Preprint, Institute of Mathematics, AS CR, Prague. 2007-12-21

Languages with Bounded Multiparty Communication Complexity *

Arkadev Chattopadhyay
McGill University, Montreal, Canada
achatt3@cs.mcgill.ca

Andreas Krebs
Universität Tübingen, Germany
mail@krebs-net.de

Michal Koucký
Mathematical Institute, Academy of Sciences, Czech Republic
koucky@math.cas.cz

Mario Szegedy
Rutgers University, New Jersey, USA
szegedy@cs.rutgers.edu

Pascal Tesson
Laval University, Québec, Canada
pascal.tesson@ift.ulaval.ca

Denis Thérien
McGill University, Montreal, Canada
denis@cs.mcgill.ca

*Research supported in part by the NFS (M. Szegedy), NSERC (A. Chattopadhyay, P. Tesson, D. Thérien), FQRNT (D. Thérien) and the Alexander von Humboldt Foundation (P. Tesson and D. Thérien). We are also grateful to Pavel Pudlák for suggesting the use of the Hales-Jewett Theorem.

Abstract

We study languages with bounded communication complexity in the multiparty "input on the forehead model" with worst-case partition. In the two-party case, languages with bounded complexity are exactly those recognized by programs over commutative monoids [20]. This can be used to show that these languages all lie in shallow ACC^0.

In contrast, we use different coding techniques to show that there are languages of arbitrarily large circuit complexity which can be recognized in constant communication by k players for $k \geq 3$. However, if a language has a neutral letter and bounded communication complexity in the k-party game for some fixed k then the language is in fact regular and we give an algebraic characterization of regular languages with this property. We also prove that a symmetric language has bounded k-party complexity for some fixed k iff it has bounded two party complexity.

1 Introduction

The "input on the forehead" multiparty model of communication, introduced by Chandra, Furst and Lipton [7], is a powerful tool in the study of branching programs [2, 6, 7] and shallow-depth Boolean circuits (among many others [11, 14, 15]). However, it is still, in many regards, not well-understood as both upper bounds [1, 12] and lower bounds [2, 7, 19] for the model appear very challenging. In particular, good lower bounds on the k-party non-interactive communication complexity of an explicit function f when $k > \log n$ have long been sought since they would yield size-lower bounds for ACC^0 circuits computing f [9], and even more modest lower bounds $\Omega(\log^3 n)$ for particular functions like Disjointness in three-party setting would imply separation of different proof systems [5].

We obtain significant insight in the multiparty model by focusing on functions that have bounded k-party complexity for $k \geq 3$ an arbitrary constant. For the two-party model, languages with bounded communication complexity have many nice characterizations [20] implying, in particular, that any language with bounded two-party complexity can be computed by very shallow ACC^0 circuits. In contrast, we show in Section 3 that there are languages with arbitrarily large uniform circuit complexity whose three-party communication complexity is bounded by a constant even for the worst-case partition of the input instances among the players. An analog result for non-uniform circuit complexity can also be derived. These languages are constructed using specially crafted *error-correcting codes*. Because of these results, we cannot expect to obtain characterizations of languages of bounded multiparty complexity which are as nice as those for the two-player case.

There are several key features that make the multiparty communication model so powerful: first, every input bit is seen by several players, second, every $(k-1)$-tuple of input positions is seen by at least one of the k players, and third, all players know the partitioning of the input, i.e., they know which positions they actually see. Multiparty communication complexity upper bounds typically rely heavily on all these properties. If we remove the first two properties then we obtain essentially the multiparty "input in the hand" model which is computationally even weaker than two-party communication model. To understand how crucial the last property is, we consider two restricted classes of languages/functions in which this advantage is in some sense taken away.

First, we consider in Section 4 languages with a *neutral letter* [4, 3], i.e. a letter which can be inserted or deleted at will in an input word. We show that every language with a neutral letter and bounded k-party communication complexity for some fixed k is regular. Furthermore, we give a characterization of this class of regular languages in terms of algebraic properties of their minimal automaton. Our results indicate that the presence of a neutral letter is thus a severe handicap in the

multiparty game and suggests that it might be easier to prove communication complexity lower bounds under this assumption.

Finally, we prove in Section 5 that the class of symmetric functions that can be computed in constant communication complexity by k-players for any fixed $k \geq 3$ is exactly the class of symmetric functions that can be computed by two players in constant communication.

Two of our main proofs rely on the same lower bound which is of independent interest: In Section 2 we show, using a Ramsey-theoretical argument reminiscent of [7], that k parties need to exchange $\omega(1)$ bits of communication to verify that their k inputs in $\{0,1\}^n$ represent a partition of $[n]$.

2 Multiparty Communication Complexity

The multiparty model of communication complexity was first introduced by Chandra, Furst and Lipton [7]. In this game, k players $P_1, \ldots, P_k$ wish to collaborate to compute a function $f : \Sigma^n \to \{0,1\}$. The n input letters are partitioned into k sets $X_1, \ldots, X_k \subseteq [n]$ and each participant P_i knows the values of all the inputs *except* the ones of X_i. This game is often referred to as the "input on the forehead" model since it is convenient to picture that player i has the letters of X_i written on his forehead, available to everyone but himself. Players exchange bits, according to an agreed upon protocol, by writing them on a public blackboard. The protocol specifies whose turn it is to speak, and what the player broadcasts is a function of the communication history and the input he has access to. The protocol's output is a function of what is on the blackboard after the protocol's termination. We denote by $D_k(f)$ the k-party communication complexity of f, i.e. the minimum number of bits exchanged in a protocol for f on the worst case input and for the worst-case partition of inputs. More generally, we consider functions $f : \Sigma^* \to \{0,1\}$ and thus view $D_k(f)$ as a function of input length.

