Simultaneous Multiparty Communication Complexity of Composed Functions

Yassine Hamoudi
IRIF, Université Paris Diderot, France.
hamoudi@irif.fr

Abstract

The Number On the Forehead (NOF) model is a multiparty communication game between k players that collaboratively want to evaluate a given function $F : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathcal{Y}$ on some input $(x_1, \ldots, x_k)$ by broadcasting bits according to a predetermined protocol. The input is distributed in such a way that each player i sees all of it except x_i (as if x_i is written on the forehead of player i). In the Simultaneous Message Passing (SMP) model, the players have the extra condition that they cannot speak to each other, but instead send information to a referee. The referee does not know the players' inputs, and cannot give any information back. At the end, the referee must be able to recover $F(x_1, \ldots, x_k)$ from what she obtained from the players.

A central open question in the simultaneous NOF model, called the $\log n$ *barrier*, is to find a function which is hard to compute when the number of players is polylog(n) or more (where the x_i's have size poly(n)). This has an important application in circuit complexity, as it could help to separate ACC^0 from other complexity classes [HG91, BGKL04]. One of the candidates for breaking the $\log n$ barrier belongs to the family of *composed functions*. The input to these functions in the k-party NOF model is represented by a $k \times (t \cdot n)$ boolean matrix M, whose row i is the number x_i on the forehead of player i and t is a block-width parameter. A symmetric composed function acting on M is specified by two symmetric n- and kt-variate functions f and g (respectively), that output $f \circ g(M) = f(g(B_1), \ldots, g(B_n))$ where B_j is the j-th block of width t of M. As the majority function $\textsc{Maj}$ is conjectured to be outside of ACC^0, Babai *et. al.* [BKL95, BGKL04] suggested to study the composed function $\textsc{Maj} \circ \textsc{Maj}_t$, with t large enough, for breaking the $\log n$ barrier (where $\textsc{Maj}_t$ outputs 1 if at least $kt/2$ bits of the input block are set to 1).

So far, it was only known that block-width $t = 1$ is not enough for $\textsc{Maj} \circ \textsc{Maj}_t$ to break the $\log n$ barrier in the simultaneous NOF model [BGKL04] (Chattopadhyay and Saks [CS14] found an efficient protocol for $t \leq$ polyloglog(n), but it requires randomness to be simultaneous). In this paper, we extend this result to any constant block-width $t > 1$ by giving a *deterministic simultaneous* protocol of cost $2^{\mathcal{O}(2^t)} \log^{2^{t+1}}(n)$ for *any* symmetric composed function $f \circ g$ (which includes $\textsc{Maj} \circ \textsc{Maj}_t$) when there are more than $2^{\Omega(2^t)} \log n$ players.

Keywords: Communication complexity, Number On the Forehead model, Simultaneous Message Passing, Log n barrier, Symmetric Composed functions.

1 Introduction

1.1 Number On the Forehead and Simultaneous models

The *Number On the Forehead* (NOF) model is a multiparty communication model introduced by Chandra, Furst and Lipton [CFL83] that generalizes the two player communication game of Yao [Yao79]. In this model, k players are given k inputs $x_1 \in \mathcal{X}_1, \ldots, x_k \in \mathcal{X}_k$ on which they want to compute some function $F : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathcal{Y}$. Each player i sees all of the input $(x_1, \ldots, x_k)$, except x_i. The situation is as if input x_i is written on the forehead of player i.

In order to collaboratively evaluate $F(x_1, \ldots, x_k)$, the players communicate by *broadcasting* bits according to a predetermined *protocol*. This protocol specifies whose turn it is to speak, and which bit is to be sent given the information exchanged so far and the input seen by the speaking player. It also determines when communication stops. At the end, all the players must be able to recover $F(x_1, \ldots, x_k)$ from the input they see and the transcript of the exchange. The cost of the protocol on input $(x_1, \ldots, x_k)$ is the number of exchanged bits, and the total cost is the worst case cost on all inputs. The k-party deterministic communication complexity of F, denoted $\mathsf{D}_k(F)$, is the cost of the most efficient protocol computing F.

In most of the settings, the x_i's are poly n-bits long (for some parameter n) and $\mathcal{Y} = \{0, 1\}$. In this case, the naive protocol is to broadcast first the entire input x_1 (this can be done by player 2), and then player 1 computes $F(x_1, \ldots, x_k)$ and sends the result to the other players. This protocol has cost $m + 1$ (where $m = \text{poly}(n)$ is the number of bits required for sending x_1), which proves $\mathsf{D}_k(F) = \mathcal{O}(\text{poly}\, n)$. Consequently, a protocol will be said to be *efficient* if it has cost $\mathcal{O}(\text{polylog}\, n)$ (i.e. we seek for exponential speed-up over the naive protocol).

Among the many variants of the previous framework (randomized, quantum, etc.), we will be interested in the *simultaneous* (or *Simultaneous Message Passing - SMP*) model [Yao79, NW93, BKL95, PRS97] in which the players cannot speak to each other but instead send information to a referee. The referee does not know the players' inputs, and cannot give any information back. At the end, the referee must be able to recover $F(x_1, \ldots, x_k)$ from what she obtained from the players. The simultaneous deterministic communication complexity is denoted $\mathsf{D}^{||}_k(F)$, and it always satisfies $\mathsf{D}_k(F) \leq \mathsf{D}^{||}_k(F)$. It has often been easier to reason first in this weaker model for proving lower bounds [BGKL04, PRS97, BPSW05, BYJKS02]. It is also more suitable and fruitful for studying certain functions, such as Equality in the two party setting [Yao79, Amb96, NS96, BK97, BCWdW01, GRd08, BGK15]. We will show in the next section that the simultaneous deterministic communication model is also closely connected to lower bound results in the complexity class ACC^0.

1.2 The log n barrier problem and ACC^0 lower bounds

The NOF model has proved to be of value in the study of many areas of computer science, such as branching programs [CFL83], Ramsey theory [CFL83], circuit complexity [HG91, BT94], quasi-random graphs [CT93], proof complexity [BPS07], etc. One of the most interesting connections, pointed out by Håstad and Goldmann [HG91] and refined in [BGKL04], is a way to derive lower bounds for the complexity class[1] ACC^0 from lower bounds in the *simultaneous* NOF model. More precisely, according to a result from Yao, Beigel and Tarui [Yao90, BT94], any function $f \in \mathsf{ACC}^0$ can be expressed as a depth-2 circuit whose top gate is a symmetric gate of fan-in $2^{\log^c n}$, and

[1] ACC_0 refers to the functions computable by constant-depth poly-size circuits with unbounded fan-in And, OR, Not and Mod_m gates (where Mod_m outputs 0 iff the sum of its inputs is divisible by m).

each bottom gate is an AND gate of fan-in $\log^d n$ (for some constants c, d). Consequently, for any partition of the input of f between $k = \log^d n - 1$ players in the *simultaneous* NOF model, there exists a partition of the AND gates between the players such that each of them sees all the input bits she needs to evaluate the gates she received. The players can then send to the referee the number of gates that evaluate to 1, which enables the referee to compute f. The total cost of this protocol is $\mathcal{O}\left(k \log\left(2^{\log^c n}\right)\right) = \mathcal{O}\left(\log^{c+d} n\right)$. Conversely, any super-polylogarithmic lower bound in the *simultaneous* NOF model for a function f and a partition of its input between polylog(n) players would imply $f \notin \mathsf{ACC}^0$.

Separating ACC^0 from other complexity classes is a central question in complexity theory. It is conjectured that ACC^0 does not contain the majority function MAJ, but the only result known so far is $\mathsf{NEXP} \not\subset \mathsf{ACC}^0$ [Wil14]. The aforementioned connection with communication complexity has motivated the search for a function which is hard to compute for $k \geq \log n$ players in the simultaneous NOF model. This problem is called the $\log n$ *barrier*.

Obtaining lower bounds in the NOF model is a challenging task, as the current methods become very weak when $k \geq \log n$. The only general lower bound technique known so far is the discrepancy method and its variants [BNS92, CT93, Raz00, She11]. One of the early application of it was an $\Omega(n/4^k)$ lower bound on the randomized complexity of the *Generalized Inner Product* (GIP) function [BNS92]. A long series of generalizations and improvements of the discrepancy method subsequently led to an $\Omega\left(\frac{\sqrt{n}}{k2^k}\right)$ (resp. $\Omega(n/4^k)$) lower bound on the randomized (resp. deterministic) complexity of the *Disjointness* (DISJ) function [Tes03, BPSW06, CA08, LS09, BH12, She16, She14, RY15]. It might seem like other lower bound arguments could prove that GIP and DISJ remain hard for $k \geq \log n$ players. However, surprising non-simultaneous [Gro94, ACFN15] and simultaneous [BGKL04, ACFN15] protocols proved that the aforementioned lower bounds are nearly optimal, and that these two functions cannot break the $\log n$ barrier. Very recently, Podolskii and Sherstov [PS17] showed that the randomized complexity of GIP and DISJ is exactly $\Theta\left(\frac{\log n}{\lceil 1+k/\log n \rceil} + 1\right)$ when $k \geq \log n$, and built a function having complexity $\Omega(\log n)$ independently of k. Although these last results do not break the $\log n$ barrier, they are the first superconstant lower bounds proved for explicit functions when $k \geq \log n$.

