance, it is tempting to conjecture that PTFs can only compute functions computable by constant depth threshold circuits. It seems that before such questions can be addressed, one needs more insight into PTFs. We currently don't know how large PTF degree can be useful for computation. In Section 4 we show that the minimal degree of a PTF within a given length bound can be bounded in terms of its integer weights, and conversely the integer weights can be bounded in terms of its degree. These bounds are obtained by setting up suitable linear programs and integer linear programs, where the variables are exponents or weights respectively, and then using known bounds on feasible basic solutions and small integer feasible solutions.

In addition to PTFs on general domains we also consider a max-plus version of PTFs. The max-plus algebra works over the max-plus semiring, which is the set of integers with the max operation playing the role of addition and the usual addition playing the role of multiplication [30, 6]. This setting arises as a "limit" case in several areas of mathematics and turns out to be helpful. In our case it turns out that max-plus PTFs are connected with PTFs over the the general domains and are moreover connected to the hierarchy of $\mathsf{AC}^0 \circ \mathsf{THR}$ circuits.

Finally we study the relations between PTFs over different general domains. Though we are unable to completely resolve the questions arising here, we still can prove some nontrivial relations. For example, we show that PTFs over domains $\{1,2\}^n$ and PTFs over $\{1,-2\}^n$ are essentially equivalent.

2 Preliminaries

2.1 Polynomial threshold functions

For given length bound $l(n)$ and degree bound $d(n)$, we let $\mathsf{PTF}_{a,b}(l(n), d(n))$ denote the class of Boolean functions on domain $\{a,b\}^n$ computed by polynomial threshold functions of length $l(n)$ and degree $d(n)$. That is $f \in \mathsf{PTF}_{a,b}(l(n), d(n))$ if and only if there is a polynomial $p(x) \in \mathbb{Z}[x]$ with $l(n)$ monomials of degree at most $d(n)$ and such that for all $x \in \{a,b\}^n$ we have $f(x) = 1$ if and only if $p(x) \geqslant 0$. Of particular interest is the case when $l(n)$ is a polynomial in n. For this reason we will abbreviate $\mathsf{PTF}_{a,b}(\mathrm{poly}(n), d(n))$ by $\mathsf{PTF}_{a,b}(d(n))$. If we do not wish to impose a degree bound we write this as $\mathsf{PTF}_{a,b}(l(n), \infty)$ and $\mathsf{PTF}_{a,b}(\infty)$, respectively. As mentioned in the

[1]Unlike the case of bounded error communication complexity, even obtaining non-explicit lower bounds pose a challenge, since counting arguments fail [24].

introduction we state our results in terms of the specific domain $\{1,2\}^n$. We remark that in most of our results one may replace $\{1,2\}$ be any other domain $\{a,b\}$, where $|a| \neq |b|$, and $a,b \neq 0$. The exceptions to this are our results about PTFs of constant length[2], namely Propositions 1, 4, and 16 as well as Theorem 17. These results hold instead assuming $\mathrm{sgn}(a) = \mathrm{sgn}(b)$. See Section 7 for further discussion of differences and equivalences between domains.

2.2 Exponential form of PTFs

We shall find it convenient to switch back to the standard domain $\{0,1\}^n$ even when considering PTFs over the domain $\{1,2\}^n$. Given variables $y_1,\ldots,y_n \in \{1,2\}$, we define corresponding variables $x_1,\ldots,x_n \in \{0,1\}$ by $x_i = \log_2(y_i)$. Correspondingly we have $y_i = 2^{x_i}$. Under this change of variables monomials turn into exponential functions, $y_1^{a_1}\ldots y_n^{a_n} = 2^{a_1x_1+\ldots+a_nx_n}$ and more generally a polynomial $P(y) = \sum_{j=1}^{l} c_j \prod_{i=1}^{n} y_i^{a_{ij}}$, turns into a weighted sum of exponential functions:

$$P(y) = \sum_{j=1}^{l} c_j 2^{\sum_{i=1}^{n} a_{ij}x_i} \ ,$$

where $a_{ij} \geqslant 0$ are the non-negative integer exponents of the polynomial. Rewriting a PTF in this way, we shall say it is in *exponential form*. We shall in general allow also for negative integer coefficients a_{ij} in the exponents. By simply multiplying the entire expression with the term $2^{\sum_{i=1}^{n} b_i x_i}$, where $b_i = \max(0, \max_j -a_{ij})$ we can make all the coefficients to be positive. This in turn requires us to redefine the degree of the polynomial to be $\max_j(\sum_{i=1}^{n}\max(0,a_i^{(j)}))+\max_j(\sum_{i=1}^{n}\max(0,-a_i^{(j)}))$. At times it may also be convenient to move the absolute value of the coefficients $|c_j|$ to the exponents as an additive term $\log_2(|c_j|)$ in order to have all coefficients of the exponential form be of magnitude 1.

2.3 Boolean functions and Circuit classes

We give here briefly for the most part standard definitions of Boolean functions and circuit classes. As is usual, when considering a Boolean function $f : \{0,1\}^n \to \{0,1\}$, unless otherwise specified we always have a family of such functions in mind, one for each input length.

The functions of most importance for this work are linear threshold functions defined by linear inequalities and their relatives defined by linear equations [14]. Let $x_1,\ldots,x_n \in \{0,1\}$ be Boolean variables. For $w = (w_1,\ldots,w_n) \in \mathbb{R}^n$ and $t \in \mathbb{R}$, we define the function $\mathrm{THR}_{w,t}$ by $\mathrm{THR}_{w,t}(x) = 1$ if and only if $\sum_{i=1}^{n} w_i x_i \geqslant t$. Similarly we define the function $\mathrm{ETHR}_{w,t}$ by $\mathrm{ETHR}_{w,t}(x) = 1$ if and only if $\sum_{i=1}^{n} w_i x_i = t$. We call w the weights and t the threshold. The case when all weights are 1 and the threshold is $n/2$ is of special interest. Define the function MAJ by $\mathrm{MAJ}(x) = 1$ if and only if $\sum_{i=1}^{n} x_i \geqslant n/2$ and the function EMAJ by $\mathrm{EMAJ}(x) = 1$ if and only if $\sum_{i=1}^{n} x_i = \lceil n/2 \rceil$. Besides these functions we consider the usual AND, OR, and XOR functions of n Boolean variables.

Let THR and ETHR denote the class of $\mathrm{THR}_{w,t}$ and $\mathrm{ETHR}_{w,t}$ functions for all w and t. Let MAJ, EMAJ, AND, OR, XOR denote the class of all MAJ, EMAJ, AND, OR and XOR functions. We may by adding a subscript to these functions and classes denote the number of inputs to them, e.g. by AND_k we denote the Boolean AND function of k Boolean variables. By ANY we denote the class

[2] Note that the XOR function can be computed by a length 1 PTF over the domain $\{1,-2\}$ but not over the domain $\{1,2\}$.

of *all* Boolean functions. This latter class will only be used with a subscript specifying a constant number of inputs, and is hence in this case equivalent to constant sized Boolean combinations.

From these Boolean functions we define unbounded fan-in Boolean circuits. Inputs are allowed to be Boolean variables or their negation, as well as the Boolean constants 0 and 1. As with Boolean functions we always have a family of Boolean circuits in mind, one for each input length n. The size of a Boolean circuit is the number of wires. Unless otherwise specified, we consider circuit families of polynomial size. A class of Boolean functions immediately define a class of Boolean circuits as families of single gate circuits. Given two classes of circuits $\mathcal{C}_1$ and $\mathcal{C}_2$ we denote by $\mathcal{C}_1 \circ \mathcal{C}_2$ the class of circuits consisting of circuits from $\mathcal{C}_1$ that is fed as inputs the output of circuits from $\mathcal{C}_2$.

3 PTFs and threshold circuits

3.1 Circuit characterizations

We first note that PTFs already of length 2 on domain $\{1,2\}^n$ can compute the class of linear threshold functions, and this is in fact an exact characterization. Furthermore, the case of polynomial degree corresponds to the case of polynomial weights.

Proposition 1. $\mathsf{PTF}_{1,2}(2,\infty) = \mathsf{THR}$ *and* $\mathsf{PTF}_{1,2}(2, poly(n)) = \mathsf{MAJ}$.

Proof. Consider first the linear threshold function $\mathrm{THR}_{w,t}$ given by the linear inequality $\sum_{i=1}^n w_i x_i - t \geqslant 0$. By raising each side of the inequality to the power of 2 immediately results in the exponential form of the desired length 2 PTF.

Conversely, consider a length 2 PTF defined by the exponential form inequality $c_1 2^{\sum_{i=1}^n a_i x_i} + c_2 2^{\sum_{i=1}^n b_i x_i} \geqslant 0$. If $\mathrm{sgn}(c_1) = \mathrm{sgn}(c_2)$ the Boolean function is constant, and hence is trivially a linear threshold function. So assume $\mathrm{sgn}(c_1) \neq \mathrm{sgn}(c_2)$. Without loss of generality let $c_1 > 0$ and $c_2 < 0$. The inequality is then equivalent to the linear inequality $\sum_{i=1}^n (a_i - b_i)x_i + \log_2 c_1 - \log_2(-c_2) \geqslant 0$, defining a linear threshold function. The specialization to polynomial degree of the PTF and polynomial weights of the linear threshold function follows immediately. □

We can generalize one direction of Proposition 1 as follows.

Lemma 2. *Any function computed by an* $\mathsf{AND}_k \circ \mathsf{THR}$ *circuit or an* $\mathsf{OR}_k \circ \mathsf{THR}$ *circuit can be computed by a PTF on domain* $\{1,2\}^n$ *of length* $k+1$. *Similarly, any function computed by an* $\mathsf{AND}_k \circ \mathsf{MAJ}$ *circuit or an* $\mathsf{OR}_k \circ \mathsf{MAJ}$ *circuit can be computed by a PTF on domain* $\{1,2\}^n$ *of length* $k+1$ *and degree poly*(n).