The information available to individual players overlaps a lot since any input letter is known to $k-1$ of the k players. Thus, the power of the multiparty model increases with the number of players involved as the fraction of inputs available to each player increases.

A subset S of $\Sigma^{X_1 \times \ldots \times X_k}$ is a *cylinder in the ith dimension* if membership in S is independent of the ith coordinate, i.e. if for all $x_1, x_2, \ldots, x_k$ and any x_i' we have $(x_1, \ldots, x_i, \ldots, x_k) \in S$ if and only if $(x_1, \ldots, x_i', \ldots, x_k) \in S$. We say that S is a *cylinder intersection* if $S = \bigcap_{1 \leq i \leq k} S_i$ where S_i is a cylinder in the ith dimension. A cylinder intersection is called *f-monochromatic* if the function f evaluates to the same value on every input instance in the intersection. The following lemma underlies all lower bound arguments for the multiparty model:

Lemma 1 (see [14]) *Let $f : \Sigma^{X_1 \times \ldots \times X_k} \to \{0,1\}$ be a function of k-inputs. Any deterministic k-party communication protocol of cost c computing f partitions the input space into at most 2^c f-monochromatic cylinder intersections corresponding to the communication exchanged on a particular input.*

We say that a set of k elements of $\Sigma^{X_1 \times \ldots \times X_k}$ forms a *star* if it is of the form:

$$(x_1', x_2, \ldots, x_k), (x_1, x_2', \ldots, x_k), \ldots, (x_1, x_2, \ldots, x_k')$$

where the x_i are values for the input bits letters in X_i for each i with $x_i \neq x_i'$. In that case, we call $(x_1, x_2, \ldots, x_k)$ the *center* of this star. These notions lead to a useful characterization of cylinder intersections.

Lemma 2 *A set $S \subseteq \Sigma^{X_1 \times \ldots \times X_k}$ is a cylinder intersection if and only if the center of any star contained in S is itself an element of S.*

A k-rectangular reduction r from $L \subseteq \{0,1\}^{n\times k}$ to $K \subseteq \{0,1\}^{l(n)\times k}$ is a k-tuple of functions $(r_1,\ldots,r_k)$ with each $r_i : \{0,1\}^n \to \{0,1\}^{l(n)}$ such that $(x_1,\ldots,x_k) \in L$ iff $(r_1(x_1),\ldots,r_k(x_k)) \in K$. We call l the length of the reduction. The following simple observation shall be useful:

Observation 3 *Let $L \subseteq \{0,1\}^{n\times k}$ and $K \subseteq \{0,1\}^{l(n)\times k}$ be languages such that there exists a rectangular reduction from L to K of length l. Then, $D_k(L)(n) \leq D_k(K)(l(n))$.*

Lower bounds for the k-party communication complexity of the functions $Part_k$ and $GIP_{k,p}$ will be particularly useful. Both functions take as input an $n \times k$ Boolean matrix A and we think of the i^{th} column of A as representing a subset x_i of $[n] = \{1,\ldots n\}$. We define $Part_k(A) = 1$ iff each row contains exactly one 1 (i.e. the x_i form a partition of $[n]$) and $GIP_{k,p} = 1$ iff the number of all-1 rows of A (i.e. the size of the intersection of the x_i) is divisible by p. It is clear that for the k-party game the worst input partition for $GIP_{k,p}$ and $Part_k$ is the one where player P_i holds the bits of column i on his forehead.

Lemma 4 ([2, 13]) *$D_k(GIP_{k,p}) = \Omega(n)$ for all constants $k, p \geq 2$.*

More precisely, the best known lower bounds for GIP are $\Omega(n/2^k)$ [8, 19] and hold even for k growing as a function of n but we only consider the case where k is constant.

We establish a lower bound on the k-party communication complexity of $Part_k$ by applying a Ramsey-theoretical result known as the Hales-Jewett Theorem. The n-tuples $v^1,\ldots,v^t \in [t]^n$ are said to form a *combinatorial line* if the v^j are distinct and for each $1 \leq i \leq n$ either all the v^j agree on position i (i.e. $v^j_i = v^{j'}_i$ for all $1 \leq j \leq j' \leq t$) or we have $v^j_i = j$ for all $1 \leq j \leq t$.

Theorem 5 (Hales-Jewett [10]) *For any integers c, t there exists an integer n such that if all vectors in $[t]^n$ are colored with c colors then there is a monochromatic combinatorial line $v^1,\ldots,v^t$ (i.e. a line whose elements all were assigned the same color).*

We now prove:

Lemma 6 *For all k, $D_k(Part_k) = \omega(1)$.*

Proof: Consider the input as a collection of k subsets of $[n]$. Every input $(S_1,\ldots,S_k) \in \mathcal{P}([n])^k$ that is *accepted* by a protocol for $Part_k$ is such that for every $1 \leq j \leq n$, the element j lies in exactly one of the S_i. Using this observation, these inputs can be put in one-to-one correspondence with n-tuples in $[k]^n$. As an example for $k = 3$ and $n = 4$, we have $Part_3(\{4\},\{1,3\},\{2\}) = 1$ and this input corresponds to the n-tuple $(2,3,2,1)$.