1.3 Composed Functions

An input $x_1, \ldots, x_k \in \{0,1\}^n$ to k players in the NOF model can be visualized as a $k \times n$ boolean matrix M where row i is the number x_i on the forehead of player i. The protocols known so far for GIP and DISJ strongly rely on the particular way these functions act on matrix M. They both consist in applying the $g = \text{AND}$ function on each of the n columns of M, followed by the $f = \text{MOD}_2$ (for GIP) or $f = \text{NOR}$ (for DISJ) function on the n resulting bits. Since GIP and DISJ do not break the $\log n$ barrier, a natural move has been to try other f and g functions, and to increase the number t of columns on which each g function applies. These are called the *composed functions*, formally defined below and depicted in Figure 1.

Definition 1 (Boolean input version)**.** *Fix a block-width parameter $t \geq 1$, and consider functions $f : \{0,1\}^n \to \{0,1\}$ and $\vec{g} = (g_1, \ldots, g_n)$ where $g_j : (\{0,1\}^t)^k \to \{0,1\}$. Given $x_1, \ldots, x_k \in \{0,1\}^{t \cdot n}$, the composed function $f \circ \vec{g}$ for k players outputs $f \circ \vec{g}(x_1, \ldots, x_k) = f(g_1(B_1), \ldots, g_n(B_n))$ where $B_j \in (\{0,1\}^t)^k$ is the j^{th} block of width t in the matrix representation M of the input. When $g = g_1 = \cdots = g_n$, we denote $f \circ \vec{g}$ by $f \circ g$.*

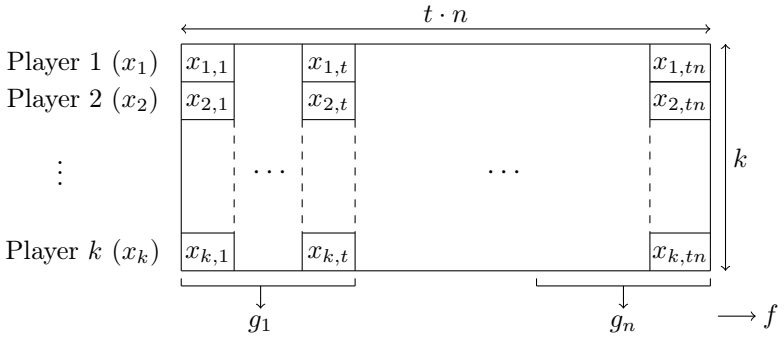

Figure 1: Matrix structure of a composed function $f \circ \vec{g}$ of block-width t.

Both $\textsc{Gip} = \textsc{Mod}_2 \circ \textsc{And}$ and $\textsc{Disj} = \textsc{Nor} \circ \textsc{And}$ are composed functions for $t = 1$, with the additional property that $\textsc{Mod}_2$, Nor and And are *symmetric* functions (i.e. invariant under any permutation of their input). Since the majority function Maj is conjectured to be outside of ACC^0, Babai *et. al.* [BKL95, BGKL04] suggested to look at $\textsc{Maj} \circ \textsc{Maj}_t$ and $\textsc{Maj} \circ \textsc{Thr}^s_t$ for breaking the $\log n$ barrier (where $\textsc{Maj}_t$ outputs 1 if at least $kt/2$ bits of the input block are set to 1, and $\textsc{Thr}^s_t(r_1, \ldots, r_k) = 1$ if $r_1 + \cdots + r_k \geq s$ for $r_1, \ldots, r_k$ seen as t-bits numbers).

Another way to look at composed functions of block-width t is to interpret each sub-row $r \in \{0,1\}^t$ of each block as a number in $\mathbb{Z}_d$, where $d = 2^t$. This representation of the input as a $k \times n$ matrix M over some set $\mathbb{Z}_d$ is sometimes more convenient to use. Below, we reformulate Definition 1 using this point of view.

Definition 2 (Integer input version)**.** *Fix an integer $d \geq 2$ and consider functions $f : \{0,1\}^n \to \{0,1\}$ and $\vec{g} = (g_1, \ldots, g_n)$ where $g_j : \mathbb{Z}_d^k \to \{0,1\}$. Given $x_1, \ldots, x_k \in \mathbb{Z}_d^n$, the composed function $f \circ \vec{g}$ for k players outputs $f \circ \vec{g}(x_1, \ldots, x_k) = f(g_1(C_1), \ldots, g_n(C_n))$ where $C_j \in \mathbb{Z}_d^k$ is the j^{th} column in the matrix representation M of the input. When $g = g_1 = \cdots = g_n$, we denote $f \circ \vec{g}$ by $f \circ g$.*

The set of all composed functions $f \circ \vec{g}$ (resp. $f \circ g$) over $\mathbb{Z}_d$ is denoted $\mathsf{ANY} \circ \overrightarrow{\mathsf{ANY}}_{\mathbb{Z}_d}$ (resp. $\mathsf{ANY} \circ \mathsf{ANY}_{\mathbb{Z}_d}$). Similarly, $\mathsf{SYM} \circ \mathsf{SYM}_{\mathbb{Z}_d}$ is the set of $f \circ g$ for symmetric f and symmetric g functions, $\mathsf{SYM} \circ \overrightarrow{\mathsf{ANY}}_{\mathbb{Z}_d}$ is the set of $f \circ \vec{g}$ for symmetric f and any $\vec{g}$, etc. If $d = 2$ (which corresponds to block-width $t = 1$), we will drop the subscript and write $\mathsf{ANY} \circ \overrightarrow{\mathsf{ANY}}$, $\mathsf{SYM} \circ \mathsf{SYM}$, etc. We have for instance $\textsc{Gip}, \textsc{Disj} \in \mathsf{SYM} \circ \mathsf{SYM}$ and $\textsc{Maj} \circ \textsc{Maj}_t, \textsc{Maj} \circ \textsc{Thr}^s_t \in \mathsf{SYM} \circ \mathsf{SYM}_{\mathbb{Z}_{2^t}}$.

The first efficient protocol for composed functions with polylog(n) or more players was given by Grolmusz [Gro94]. It is a *non-simultaneous* protocol of cost $\mathcal{O}\left(\log^2 n\right)$ for any composed function in $\mathsf{SYM} \circ \textsc{And}$ (the inner function is fixed to be And) when $k \geq \log n$. The study of composed functions with symmetric outer function f was subsequently continued, as it captures many other interesting cases in communication complexity. Babai *et. al.* [BKL95] proposed first $\textsc{Maj} \circ \textsc{Maj}_1$ as a candidate to break the $\log n$ barrier. However, in a subsequent work [BGKL04], they found a *simultaneous* protocol that applies to $\mathsf{SYM} \circ \mathsf{COMP}^c$ (where COMP^c holds for *c-compressible symmetric* functions[2], a subset of SYM that contains Maj and And). It has cost $\mathcal{O}\left(\log^{2+c} n\right)$

[2]A class $\mathscr{G}$ (parameterized by k) of symmetric functions $g : \{0,1\}^k \to \{0,1\}$ is c-compressible if for any function $g \in \mathscr{G}$, set $S \subsetneq \{1, \ldots, k\}$ and input $(x_i)_{i \in S} \in \{0,1\}^{|S|}$ there is a message m_S of size $\mathcal{O}(1) + c \log(k - |S|)$ such that $g(x_1, \ldots, x_k)$ can be computed for any $(x_i)_{i \in \{1,\ldots,k\} \setminus S} \in \{0,1\}^{k-|S|}$ from knowledge of m_S and $(x_i)_{i \in \{1,\ldots,k\} \setminus S}$. The $\textsc{Maj}_1$ and $\textsc{Thr}^s_1$ functions are 1-compressible [BGKL04].

when $k > 1 + \log n$. Later, Ada *et. al.* [ACFN15] generalized this result to $\mathsf{SYM} \circ \overrightarrow{\mathsf{ANY}}$, with a *simultaneous* protocol of cost $\mathcal{O}\left(\log^3 n\right)$ for $k > 1 + 2\log n$ players. The only protocol known so far for block-width $t > 1$ has been discovered by Chattopadhyay and Saks [CS14]. It has cost $\mathcal{O}\left(d \log n \log(dn)\right)$ for $\mathsf{SYM} \circ \overrightarrow{\mathsf{ANY}}_{\mathbb{Z}_d}$ when $k > 1 + d\log(3n)$ (which is efficient for $d \leq \text{polylog}\, n$). However, it is *not simultaneous* in the deterministic setting (the authors showed how to make it simultaneous using shared randomness between the players). Thus, none of these previous results prevents from breaking the $\log n$ barrier in the SMP model with composed functions of block-width as small as $t = 2$. The goal of this paper is to rule out this possibility for all symmetric composed functions of constant block-width $t > 1$.