Proof. We prove the case of $\mathsf{OR}_k \circ \mathsf{THR}$ circuits. The case of $\mathsf{AND}_k \circ \mathsf{THR}$ circuits then follow by using De Morgan's rules. Let the k threshold functions be given by the sign of the linear expressions $l_1(x), \ldots, l_k(x)$, all with integer coefficients. Let $c = \lceil \log_2 k \rceil + 2$, and define new linear expressions $\widehat{l_1}(x), \ldots, \widehat{l_k}(x)$ by $\widehat{l_i}(x) = c(l_i(x) + 1)$. We can then define the PTF by the exponential form

$$E(x) = 2^{\widehat{l_1}(x)} + \cdots + 2^{\widehat{l_k}(x)} - 2k \ .$$

In case there exist i such that $l_i(x) \geqslant 0$, then $\widehat{l_i}(x) \geqslant c$, and hence $E(x) \geqslant 2^c - 2k > 0$. In case for all i we have $l_i(x) < 0$, then we also have $\widehat{l_i}(x) \leqslant 0$ for all i, and hence $E(x) \leqslant k - 2k < 0$. Again the specialization to polynomial degree of the PTF and polynomial weights of the linear threshold functions follows immediately. □

A basic closure property of PTFs, being sign representations, is that the parity of PTFs is also a PTF.

Lemma 3. *If the Boolean functions $f_1, \ldots, f_k$ are computed by PTFs on domain $\{1,2\}^n$ of length s, then $\mathrm{XOR}(f_1, \ldots, f_k)$ is computed by a PTF on domain $\{1,2\}^n$ of length s^k. Furthermore, if each of the k given PTFs are of degree at most d, the resulting PTF is of degree at most kd.*

Proof. If $E_1(x), \ldots, E_k(x)$ are the exponential forms of the PTFs computing the Boolean functions $f_1, \ldots, f_k$ then $-\prod_{j=1}^{k}(-E_j(x))$ is an exponential form of a PTF computing the function $\mathrm{XOR}(f_1(x), \ldots, f_k(x))$. □

Remark. We note that classes $\mathsf{THR} \circ \mathsf{THR}$, $\mathsf{MAJ} \circ \mathsf{MAJ}$ and so on are also closed under the parity function of constant fan-in. To see this, one can switch to ETHR and EMAJ gates on the bottom level and use the fact that they are closed under AND_k using results of [14].

We can now characterize the Boolean functions computed by constant length PTFs on domain $\{1,2\}^n$ as the class of constant size Boolean combinations of linear threshold functions. This class of functions was considered earlier in the setting of learning in [16].

Proposition 4. $\mathsf{PTF}_{1,2}(O(1), \infty) = \mathsf{ANY}_{O(1)} \circ \mathsf{THR}$, *and* $\mathsf{PTF}_{1,2}(O(1), poly(n)) = \mathsf{ANY}_{O(1)} \circ \mathsf{MAJ}$

Proof. Consider a function $g(f_1(x), \ldots, f_k(x))$, where each f_i is a linear threshold function. First express $g(y)$ as a k-variate polynomial $p(y)$ over $\mathbb{Z}_2$. By Lemma 2, the function of x defined by each monomial of p can be computed by a PTF of length $k+1$. Using Lemma 3 we may now express the sum over $\mathbb{Z}_2$ of these at most 2^k functions by a PTF of length $(k+1)^{2^k}$.

Conversely, we will show how to evaluate the sign of a PTF of length k by a constant size Boolean combination of linear threshold functions. We will describe this evaluation by a decision tree, where each decision node is determined by a linear threshold function. Consider the PTF in its exponential form

$$s_1 2^{l_1(x)} + \cdots + s_k 2^{l_k(x)}$$

where $s_i \in \{-1, 1\}$, and the coefficients of every l_i, except possibly for the constant term, are integer. Define $c = \lceil \log_2 k \rceil$, and consider an input x. First we determine using a decision tree of depth $O(k)$, indices i_1 and i_2 such that

$$l_{i_1}(x) \geqslant l_{i_2}(x) \geqslant l_j(x)$$

for all $j \notin \{i_1, i_2\}$. In case $l_{i_1}(x) \geqslant c + l_{i_2}(x)$ the sign of $E(x)$ is precisely s_{i_1}. Otherwise we have that $l_{i_1}(x) = d + l_{i_2}$ for some number $0 \leqslant d < c$. Note that there are at most c such possible d, since the coefficients of $l_{i_1}(x)$ and $l_{i_2}(x)$ are, except possibly for the constant term, integer. We may consider each of these at most c cases separately after using a depth $\lceil \log_2 c \rceil$ decision tree. For such a fixed d we may now rewrite the exponential form as

$$(s_{i_1} 2^d + s_{i_2}) 2^{l_{i_2}(x)} + \sum_{j \notin \{i_1, i_2\}} s_j 2^{l_j(x)} = s 2^{l(x)} + \sum_{j \notin \{i_1, i_2\}} s_j 2^{l_j(x)}$$

for appropriate s and l, depending on s_{i_1}, s_{i_2}, and l_{i_2}, but otherwise independent of the specific x. Since this is an expression of the same type, but of length $k-1$ we may continue as above. Doing this we obtain a decision tree of depth $O(k^2)$ that determines the sign of the expression. This gives in turn a Boolean combination of $2^{O(k^2)}$ linear threshold functions. Again, the specialization to polynomial degree of the PTF and polynomial weights of the linear threshold functions follows immediately. □

In the case of polynomial degree we can characterize the Boolean functions computed by polynomial length PTFs on domain $\{1,2\}^n$ as a class of depth 2 threshold functions with polynomially bounded weights on the bottom level.

Theorem 5.
$$\mathsf{PTF}_{1,2}(poly(n)) = \mathsf{THR} \circ \mathsf{MAJ}$$

Proof. We first construct a PTF given a $\mathsf{THR}\circ\mathsf{MAJ}$ circuit. Suppose that the output threshold gate is given by the inequality $(\sum_{k=1}^{s} w_k y_k) - t \geqslant 0$. Let $l_1(x), \ldots, l_s(x)$ be the linear expressions defining the s majority gates with integer coefficients. Let $p(n)$ be a polynomial such that $|l_k(x)| \leqslant p(n)$ for all $x \in \{0,1\}^n$ and all k.

Let $m = 2p(n) + 1$ and define the $m \times m$ matrix $A = (a_{ij})$ by $a_{ij} = 2^{(i-p(n)-1)(j-1)}$ for $i, j = 1, \ldots, m$. Note that A is a Vandermonde matrix with distinct rows, and hence A is invertible. Define $u = (\underbrace{0, \ldots, 0}_{p(n)}, \underbrace{1, \ldots, 1}_{p(n)+1})^{\mathsf{T}}$, and let $v = A^{-1}u$. We now define PTFs by the exponential forms $E_1(x), \ldots, E_s(x)$ given by

$$E_k(x) = \sum_{j=1}^{m} v_j 2^{(j-1)l_k(x)} \quad . \tag{1}$$

By construction we have $E_k(x) = u_{l_k(x)+p(n)+1}$. In other words, whenever $l_k(x) < 0$ we have $E_k(x) = 0$ and whenever $l_k(x) \geqslant 0$ we have $E_k(x) = 1$. We then obtain a PTF for the entire circuit by the exponential form $E(x) = (\sum_{k=1}^{s} w_k E_k(x)) - t$.

Conversely consider a PTF given by its exponential form $E(x) = \sum_{k=1}^{s} c_k 2^{l_k(x)}$, where the coefficients of $l_k(x)$ are positive integers of polynomial magnitude. Thus there is a polynomial $p(n)$ such that $0 \leqslant l_k(x) \leqslant p(n)$ for all $x \in \{0,1\}^n$ and all k. We now construct a $\mathsf{THR} \circ \mathsf{EMAJ}$ circuit is follows. For every $k \in \{1, \ldots, s\}$ and for every $j \in \{0, \ldots, p(n)\}$, we take an EMAJ gate deciding whether $l(x) = j$, and then feed the output of this gate into the output THR gate with weight $c_k 2^j$. This $\mathsf{THR} \circ \mathsf{EMAJ}$ circuit is then easily converted into a $\mathsf{THR} \circ \mathsf{MAJ}$ circuit [14]. $\square$

As a byproduct of this result we get the following interesting structural result about depth 2 threshold circuits.

Lemma 6. *Any polynomial size circuit in* $\mathsf{THR} \circ \mathsf{MAJ}$ *is equivalent to a polynomial size circuit of the same form such that all majority gates on the bottom level are monotone.*

Proof. Starting with $\mathsf{THR} \circ \mathsf{MAJ}$ circuit we first switch to a PTF in exponential form, then by multiplying the whole expression by a large enough term $2^{c\sum_i x_i}$ we make all coefficients in the exponents positive and then we switch back to $\mathsf{THR}\circ\mathsf{MAJ}$ circuit. Note that the proof of Theorem 5 gives the circuit with monotone bottom level. $\square$

The previous lemma can be actually proved directly by the same techniques as in the proof of Theorem 5. This direct proof actually gives also the analogous result for $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits. We present this in Appendix A.

3.2 Lower bounds for PTFs

The sign rank of a real matrix $A = (a_{ij})$ with nonzero entries is the minimum possible rank of a real matrix $B = (b_{ij})$ of same dimensions as A satisfying $\operatorname{sgn}(a_{ij}) = \operatorname{sgn}(b_{ij})$ for all i, j. We are interested in the sign rank of matrices defined from Boolean functions. Let $f : \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ be a Boolean function of $2n$ bits partitioned in two blocks each of n bits. We associate with f a $2^n \times 2^n$ matrix M_f, the "communication matrix", indexed by $x, y \in \{0,1\}^n$ and defined by $(M_f)_{x,y} = f(x,y)$.