Suppose that the k-party communication complexity of $Part_k$ is bounded, for some k, by a constant c. To every input accepted by a protocol for $Part_k$, (i.e. to every element in $[k]^n$), we can assign one of 2^c colors corresponding to the communication history resulting from that particular input. If n is large enough then by the Hales-Jewett Theorem this set contains a monochromatic combinatorial line $v^1,\ldots,v^k$. Let $T \subseteq [n]$ be the (non-empty) set of positions on which the v^j differ and for each $i \leq k$ denote as S_i the set of positions on which all the v^j are i. By definition of the above one-to-one correspondence, we have that $T, S_1,\ldots,S_k$ form a partition of $[n]$ and all the inputs $(S_1 \cup T, S_2,\ldots S_k)$, $(S_1, S_2 \cup T,\ldots S_k)$, $\ldots$, $(S_1, S_2,\ldots S_k \cup T)$ induce the same communication history. Since these inputs form a star, Lemma 2 guarantees that its center $(S_1, S_2,\ldots S_k)$ *also* induces that same communication and must thus belong to $Part_k$. However $S_1 \cup \ldots \cup S_k = [n] - T \neq [n]$ so we obtain a contradiction. ∎

Note that an $n \times k$ matrix A belongs to $Part_k$ iff none of its rows contains two 1 and the total number of 1 entries in A is n. If $k \geq 3$ then k players can check the first condition using k bits of communication since any pair of input bits is accessible to at least one player. They are then left with verifying that the sum of the input bits is n which can, surprisingly, be achieved with a communication cost much less than the trivial $O(\log n)$ [7].

3 Functions with bounded multi-party complexity but high time/space complexity

In this section we exhibit languages of arbitrarily large computational complexity but with bounded multiparty communication complexity. For a language L and an *encoding* $C : \{0,1\}^* \to \{0,1\}^*$, we denote by $C(L)$ the set $\{C(x);\ x \in L\}$. We prove that for a suitably chosen error-correcting code C, any language L is such that its encoding $C(L)$ has bounded multiparty communication complexity. We will choose C such that the corresponding encoding and decoding function are efficiently computable and hence the complexities of L and $C(L)$ will be closely related.

As a warm-up we start with the *unary encoding* C_U defined as follows: for $x \in \{0,1\}^*$, $C_{\mathrm{U}}(x) = 0^x 10^{2^n - x - 1}$, where n is the length of x and x is interpreted as an integer between 0 and $2^n - 1$. Hence, C_{U} encodes bit strings of length n into strings of length 2^n having a single 1 in a one-to-one way.

Lemma 7 *For any language L and integer $k \geq 3$, $D_k(C_{\mathrm{U}}(L)) \leq 3$.*

Proof: Without loss of generality $k = 3$. On an input w that is split among the three parties, the players need to verify two things: 1) whether w is a valid encoding of some string x, and 2) whether the corresponding string x is in L. To verify the first property the players only need to check whether at least one of them sees a 1 and whether none of them sees two or more 1s. They can communicate their observations regarding this using six bits in total. Next, one of the players who sees the one, determines the unique string x with $C_{\mathrm{U}}(x) = w$. He can do this solely based on the position of the one since he knows how w is partitioned. This player can also determine whether $x \in L$ and hence $w \in C_{\mathrm{U}}(L)$. He communicates his conclusion to the other parties by sending one more bit. Hence in total players exchange at most seven bits.

■

The disadvantage of the unary encoding is its inefficiency: because codewords are exponentially longer than the words they encode, we cannot provide efficient reductions between L and $C(L)$. A better encoding can be obtained by concatenating Reed-Solomon codes with the unary encoding. In the 3-party scenario at least one of the parties has on its forehead at least a $1/3$-fraction of the input. Hence, if the chosen encoding has the property that from an arbitrary $1/3$-fraction of the input the whole word can be reconstructed (assuming the input is an encoding of some word, i.e., assuming that the input is a codeword) the other two parties can reconstruct the whole input and verify whether the parts on remaining foreheads are consistent with such an input. With the proper choice of parameters Reed-Solomon codes have this property.

Let n be a large enough integer, $m = \lceil \log_2 3n \rceil$ and $d = n/m$. Any string $x \in \{0,1\}^n$ can be interpreted as a sequence of d elements from $GF[2^m]$. Define p_x to be the degree $d-1$ polynomial over $GF[2^m]$ whose coefficients are given by x. Define the Reed-Solomon encoding by $C_{\mathrm{RS}}(x) = p_x(g_0)p_x(g_1)\cdots p_x(g_{3d-1})$, where $GF[2^m] = \{g_0, g_1, \ldots, m_{2^m-1}\} = \{0,1\}^m$. Furthermore, define the concatenation of the Reed-Solomon encoding with the unary encoding by $C_{\mathrm{RS}\circ\mathrm{U}}(x) = C_{\mathrm{U}}(p_x(g_0))\cdots C_{\mathrm{U}}(p_x(g_{3d-1}))$. Codewords thus consist of $3d$ blocks of 2^m bits (corresponding to

the $3d$ symbols of the Reed-Solomon encoding) with each block containing exactly one 1. Thus, $C_{\mathrm{RS}\circ\mathrm{U}}$ encodes strings of length n into strings of length $O(n^2)$. Furthermore, $C_{\mathrm{RS}\circ\mathrm{U}}$ can be encoded and decoded in polynomial time and so the languages L and $C_{\mathrm{RS}\circ\mathrm{U})}(L)$ are polynomial-time equivalent. Note that the decoding task at hand does not require us to perform error correction in the usual sense: we simply want to identify if an input is a codeword (since we reject all words that are not codewords) and we only care about decoding true codewords.