1.4 Summary of Results and Comparison to Previous Protocols

Below, we describe our main results, and summarize in Table 2 the complexity of all the known protocols for composed functions. Then, we review the main ideas used in the previous literature, and we explain how we differ from them.

Our results In this paper, we describe the first *deterministic simultaneous* protocol for symmetric composed functions of block-width $t > 1$. Our result is divided into two parts. We first give (Section 3.1) a protocol of cost $\mathcal{O}\left(k(k+d)^{d-1} \log n\right)$ for $\mathsf{SYM} \circ \mathsf{SYM}_{\mathbb{Z}_d}$ when the number of players is $k \geq 4^{d-1} \log n$. In a second time (Section 3.2), we build upon this result to give a simultaneous protocol of cost $2^{\mathcal{O}(d)} \log^{2 \cdot 2^{\lceil \log d \rceil}}(n)$ for $\mathsf{SYM} \circ \overrightarrow{\mathsf{SYM}}_{\mathbb{Z}_d}$ when $k \geq 4^{2d} \log n$. Unlike the first protocol, this last result also works with different inner functions $g_1, \ldots, g_n$ and it is efficient even if k is super-polylogarithmic.

	Supported functions	Complexity of the protocol	Simultaneous	Number of players required
Grolmusz [Gro94]	$\mathsf{SYM} \circ \textsc{And}$	$\mathcal{O}\left(\log^2 n\right)$	No	$k \geq \log n$
Babai *et. al.* [BGKL04]	$\mathsf{SYM} \circ \mathsf{COMP}^c$	$\mathcal{O}\left(\log^{2+c} n\right)$	Yes	$k > 1 + \log n$
Ada *et. al.* [ACFN15]	$\mathsf{SYM} \circ \overrightarrow{\mathsf{ANY}}$	$\mathcal{O}\left(\log^3 n\right)$	Yes	$k > 1 + 2\log n$
C. and Saks [CS14]	$\mathsf{SYM} \circ \overrightarrow{\mathsf{ANY}}_{\mathbb{Z}_d}$	$\mathcal{O}\left(d \log n \log(dn)\right)$	No	$k > 1 + d\log(3n)$
This work	$\mathsf{SYM} \circ \overrightarrow{\mathsf{SYM}}_{\mathbb{Z}_d}$	$2^{\mathcal{O}(d)} \log^{4d}(n)$	Yes	$k \geq 4^{2d} \log n$

Figure 2: Deterministic protocols for different families of composed functions. The top three results apply only to block-width 1 (i.e. $d = 2$), whereas the last two results work for any d. Note that the protocol of [CS14] can be made simultaneous using shared randomness between the players.

Adjacent vertices of the $\{0,1\}^n$ hypercube. For block-width $t = 1$ and an input matrix $M \in \{0,1\}^{k \times n}$, denote n_c the number of times column $c \in \{0,1\}^k$ occurs in M. Grolsmusz [Gro94] noticed that if $c_1, \ldots, c_m$ is a sequence of adjacent vertices of the $\{0,1\}^k$ hypercube (i.e. c_{l+1} differs from c_l by exactly one coordinate) then $n_{c_1} = \left(\sum_{l=1}^{m-1} (-1)^{l+1} (n_{c_l} + n_{c_{l+1}})\right) + (-1)^{m+1} n_{c_m}$. Moreover, if position i is the coordinate at which c_l and c_{l+1} differ, then the quantity $n_{c_l} + n_{c_{l+1}}$ is known by player i. This leads to a straightforward simultaneous protocol of cost $\mathcal{O}\left(k \log n\right)$ for computing

n_{c_1}, provided that n_{c_m} is known by the referee. In his initial work, Grolsmusz [Gro94] gave a non-simultaneous way to find some initial n_{c_m}. Ada *et. al.* [ACFN15] noticed later that this step can be made simultaneous using the protocol of Babai *et. al.* [BGKL04], and that the idea of Grolsmusz (initially designed for $\mathsf{SYM} \circ \textsc{And}$) easily adapts to $\mathsf{SYM} \circ \overrightarrow{\mathsf{ANY}}$. Unfortunately, this "hypercube view" does not generalize to block-width $t > 1$: for each i and $c \in (\{0,1\}^t)^k$, the number of vertices that differ from c only at position i is now $2^t - 1 > 1$. It is easy to see that writing a similar telescoping sum as above, in which each term would be known by a player, is no longer possible.

Counting up to symmetry. Given a $k \times n$ matrix M over $\mathbb{Z}_d$, for all $0 \leq e_1 + \cdots + e_{d-1} \leq k$ denote $y_{e_1,\ldots,e_{d-1}}$ the number of columns of M with exactly e_s occurrences of each $s \in \mathbb{Z}_d \backslash \{0\}$ (we do not put e_0 since it is always equal to $k - (e_1 + \cdots + e_{d-1})$). These numbers provide less information than the n_c's defined above, but they still unable us to compute $f \circ g(M)$ for all $f \circ g \in \mathsf{SYM} \circ \mathsf{SYM}_{\mathbb{Z}_d}$. If M is distributed between k players in the NOF model (player i does not see row i), a naive *simultaneous* protocol is to have each player i send the number of columns $a^i_{e_1,\ldots,e_{d-1}}$ which contain, *from her point of view*, exactly e_s occurrences of each element $s \in \{1, \ldots, d-1\}$ (for all $e_1 + \cdots + e_{d-1} \leq k-1$). Babai *et. al.* [BGKL04] analyzed this protocol in the case $d = 2$, and showed that it gives the referee enough information to recover the $y_{e_1,\ldots,e_{d-1}}$'s, provided that $k > 1 + \log n$. In Section 3.1, we extend this analysis to any $d > 2$. The core of the proof, as in [BGKL04], is to define a specific equation (using the $a^i_{e_1,\ldots,e_{d-1}}$'s) whose only integral solution is the $y_{e_1,\ldots,e_{d-1}}$'s.

The shifted basis technique. The only protocol [CS14] known prior to this work for block-width $t > 1$ is based on the following observation: given polynomial representations of the inner functions g_j (over variables $x_{1,j}, \ldots, x_{k,j}$), each term involving strictly less than k variables can be evaluated on input matrix M by at least one player (in fact, by all the players that have one of the missing variables on their foreheads). The key idea of [CS14] is to get rid of the remaining terms by expressing the g_j in a c-shifted basis where all terms of degree k will evaluate to 0 on M (shifting for instance monomial $x_{1,j} \cdots x_{k,j}$ by $c = (s_1, \ldots, s_k)$ means to replace it with $(x_{1,j} - s_1) \cdots (x_{k,j} - s_k)$). To this end, it would suffice to find some c that shares at least one coordinate in common with each column of M. Provided that k is large enough, [CS14] showed that a randomly picked c has this property with high probability. This gives rise to a simultaneous protocol for $\mathsf{SYM} \circ \overrightarrow{\mathsf{ANY}}_{\mathbb{Z}_d}$ *if* the players have access to a shared random string. In the deterministic setting (no shared randomness), it is not known how to make this protocol simultaneous.