Lemma 7. *Assume $f : \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ is computed by a PTF on domain $\{1,2\}^n \times \{1,2\}^n$ of length s. Then the matrix M_f has sign rank at most s.*

Proof. Consider the PTF computing f in exponential form $E(x,y) = \sum_{j=1}^{s} c_j 2^{l_j(x,y)}$. For each j, define the $2^n \times 2^n$ matrix B_j, indexed by $x, y \in \{0,1\}^n$, defined by $(B_j)_{x,y} = 2^{l_j(x,y)}$. From this definition we immediately have that the matrix $B = \sum_{j=1}^{s} c_j B_j$ is a sign representation of M_f. Now note that we can write $l_j(x,y) = l_j^{(1)}(x) + l_j^{(2)}(y)$. Hence B_j is an outer product, $B_j = b_j^{(1)} b_j^{(2)\mathsf{T}}$, where $b_j^{(1)}, b_j^{(2)} \in \{0,1\}^n$ are defined by $(b_j^{(1)})_x = 2^{l_j^{(1)}(x)}$ and $(b_j^{(2)})_y = 2^{l_j^{(2)}(y)}$. It follows that B_j is of rank at most 1, and hence $\operatorname{rank}(B) \leqslant \sum_{j=1}^{s} \operatorname{rank}(B_j) \leqslant s$. $\square$

Thus lower bounds on the sign rank of communication matrices of Boolean functions directly implies length lower bounds for PTFs on domain $\{1,2\}^n$ not depending on the degree and weights. Strong lower bounds are now known for several of Boolean functions. We mention two of particular interest. Forster [8] proved that the sign rank of the $2^n \times 2^n$ matrix corresponding to the inner product mod 2 function, $\mathrm{IP}_2(x,y)$, has sign rank $2^{\frac{n}{2}}$. Razborov and Sherstov [26] proved that the sign rank of the $2^{m^3} \times 2^{m^3}$ matrix corresponding to the Boolean function $f_m(x,y) = \bigwedge_{i=1}^{m} \bigvee_{j=1}^{m^2} (x_{ij} \wedge y_{ij})$ is $2^{\Omega(m)}$. Combining these results with Lemma 7 we have the following.

Corollary 8. *Any PTF on domain $\{1,2\}^n \times \{1,2\}^n$ computing IP_2 requires length $2^{\frac{n}{2}}$. Any PTF on domain $\{1,2\}^{m^3} \times \{1,2\}^{m^3}$ computing f_m requires length $2^{\Omega(m)}$.*

Sign rank was previously used to give the first lower bounds for $\mathsf{THR} \circ \mathsf{MAJ}$ circuits and sign rank remains the only known method for obtaining such lower bounds. Since PTFs can compute all functions computed by $\mathsf{THR} \circ \mathsf{MAJ}$ circuits already with polynomial degree by Theorem 5, Corollary 8 indicates that the lower bound technique of sign rank is applicable to more general models of computation. Showing that these models are indeed stronger would require a different lower bound method for $\mathsf{THR} \circ \mathsf{MAJ}$ circuits. Instead we will relate the question whether PTFs with no degree restrictions are more expressive than PTFs of polynomial degree to a question about threshold circuits.

For this we will need the following lemma.

Lemma 9. *For any s we have $\mathsf{THR}_s \circ \mathsf{ETHR} \subseteq \mathsf{PTF}_{1,2}(s+1, \infty) \circ \mathsf{AND}_2$.*
In particular, $\mathsf{THR} \circ \mathsf{THR} \subseteq \mathsf{PTF}_{1,2}(\infty) \circ \mathsf{AND}_2$.

Proof. From [14] we have $\mathsf{THR} \circ \mathsf{THR} = \mathsf{THR} \circ \mathsf{ETHR}$, so the second statement follows from the first. Consider a $\mathsf{THR}_s \circ \mathsf{ETHR}$ circuit C with ETHR gates $g_1, \ldots, g_s$ defined by integer linear expressions $l_1(x), \ldots, l_s(x)$, and suppose the output THR gate is given by $\operatorname{sgn}(\sum_{j=1}^{s} w_j y_j - t)$, where $w_j \neq 0$ for all j. Let $0 < m \leqslant \min_{y \in \{0,1\}^s} |\sum_{j=1}^{s} w_j y_j - t|$ be such that $m/(2s|w_j|) < 1$ for all j, and let also $m_j = \min_{\{x \in \{0,1\}^n | l_j(x) \neq 0\}} l_j(x)^2$. Define polynomials $p_1, \ldots, p_s$ by $p_j(x) = \lfloor \log_2(m/(2s|w_j|))/m_j \rfloor l_j(x)^2$. Then when $l_j(x) = 0$ we have $2^{p_j(x)} = 1$ and when $l_j(x) \neq 0$ we have

$2^{p_j(x)} \leqslant m/(2s|w_j|)$. Let $x \in \{0,1\}^n$ and define $y \in \{0,1\}^s$ be $y_j = 1$ if and only if $l_j(x) = 0$. Then we have $|y_j - 2^{p_j(x)}| \leqslant m/2s|w_j|$ and further

$$\left|\sum_{j=1}^{s} w_j y_j - \sum_{j=1}^{s} w_j 2^{p_j(x)}\right| \leqslant \frac{m}{2} \ .$$

from which it follows that the sign of the expression $\sum_{j=1}^{s} w_j 2^{p_j(x)} - t$ corresponds to the output of the circuit. On the other hand, it is easy to see that after opening the brackets in the exponents this expression corresponds to a $\mathsf{PTF}_{1,2}(s+1,\infty) \circ \mathsf{AND}_2$ circuit. $\square$

Proposition 10. $\mathsf{PTF}_{1,2}(\mathit{poly}(n)) \subsetneq \mathsf{PTF}_{1,2}(\infty)$ *unless* $\mathsf{THR} \circ \mathsf{THR} \subseteq \mathsf{THR} \circ \mathsf{MAJ} \circ \mathsf{AND}_2$.

Proof. Assume $\mathsf{PTF}_{1,2}(\mathrm{poly}(n)) = \mathsf{PTF}_{1,2}(\infty)$. Then

$$\mathsf{THR} \circ \mathsf{THR} \subseteq \mathsf{PTF}_{1,2}(\infty) \circ \mathsf{AND}_2 = \mathsf{PTF}_{1,2}(\mathrm{poly}(n)) \circ \mathsf{AND}_2 = \mathsf{THR} \circ \mathsf{MAJ} \circ \mathsf{AND}_2,$$

where the first inclusion follows from Lemma 9 and the last equality follows from Theorem 5. $\square$

We tend to consider the inclusion $\mathsf{THR} \circ \mathsf{THR} \subseteq \mathsf{THR} \circ \mathsf{MAJ} \circ \mathsf{AND}_2$ as being unlikely to hold. Note that this would also mean $\mathsf{THR} \circ \mathsf{THR} \circ \mathsf{AND} = \mathsf{THR} \circ \mathsf{MAJ} \circ \mathsf{AND}$.

3.3 Sign complexity of tensors and depth 2 threshold circuits

In this section we define the notion of sign complexity of an arbitrary order tensor, generalizing sign rank of matrices. The definition is made with the aim of capturing the notion of k-party unbounded error communication complexity given in Section 6. For simplicity we give the definition for the special case of order 3 tensors. The extension to tensors of any order k is direct.

Let $A = (a_{ijk})$ be an order 3 tensor. We say that A is a *cylinder tensor* if there is an order 2 tensor $A' = (a'_{ij})$ such either $a_{ijk} = a'_{jk}$, for all i, j, k, $a_{ijk} = a'_{ik}$ for all i, j, k, or $a_{ijk} = a'_{ij}$, for all i, j, k. In other words an order 3 tensor is a cylinder tensor if there are two indices such that every entry depends only on the value of these two indices. An order 3 tensor A is a *cylinder product* if it can be written as a Hadamard product $A_1 \odot A_2 \odot A_3$ where A_1,A_2, and A_3 are cylinder tensors. That is, $a_{ijk} = a^{(1)}_{jk} a^{(2)}_{ik} a^{(3)}_{ij}$, for all i, j, k, where $A_1 = (a^{(1)}_{jk})$,$A_2 = (a^{(2)}_{ik})$, $A_3 = (a^{(3)}_{ij})$.

The *sign complexity* of an order 3 tensor $A = (a_{ijk})$ is the minimum r such that there exist cylinder product tensors $B_1, \ldots, B_r$, with $B_\ell = (b^{(\ell)}_{ijk})$, such that $\mathrm{sgn}(a_{ijk}) = \mathrm{sgn}\left(b^{(1)}_{ijk} + \cdots + b^{(r)}_{ijk}\right)$, for all i, j, k. The *uniform sign complexity* is defined in the same way, except we require that each cylinder product tensor $B_1, \ldots, B_r$ is a Hadamard product of cylinder tensors in which all entries have the same sign.

Remark. Recalling that the rank of a matrix A is also the minimum number r such that A can be written as a sum of r matrices of rank 1, we see that the sign complexity of a tensor specializes to sign rank for matrices.

In Section 6 we generalize unbounded error communication complexity to the multi-party number-on-the-forehead setting, and show that uniform sign complexity of tensors precisely captures communication complexity in this setting, in the same way that sign rank captures the communication complexity in the two-party setting. This also means that the sign complexity essentially captures communication complexity in this setting.

Lemma 11. *The uniform sign complexity an order 3 tensor is at most 8 times its sign complexity (for a tensor of general order k the factor is 2^k).*

Proof. Simply rewrite each Hadamard product of cylinder tensors $A_1 \odot A_2 \odot A_3$ into a sum of 8 such products by expanding $(A_1^+ - A_1^-) \odot (A_2^+ - A_2^-) \odot (A_3^+ - A_3^-)$, where $A_\ell^+, A_\ell^- \geqslant 0$ with $A_\ell = A_\ell^+ - A_\ell^-$ for $\ell = 1, 2, 3$. $\square$

We are interested in the sign complexity of tensors defined from Boolean functions. Namely, for a Boolean function $f : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ we associate with f a $2^n \times 2^n \times 2^n$ tensor T_f, the "communication tensor", indexed by $x, y, z \in \{0,1\}^n$ and defined by $(T_f)_{xyz} = f(x, y, z)$.