Lemma 8 *For any language L and any $k \geq 3$, $D_k(C_{\mathrm{RS}\circ\mathrm{U}}(L)) \leq 6$*

Proof: Without loss of generality $k = 3$. Let $m = \lceil \log_2 3n \rceil$ and $d = n/m$. To check if an input is a codeword, the players can easily check that there are never two 1s in a single block of input bits. They cannot, however, verify at constant cost that each of the $3d$ blocks contains *at least* one 1 since this task is essentially the partition problem whose complexity we lower bounded in Lemma 6. We proceed differently: an input w of length $3d \cdot 2^m$ can only be a codeword if at least one player (say player 1) has on its forehead d ones and this player can be identified with three bits of communication. These d ones determine d elements of $GF[2^m]$ hence players 2 and 3 can each privately reconstruct from them the unique degree $d-1$ polynomial p that coincides with these elements. Players 2 and 3 now know that if the input is a codeword then it must the one corresponding to p and player 2 can check that the bits on players 1 and 3's foreheads are consistent with that hypothesis while player 3 can similarly cross-check the input bits on player 2's forehead. If this cross-checking procedure is successful, player 2 can determine the unique x such that $p_x = p$, verify $x \in L$ and send the result to all parties. Altogether the parties need to communicate only six bits in order to decide if their input is from $C_{\mathrm{RS}\circ\mathrm{U}}(L)$. ∎

As an immediate corollary to this lemma and the fact that the complexity of $C_{\mathrm{RS}\circ\mathrm{U}}(L)$ is polynomially related to the complexity of L we obtain:

Corollary 9 *The class of languages with bounded multi-party communication complexity contains languages with arbitrarily large time and space complexity.*

In order to obtain also languages with essentially the largest possible circuit complexity we need codes that map n bits into $O(n)$ bits. We can obtain such codes by concatenating Reed-Solomon codes with codes provided by the following lemma and the unary code C_U.

Lemma 10 *For any integer $n \geq 1$, there exists a linear map $C_8 : \{0,1\}^n \to GF[8]^{39n}$ such that every $w \in C_8(\{0,1\}^n)$ is uniquely determined by any one-third of its coordinates.*

By concatenating C_{RS} with C_8 and C_{U} we obtain the code $C_{\mathrm{RS}\circ 8\circ\mathrm{U}}$ with polynomial time encoding and decoding that maps n bit strings into $O(n)$ bit strings.

Corollary 11 *For any $k \geq 3$, the class of languages with bounded k-party communication complexity contains languages with $2^{\Omega(n)}$ circuit complexity.*

4 Languages with a neutral letter

A language $L \in \Sigma^*$ is said to have a *neutral letter* e if for all $u, v \in \Sigma^*$ we have $uv \in L$ iff $uev \in L$. Thus, adding or deleting e anywhere in a word w does not affect membership in L. If a language has a neutral letter then membership in L cannot depend, as in Lemma 7, on having specific value on a

specific input position and, at least intuitively, this seems to take away a lot of the power inherent to the multiparty communication model. The neutral letter hypothesis was helpful in obtaining length lower bounds on bounded-width branching programs [4] and was central to the Crane-Beach Conjecture [3]. In this section, we give a precise characterization of languages with a neutral letter that have bounded k-party complexity for some fixed k. We first show that all such languages must be regular and then characterize them in terms of algebraic properties of their minimal automaton.

4.1 Proving Regularity

Let $C \geq 0$ be an integer and let $\mathcal{G}$ be a family of functions over Σ^* with finite range R. We say that inputs with weight at most C *determine* the functions of $\mathcal{G}$ if every function $g : \Sigma^{\leq C} \to R$ has at most one extension to Σ^* in $\mathcal{G}$. Now, let $\mathcal{C}_{k,c}$ be the family of functions with a neutral letter and k-party communication complexity at most c. We show:

Lemma 12 *There is a constant $C = C(k, c)$ such that functions of $\mathcal{C}_{k,c}$ are determined by inputs of weight at most C.*

We obtain this lemma as a corollary to

Lemma 13 *For any $C > 0$ if the functions of $\mathcal{C}_{k,c}$ are not determined by inputs of size C then $Part_k$ can be solved by k parties with $2c + 2$ communication for sets of size C' for some $C' \geq C$.*

Lemma 13 implies Lemma 12, since if there were no bound $C(k, c)$ as stated in Lemma 12, then $Part_k$ would have k-party communication complexity at most $2c + 2$ for arbitrary set size, resulting in a contradiction with Lemma 6.