Different inner functions, and reducing the number of players. The communication complexity is expected to decrease as k grows up (since the overlap of information among the players increases). However, this fact is not reflected in the cost of our first protocol (Section 3.1). This issue is closely related to that of having different inner functions $g_1, \ldots, g_n$. Indeed, the problem of computing $f \circ g \in \mathsf{SYM} \circ \mathsf{SYM}_{\mathbb{Z}_d}$ with k players on a matrix $M \in \mathbb{Z}_d^{k \times n}$ can be changed into computing $f \circ (\widetilde{g_1}, \ldots, \widetilde{g_n}) \in \mathsf{SYM} \circ \overrightarrow{\mathsf{SYM}}_{\mathbb{Z}_d}$ with the first $\ell < k$ players on the submatrix $\widetilde{M} \in \mathbb{Z}_d^{\ell \times n}$ (first ℓ rows of M), where $\widetilde{g_j} : \mathbb{Z}_d^\ell \to \{0,1\}$ is defined as $\widetilde{g_j}(u) = g(u \cdot v_j)$ and v_j is the values occurring from row $\ell + 1$ to k in the j-th column of M (note that the new $\widetilde{g_j}$ functions are still symmetric, but unknown to the referee). Our first protocol cannot handle different inner functions, but this issue will be solved in Section 3.2 where we describe a protocol for $\mathsf{SYM} \circ \overrightarrow{\mathsf{SYM}}_{\mathbb{Z}_d}$ based on a new use of the *polynomial representations* (different than [CS14]). We will show that each inner function $\widetilde{g_j}$ can be represented into a (small) basis of symmetric functions $\{m_a\}_a$ (Section 2), which will allow us to

split the problem of computing $f \circ (\widetilde{g_1}, \ldots, \widetilde{g_n})$ on $\widetilde{M}$ into computing each $f \circ m_a \in \mathsf{SYM} \circ \mathsf{SYM}_{\mathbb{Z}_d}$ on a well-chosen matrix $\widetilde{M_a}$. This last step can be done with the initial protocol of Section 3.1.

2 Polynomial Representations for Symmetric Functions

Throughout this paper, $\mathbb{Z}_d$ will denote the set of integers $\{0, \ldots, d-1\}$ and $\mathbb{F}_p$ is the finite field with p elements. Furthermore, a function $f : \mathcal{X}^m \to \mathcal{Y}$ is said to be *m-symmetric* (or *symmetric*) if it is invariant under any permutation of the input variables (i.e. for any input $(x_1, \ldots, x_m)$ and permutation $\sigma \in S_m$, we have $f(x_1, \ldots, x_m) = f(x_{\sigma(1)}, \ldots, x_{\sigma(m)})$).

The protocol designed in Section 3.2 for composed functions $f \circ \vec{g}$ requires a concise polynomial representation of the inner functions $g_1, \ldots, g_n : (\{0,1\}^t)^k \to \{0,1\}$. Informally, we look for a field K and polynomials $G_j \in K[X]$ with variables $X = (x_{u,v})_{1 \leq u \leq k, 1 \leq v \leq t}$, such that:

(a) for all $x \in (\{0,1\}^t)^k$, $g_j(x) = G_j(x)$

(b) the order of K is at least $n+1$ (so that the set $\{0, \ldots, n\}$ of values taken by $\sum_j g_j(x^{(j)})$ for $x^{(1)}, \ldots, x^{(n)} \in (\{0,1\}^t)^k$ can be embedded into K)

(c) the G_j polynomials can be represented in a basis of size $\mathcal{O}(\operatorname{poly} k)$ when t is constant

(d) the values of the coefficients of the G_j polynomials in this basis are less than n^c, for some absolute constant c independent of k and t.

The first step towards this end is to look at the usual *$\mathbb{R}$-multilinear representation* (also called *Fourier expansion* [O'D14]) of a function $g : (\{0,1\}^t)^k \to \{0,1\}$. For each $a = (a_{u,v})_{1 \leq u \leq k, 1 \leq v \leq t} \in (\{0,1\}^t)^k$ we define the *indicator polynomial* $1_{\{a\}}(x)$ to be $1_{\{a\}}(x) = \prod_{1 \leq u \leq k, 1 \leq v \leq t} (1 - a_{u,v} + (2a_{u,v} - 1)x_{u,v})$. It is easy to see that it takes value 1 when $x = a$ and value 0 when $x \in (\{0,1\}^t)^k \setminus \{a\}$. Consequently, we have $g(x) = \sum_{a \in (\{0,1\}^t)^k} g(a) 1_{\{a\}}(x)$ for all $x \in (\{0,1\}^t)^k$. If we let x^a be the monomial $\prod_{(u,v); a_{u,v}=1} x_{u,v}$, then there exist real coefficients $\widehat{g}(a)$ such that it can be rewritten as the following multilinear polynomial

$$g(x) = \sum_{a \in (\{0,1\}^t)^k} \widehat{g}(a) x^a \tag{1}$$

Moreover, the $\widehat{g}(a)$ coefficients are given by the Möbius inversion formula

$$\widehat{g}(a) = \sum_{a' \subseteq a} (-1)^{|a| - |a'|} g(a') \tag{2}$$

where $|a|$ is the number of 1 in $a \in (\{0,1\}^t)^k$, and $a' \subseteq a$ means $a'_{u,v} = 0$ whenever $a_{u,v} = 0$.

Polynomial (1) is called the *$\mathbb{R}$-multilinear representation* of function g. It satisfies requirements (a) and (b) above, but not requirement (c). Indeed, these polynomials are expressed in the basis of monomials $\{x^a\}_{a \in (\{0,1\}^t)^k}$ which has size $2^{t \cdot k}$.

In order to reduce the size of the basis, we restrict ourselves to the k-symmetric functions $g : (\{0,1\}^t)^k \to \{0,1\}$ (as will be the case in Section 3.2). This condition leads to the following equalities between coefficients.

Lemma 1. *For any $a = (a_1, \ldots, a_k) \in (\{0,1\}^t)^k$ and any permutation $\sigma \in S_k$, if $g : (\{0,1\}^t)^k \to \{0,1\}$ is a k-symmetric function then the coefficients $\widehat{g}(a)$ and $\widehat{g}(\sigma(a))$ in the $\mathbb{R}$-multilinear representation of g are equal (where $\sigma(a) = (a_{\sigma(1)}, \ldots, a_{\sigma(k)})$).*

Proof. The proof is direct from Equation (2). □

This lemma motivates the definition of the following polynomials, that will be used to obtain a basis for the k-symmetric functions over $(\{0,1\}^t)^k$.

Definition 3. *Given $a \in (\{0,1\}^t)^k$, the* monomial k-symmetric polynomial *$m_a(x)$ over variables $(x_{u,v})_{1 \le u \le k, 1 \le v \le t}$ is defined to be the sum of all the distinct monomials $x^{\sigma(a)}$ where $\sigma \in S_k$ ranges over all the permutations.*

Example 1. *If $(t,k) = (2,3)$ and $a = ((1,1),(0,1),(0,1))$ then $m_a(x) = x_{1,1}x_{1,2}x_{2,2}x_{3,2} + x_{1,2}x_{2,1}x_{2,2}x_{3,2} + x_{1,2}x_{2,2}x_{3,1}x_{3,2}$.*

According to Lemma 1, any k-symmetric function $g : (\{0,1\}^t)^k \to \{0,1\}$ can be expressed as a linear combination of monomial k-symmetric polynomials. From this observation, we can derive a basis for the k-symmetric functions by taking all the *distinct* monomial k-symmetric polynomials. We specify a subset of elements $a \in (\{0,1\}^t)^k$ that corresponds to this basis.

Definition 4. *We define a tuple $a = (a_1, \ldots, a_k) \in (\{0,1\}^t)^k$ to be sorted, if $|a_u| \le |a_{u'}|$ for all $1 \le u \le u' \le k$, and $a_u \le_{lex} a_{u'}$ whenever $|a_u| = |a_{u'}|$ (where $|a_u|$ is the Hamming weight of a_u, and $\le_{lex}$ is the lexicographic order over $\{0,1\}^t$). The set of all the sorted tuples over $(\{0,1\}^t)^k$ is denoted $\mathscr{S}(t,k)$.*

Lemma 2. *The set $\{m_a(x) : a \in \mathscr{S}(t,k)\}$ is a basis for the k-symmetric functions $g : (\{0,1\}^t)^k \to \{0,1\}$. Moreover, it has size $\binom{k+2^t-1}{2^t-1}$.*

Proof. It is straightforward to see that all the possible monomial k-symmetric polynomials belong to $\{m_a(x) : a \in \mathscr{S}(t,k)\}$, and that no two elements in this set have a monomial in common. Thus, it is a basis for the k-symmetric functions.