Proposition 12. *Assume that $f : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ is computed by a $\mathsf{THR}_s \circ \mathsf{ETHR}$ circuit. Then the sign complexity of T_f is at most $s+1$.*

Proof. From Lemma 9 we have that

$$\operatorname{sgn}\left(\sum_{j=1}^{s+1} w_j 2^{p_j(x,y,z)}\right) = f(x,y,z) \ ,$$

for all $x, y, z \in \{0,1\}^n$, where $p_j(x,y,z)$ are degree 2 polynomials. Now notice that we can rewrite each p_j as $p_j(x,y,z) = p_j^{(1)}(y,z) + p_j^{(2)}(x,z) + p_j^{(3)}(x,y)$ and we can rewrite the above as

$$\operatorname{sgn}\left(\sum_{j=1}^{s+1} w_j 2^{p_j^{(1)}(y,z)} 2^{p_j^{(2)}(x,z)} 2^{p_j^{(3)}(x,y)}\right) = f(x,y,z) \ .$$

Since each of the exponential expressions $w_j 2^{p_j^{(1)}(y,z)}$, $2^{p_j^{(2)}(x,z)}$, and $2^{p_j^{(3)}(x,y)}$ define $2^n \times 2^n \times 2^n$ cylinder tensors, this shows that the sign complexity of T_f is at most $s+1$. $\square$

Using the result of [14] that $\mathsf{THR} \circ \mathsf{THR} = \mathsf{THR} \circ \mathsf{ETHR}$ this translates to a statement about $\mathsf{THR} \circ \mathsf{THR}$ circuits. Inspection of the proof of [14, Theorem 7] shows that a $\mathsf{THR} \circ \mathsf{THR}$ circuit where the output gate has fan-in s may be converted to an equivalent $\mathsf{THR} \circ \mathsf{ETHR}$ circuit where the output gate has fan-in $O(sn^3 \log n)$. From this, Proposition 12 above, and the results of Section 6 we obtain the following.

Corollary 13. *Assume that $f : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ has unbounded error 3-player NOF communication complexity c. Then every $\mathsf{THR} \circ \mathsf{ETHR}$ computing f must contain $2^c/poly(n)$ THR gates.*

Remark. The above result can be generalized to $\mathsf{THR} \circ \mathsf{THR} \circ \mathsf{AND}_k$ circuits by considering communication protocols with $2k+1$ parties. If lower bounds for such circuits could be obtained for increasing k, they could using the switching lemma be generalized to $\mathsf{THR} \circ \mathsf{THR} \circ \mathsf{AND}$ circuits, or even $\mathsf{THR} \circ \mathsf{THR} \circ \mathsf{AC}^0$ circuits (cf. [25, 13]).

4 Weights and Degree

In this section we among other things address the question of the minimal degree of a PTF computing a given Boolean function. Currently we are unable to give an upper bound on the degree required to compute a Boolean function given a bound on the length. That is, we don't know if $\mathsf{PTF}_{1,2}(\text{poly}(n), d(n)) = \mathsf{PTF}_{1,2}(\infty)$ for any function $d(n)$, and in particular we don't know if $\mathsf{PTF}_{1,2}(\text{poly}(n)) = \mathsf{PTF}_{1,2}(\infty)$, though as we have indicated we believe the latter to be false. We first show that the degree can be bounded in terms of the weight, and conversely the weight can be bounded in terms of the degree. We currently know of no method to bound the degree and weight simultaneously in terms of the length. Note that the proof of Proposition 4 together with the upper bound $n^{O(n)}$ on the weight of linear threshold functions [21] imply that for any PTF P of constant length there is another PTF P' of constant length and degree $n^{O(n)}$ computing the same function. However the precise length of P' is exponential in the length of P. We are able to avoid the exponential increase only for length 3, and we feel the proof of this gives some indications about the difficulty of the general problem. The final result of this section is an exponential degree lower bound for PTFs of constant length.

4.1 Relationship between minimal weight and minimal degree

The proof by Muroga, Toda and Takasu [22] (cf. [21]), showing that linear threshold functions needs integer weights of magnitude no more than $(n+1)^{(n+1)/2}/2^n$ can readily be adapted to PTFs on domain $\{1,2\}^n$ to give a bound on weight in terms of degree.

Proposition 14. *Suppose P is a PTF of degree d and length s. Then there is another PTF P' of degree d and length s having weight at most $s^{s/2}2^{ds}$ such that $\operatorname{sgn}(P(x)) = \operatorname{sgn}(P'(x))$ for all $x \in \{1,2\}^n$. Furthermore the set of monomials of P' is the same of P.*

Proof. Suppose $P(x) = \sum_{j=1}^{s} w_j \prod_{i=1}^{n} x_i^{a_{ij}}$. We construct system of linear inequalities in variables $w'_1, \ldots, w'_s$, with an inequality for each $x \in \{1,2\}^n$. Let $P'(x) = \sum_{j=1}^{s} w'_j \prod_{i=1}^{n} x_i^{a_{ij}}$. For each $x \in \{1,2\}^n$, if $P(x) > 0$ we take the inequality $P'(x) \geqslant 1$, and if $P(x) < 0$ we take the inequality $P'(x) \leqslant -1$. This set of inequalities has a solution (namely any sufficiently large multiple of w), and any solution of the inequalities gives a set of weights defining an equivalent PTF P'. Let w' be a solution that maximizes the number of inequalities satisfied with equality. Then it is easy to see that w' is uniquely determined by these equalities. In other words, w' is a solution to a linear system $Aw' = b$, where A is a $s \times s$ matrix with entries from $\{0, 1, \ldots, 2^d\}$ and $b \in \{-1,1\}^s$. By Cramer's rule, $w'_j = \det(A_j)/\det(A)$, where A_j is obtained from A by replacing column j by b. By Hadamard's inequality $|\det(A_j)| \leqslant s^{s/2}2^{ds}$, and we can now just clear the common denominator obtaining the claimed integer weights. □

A more complicated proof can give us a bound in another direction.

Proposition 15. *Suppose P is a PTF having integer coefficients, weight W and length s with n variables. Then there is another PTF P' of weight W and length s having degree at most*

$$(sn+1)(sn)^{sn/2}\lceil \log_2 s + \log_2 W \rceil^{sn}$$

such that $\operatorname{sgn}(P(x)) = \operatorname{sgn}(P'(x))$ for all $x \in \{1,2\}^n$. Furthermore the (multi)set of weights of P' is the same of P.

Proof. Consider P in its exponential form $\sum_{j=1}^{s} w_j 2^{l_j(x)}$, where $l_j(x) = a_{1j}x_1 + \cdots + a_{nj}x_n$, with $a_{ij} \geqslant 0$, and $w_j \in \mathbb{Z}$ with $|w_j| \leqslant W$. We construct a system of linear inequalities and equations in variables a'_{ij}, with a number of inequalities and equations produced for each $x \in \{0,1\}^n$. Let $l'_j(x) = a'_{1j}x_1 + \cdots + a'_{nj}x_n$, and P' be the PTF given by the exponential form $\sum_{j=1}^{s} w_j 2^{l'_j(x)}$. Define $c = \lceil \log_2 s + \log_2 W \rceil$.

Consider a fixed $x \in \{0,1\}^n$. We will consider the values $l_1(x), l_2(x), \ldots, l_s(x)$ in sorted order. For simplicity of notation, assume that in fact we have

$$l_1(x) \geqslant l_2(x) \geqslant \ldots \geqslant l_s(x) \ .$$

We partition this ordered sequence into blocks of consecutive values. Let $1 = i_1 \leqslant i_2 \leqslant \ldots \leqslant i_m \leqslant i_{m+1} = s+1$ be the endpoint of m such blocks, where block k consists of the indices $i_k \leqslant j < i_{k+1}$. We choose the partition in such a way that

$$l_j(x) < l_{j+1}(x) + c \ ,$$

for $i_k \leqslant j < i_{k+1} - 1$ with $1 \leqslant k \leqslant m$, and

$$l_{i_k - 1}(x) \geqslant l_{i_k}(x) + c \ ,$$

for $1 < k \leqslant m$. In other words, in each block the difference between consecutive values is less than c, whereas the difference between the endpoints values of different blocks is at least c.

We form an equality for consecutive values from the same block and an inequality otherwise. For $i_k \leqslant j < i_{k+1} - 1$ we take the equality

$$l'_j(x) - l'_{j+1}(x) = l_j(x) - l_{j+1}(x) \ ,$$

and for $1 < k \leqslant m$ we take the inequality

$$l'_{i_k - 1}(x) - l'_{i_k}(x) \geqslant c \ .$$

In total this gives a system of $2^n(s-1)$ equations and inequalities in sn variables. We add to this positivity constraints, $a'_{ij} \geqslant 0$. All coefficients are either -1 or 1, and the right-hand sides are all integers in the range $\{0, 1, \ldots, c\}$. Let A be the $(2^n(s-1) + sn) \times sn$ matrix of coefficients and let $b \in \mathbb{Z}^{2^n(s-1)+sn}$ be the right-hand sides. We know already that the system of equations and inequalities has an integer solution, namely that given by P. Furthermore, any integer solution gives a set of coefficients to the linear forms defining an equivalent PTF. To see this, note the inequalities guarantee that for any block, when its contribution is nonzero it is guaranteed to dominate the contribution of all following blocks. We can now use a bound of von zur Gathen and Sieveking [31] to estimate an integer solution with small entries. For this let M be the maximum magnitude of a $m \times m$ minor of the matrix $\begin{bmatrix} A & b \end{bmatrix}$, where $m \leqslant sn$. Then the system has an integer solution a' with $a'_{ij} \leqslant (sn+1)M$. By Hadamard's inequality, $M \leqslant (sn)^{sn/2} c^{sn}$, and hence the entries of a' are bounded by $(sn+1)(sn)^{sn/2} c^{sn}$. $\square$

From Propositions 14 and 15 we for example have that for a PTF of length poly(n), if the degree is at most $2^{\text{poly}(n)}$ then the weight can be assumed to be $2^{2^{\text{poly}(n)}}$, and vise versa.

4.2 Exponential upper bound for the degree of size 3 threshold gates.

Proposition 16. *Any PTF of length 3 over the domain $\{1,2\}^n$ is equivalent to a length 3 PTF with degree $n^{O(n)}$.*

Proof. Consider a PTF of length 3 in its exponential form

$$E(x) = w_1 2^{l_1(x)} + w_2 2^{l_2(x)} + w_3 2^{l_3(x)}$$

When all w_i's have the same sign, the Boolean function computed is constant, so assume this is not the case. Thus either 1 or 2 of the w_i's are positive. We consider the case when 2 are positive. The other case is analogous. We can thus rewrite $E(x)$ in the form

$$E(x) = 2^{A(x)+\alpha} + 2^{B(x)+\beta} - 2^{C(x)+\gamma}$$

where the $A(x)$,$B(x)$, and $C(x)$ are affine linear forms with positive integer coefficients, and $0 \leqslant \alpha, \beta, \gamma < 1$. In fact, by dividing by 2^γ, we may assume $\gamma = 0$.