Proof:(Lemma 13) For any word $w \in \Sigma^*$, we shall denote by w_e the word obtained from w by deleting all occurrences of e in w. The ith letter of w will be denoted by w^i. Also, for k words $w_1, \ldots, w_k$, each of length ℓ, let $w = w_1 \Diamond \ldots \Diamond w_k$ denote the word obtained by interleaving the k words in the following way : $|w| = \ell k$ and for all $1 \leq i \leq \ell k$, $w^i = w_j^m$ if $i = (m-1)k+j$ with $0 < j < k+1$. Let us assume that f and g are in $\mathcal{C}_{k,c}$, such that they are not identical, but the minimal string $v \in \{\Sigma - e\}^*$ such that $f(v) \neq g(v)$ has length at least C. We consider the following k-party communication problem: each player gets $|v|$ bits on their forehead and let us denote the input on player i's forehead by y_i. Note that from our comments following Lemma 6, the function $Part_k$ requires unbounded k-wise complexity even if the input sets are known to be pairwise disjoint. Consider a family of k sets $I_1, \ldots, I_k \subseteq \{1, \ldots, |v|\} = [|v|]$, such that $I_i \cap I_j = \emptyset$ for all $i \neq j$. For each such choice of k sets, we assign foreheads of the players in the following way: $y_i^j = v^j$ if $j \in I_i$, otherwise $y_i^j = e$. We define the function $h(y_1, \ldots, y_k) = 1$ iff the corresponding family of k subsets partitions $[|v|]$, i.e., $\cup_{i=1}^k I_i = [|v|]$. Notice that h is exactly the partition problem for a basis set of size $|v| \geq C$. The reduction $(I_1, \ldots, I_k) \to (y_1, \ldots, y_k)$ is a rectangular reduction. We claim that $h(y_1, \ldots, y_k) = 1$ iff $f(y_1 \Diamond \ldots \Diamond y_k) \neq g(y_1 \Diamond \ldots \Diamond y_k)$.

To see this we use the minimality property of v: on words of length less than $|v|$ f and g agree. For $y = y_1 \Diamond \ldots \Diamond y_k$ we have $|y_e| = |v|$ only if $\cup_{i=1}^k I_i = [|v|]$ and in that case $y_e = v$ and $f(y) \neq g(y)$. Otherwise, we have $|y_e| < |v|$ and therefore $f(y) = g(y)$.

The function $f(v) \neq g(v)$ can be computed with 2c+2 bits of communication by running the c bit protocol on f and g separately. For $Part_k$ we also need to verify using two extra bits of communication that no row contains two ones. ∎

Let $f : \Sigma^* \to R$ be a function in $\mathcal{C}_{k,c}$: For a word $w \in \Sigma^*$, we define the function $f_w : \Sigma^* \to R$ by $f_w(z) = f(wz)$. All the f_w are also in $\mathcal{C}_{k,c}$ and so the functions $\{f_w\}$ are determined by inputs of length at most C. It follows that the equivalence relation on Σ^* defined by $u \sim v$ iff $f(uz) = f(vz)$ for all $z \in \Sigma^*$ has at most $(|\Sigma| + 1)^C$ equivalence classes. It is well-known that if $\sim$ has finite index then f is regular and we obtain

Theorem 14 *If f is a function with a neutral letter such that $D_k(f) = O(1)$ for some fixed k, then f is regular.*

4.2 Regular languages with bounded complexity

A monoid M is a set with a binary associative operation (i.e. a semigroup) and a distinguished identity element 1_M. A language $L \subseteq \Sigma^*$ is *recognized* by a finite monoid M if there is a morphism ϕ from the free monoid Σ^* to M and a set $F \subseteq M$ such that $L = \phi^{-1}(F)$. A restatement of Kleene's Theorem asserts that L is regular iff it is recognized by some finite monoid. If L is regular, the *syntactic monoid of* L (denoted $M(L)$) is the transformation monoid of L's minimal automaton [16] and is the smallest monoid recognizing L.

The *word problem* for M is the function $eval$ which maps a string $w = w_1 \dots w_n \in M^*$ to the product $eval(w_1 \dots w_n) = w_1 \cdot w_2 \cdot \dots \cdot w_n$. We define the k-party *communication complexity of* M, denoted $D_k(M)$ as the communication complexity of its word problem. Two of the authors gave a complete classification result for the two-party communication complexity of finite monoids [21] and this led to a similar classification for the two-party complexity of regular languages. The communication complexity of monoids was first studied in [18] from which we use the following lemma.

Lemma 15 *Let L be a regular language with a neutral letter and let $M = M(L)$ be its syntactic monoid. Then for any $k \geq 2$ we have $D_k(L) = \Theta(D_k(M))$.*

A finite group is *nilpotent* if it is the direct product of p-groups and a monoid lies in the class $\overline{\mathbf{G_{nil}}}$ if all its subgroups are nilpotent. The class $\mathbf{DO}$ consists of monoids satisfying the identity $(xy)^\omega(yx)^\omega(xy)^\omega = (xy)^\omega$.

Lemma 16 *If M is a finite monoid outside of $\mathbf{DO}$ then $D_k(M) = \omega(1)$ for all k.*

This lemma is proved in the appendix by showing that if M lies outside $\mathbf{DO}$ then for any k there exists a rectangular reduction of linear length from either $GIP_{k,p}$ (for a suitably chosen p) or $Part_k$ to the word problem of M.