Consider the total order $\prec$ over $\{0,1\}^t$ defined as $a_u \prec a_{u'}$ if and only if $|a_u| \le |a_{u'}|$, or $|a_u| = |a_{u'}|$ and $a_u \le_{lex} a_{u'}$. Each $a \in \mathscr{S}(t,k)$ can be seen as a (distinct) non-decreasing sequence of length k from the totally ordered set $(\{0,1\}^t, \prec)$ of size 2^t. The total number of such sequences is known to be $\binom{k+2^t-1}{2^t-1}$. □

Finally, given a parameter n, we want the coefficients of the k-symmetric functions in the chosen basis to be less than n^c for some constant c independent of k and t (requirement (d)). To this end, it suffices to reformulate the previous results over a field $\mathbb{F}_p$, for some prime $p \in (n, 2n)$. We obtain the following polynomial representation for k-symmetric functions:

Proposition 3. *Any k-symmetric function $g : (\{0,1\}^t)^k \to \{0,1\}$ can be written as*

$$g(x) = \sum_{a \in \mathscr{S}(t,k)} c_a(g) \cdot m_a(x) \mod p$$

where $p \in (n, 2n)$ is prime, $c_a(g) \in \mathbb{F}_p$ and m_a is the monomial k-symmetric polynomial corresponding to the sorted tuple a. Moreover, $\mathscr{S}(t,k)$ has size $\binom{k+2^t-1}{2^t-1}$.

3 Simultaneous Protocol for $\mathsf{SYM} \circ \overrightarrow{\mathsf{SYM}}_{\mathbb{Z}_d}$

We now describe in detail our simultaneous protocol for symmetric composed functions. The result is divided into two parts. We first give in Section 3.1 a protocol of cost $\mathcal{O}\left(k(k+d)^{d-1}\log n\right)$ for $\mathsf{SYM} \circ \mathsf{SYM}_{\mathbb{Z}_d}$ when $k \geq 4^{d-1}\log n$. This is a generalization of the idea of [BGKL04], which was based on solving a particular equation. We build upon this result in Section 3.2 to give an efficient protocol of cost $\mathcal{O}\left(\log^{4d}(n)\right)$ for $\mathsf{SYM} \circ \overrightarrow{\mathsf{SYM}}_{\mathbb{Z}_d}$ when $k \geq 4^{2d}\log n$ and d is constant. This last result uses the protocol of Theorem 4 as a subroutine, and the polynomial representations described in Section 2.

3.1 The Equation Solving part

We extend the protocol for $\mathsf{SYM} \circ \mathsf{SYM}_{\mathbb{Z}_2}$ from [BGKL04] to any $d > 1$. It applies to all functions in $\mathsf{SYM} \circ \mathsf{SYM}_{\mathbb{Z}_d}$ as long as $k \geq 4^{d-1}\log n$, but it is not efficient if d is nonconstant or if the number k of players is super-polylogarithmic (we will remove this last condition in the next section). For convenience in the proof, we state the result over $\mathbb{Z}_{d+1}$ instead of $\mathbb{Z}_d$:

Theorem 4. *Let M be a $k \times n$ matrix over $\mathbb{Z}_{d+1}$, where $n \geq 2$ and $d \geq 1$. For $0 \leq e_1 + \cdots + e_d \leq k$, denote $y_{e_1,\ldots,e_d}$ the number of columns of M with exactly e_s occurrences of each $s \in \mathbb{Z}_{d+1}\backslash\{0\}$. For each $i = 1, \ldots, k$, let player i see all of M except row i. If $k \geq 4^d \log n$ then there exists a deterministic simultaneous NOF protocol of cost $k\binom{k+d}{d}\lceil \log n \rceil$, at the end of which the referee knows all the $y_{e_1,\ldots,e_d}$'s.*

Proof. The communication part of the protocol is pretty simple: each player i sends to the referee the number of columns $a^i_{e_1,\ldots,e_d}$ which contain, *from her point of view* (i.e. without taking row i into account), exactly e_s occurrences of each element $s \in \{1, \ldots, d\}$ (for all $e_1 + \cdots + e_d \leq k-1$).

The referee computes then $b_{e_1,\ldots,e_d} = \sum_{i=1}^{k} a^i_{e_1,\ldots,e_d}$ (for all $e_1 + \cdots + e_d \leq k-1$). The important thing to note is that these numbers must verify the following equalities:

$$\begin{cases} (k - (e_1 + \cdots + e_d))y_{e_1,\ldots,e_d} + \sum_{s=1}^{d} (e_s + 1)y_{e_1,\ldots,e_{s-1},e_s+1,e_{s+1},\ldots,e_d} = b_{e_1,\ldots,e_d} \\ 0 \leq e_1 + \cdots + e_d \leq k-1 \end{cases} \tag{3}$$

To see why it is true, consider a column C of M that contributes to a given $b_{e_1,\ldots,e_d}$. Either C contains exactly e_s occurrences of each element $s \in \{1, \ldots, d\}$, or there is one $s' \in \{1, \ldots, d\}$ that occurs $e_{s'} + 1$ times in C (the other s having exactly e_s occurrences in C). In the first case, C contributes to $y_{e_1,\ldots,e_d}$ and to the quantity $a_i(e_1, \ldots, e_d)$ of each player i having a 0 entry of C on her forehead (there are $k - (e_1 + \cdots + e_d)$ such players). In the second case, C contributes to $y_{e_1,\ldots,e_{s'-1},e_{s'}+1,e_{s'+1},\ldots,e_d}$ and to the quantity $a_i(e_1, \ldots, e_d)$ of each player i having a s' entry of C on her forehead (there are $e_{s'} + 1$ such players). Thus, the total contribution for $b_{e_1,\ldots,e_d}$ is $(k - (e_1 + \cdots + e_d))y_{e_1,\ldots,e_d} + \sum_{s'=1}^{d}(e_{s'} + 1)y_{e_1,\ldots,e_{s'-1},e_{s'}+1,e_{s'+1},\ldots,e_d}$.

Equalities (3) can be seen as a system of equations whose unknowns are the $y_{e_1,\ldots,e_d}$'s. Since the referee is not computationally restricted she can enumerate all the integral solutions, but she does not know which one corresponds to matrix M. The key lemma is to show that Equations (3), under mild constraints

$$y_{e_1,\ldots,e_d} \geq 0,\ 0 \leq e_1 + \cdots + e_d \leq k \quad \textit{and} \quad \sum_{e_1+\cdots+e_d \leq k} y_{e_1,\ldots,e_d} \leq n \tag{4}$$

have at most one integral solution when $k \geq 4^d \log n$. We prove it by induction on d (the base case $d = 1$ corresponds to the work of [BGKL04], the induction step is more involved and is given in Appendix A). Consequently, the referee is able to know unambiguously the correct $y_{e_1,\ldots,e_d}$'s that correspond to M.

This protocol is clearly simultaneous since the players do not need to talk to each other. Each of the k players sends $\binom{k+d}{d}$ numbers $a_i(e_1, \ldots, e_d) \leq n$. Thus the total communication cost is at most $k\binom{k+d}{d}\lceil \log n \rceil$. □

Corollary 5. *Let $n \geq 2$, $d \geq 2$ and suppose $k \geq 4^{d-1} \log n$. There is a deterministic simultaneous NOF protocol of cost $k\binom{k+d-1}{d-1}\lceil \log n \rceil$, at the end of which the referee can compute* all *composed functions $f \circ g \in \mathsf{SYM} \circ \mathsf{SYM}_{\mathbb{Z}_d}$ of her choice.*

This result can also be adapted to the case of $k < 4^d \log n$ players by splitting the initial matrix into sufficiently many parts. Previously, Ada *et. al.* [ACFN15] also generalized their work to any number k of players, by giving a protocol of cost $\mathcal{O}\left(n/2^k \cdot \log n + k \log n\right)$ for $\mathsf{SYM} \circ \overrightarrow{\mathsf{ANY}}$. However, it was not simultaneous and it does not apply to $t > 1$.

Proposition 6. *Let M be a $k \times n$ matrix over $\mathbb{Z}_{d+1}$, where $n \geq 2$ and $d \geq 1$. For $0 \leq e_1 + \cdots + e_d \leq k$, denote $y_{e_1,\ldots,e_d}$ the number of columns of M with exactly e_s occurrences of each $s \in \mathbb{Z}_{d+1} \backslash \{0\}$. For each $i = 1, \ldots, k$, let player i see all of M except row i. If $4^d \leq k < 4^d \log n$ then there exists a deterministic simultaneous NOF protocol of cost at most $\mathcal{O}\left(\frac{n}{2^{k/4^d}} \cdot (k+d)^{d+2}\right)$, at the end of which the referee knows all the $y_{e_1,\ldots,e_d}$'s.*

Proof. We split M into $\left\lceil \frac{n}{\lfloor 2^{k/4^d} \rfloor} \right\rceil$ matrices, each of size $k \times \left\lfloor 2^{k/4^d} \right\rfloor$ (except one matrix that can have less columns). These matrices have few enough columns to apply (separately) the protocol of Theorem 4 on them. The $y_{e_1,\ldots,e_d}$'s for the original matrix M are computed by recombining all the obtained results. The total cost is $\mathcal{O}\left(\frac{n}{2^{k/4^d}} \cdot k\binom{k+d}{d} \log\left(2^{k/4^d}\right)\right)$. □

3.2 The Polynomial Representation part

Using the polynomial representation of Proposition 3, we give a protocol that improves upon Corollary 5 in two ways: it is still efficient when k is super-polylogarithmic, and the inner functions $g_1, \ldots, g_n$ can be different (i.e. it applies to $\mathsf{SYM} \circ \overrightarrow{\mathsf{SYM}}_{\mathbb{Z}_d}$ instead of $\mathsf{SYM} \circ \mathsf{SYM}_{\mathbb{Z}_d}$).