For each $x \in \{0,1\}^n$ we produce a number of linear equations and inequalities. We have 3 cases. In case $A(x) \geqslant C(x)$ or $B(x) \geqslant C(x)$ we take that inequality. Note that $A(x) \geqslant C(x)$ or $B(x) \geqslant C(x)$ alone implies that $E(x) > 0$.

In case both $A(x) \leqslant C(x) - 2$ and $B(x) \leqslant C(x) - 2$ we take both these inequalities. Note that $A(x) \leqslant C(x) - 2$ and $B(x) \leqslant C(x) - 2$ together implies that $E(x) < 0$.

The last case is that either $A(x) = C(x) - 1$ or $B(x) = C(x) - 1$. Consider first the subcase of $A(x) = C(x) - 1$. We may then rewrite

$$E(x) = 2^{B(x)+\beta} - 2^{C(x)}(1 - 2^{\alpha-1})$$

Thus $E(x) \geqslant 0$ if and only if

$$2^{B(x)} \geqslant 2^{C(x)}(1 - 2^{\alpha-1})2^{-\beta}$$

which holds if and only if

$$B(x) \geqslant C(x) + \log_2(1 - 2^{\alpha-1}) - \beta \ .$$

Since the coefficients of $B(x)$ and $C(x)$ are integer, this holds if and only if

$$B(x) \geqslant C(x) + \lceil \log_2(1 - 2^{\alpha-1}) - \beta \rceil \ .$$

Note that $\lceil \log_2(1 - 2^{\alpha-1}) - \beta \rceil$ might be of large magnitude, and thus we do not want to have such a number in our system. Instead we introduce a new integer variable k_A and add to our system the equality $A(x) = C(x) - 1$ and the inequalities $B(x) \geqslant C(x) - k_A$ and $k_A \geqslant 1$.

The sub-case of $B(x) = C(x) - 1$ is similar, and $E(x) \geqslant 0$ if and only if

$$A(x) \geqslant C(x) + \lceil \log_2(1 - 2^{\beta-1}) - \alpha \rceil \ .$$

We add here in a similar way the equality $B(x) = C(x) - 1$ and the inequalities $A(x) \geqslant C(x) - k_B$ and $k_B \geqslant 1$.

Finally we add positivity constraints to all coefficients of variables in exponents. In this way we end up with a system of at most $2 \cdot 2^n + 3n + 2$ equations and inequalities in $3(n+1) + 2 = 3n + 5$ variables. Similarly as in the proof of Proposition 15 we may find an integer solution with entries of bounded magnitude. The bound we can achieve here by the result of von zur Gathen and Sieveking [31] is $(3n+6)(3n+5)^{(3n+5)/2}2^{3n+5} = n^{O(n)}$.

Given such an integer solution $A'(x), B'(x), C'(x), k'_A, k'_B$ we now wish to find α' and β' solving the equations

$$\begin{aligned} -k_A &= \lceil \log_2(1 - 2^{\alpha'-1}) - \beta' \rceil \\ -k_B &= \lceil \log_2(1 - 2^{\beta'-1}) - \alpha' \rceil \end{aligned} \tag{2}$$

or equivalently

$$\begin{aligned} 2^{-k_A-1} &< (1 - 2^{\alpha'-1})2^{-\beta'} \leqslant 2^{-k_A} \\ 2^{-k_B-1} &< (1 - 2^{\beta'-1})2^{-\alpha'} \leqslant 2^{-k_B} \end{aligned} \tag{3}$$

This can be done as follows. First, let $\log_2(3)-1 \leqslant \alpha_0, \beta_0 < 1$ be given by $\alpha_0 = \log_2(1-2^{-k_A-1})+1$ and $\beta_0 = \log_2(1-2^{-k_B-1})+1$. Then we have $1-2^{\alpha_0-1} = 2^{-k_A-1}$ and $1-2^{\beta_0-1} = 2^{-k_B-1}$. Choose $0 < \epsilon < \min(1-\alpha_0, 1-\beta_0)$, and let $\alpha' = \log_2(1-2^{-k_A-\epsilon})+1$ and $\beta' = \log_2(1-2^{-k_B-\epsilon})+1$. Then we have $1-2^{\alpha'-1} = 2^{-k_A-\epsilon}$ and $1-2^{\beta'-1} = 2^{-k_B-\epsilon}$. Note also that $\alpha' < \alpha_0$ and $\beta' < \beta_0$ which implies $0 < \alpha', \beta' < 1-\epsilon$. Combining this we see that α' and β' satisfies the equations (3). $\square$

Remark. One may in fact solve the equations (2) without the ceiling operation except when exactly one of k_A and k_B is 1. In case $k_A = k_B = 1$ we have the solution $\alpha' = \beta' = 0$. In case $k_A, k_B > 1$ we have the solution

$$\begin{aligned} \alpha' &= \log_2\left(\frac{2 - 2^{2-k_A}}{1 - 2^{2-k_A-k_B}}\right) \\ \beta' &= \log_2\left(\frac{2 - 2^{2-k_B}}{1 - 2^{2-k_A-k_B}}\right) \quad . \end{aligned}$$

4.3 Exponential lower bound for the degree of constant-size threshold gates

The ODD-MAX-BIT function (abbreviated here by OMB) was defined by Beigel [3] as $\mathrm{OMB}(x_1, \ldots, x_n) = 1$ if and only if $(\max\{i \mid x_i = 1\} \mod 2) = 1$. In this section we prove a degree lower bound of the form $2^{\Omega(n^\epsilon)}$ for any PTF of constant length s computing OMB, where ϵ depends on s.

By Proposition 4 we have $\mathsf{PTF}_{1,2}(O(1), \infty) = \mathsf{ANY}_{O(1)} \circ \mathsf{THR}$. Inspecting the proof we can observe that a PTF of individual degree at most d and length k is in fact turned into a Boolean combination of $2^{O(k^2)}$ threshold functions, where these threshold functions have integer weights of magnitude at most d. We can rewrite this Boolean combination of linear threshold functions by a constant size DNF, i.e. we obtain an $\mathsf{OR}_{2^{2^{O(k^2)}}} \circ \mathsf{AND}_{2^{O(k^2)}} \circ \mathsf{THR}$ circuit where all threshold gates are computed with integer weights of magnitude at most d. We can thus just give weight lower bounds for $\mathsf{OR}_{O(1)} \circ \mathsf{AND}_{O(1)} \circ \mathsf{THR}$ circuits.

Theorem 17. *Any circuit in the class* $\mathsf{OR}_k \circ \mathsf{AND}_l \circ \mathsf{THR}$ *computing* OMB *function on n variables require weights of size* $2^{\Omega(n^{1/kl})}$.

Proof. In this proof we will identify inputs $x \in \{0,1\}^n$ to the function OMB with the subsets $X \subseteq [n]$ given by $X = \{i \mid x_i = 1\}$, and vice versa.

Define $m = n^{1/kl}$. In the course of the proof we will construct a sequence of disjoint subsets $A_0, A_1, \ldots, A_m \subseteq [n]$ such that $a_0 < \ldots < a_m$ whenever $a_0 \in A_0, \ldots, a_m \in A_m$. In addition to these we will identify a particular THR gate in the circuit such that on inputs of the form

$X_i = A_0 \cup A_i \cup A_{i-1} \cup A_{i-3} \cup A_{i-5} \cup \ldots$, the THR-gate is 1 for odd i and is 0 for even i. With this sequence in hand we will be able to prove the weight lower bound in a standard way.

Break the set $[n]$ into m equal-sized blocks $B_1, \ldots, B_m$ of consecutive elements. Let $A_0 = \emptyset$. Consider the first AND gate in the circuit and the first of its THR gates. Let us denote it by t_1. We shall attempt to find the sequence of sets $A_1, \ldots, A_m$ for t_1. We shall choose these one by one, $i = 1, \ldots, m$, until we either succeed or fail for some i. When i is odd we wish to find $A_i \subseteq B_i$ such that $\mathrm{OMB}(X_i) = 1$ and $t_1(X_i) = 1$. When i is even we wish to find $A_i \subseteq B_i$ such that $\mathrm{OMB}(X_i) = 0$ and $t_1(X_i) = 0$. If on each step we can find such A_i we are done.

Suppose that on some step i we fail to find A_i. We shall then add to A_0 the sets $A_{i-1}, A_{i-3}, A_{i-5}, \ldots$ and repeat the whole argument restricted to block B_i, which we divide into m new blocks. The point is that before doing this we may simplify the circuit. Namely, when i is odd, we have for all $A_i \subseteq B_i$ for which $\mathrm{OMB}(X_i) = 1$ that $t_1(X_i) = 0$. Thus we may eliminate t_1 together with its parent AND gate from the circuit when restricting input to B_i and the bits in the set A_0. Since the circuit actually computes the OMB function there will still be other $\mathsf{AND} \circ \mathsf{THR}$ sub-circuits remaining. Similarly, when i is even, we have for all $A_i \subseteq B_i$ for which $\mathrm{OMB}(X_i) = 0$ that $t_1(X_i) = 1$. Thus we may eliminate t_1 from the circuit when restricting input to B_i and the bits in the set A_0. Since the circuit actually computes the OMB function there will still be another THR gate in the circuit that is a sibling of t_1.

We need to proceed to the sub-block at most $kl-1$ times since each time we reduce the number of threshold gates by at least 1. It is not hard to see that by the choice of parameters we finally get the sequence of sets as desired. Let us now consider the threshold gate corresponding to this sequence. Suppose it is given by the inequality $\sum_j w_j x_j > t$, where the variables range only over the proper block, and all variables outside are fixed. Define $W_{A_i} = \sum_{j \in A_i} w_j$, where $i = 1, \ldots, m$. Then from the inputs $X_1, \ldots, X_m$ we get the sequence of inequalities

$$W_{A_1} > t, \quad W_{A_2} + W_{A_1} < t, \quad W_{A_3} + W_{A_2} > t,$$
$$W_{A_4} + W_{A_3} + W_{A_1} < t, \quad \ldots \ ,$$

and from these inequalities the desired lower bound $2^{\Omega(n^{1/kl})}$ follows easily. □

5 Max-plus PTFs

In this section we introduce one more complexity class which is a "limit" class of our PTF classes in the flavor of max-plus algebra. It turns out this class is related to the Boolean circuits of the form $\mathsf{AC}^0 \circ \mathsf{THR}$.