Theorem 17 ([18]) *Let G be a group. If G is in $\mathbf{G_{nil}}$ then there exists a constant $k \geq 2$ such that $D_k(G) = O(1)$. Otherwise $D_k(G) = \Omega(n)$ for all k.*

In this case also, the lower bound is obtained through a rectangular reduction from $GIP_{k,p}$ to the word problem of any non-nilpotent finite group. The upper bound, on the other hand, stems from a combinatorial description of languages recognized by nilpotent groups. We say that a word $u = a_1 \dots a_t$ with $a_i \in \Sigma$ is a *subword* of the word w if w can be factorized as $w_0 a_1 w_1 \dots w_{t-1} a_t w_t$ and we denote by $\binom{w}{u}$ the number of such factorizations. We say that a language L *counts subwords of length k modulo m* if membership of w in L depends on the values modulo m of $\binom{w}{u_1}, \dots, \binom{w}{u_t}$ for

some u_i with $|u_i| \leq k$. One can show that the syntactic monoid of a regular language L is a nilpotent group iff there exist $k, m \geq 2$ such that L counts subwords of length k modulo m [23].

For $a \in \Sigma$ and $L, K \subseteq \Sigma^*$, we say that the concatenation LaK is *perfectly unambiguous* if $L \subseteq (\Sigma - \{a\})^*$ or $K \subseteq (\Sigma - \{a\})^*$. If LaK is perfectly unambiguous then any $w \in LaK$ can be uniquely factorized as $w_L a w_K$ with $w_L \in L$ and $w_K \in K$ since the a can only be the first or last occurrence of a in w. We now denote as $\mathcal{V}_\Sigma$ the smallest class of regular languages over Σ that contains both the subword-counting languages and the languages Σ_0^* for each $\Sigma_0 \subseteq \Sigma$ and which is closed under Boolean operations and perfectly unambiguous concatenations. The following lemma can be inferred from more general results of [21].

Lemma 18 *A language $L \subseteq \Sigma^*$ is recognized by a monoid in $\mathbf{DO} \cap \overline{\mathbf{G_{nil}}}$ iff it is in $\mathcal{V}_\Sigma$.*

We can now give a characterization of monoids that have bounded multiparty communication complexity for some suitably large constant k.

Theorem 19 *Let $L \subseteq \Sigma^*$ be a regular language with a neutral letter and syntactic monoid M. If M lies in $\mathbf{DO} \cap \overline{\mathbf{G_{nil}}}$ then there exists a constant k such that $D_k(L) = O(1)$. Otherwise, we have $D_k(L) = \omega(1)$ for all k.*

Proof: To obtain the upper bound, it suffices to show, by Lemma 18, that every language in $\mathcal{V}_\Sigma$ has bounded k-party complexity for some k and we argue from the definition of $\mathcal{V}_\Sigma$.

First, any language Σ_0^* has bounded two-party communication complexity since the players only need to check that the input letters they have access to indeed belong to Σ_0. Furthermore, if K counts subwords of length k modulo m, then $D_{k+1}(K) = O(1)$ because every k-tuple of input letters is available to at least one player in the $(k+1)$-party game and the value of $\binom{w}{u}$ modulo m can thus be computed with communication $k \cdot \lceil \log m \rceil$ if $|u| \leq k$. Obviously, Boolean combinations of languages with bounded k-party complexity also have bounded k-party complexity and it remains to show that if L and K have bounded k-party complexity and $L \subseteq (\Sigma - \{a\})^*$ then LaK has bounded $(k+1)$-party complexity. The players proceed as follows: each party broadcasts the identity of the player which, in their opinion, holds on the forehead the first occurrence of a in the input string. This requires $k \cdot \lceil \log k \rceil$ bits of communication and the player holding that first occurrence will be the only dissenting voice since that letter can be seen by all other parties. Since $k + 1 \geq 3$, the k remaining players now know the position of the first a and they can simulate the k-party protocols for L and K on the prefix and suffix at constant cost.

For the lower bound, if M is not in $\mathbf{DO}$ then $D_k(M) = \omega(1)$ for all k by Lemma 16. If M contains a non-nilpotent group G then $D_k(G) = \Omega(n)$ for all k by Theorem 17 and we clearly have $D_k(M) \geq D_k(G)$. So for all k, we have $D_k(M) = \omega(1)$ and, by Lemma 15, $D_k(L) = \omega(1)$. ∎

Combining this result with Theorem 14 we get

Theorem 20 *If L is a language with a neutral letter and bounded k-party communication complexity for some fixed k then L is regular and $M(L) \in \mathbf{DO} \cap \overline{\mathbf{G_{nil}}}$.*

It is worth noting that the class $\mathbf{DO} \cap \overline{\mathbf{G_{nil}}}$ is decidable. Moreover, the corresponding regular languages also have a logical characterization in terms of two-variable formulas [22] and one can easily see from the definition of $\mathcal{V}_\Sigma$ that these languages all lie in ACC^0.

5 Symmetric Functions

For $w \in \Sigma^*$, let us denote as $|w|_a$ the number of occurrences of a in w. A function $f : \Sigma^* \to \{0,1\}$ is *symmetric* if the value of f depends only on the values $|w|_a$ for $a \in \Sigma$. Intuitively $k \geq 3$ parties trying to compute a symmetric function can hardly take advantage of the fact that any $(k-1)$-tuple of input positions is seen by at least one player. In this section, we formalize this idea by showing that any symmetric function with bounded k-party communication complexity for some fixed k in fact has bounded two-party complexity.