Theorem 7. *Let $n \geq 2$, $d \geq 2$ and suppose $k \geq 4^{2^{\lceil \log d \rceil}} \log n$. For any composed function $f \circ \vec{g} \in \mathsf{SYM} \circ \overrightarrow{\mathsf{SYM}}_{\mathbb{Z}_d}$ there exists a deterministic simultaneous NOF protocol that computes it with cost $4^{2^{\lceil \log d \rceil + 2}} \log^{2 \cdot 2^{\lceil \log d \rceil}}(n)$.*

Proof. Let $\vec{g} = (g_1, \ldots, g_n)$. In order to use the polynomial representation of Section 2, we change the range of the g_j functions as $g_j : (\{0,1\}^t)^k \to \{0,1\}$, where $t = \lceil \log d \rceil$. This requires to encode each number $x \in \mathbb{Z}_d$ as an element $\bar{x} \in \{0,1\}^t$. If d is not a power of two then some $y \in \{0,1\}^t$ will not correspond to any $x \in \mathbb{Z}_d$. We extend each g_j as the zero function on inputs that contain such numbers (note that the functions are still k-symmetric).

The input is now a $k \times (t \cdot n)$ boolean matrix M. Each function g_j acts on the j^{th} block of M, which will be denoted $B_j \in (\{0,1\}^t)^k$. Let $\ell = 4^{2^t} \log n$, so that only the first ℓ players are

going to speak. For each block B_j, if we let $v_j \in (\{0,1\}^t)^{(k-\ell)}$ be the sub-block occurring from row $\ell+1$ to k, then $g_j : (\{0,1\}^t)^k \to \{0,1\}$ induces a new function $\widetilde{g_j} : (\{0,1\}^t)^\ell \to \{0,1\}$ such that $\widetilde{g_j}(u) = g_j(u \cdot v_j)$. Moreover, $\widetilde{g_j}$ is still a symmetric function. Thus, our task reduces to find an efficient simultaneous protocol for $f \circ (\widetilde{g_1}, \dots, \widetilde{g_n})$ with $\ell = 4^{2^t} \log n$ players. We denote $\widetilde{M}$ the $\ell \times (t \cdot n)$ submatrix of M on which we now work, and $\widetilde{B_j} \in (\{0,1\}^t)^\ell$ is the sub-block of B_j corresponding to $\widetilde{M}$.

We cannot apply directly the protocol of Theorem 4, since it only works for equal inner functions $\widetilde{g_1} = \cdots = \widetilde{g_n}$. Instead, we use first Proposition 3 on the $\widetilde{g_j}$ functions: for each $j \in \{1, \dots, n\}$ there exist coefficients $(c_a(\widetilde{g_j}))_{a \in \mathscr{S}(t,\ell)}$ over $\mathbb{F}_p$ such that $\widetilde{g_j}(x) = \sum_{a \in \mathscr{S}(t,\ell)} c_a(\widetilde{g_j}) \cdot m_a(x) \mod p$ where $p \in (n, 2n)$, m_a is the monomial k-symmetric polynomial corresponding to the sorted tuple a and $|\mathscr{S}(t,\ell)| = \binom{\ell+2^t-1}{2^t-1}$. The coefficients $c_a(\widetilde{g_j})$ are known by the first ℓ players, but not by the referee (since they depend on rows $\ell+1$ to k of M).

For each $a \in \mathscr{S}(t,\ell)$, the players build a new matrix $\widetilde{M_a}$ of size $\ell \times (c_a(\widetilde{g_1}) + \cdots + c_a(\widetilde{g_n}))$ where block $\widetilde{B_j}$ from $\widetilde{M}$ is copied $c_a(\widetilde{g_j}) \in [0, 2n)$ times. Note that $\widetilde{M_a}$ has at most $2n^2$ blocks, and there are enough players $\ell = 4^{2^t} \log n$ for applying (the boolean input version of) the simultaneous protocol of Theorem 4. It allows the referee to know the number of blocks of $\widetilde{M_a}$ which are equal —up to row permutation— to any $\widetilde{B} \in (\{0,1\}^t)^\ell$. This information is sufficient to compute $\sum_{j=1}^n c_a(\widetilde{g_j}) \cdot m_a(\widetilde{B_j})$ since the m_a functions are k-symmetric.

Finally, the referee sums these quantities modulo p over all a. It gives her $\sum_{a \in \mathscr{S}(t,\ell)}$ $\sum_{j=1}^n c_a(\widetilde{g_j}) \cdot m_a(\widetilde{B_j}) \mod p = \sum_{j=1}^n \widetilde{g_j}(\widetilde{B_j}) \mod p$. Since $\sum_{j=1}^n \widetilde{g_j}(\widetilde{B_j}) \leq n$ and $p > n$, it equals $\sum_{j=1}^n \widetilde{g_j}(\widetilde{B_j}) = \sum_{j=1}^n g_j(B_j)$. Knowing this, the referee can compute $f \circ (g_1, \dots, g_n)(M)$ since f is symmetric.

Regarding the cost of the protocol, we applied $|\mathscr{S}(t,\ell)| = \binom{\ell+2^t-1}{2^t-1}$ times the protocol of Theorem 4, with ℓ players and inputs of size at most $2n^2$. Thus the total cost is at most $\binom{\ell+2^t-1}{2^t-1} \cdot \ell\binom{\ell+2^t-1}{2^t-1}\lceil \log 2n^2 \rceil \leq \ell(\ell+2^t)^{2^{t+1}-2} \log n$. Since $\ell = 4^{2^t} \log n$ and $t = \lceil \log d \rceil$, this is less than $4^{2^t+(2^t+1)(2^{t+1}-2)} \log^{2^{t+1}}(n) \leq 4^{2^{\lceil \log d \rceil+2}} \log^{2 \cdot 2^{\lceil \log d \rceil}}(n)$. □

4 Conclusion and Open Problems

One of the main open problems in communication complexity remains to find a function which is hard to compute for $k \geq \log n$ players in the simultaneous Number On the Forehead model. We discarded this possibility for the composed functions in $\mathsf{SYM} \circ \overrightarrow{\mathsf{SYM}}_{\mathbb{Z}_d}$ (for constant d) by giving the first efficient *deterministic simultaneous* protocol for composed functions of block-width $t > 1$. In the non-simultaneous setting, the best result so far applies to $\mathsf{SYM} \circ \overrightarrow{\mathsf{ANY}}_{\mathbb{Z}_d}$ and $d = \mathcal{O}(\text{polylog}\, n)$ [CS14]. Extending these protocols to larger d, bigger families of composed functions or to the simultaneous setting (for [CS14]) would give a better insight on composed functions. Indeed, it is conjectured that the $\log n$ barrier can be broken by such functions for large d, two of the candidates being $\textsc{Maj} \circ \textsc{Maj}_t$ and $\textsc{Maj} \circ \textsc{Thr}_t^s$.

Note that both the Equation Solving and the Polynomial Representation parts of our protocol are bottleneck for handling non-constant d in our result. It could be interesting to restrict to smaller families than symmetric functions (or to choose specific inner or outer functions, such as threshold functions), or to find other relevant equations that could be solved by the referee with fewer information than in our protocol.

Apart from composed functions, there are a few other candidates for breaking the $\log n$ barrier. Some of them are matrix related problems, such as deciding the top-left entry of the multiplication of k matrices in $\mathbb{F}_2^{n\times n}$ (an $\Omega(n/2^k)$ lower bound has been obtained by Raz [Raz00]). More recently, Gowers and Viola [GV15] studied the interleaved group products, where each player receives a tuple $(x_{i,1},\ldots,x_{i,n})$ in $G = SL(2,q)$, with the promise that $\prod_{i=1}^{n} x_{1,i}\cdots x_{k,i} = g$ *or* h. Finding which is the case has cost $\Omega(n\log|G|)$ when $k=2$, and it is conjectured to remain hard for larger k.