Consider some integer polynomial $p(x) \in \mathbb{Z}[x]$ and consider some domain $\{1, b\}$. This polynomial computes some function in $\mathsf{PTF}_{1,b}(l, d)$, where l is the length of p and d is the degree of p. We start with some preliminary work to make p be in "general position". First we can assume that $p(x) \neq 0$ for all $x \in \{1, b\}^n$. For this just consider $2p(x) + 1$ instead of $p(x)$. Next by varying the coefficients of p by small rational numbers and multiplying the whole polynomial by large enough integer to make the coefficients integer again we can assume that for each $x \in \{1, b\}^n$ none of the two monomials are equal.

Now consider the exponential form of p

$$P(x) = \sum_{i=1}^{l_1} b^{L_i(x)} - \sum_{j=1}^{l_2} b^{M_j(x)}, \quad L_i(x) = \sum_{k=1}^{n} v_{ki} x_k + v_{0i}, \quad M_j(x) = \sum_{k=1}^{n} u_{kj} x_k + u_{0j}.$$

Note that here we put the coefficients of the polynomial to the exponents.

Suppose now that we start to increase b (for convenience assume that b is positive). Then the function computed by p might start changing. Since all linear forms are different for all inputs $x \in \{0,1\}^n$ it is easy to see that for large enough b and for any x the sign of $P(x)$ is equal to the sign of the largest monomial. That is, $P(x) \geqslant 0$ if and only if

$$\max_{i=1,\ldots,l_1} (L_i(x)) \geqslant \max_{j=1,\ldots,l_2} (M_j(x)) \tag{4}$$

Definition 1. By max-plus PTFs we denote expressions of the form (4). The length of the PTF (4) is $l_1 + l_2$, the degree is the maximal sum of absolute values of all coefficients of $L_1, \ldots, L_{l_1}$ and $M_1, \ldots, M_{l_2}$ except the constant term. Note that max-plus PTFs are essentially just polynomial inequalities in the max-plus algebra.

We let $\mathsf{mpPTF}(l(n), d(n))$ to be the set of Boolean functions computable by max-plus PTFs of length $l(n)$ and degree $d(n)$. As before we denote by $\mathsf{mpPTF}(d(n))$ the set of Boolean functions computable by max-plus PTFs of length $\mathrm{poly}(n)$ and degree $d(n)$. Here $d(n)$ might be ∞.

It turns out that the class $\mathsf{mpPTF}(\infty)$ is related to the circuits of the form $\mathsf{AC}^0 \circ \mathsf{THR}$. More specifically, we can consider the hierarchy of polynomial size circuits of the form $\Sigma_d \circ \mathsf{THR}$ and $\Pi_d \circ \mathsf{THR}$, where Σ_d and Π_d are the classes of depth d circuits of interchanging layers of AND and OR gates of unbounded fan-in, Σ_d having OR gate at the top and Π_d having AND gate at the top. In this hierarchy lower bounds are known only for the case $d = 1$, that is for the circuits of the form $\mathsf{AND} \circ \mathsf{THR}$ and $\mathsf{OR} \circ \mathsf{THR}$ (the standard proof of the lower bound on the size of DNF for the parity function works). For the circuits of the form $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{THR}$ and $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{THR}$ no superpolynomial lower bounds for explicit functions are known. It turns out that the class $\mathsf{mpPTF}(\infty)$ lies between these two levels of the hierarchy.

First we note some trivial property of max-plus PTF classes.

Proposition 18. *$\mathsf{mpPTF}(\infty)$ and $\mathsf{mpPTF}(poly(n))$ are closed under negation.*

Proof. This follows directly from the definition of max-plus PTFs and the fact that any max-plus PTF can be easily reconstructed in such a way that for all $x \in \{0,1\}^n$ the equality does not hold in (4). $\square$

Lemma 19. $\mathsf{AND} \circ \mathsf{THR}, \mathsf{OR} \circ \mathsf{THR} \subseteq \mathsf{mpPTF}(\infty)$*, and*
$\mathsf{mpPTF}(\infty) \subseteq \mathsf{AND} \circ \mathsf{OR} \circ \mathsf{THR}, \mathsf{OR} \circ \mathsf{AND} \circ \mathsf{THR}$

Proof. Assume that we have a circuit of the form $\mathsf{OR} \circ \mathsf{THR}$ and $l_1 \geqslant 0, \ldots, l_k \geqslant 0$ are the threshold gates on the bottom level. Then it is clear that the max-plus PTF

$$\max(l_1, \ldots, l_k) \geqslant 0$$

computes the same function. For the class $\mathsf{AND} \circ \mathsf{THR}$ an analogous proof works. Alternatively, the inclusion follows from Proposition 18.

Now consider the max-plus PTF (4). It is true if and only if there is $i \in [l_1]$ such that for all $j \in [l_2]$ it is true that $L_i(x) \geqslant M_j(x)$. That is, this PTF is equivalent to the formula

$$\bigvee_{i=1}^{l_1} \bigwedge_{j=1}^{l_2} (L_i(x) \geqslant M_j(x)),$$

which is clearly in the class $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{THR}$. Again, the inclusion in $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{THR}$ follows from Proposition 18. $\square$

Now we show that min-plus PTFs are not stronger than usual PTFs.

Lemma 20. *For all $b > 1$ there is a constant C such that* $\mathsf{mpPTF}(l(n), d(n)) \subseteq \mathsf{PTF}_{1,b}(l(n), C \cdot d(n) \log l(n))$.

Proof. Suppose we are given a max-plus PTF (4). First note that we can change the linear forms in such a way that for each $x \in \{0,1\}^n$ equality does not hold. For this we can just for each i consider $2L_i + 1$ instead of L_i and for each j consider $2M_j$ instead of M_j. This increases the degree at most by a constant factor.

After that we can consider the PTF P in exponential form over $\{1, b\}$:

$$P(x) = \sum_{i=1}^{l_1} b^{C \log l(n) L_i(x)} - \sum_{j=1}^{l_2} b^{C \log l(n) M_j(x)} \quad .$$

It is easy to see that if C is such that $b^{C \log l(n)} > l(n)$ then for each $x \in \{0,1\}^n$ the absolute value of the largest monomial in $P(x)$ is greater than the absolute value of the sum of all monomials of the opposite sign. Thus the sign of $P(x)$ is equal to the sign of the largest monomial in it and $P(x) \geqslant 0$ if and only if (4) is true. $\square$

From the lemma above and Corollary 8 we immediately obtain the following corollary.

Corollary 21. *Any max-plus PTF computing IP_2 requires length $2^{\frac{n}{2}}$. Any max-plus PTF computing f_m requires length $2^{\Omega(m)}$.*

Thus $\mathsf{mpPTF}(\infty)$ is an intermediate class in the $\mathsf{AC}^0 \circ \mathsf{THR}$ hierarchy for which we do know a lower bound. On the other hand the class is still rather strong. We show that it contains some functions which are complicated for other complexity classes.

Lemma 22. $\mathsf{PARITY} \in \mathsf{mpPTF}(\mathit{poly}(n))$, $\mathsf{OMB} \circ \mathsf{THR} \subseteq \mathsf{mpPTF}(\infty)$.

Proof. The fact that the parity is computable by mpPTF follows already from the analysis of Basu et al. [1] (Theorem 4.1). For the sake of completeness we sketch the construction below. Define $X = x_1 + \ldots + x_n$ and let

$$L_i(x) = l_i(X) = i(X - i) + C_i$$

for $i = 0, \ldots n$, where $C_0 = 0$ and other C_i are defined recursively by $C_i = l_{i-1}(i) + 1/2$. Linear forms L_i are arranged in such a way that on the input with i ones L_i is the largest of the forms being by $1/2$ greater than the previous linear form. Then if we put all L_i with odd i on the left-hand side of (4) and all L_i with even i on the right-hand side, it is not hard to see that the resulting max-plus PTF computes PARITY.

Suppose now we have a function $f \in \mathsf{OMB} \circ \mathsf{THR}$ and $l_1 \geqslant 0, \ldots, l_k \geqslant 0$ are the threshold gates on the bottom level. That is $f(x) = 1$ if and only if the largest i such that $l_i(x) \geqslant 0$ is odd (if there is no such i the function is 0). First as usual we can assume that all l_i never evaluate to zero. Let

$$B = \max_{i=1,\ldots,k,\ x \in \{0,1\}^n} |l_i(x)|.$$

Consider $l'_i(x) = B^{i-1} l_i(x)$, a then the max-plus PTF for which on the left-hand side we have all l'_i for odd i and on the right-hand side we have all l'_i for even i and in addition a constant zero linear form. It is not hard to see that for any x the positive linear form with the greatest number is greater than all other linear forms and thus we have constructed a max-plus PTF for the function f. □

We note that Buhrman et al. [5] proved that the function OMB $\circ \mathsf{AND}_2$ is not in the complexity class $\mathsf{MAJ} \circ \mathsf{MAJ}$. Thus we have the following corollary.

$$\mathsf{mpPTF}(\infty) \not\subseteq \mathsf{MAJ} \circ \mathsf{MAJ}.$$

Besides max-plus PTFs which are just polynomial inequalities in the max-plus algebra, we can consider systems of max-plus polynomial inequalities. Clearly they can be expressed by AND's of max-plus PTFs. Using the results above we can precisely characterize the power of the systems of max-plus PTFs in terms of $\mathsf{AC}^0 \circ \mathsf{THR}$ circuits.

Corollary 23. *The functions computed by systems of max-plus PTFs are exactly those computed by* $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{THR}$ *circuits.*

Proof. Functions computed by systems of max-plus PTFs are also computed by $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{THR}$ circuits, since by Lemma 19 each function computed by a PTF is computed by a $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{THR}$ circuit, and the system is just an AND of its PTFs. On the other hand, systems of max-plus PTFs can compute every function computed by $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{THR}$ circuits since, again by Lemma 19 we can represent $\mathsf{OR} \circ \mathsf{THR}$ circuits by PTFs and then we can simulate the AND gate by a system. □

6 Sign complexity of tensors and multi-party NOF communication complexity

We show here how the notion of unbounded error two-party communication complexity of Paturi and Simon [24] and its matrix characterization can be extended from the two-party setting to the multi-party number-on-the-forehead setting introduced by Chandra, Furst, and Lipton [7]. Once a suitable definition of one-way protocols together with the definition of sign complexity from Section 3.3 is in place, the extension is rather straightforward, following Paturi and Simon [24]. For simplicity, as was similarly done in Section 3.3, we give the definitions for the special case of three parties. The extension to any number of parties k is direct.