For the sake of simplicity we will first deal with functions whose input variables are boolean. The *weight* of an input x is $|x|_1$ which we will simply denote by $|x|$. To any symmetric function $f : \{0,1\}^n \to \{0,1\}$ we can naturally associate the function $\overline{f} : \{0,\ldots,n\} \to \{0,1\}$ such that $f(x) = \overline{f}(|x|)$ for every $x \in \{0,1\}^n$. A symmetric boolean function f on n variables will be called $(t,r,p)-$periodic if $\overline{f}(a) = \overline{f}(a+p)$ for $t \leq a \leq n-r$.

Theorem 21 *If $f : \{0,1\}^n \to \{0,1\}$ is symmetric and has bounded k-party communication complexity then in fact f has bounded two-party complexity.*

In the appendix, we show how to extend this theorem to symmetric functions with non-Boolean domains. The result in the Boolean case is established through the following.

Lemma 22 *For any constants k, c with $k \geq 2$ there is an integer $N_k = N(k,c)$ such that every symmetric boolean function f that has a k-party simultaneous protocol of complexity c for the input partition in which players $X_1,\ldots,X_{k-1}$ each get N_k bits and player X_k gets the remaining $n-(k-1)N_k$ bits is (t_f, r_f, ρ_f)-periodic for some $t_f, r_f \leq (k-1)N_k$ and some $\rho_f \leq N_k$.*

Theorem 21 then follows by observing that a (t,r,ρ)-periodic function has 2-party simultaneous communication complexity roughly $2 \cdot \lceil \log(t+r+\rho) \rceil$. We only discuss a rough outline of the proof of Lemma 22 which is given in full detail in the appendix. We argue by induction on the number of players k: the base case $k = 2$ is a result of [20]. For the induction step, we use an idea similar to [17] and proceed by "player elimination". More precisely, we use Ramsey theory to show that if f has a $k+1$-party protocol of bounded cost c then there exists a sufficiently large set of inputs $\mathcal{P}$ for the foreheads of the first k players (i.e. for the information viewed by player P_{k+1}) on which player P_{k+1} always sends the same communication. This renders the $(k+1)$st player irrelevant if the input lies in $\mathcal{P}$ and this allows us to show that a symmetric function closely related to f can now be computed at communication cost c but using only k-parties.

References

[1] A. Ambainis. Upper bounds on multiparty communication complexity of shifts. In *Proc. 13th Symp. on Theoretical Aspects of Comp. Sci.*, pages 631–642, 1996.

[2] 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.

[3] D. A. M. Barrington, N. Immerman, C. Lautemann, N. Schweikardt, and D. Thérien. The Crane Beach conjecture. In *Proc. 16th Symp. on Logic in Comp. Sci. (LICS-01)*, pages 187–196, 2001.

[4] D. A. M. Barrington and H. Straubing. Superlinear lower bounds for bounded-width branching programs. *J. Comput. Syst. Sci.*, 50(3):374–381, 1995.

[5] P. Beame, T. Pitassi, and N. Segerlind. Lower bounds for lovász-schrijver systems and beyond follow from multiparty communication complexity. In *Proc. 32nd Int. Conf. on Automata, Languages and Programming (ICALP'05)*, pages 1176–1188, 2005.

[6] P. Beame and E. Vee. Time-space tradeoffs multiparty communication complexity and nearest neighbor problems. In *34th Symp. on Theory of Computing (STOC'02)*, pages 688–697, 2002.

[7] A. K. Chandra, M. L. Furst, and R. J. Lipton. Multi-party protocols. In *Proc. 15th ACM Symp. on Theory of Computing (STOC'83)*, pages 94–99, 1983.

[8] F. Chung and P. Tetali. Communication complexity and quasi-randomness. *SIAM J. Discrete Math.*, 6(1):110–123, 1993.

[9] M. Goldmann and J. Håstad. Monotone circuits for connectivity have depth (log)$^{2\text{-}(1)}$. *SIAM J. Comput.*, 27(5):1283–1294, 1998.

[10] R. L. Graham, B. L. Rotschild, and J. H. Spencer. *Ramsey Theorey*. Series in Discrete Mathematics. Wiley Interscience, 1980.

[11] V. Grolmusz. Separating the communication complexities of MOD m and MOD p circuits. In *Proc. 33rd IEEE FOCS*, pages 278–287, 1992.

[12] V. Grolmusz. The BNS lower bound for multi-party protocols in nearly optimal. *Information and Computation*, 112(1):51–54, 1994.

[13] V. Grolmusz. A weight-size trade-off for circuits and MOD m gates. In *Proc. 26th ACM STOC*, pages 68–74, 1994.

[14] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[15] N. Nisan. The communication complexity of treshold gates. In *Combinatorics, Paul Erdös is Eighty, Vol. 1*, pages 301–315, 1993.

[16] J.-E. Pin. Syntactic semigroups. In *Handbook of language theory*, volume 1, chapter 10, pages 679–746. Springer Verlag, 1997.

[17] P. Pudlák. An application of Hindman's theorem to a problem on communication complexity. *Combinatorics, Probability and Computing*, 12(5–6):661–670, 2003.