Acknowledgements

This work was initiated during a visit to Carnegie Mellon University. The author is very grateful to Anil Ada, who introduced him to the $\log n$ barrier problem and the $\textsc{Maj}\circ\textsc{Maj}_t$ conjecture for composed functions. He also thanks him for helpful discussions on this subject, as well as the anonymous referees for their valuable comments and suggestions which helped to improve this paper.

References

[ACFN15] A. Ada, A. Chattopadhyay, O. Fawzi, and P. Nguyen. The NOF multiparty communication complexity of composed functions. *Computational Complexity*, 24(3):645–694, 2015.

[Amb96] A. Ambainis. Communication complexity in a 3-computer model. *Algorithmica*, 16(3):298–301, 1996.

[BCWdW01] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. *Phys. Rev. Lett.*, 87:167902, 2001.

[BGK15] R. C. Bottesch, D. Gavinsky, and H. Klauck. Equality, revisited. *CoRR*, abs/1511.01211, 2015.

[BGKL04] L. Babai, A. Gál, P. G. Kimmel, and S. V. Lokam. Communication complexity of simultaneous messages. *SIAM J. Comput.*, 33(1):137–166, 2004.

[BH12] P. Beame and T. Huynh. Multiparty communication complexity and threshold circuit size of AC0. *SIAM Journal on Computing*, 41(3):484–518, 2012.

[BK97] L. Babai and P. G. Kimmel. Randomized simultaneous messages: solution of a problem of Yao in communication complexity. In *Proceedings of Computational Complexity. Twelfth Annual IEEE Conference*, pages 239–246, 1997.

[BKL95] L. Babai, P. G. Kimmel, and S. V. Lokam. Simultaneous messages vs. communication. In *12th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 361–372. Springer, 1995.

[BNS92] L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Comput. Syst. Sci.*, 45(2):204–232, 1992.

[BPS07] P. Beame, T. Pitassi, and N. Segerlind. Lower bounds for Lovász–Schrijver systems and beyond follow from multiparty communication complexity. *SIAM J. Comput.*, 37(3):845–869, 2007.

[BPSW05] P. Beame, T. Pitassi, N. Segerlind, and A. Wigderson. A direct sum theorem for corruption and the multiparty NOF communication complexity of set disjointness. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, CCC '05, pages 52–66. IEEE Computer Society, 2005.

[BPSW06] P. Beame, T. Pitassi, N. Segerlind, and A. Wigderson. A strong direct product theorem for corruption and the multiparty communication complexity of disjointness. *Comput. Complex.*, 15(4):391–432, 2006.

[BT94] R. Beigel and J. Tarui. On ACC. *Computational Complexity*, 4(4):350–366, 1994.

[BYJKS02] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. Information theory methods in communication complexity. In *Proceedings 17th IEEE Annual Conference on Computational Complexity*, pages 72–81, 2002.

[CA08] A. Chattopadhyay and A. Ada. Multiparty communication complexity of disjointness. *arXiv preprint arXiv:0801.3624*, 2008.

[CFL83] A. K. Chandra, M. L. Furst, and R. J. Lipton. Multi-party protocols. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, pages 94–99. ACM, 1983.

[CS14] A. Chattopadhyay and M. E. Saks. The power of super-logarithmic number of players. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2014.

[CT93] F. R. K. Chung and P. Tetali. Communication complexity and quasi randomness. *SIAMJDiscreteMath*, 6(1):110–123, 1993.

[GRd08] D. Gavinsky, O. Regev, and R. de Wolf. Simultaneous communication protocols with quantum and classical messages. *Chicago Journal of Theoretical Computer Science*, 7, 2008.

[Gro94] V. Grolmusz. The BNS lower bound for multi-party protocols is nearly optimal. *Information and Computation*, 112:51–54, 1994.

[GV15] T. Gowers and E. Viola. The communication complexity of interleaved group products. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC '15, pages 351–360. ACM, 2015.

[HG91] J. Håstad and M. Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1(2):113–129, 1991.

[LS09] T. Lee and A. Shraibman. Disjointness is hard in the multiparty number-on-the-forehead model. *Computational Complexity*, 18(2):309–336, 2009.

[NS96] I. Newman and M. Szegedy. Public vs. private coin flips in one round communication games (extended abstract). In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 561–570. ACM, 1996.

[NW93] N. Nisan and A. Wigderson. Rounds in communication complexity revisited. *SIAM J. Comput.*, 22(1):211–219, 1993.

[O'D14] R. O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[PRS97] P. Pudlák, V. Rödl, and J. Sgall. Boolean circuits, tensor ranks, and communication complexity. *SIAM J. Comput.*, 26(3):605–633, 1997.

[PS17] V. V. Podolskii and A. A. Sherstov. Inner product and set disjointness: Beyond logarithmically many parties. *CoRR*, abs/1711.10661, 2017.

[Raz00] R. Raz. The BNS-Chung criterion for multi-party communication complexity. *Computational Complexity*, 9:2000, 2000.

[RY15] A. Rao and A. Yehudayoff. Simplified lower bounds on the multiparty communication complexity of disjointness. In *Proceedings of the 30th Conference on Computational Complexity*, CCC '15, pages 88–101, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[She11] A. A. Sherstov. The pattern matrix method. *SIAM Journal on Computing*, 40(6):1969–2000, 2011.

[She14] A. A. Sherstov. Communication lower bounds using directional derivatives. *J. ACM*, 61(6):34:1–34:71, 2014.

[She16] A. A. Sherstov. The multiparty communication complexity of set disjointness. *SIAM Journal on Computing*, 45(4):1450–1489, 2016.

[Tes03] P. Tesson. *Computational Complexity Questions Related to Finite Monoids and Semigroups*. PhD thesis, Montreal, Canada, 2003.

[Wil14] R. Williams. Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1):2:1–2:32, 2014.

[Yao79] A. Yao. Some complexity questions related to distributive computing. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 209–213. ACM, 1979.

[Yao90] A. Yao. On ACC and threshold circuits. In *Proceedings 31st Annual Symposium on Foundations of Computer Science*, pages 619–627 vol.2, 1990.

A Lemma for the Equation Solving part

In this section, we prove the following lemma:

Lemma 8. *Let $n \geq 2$, $d \geq 1$ and $k \geq 4^d \log n$. Let $(b_{e_1,\ldots,e_d})_{0 \leq e_1+\cdots+e_d \leq k-1}$ be integers. Consider the following system of equations:*

$$\begin{cases} (k-(e_1+\cdots+e_d))y_{e_1,\ldots,e_d} + \sum_{s=1}^{d}(e_s+1)y_{e_1,\ldots,e_{s-1},e_s+1,e_{s+1},\ldots,e_d} = b_{e_1,\ldots,e_d} \\ 0 \leq e_1+\cdots+e_d \leq k-1 \end{cases} \tag{5}$$

Assume further that

$$y_{e_1,\dots,e_d} \geq 0,\ 0 \leq e_1 + \cdots + e_d \leq k \quad \textit{and} \quad \sum_{e_1+\cdots+e_d \leq k} y_{e_1,\dots,e_d} \leq n \tag{6}$$

Then, under constraints (6), the system of equations (5) has at most one integral solution.

In fact, we are going to show a stronger result:

Lemma 9. *Let $n \geq 2$, $d \geq 1$ and $k > 4^d \log n - d$. Let $(b_{e_1,\dots,e_d})_{0 \leq e_1+\cdots+e_d \leq k-1}$ be integers. Consider the following system of equations:*

$$\begin{cases} (k - (e_1 + \cdots + e_d)) z_{e_1,\dots,e_d} + \sum_{s=1}^{d} (e_s + 1) z_{e_1,\dots,e_{s-1},e_s+1,e_{s+1},\dots,e_d} = 0 \\ 0 \leq e_1 + \cdots + e_d \leq k - 1 \end{cases} \tag{7}$$

Assume further that

$$\sum_{e_1+\cdots+e_d \leq k} |z_{e_1,\dots,e_d}| \leq 2n \tag{8}$$

Then, under constraints (8), the system of equations (7) cannot have a non-zero integral solution.

Proof that Lemma 9 implies Lemma 8. Assume by contradiction that Equations (5) under Constraints (6) have two different integral solutions $y = (y_{e_1,\dots,e_d})_{0 \leq e_1+\cdots+e_d \leq k}$ and $y' = (y'_{e_1,\dots,e_d})_{0 \leq e_1+\cdots+e_d \leq k}$ for $k \geq 4^d \log n$. Define $z_{e_1,\dots,e_d} = y_{e_1,\dots,e_d} - y'_{e_1,\dots,e_d}$. It is easy to see that it must verify (7), and since $y \neq y'$ there is at least one $z_{e_1,\dots,e_d} \neq 0$. Finally, since $z_{e_1,\dots,e_d} = |y_{e_1,\dots,e_d} - y'_{e_1,\dots,e_d}| \leq y_{e_1,\dots,e_d} + y'_{e_1,\dots,e_d}$, we have

$$\sum_{e_1+\cdots+e_d \leq k} |z_{e_1,\dots,e_d}| \leq \sum_{e_1+\cdots+e_d \leq k} (y_{e_1,\dots,e_d} + y'_{e_1,\dots,e_d}) \leq 2n$$

□

Proof of Lemma 9. We prove the result by induction on d. The base case has already been established in [BGKL04], we recall it for completeness.