In the three-party communication model it is the goal of three players A, B, and C to compute a Boolean function $f(x, y, z)$. The input to f is distributed between the players such that player A receives x, player B receives y and player C receives z. Here we are in the number-on-the-forehead model, meaning that each player can see all inputs except his own. Also we are in the unbounded error model, meaning that each player has access to *private*[3] randomness. The players can communicate by broadcasting messages. The communication should proceed according to a fixed protocol P. The protocol specifies for each sequence of bits sent by the players so far (and depending only on these) the following information: Whether the protocol is over, and in that case the output $P(x, y, z)$ of the protocol. In case the protocol is not over, which player is to send the next bit of communication as well as the distribution of this bit. This distribution depends on

[3]In case of public randomness the model becomes trivial [24].

the inputs visible to the player, the sequence of bits sent by the players so far, and it is sampled using private randomness to give the actual bit sent by the player. The protocol P computes the Boolean function f if $\Pr[P(x,y,z) = f(x,y,z)] > \frac{1}{2}$, where the probability is taken over the internal randomness of the players. A transcript of the protocol P is the concatenation of messages sent during an execution of the protocol. The communication complexity $c(P)$ of the protocol P is the length of the largest possible transcript for any possible input (x,y,z).

By a one-way protocol P' we mean the following restricted type of protocols. At first, player A sends a message. His possible messages $M_{P'}$ are divided into two sets $M^0_{P'}$ and $M^1_{P'}$, intuitively describing the intended output by player A. Suppose the message of player A is $m \in M^b_{P'}$. Then players B and C each send a single bit independently of each other, intuitively indicating whether they agree with the intended output. If both these bits are 1, the output $P'(x,y,z)$ of the protocol is b, and otherwise the output of the protocol is $1-b$. The communication complexity $c(P')$ of the one-way protocol P' is the largest possible length of the message sent by player A.

Analogously to two-party case it turns out that in the unbounded error setting, one way protocols very precisely captures the power of general protocols.

Lemma 24. *For any protocol P there is a one way protocol P' computing the same function and such that $c(P') \leqslant c(P) + 1$.*

Proof. Let T_P denote the possible transcripts of the protocol P. Divide this set of transcripts into the transcripts T^0_P with output 0 and the transcripts T^1_P with output 1. For $\alpha \in T_P$ we denote by $\Pi_A(P,\alpha,y,z)$ the probability that α is consistent with the messages of A during a run of the protocol on inputs y,z. We define Π_B and Π_C analogously. It is then not hard to see that $\Pi_A(P,\alpha,y,z)\Pi_B(P,\alpha,x,z)\Pi_C(P,\alpha,x,y)$ is exactly the probability with which the transcript α is produced by the protocol on input (x,y,z). For $b \in \{0,1\}$, define $d^b_{y,z} = \sum_{\alpha \in T^b_P} \Pi_A(P,\alpha,y,z)$ and let $d = \max_{y,z} d^1_{y,z}$.

We are now ready to construct the one-way protocol P'. The set of messages of player A consists of T_P together with a new message $\gamma \notin T_P$. That is $M_{P'} = T_P \cup \{\gamma\}$. We let $M^0_{P'} = T^0_P$ and $M^1_{P'} = T^1_P \cup \{\gamma\}$.

For $\beta \in M_{P'}$ we denote by $\Pi_A(P',\beta,y,z)$ the probability that player A sends the message β. We denote by $\Pi_B(P',\beta,x,z)$ and $\Pi_C(P',\beta,x,y)$ the probability that player B and C send the bit 1 upon seeing message β. We define these probabilities as follows:

$$\begin{aligned}
\Pi_A(P',\alpha,y,z) &= \frac{1}{2d}\Pi_A(P,\alpha,y,z) && \text{for } \alpha \in T^1_P\\
\Pi_A(P',\gamma,y,z) &= \frac{1}{2}\left(1 - \frac{d^1_{y,z}}{d}\right) &&\\
\Pi_A(P',\alpha,y,z) &= \left(\frac{1}{2d^0_x}\right)\Pi_A(P,\alpha,y,z) && \text{for } \alpha \in T^0_P\\
\Pi_B(P',\alpha,x,z) &= \Pi_B(P,\alpha,x,z) && \text{for } \alpha \in T^1_P\\
\Pi_B(P',\gamma,x,z) &= 0 &&\\
\Pi_B(P',\alpha,x,z) &= \frac{1}{2d} && \text{for } \alpha \in T^0_P\\
\Pi_C(P',\alpha,x,y) &= \Pi_C(P,\alpha,x,y) && \text{for } \alpha \in T^1_P\\
\Pi_C(P',\gamma,x,y) &= 0 &&
\end{aligned}$$

$$\Pi_C(P', \alpha, x, y) = 1 \qquad \text{for } \alpha \in T_P^0$$

With these in place we can now compute the probability that P' gives as output 1 in terms of the probability that P gives as output 1.

$$\begin{aligned}
\Pr[P'(x,y,z)=1] = &\sum_{\beta \in M_{P'}^1} \Pi_A(P', \beta, y, z)\Pi_B(P', \beta, x, z)\Pi_C(P', \beta, x, y) \\
&+ \sum_{\beta \in M_{P'}^0} \Pi_A(P', \beta, y, z)\left(1 - \Pi_B(P', \beta, x, z)\Pi_C(P', \beta, x, y)\right) \\
= &\sum_{\alpha \in T_P^1} \frac{1}{2d}\Pi_A(P, \alpha, y, z)\Pi_B(P, \alpha, x, z)\Pi_C(P, \alpha, x, y) \\
&+ \sum_{\alpha \in T_P^0} \left(\frac{1}{2d_{y,z}^0}\right)\Pi_A(P, \alpha, y, z)\left(1 - \frac{1}{2d}\right) \\
= &\frac{\Pr[P(x,y,z)=1]}{2d} + \frac{1}{2}\left(1 - \frac{1}{2d}\right) \\
= &\frac{1}{2} + \frac{\Pr[P(x,y,z)=1] - \frac{1}{2}}{2d}
\end{aligned}$$

The complexity of P' is at most by one larger than the complexity of P as claimed, since the number of possible messages is one larger than the number of possible transcripts. $\square$

Next we characterize the one-way communication complexity of a Boolean function f in terms of the uniform sign complexity of the communication tensor T_f as defined in Section 3.3.

Proposition 25. *Let $f : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ be a non-constant Boolean function. Let s be the uniform sign complexity of the associated communication tensor T_f and let c be the one-way unbounded error communication complexity of f. Then*

$$\lceil \log_2 s \rceil \leqslant c \leqslant \lceil \log_2(s+1) \rceil.$$

Proof. Assume first that we are given a one way communication protocol P of communication complexity c. With each message $\alpha \in M_P$ we associate a cylinder product tensor A^α in the following way. Define non-negative cylinder tensors $A_1^\alpha(x,y,z) = \Pi_A(P, \alpha, y, z)$, $A_2^\alpha(x,y,z) = \Pi_B(P, \alpha, x, z)$, $A_3^\alpha(x,y,z) = \Pi_C(P, \alpha, x, y)$ and let $A^\alpha(x,y,z) = A_1^\alpha(x,y,z)A_2^\alpha(x,y,z)A_3^\alpha(x,y,z)$. Then it is not hard to see that

$$T_f = \operatorname{sgn}\left(\sum_{\alpha \in M_P^1} A^\alpha - \frac{1}{2}\mathbb{1}\right),$$

where by $\mathbb{1}$ we denote the tensor in which all entries are 1. Since the messages of A form a prefix free set from $\{0,1\}^c$ the set M_P consists of at most 2^c messages. Furthermore, since f is non-constant there should be at least one message in M_P^0. It follows that T_f is the sum and difference of at most 2^c non-negative cylinder tensors, and hence $s \leqslant 2^c$.

In the other direction, suppose we are given a representation

$$T_f = \mathrm{sgn}((-1)^{t_1} B^1 + \ldots + (-1)^{t_s} B^s),$$

where $B^l(xyz) = D_1^l(y,z) D_2^l(x,z) D_3^l(x,y)$, $t_l \in \{0,1\}$, and D_1^l,D_2^l, and D_3^l are non-negative order 2 tensors.

Observe that we can assume that for each (x,y,z)

$$\sum_{l=1}^{s} D_2^l(x,z) = \sum_{l=1}^{s} D_3^l(x,y) = 1.$$

Indeed, if this is not true for one of the sums we can multiply all its terms by the same number, and this does not change the sign. Next define $E^0(y,z) = \sum_{l\colon t_l=0} D_1^l(y,z)$ and $E^1(y,z) = \sum_{l\colon t_l=1} D_1^l(y,z)$. By the same observation as above we can further assume that $\max\{E^0(y,z), E^1(y,z)\} = 1/2$.

Now we are ready to construct the protocol. The messages of player A will be chosen from the set $\{0,1,\ldots,s\}$. On input (y,z) Player A sends message $l > 0$ with probability $D_1^l(y,z)$, and send message 0 with the remaining probability $1 - E^0(y,z) - E^1(y,z)$. If $E^0(y,z) < 1/2$ we let $t_0 = 0$, otherwise we let $t_0 = 1$. The intended output of message l is $1 - t_l$.

Upon receiving the message $l > 0$, player B sends 1 with probability $D_2^l(x,z)$ and player C sends 1 with probability $D_3^l(x,y)$. Upon receiving message 0, player B and player C send 1 with probability 1.

By direct computation it is not hard to see that the probability of output 1 is equal to $1/2 + \sum_l (-1)^{t_l} B^l(x,y,z)$. Since the protocol requires A to send at most $s+1$ different messages the complexity of the protocol is at most $\lceil \log_2(s+1) \rceil$. □

7 Relations between the domains

Though we can not completely solve the problem of the relations between the domains, we still have some partial result in this direction. In particular we show that the domains $\{1,2\}$ and $\{1,-2\}$ are essentially equivalent. We start with simple observations.

Proposition 26. *For all $a,b \in \mathbb{R}$ such that $a,b \neq 0$ and $|a| \neq |b|$ we have that* $\mathsf{PTF}_{a,b}(poly(n)) = \mathsf{PTF}_{1,2}(poly(n))$.