Base case (d = 1). We denote $(z_i)_{0 \leq i \leq k}$ the variables. Equations (7) under Constraints (8) become

$$\begin{cases} (k-i) z_i + (i+1) z_{i+1} = 0,\ i = 0, 1, \dots, k-1 \\ \sum_{i=0}^{k} |z_i| \leq 2n \end{cases}$$

Thus, $z_1 = -k z_0 = -\binom{k}{1} z_0$, $z_2 = -\frac{k-1}{2} z_1 = \binom{k}{2} z_0$, and more generally $z_i = (-1)^i \binom{k}{i} z_0$. Consequently, if $(z_i)_{0 \leq i \leq k}$ is a nonzero integral solution, then $z_0 \neq 0$ and $|z_i| = \binom{k}{i} |z_0| \geq \binom{k}{i}$ for all i. We obtain a contradiction: $2n \geq \sum_{i=0}^{k} |z_i| \geq \sum_{i=0}^{k} \binom{k}{i} = 2^k > 2^{4 \log n - 1} > 2n$. Thus, Lemma 9 is true for $d = 1$.

Induction step. Assuming that Lemma 9 is true for $d-1$, we prove that it is also the case for $d \geq 2$. Suppose by contradiction that Equations (7) under Constraints (8) have a non-zero integral solution $z = (z_{e_1,\dots,e_d})_{0 \leq e_1+\cdots+e_d \leq k}$ for $k > 4^d \log n - d$. As in the proof of the base case, we want to show $\sum_{e_1+\cdots+e_d \leq k} |z_{e_1,\dots,e_d}| > 2n$, which would be a contradiction.

To this end, we are going to focus for each $0 \leq i \leq k$ on the largest element of $\{|z_{e_1,\dots,e_d}| : e_1 + \cdots + e_d = i\}$. We define

$$Z_i = \max_{e_1+\cdots+e_d=i} |z_{e_1,\dots,e_d}| \quad \text{and} \quad k^+ = \min\{i : Z_i \neq 0\}$$

Since z is a nonzero solution, k^+ is well defined. We conduct the proof as follows:

(a) Using the induction hypothesis, we show that the first nonzero Z_i must occur for $i = k^+ \leq 4^{d-1} \log n - (d-1)$.

(b) The sequence $(Z_i)_i$ verifies $\frac{k-i}{i+d} Z_i \leq Z_{i+1}$.

(c) Using the two previous results, we prove $\sum_{i=0}^{k} Z_i > 2n$.

The contradiction comes then from $\sum_{i=0}^{k} Z_i \leq \sum_{e_1+\cdots+e_d \leq k} |z_{e_1,\dots,e_d}| \leq 2n$

Proof of (a). Assume $k^+ > 0$ (otherwise the result is trivial). According to Equations (7), and knowing that $z_{e_1,\dots,e_d} = 0$ whenever $e_1 + \cdots + e_d < k^+$, we have

$$\sum_{s=1}^{d} (e_s + 1) z_{e_1,\dots,e_{s-1},e_s+1,e_{s+1},\dots,e_d} = 0$$

for all $e_1 + \cdots + e_d = k^+ - 1$. If we set apart the last term $z_{e_1,\dots,e_{d-1},e_d+1}$, we obtain

$$(k^+ - (e_1 + \cdots + e_{d-1})) z_{e_1,\dots,e_{d-1},e_d+1} + \sum_{s=1}^{d-1} (e_s + 1) z_{e_1,\dots,e_{s-1},e_s+1,e_{s+1},\dots,e_d} = 0$$

Let $z'_{e_1,\dots,e_{d-1}} = z_{e_1,\dots,e_{d-1},k^+-(e_1+\cdots+e_{d-1})}$ for all $0 \leq e_1 + \cdots + e_{d-1} \leq k^+$. We can change the variables in the previous equation as follows

$$\begin{cases} (k^+ - (e_1 + \cdots + e_{d-1})) z'_{e_1,\dots,e_{d-1}} + \sum_{s=1}^{d-1} (e_s + 1) z'_{e_1,\dots,e_{s-1},e_s+1,e_{s+1},\dots,e_{d-1}} = 0 \\ 0 \leq e_1 + \cdots + e_{d-1} \leq k^+ - 1 \end{cases}$$

This is equivalent to Equations (7) at rank $d-1$. Moreover, $\sum_{e_1+\cdots+e_{d-1} \leq k^+} |z'_{e_1,\dots,e_{d-1}}| \leq 2n$, and there exists $e_1+\cdots+e_d = k^+$ such that $z_{e_1,\dots,e_d} \neq 0$ (by definition of k^+), i.e. $z'_{e_1,\dots,e_{d-1}} \neq 0$. Consequently, it corresponds to a nonzero integral solution to Equations (7) under Constraints (8) at rank $d-1$ with parameter k^+. According to our induction hypothesis it implies $k^+ \leq 4^{d-1} \log n - (d-1)$.

Proof of (b). Setting apart $z_{e_1,\dots,e_d}$ in Equations (7), and using the triangle inequality, we obtain

$$(k - (e_1 + \cdots + e_d)) |z_{e_1,\dots,e_d}| \leq \sum_{s=1}^{d} (e_s + 1) |z_{e_1,\dots,e_{s-1},e_s+1,e_{s+1},\dots,e_d}|$$

for all $e_1 + \cdots + e_d \leq k$. In particular, if we choose $e_1 + \cdots + e_d$ such that $Z_{e_1+\cdots+e_d} = |z_{e_1,\dots,e_d}|$ then

$$\begin{aligned} (k - (e_1 + \cdots + e_d)) Z_{e_1+\cdots+e_d} &\leq \sum_{s=1}^{d} (e_s + 1) |z_{e_1,\dots,e_{s-1},e_s+1,e_{s+1},\dots,e_d}| \\ &\leq \sum_{s=1}^{d} (e_s + 1) Z_{e_1+\cdots+e_d+1} \\ &\leq (e_1 + \cdots + e_d + d) Z_{e_1+\cdots+e_d+1} \end{aligned}$$

Thus $(k-i)Z_i \le (i+d)Z_{i+1}$, where $i = e_1 + \cdots + e_d$.

Proof of (c). Using (b), first note for $i > k^+$ that

$$\begin{aligned} Z_i &\ge \frac{k-(i-1)}{(i-1)+d} \cdot \frac{k-(i-2)}{(i-2)+d} \cdots \frac{k-k^+}{k^++d} \cdot Z_{k^+} \\ &= \frac{(k-k^+)!}{(k-i)!} \cdot \frac{(k^++d-1)!}{(i+d-1)!} \cdot Z_{k^+} \\ &= \frac{(k+d-1)!}{(k-i)!(i+d-1)!} \cdot \frac{(k-k^+)!(k^++d-1)!}{(k+d-1)!} \cdot Z_{k^+} \\ &= \binom{k+d-1}{i+d-1}\binom{k+d-1}{k^++d-1}^{-1} \cdot Z_{k^+} \\ &\ge \binom{k+d-1}{i+d-1}\binom{k+d-1}{k^++d-1}^{-1} \qquad \text{since } Z_{k^+} \ge 1 \end{aligned}$$

According to (a) and our initial hypothesis on k, we have $k^+ + d - 1 \le 4^{d-1}\log n \le (k+d-1)/4$. Thus $\sum_{i=k^+}^{k} \binom{k+d-1}{i+d-1} \ge \frac{1}{2} \cdot \sum_{i=0}^{k+d-1} \binom{k+d-1}{i} = 2^{k+d-2}$ and $\binom{k+d-1}{k^++d-1}^{-1} \ge 2^{-(k+d-1)H(1/4)}$ (using the well-known bound $\binom{m}{\alpha m} \le 2^{mH(\alpha)}$ where $H(\alpha) = -\log(\alpha^\alpha(1-\alpha)^{1-\alpha})$). Consequently, since $d \ge 2$ and $n \ge 2$, we obtain $\sum_{i=k^+}^{k} Z_i \ge 2^{(1-H(1/4))(k+d-1)-1} \ge 2^{(1-H(1/4))4^d \log n - 1} > 2n$. □