Proof. This is a direct corollary of Theorem 5 (note that the proof of this theorem works for all specified domains). □

Proposition 27. *For all $a,b \in \mathbb{R}$ such that $a,b \neq 0$ and $|a| \neq |b|$ we have that* $\mathsf{PTF}_{a,b}(l(n),d(n)) = \mathsf{PTF}_{1,\frac{b}{a}}(l(n),d(n))$

Proof. Just note that if $y \in \{a,b\}$ then $z = y/a$ is the corresponding variable over $\{1, \frac{b}{a}\}$. □

Lemma 28. *For all $a,b \in \mathbb{R}$ such that $a,b \neq 0$ and $|a| \neq |b|$ and for any natural number k we have* $\mathsf{PTF}_{a,b}(\infty) = \mathsf{PTF}_{a^k,b^k}(\infty)$.

Proof. Note that $\mathsf{PTF}_{a^k,b^k}(\infty) \subseteq \mathsf{PTF}_{a,b}(\infty)$ since everything that can be computed over $\{a^k, b^k\}$ can be computed over $\{a, b\}$. Indeed, just consider the same polynomial but increase all powers by the factor of k.

To prove the other direction first note that by Proposition 27 we can safely assume that $a = 1$.

Next noting that for $y \in \{1, b\}$ defining $x \in \{0, 1\}$ to be $x = \log_{|b|}(|y|)$ we have that $y = b^x$. So we can consider exponential forms analogously to the case of $\{1, 2\}$ domain.

Consider some function f from $\mathsf{PTF}_{1,b}$ and consider the polynomial p computing it over $\{1, b\}$. We will show how to construct the polynomial q such that $p(x) = q(x)$ for all $x \in \{1, b\}^n$, all degrees in all monomials of q are divisible by k and finally the length of q are greater than the length of p by at most polynomial factor. For this consider one of the monomials C of p in the exponential form:

$$C = b^{a_1x_1+a_2x_2+\ldots+a_nx_n}.$$

Here all coefficients a_i are integer. Denote by r_i the residue of a_i modulo k and let $q_i = (a_i - r_i)/k$. Then

$$C = b^{r_1x_1+r_2x_2+\ldots+r_nx_n} \cdot \left(b^k\right)^{q_1x_1+q_2x_2+\ldots+q_nx_n}. \tag{5}$$

Denote the first multiplier by D. Note that the degree of D is polynomially bounded, more precisely it is at most kn. By repeating the argument of the first part of the proof of Theorem 5 we can construct a polynomial $p_C \in \mathbb{Z}[x]$ of polynomial size and polynomial degree such that if we consider it as a PTF over $\{1, b^k\}$ and consider it's exponential form p'_C, then $p'_C(x) = D(x)$ for all $x \in \{0, 1\}^n$. More precisely, for this we construct the polynomial in the same way as in (1) but we let the base of the exponent be b^k in the matrix A and we let u be the vector of possible values of D.

Substituting p'_C instead of D in (5) we obtain a representation of C over $\{1, b^k\}$. Repeating this argument for each monomial we prove the lemma. □

From this in particular we can deduce that

$$\mathsf{PTF}_{1,2}(\infty) = \mathsf{PTF}_{1,2^2}(\infty) = \mathsf{PTF}_{1,(-2)^2}(\infty) = \mathsf{PTF}_{1,-2}(\infty).$$

References

[1] S. Basu, N. Bhatnagar, P. Gopalan, and R. J. Lipton. Polynomials that sign represent parity and descartes' rule of signs. *Computational Complexity*, 17(3):377–406, 2008.

[2] R. Beigel. The polynomial method in circuit complexity. In *Proceedings of the Eighth Annual Structure in Complexity Theory Conference*, pages 82–95. IEEE Computer Society Press, 1993.

[3] R. Beigel. Perceptrons, PP, and the polynomial hierarchy. *Computational Complexity*, 4(4):339–349, 1994.

[4] J. Bruck and R. Smolensky. Polynomial threshold functions, AC^0 functions, and spectral norms. *SIAM Journal on Computing*, 21(1):33–42, 1992.

[5] H. Buhrman, N. K. Vereshchagin, and R. de Wolf. On computation and communication with small bias. In *IEEE Conference on Computational Complexity*, pages 24–32, 2007.

[6] P. Butkovič. *Max-linear Systems: Theory and Algorithms.* Springer, 2010.

[7] A. K. Chandra, M. L. Furst, and R. J. Lipton. Multi-party protocols. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, pages 94–99. ACM, 1983.

[8] J. Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and System Sciences*, 65(4):612–625, 2002.

[9] J. Forster, M. Krause, S. V. Lokam, R. Mubarakzjanov, N. Schmitt, and H.-U. Simon. Relations between communication complexity, linear arrangements, and computational complexity. In *Proceedings of the 21st Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 2245 of *Lecture Notes in Computer Science*, pages 171–182. Springer, 2001.

[10] M. Goldmann. On the power of a threshold gate at the top. *Information Processing Letters*, 63(6):287–293, 1997.

[11] M. Goldmann, J. Håstad, and A. A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2(4):277–300, 1992.

[12] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46(2):129–154, 1993.

[13] K. A. Hansen and P. B. Miltersen. Some meet-in-the-middle circuit lower bounds. In *29th International Symposium on Mathematical Foundations of Computer Science*, volume 3153 of *Lecture Notes in Computer Science*, pages 334–345. Springer, 2004.

[14] K. A. Hansen and V. V. Podolskii. Exact threshold circuits. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity*, pages 270–279. IEEE Computer Society, 2010.

[15] J. Håstad and M. Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:113–129, 1991.

[16] A. R. Klivans, R. O'Donnell, and R. A. Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer and System Sciences*, 68(4):808–840, 2004.

[17] A. R. Klivans and R. A. Servedio. Learning DNF in time $2^{\widetilde{O}(n^{1/3})}$. *Journal of Computer and System Sciences*, 68(2):303–318, 2004.

[18] M. Krause and P. Pudlák. On the computational power of depth-2 circuits with threshold and modulo gates. *Theoretical Computer Science*, 174(1–2):137–156, 1997.

[19] M. Krause and P. Pudlák. Computing boolean functions by polynomials and threshold circuits. *Computational Complexity*, 7(4):346–370, 1998.

[20] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry.* MIT Press, 1969.

[21] S. Muroga. *Threshold Logic and its Applications.* John Wiley & Sons, Inc., 1971.

[22] S. Muroga, I. Toda, and S. Takasu. Theory of majority decision elements. *Journal of the Franklin Institute*, 271:376–418, 1961.

[23] N. Nisan. The communication complexity of threshold gates. In V. T. S. D. Miklós, T. Szönyi, editor, *Combinatorics, Paul Erdős is Eighty*, volume 1 of *Mathematical Studies 1*, pages 301–315. Bolyai Society, 1993.

[24] R. Paturi and J. Simon. Probabilistic communication complexity. *Journal of Computer and System Sciences*, 33(1):106–123, 1986.

[25] A. Razborov and A. Wigderson. $n^{\Omega(\log n)}$ lower bounds on the size of depth-3 threshold circuits with AND gates at the bottom. *Information Processing Letters*, 45(6):303–307, 1993.

[26] A. A. Razborov and A. A. Sherstov. The sign-rank of AC^0. *SIAM Journal on Computing*, 39(5):1833–1855, 2010.

[27] M. E. Saks. Slicing the hypercube. In K. Walker, editor, *Surveys in Combinatorics*, number 187 in London Mathematical Society Lecture Note Series. Cambridge University Press, 1993.

[28] A. A. Sherstov. Communication lower bounds using dual polynomials. *Bulletin of the EATCS*, 95:59–93, 2008.

[29] A. A. Sherstov. Separating AC^0 from depth-2 majority circuits. *SIAM Journal on Computing*, 38(6):2113–2129, 2009.

[30] D. Speyer and B. Sturmfels. Tropical Mathematics. *ArXiv Mathematics e-prints*, Aug. 2004.

[31] J. von zur Gathen and M. Sieveking. A bound on the solutions of linear integer equalities and inequalities. *Proceedings of the American Mathematical Society*, 72(1):155–158, 1978.

A Threshold circuits with monotone gates

Lemma 29. *Any polynomial size circuit in* $\mathsf{MAJ} \circ \mathsf{MAJ}$ *is equivalent to a polynomial size circuit of the same form such that all majority gates on the bottom level are monotone (and we allow the output majority gate to negate some of its inputs).*

Proof. The proof is analogous to the proof of Theorem 5 and simplifies to the following. Consider some $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuit and consider a majority gate $l(x) = \sum_i a_i x_i + a_0 \geqslant 0$ on the bottom level. We will express this majority gate as a linear combination of polynomially many monotone majority gates with polynomially bounded coefficients. For this denote by $A(x)$ the sum of the positive monomials in $l(x)$ and by $B(x)$ the negated sum of the negative monomials in $l(x)$. That is $l(x) = A(x) - B(x)$ and all coefficients in A and B are positive. The coefficients in A and B are polynomially bounded and thus these two polynomials obtain at most polynomially many different values on $\{0,1\}^n$. Define $c = \max_{x \in \{0,1\}^n} A(x)$ and let us consider linear form $M(x) = A(x) + c(2B(x)+1)$ this linear form also obtains at most polynomial number of values and moreover it is easy to see that if for some $x, x' \in \{0,1\}^n$ $(A(x), B(x)) \neq (A(x'), B(x'))$ then $M(x) \neq M(x')$. Now consider all possible values of the pair $(A(x), B(x))$ such that $l(x) \geqslant 0$ and let $t_1, \ldots, t_k$ be the corresponding values of M. It is easy to see that $l(x) \geqslant 0$ if and only if for exactly one i we have $M(x) - t_i = 0$. That is, we have expressed our majority gates as a sums of exact majority gates $M(x) - t_1 = 0, \ \ldots, M(x) - t_k = 0$. Finally noting that the value of exact majority gate with integer coefficients $N(x) = 0$ is equal to the difference of two majority gates $N(x) \geqslant 0$ and $N(x) \geqslant 1/2$ we obtain the desired result.

We note that the same proof works also for $\mathsf{THR} \circ \mathsf{MAJ}$ circuits. ☐

ECCC ISSN 1433-8092

http://eccc.hpi-web.de