[18] J.-F. Raymond, P. Tesson, and D. Thérien. An algebraic approach to communication complexity. *Lecture Notes in Computer Science (ICALP'98)*, 1443:29–40, 1998.

[19] R. Raz. The BNS-Chung criterion for multi-party communication complexity. *Computational Complexity*, 9(2):113–122, 2000.

[20] M. Szegedy. Functions with bounded symmetric communication complexity, programs over commutative monoids, and ACC. *J. Comput. Syst. Sci.*, 47(3):405–423, 1993.

[21] P. Tesson and D. Thérien. Complete classifications for the communication complexity of regular languages. *Theory of Computing Systems*, 38(2):135–159, 2005.

[22] P. Tesson and D. Thérien. Restricted two-variable sentences, circuits and communication complexity. In *Proc. 32nd Int. Conf. on Automata, Languages and Programming (ICALP'05)*, pages 526–538, 2005.

[23] D. Thérien. Subword counting and nilpotent groups. In *Combinatorics on Words: Progress and Perspectives*, pages 195–208. Academic Press, 1983.

Appendix

We give here proofs of lemmas that were omitted in the extended abstract.

Proof of Lemma 10

Lemma 10 *For any integer $n \geq 1$, there exists a linear map $C_8 : \{0,1\}^n \to GF[8]^{39n}$ such that every $w \in C_8(\{0,1\}^n)$ is uniquely determined by any one-third of its coordinates.*

Proof:

To prove the existence of our code we only need to prove the following claim.

Claim For $c \geq 37$, with high probability a random matrix over $GF[8]$ of dimension $n \times cn$ has the property that each submatrix of dimension $n \times cn/3$ has rank n.

For any $n' < n$, n' vectors over $GF[8]$ of length $cn/3$ span less than 8^n different vectors. Thus the probability that a new random vector of length $cn/3$ falls into the space spanned by these vectors is at most $8^{n-cn/3}$. Hence, the probability that a random matrix over $GF[8]$ of dimension n by $cn/3$ is of rank less than n is at most $n \cdot 8^{n-cn/3}$. (We pick the vectors step by step and at each step we fail to pick a linearly independent vector with probability at most $8^{n-cn/3}$.) Thus the expected number of singular n by $cn/3$ submatrices of a random matrix of dimension n by cn is at most $n \cdot 8^{n-cn/3} \cdot \binom{cn}{cn/3}$. Since $\binom{cn}{cn/3} \leq 2^{\mathrm{H}(1/3)cn}$, if $c \geq 37$ then $3 - c + \mathrm{H}(1/3)c < 0$ and the expected number of singular submatrices is $2^{-\epsilon n}$ for some $\epsilon > 0$. The claim follows. ∎

By concatenating C_{RS} with C_8 and C_{U} we obtain the code $C_{\mathrm{RS}\circ 8\circ \mathrm{U}}$ with polynomial time encoding and decoding that maps n bit strings into $O(n)$ bit strings. (Note C_8 can be constructed by brute force in polynomial time as it is used only for strings of logarithmic length. Further speed-up can be achieved using $C_{\mathrm{RS}\circ\mathrm{RS}\circ 8\circ \mathrm{U}}$ codes where one would need to construct C_8 only for strings of log-log length. Using Fast Fourier Transform, $C_{\mathrm{RS}\circ\mathrm{RS}\circ 8\circ \mathrm{U}}$ can be encoded and decoded in time close to linear.)

Proof of Lemma 16

We want to establish

Lemma 16 *If M is a finite monoid outside of* **DO** *then $D_k(M) = \omega(1)$ for all k.*

Recall from Section 4 that **DO** is the class of finite monoids satisfying $(xy)^\omega(yx)^\omega(xy)^\omega = (xy)^\omega$ for some $\omega \geq 1$. The following lemma (see e.g. [21]) gives a more useful characterization of **DO**. An element $e \in M$ is *idempotent* if $e = e^2$.

Lemma 23 *If the finite monoid M is not in* **DO** *then either*

1. *There exist idempotents $a, b \in M$ and an integer $p \geq 2$ such that $(aba)^p = a$ but $(aba)^t \neq a$ if $1 \leq t \leq p-1$;*
2. *There exist elements $a, b \in M \times M$ such that ab is idempotent but for all $x, y \in M \times M$ we have $xa^2y \neq ab$ and $xb^2y \neq ab$.*

We can now proceed to establish Lemma 16.

Proof: (*Lemma 16*)

Suppose first that there are idempotents $a, b \in M$ such that $(aba)^p = a$ but $(aba)^t \neq a$ if $1 \leq t \leq p-1$. We claim that for any k there is a linear-length rectangular reduction from $GIP_{k,p}$ to the word problem of M. The reduction maps an $n \times k$ instance A of $GIP_{k,p}$ to a string of $(k+2)n$ elements of M with each block of $k+2$ elements corresponding to a column of A. The first and last elements of each block are always a and the $(i+1)$th element of the block is a b if the ith bit of the column is 0 and the identity 1_M otherwise. Since b is idempotent, the output of each such block thus multiplies out to aba if some bit in the column is 0 and to a otherwise. Hence, the value of the whole product is a iff the number of all 1 columns is 0 modulo p. Since $D_k(GIP_{k,p}) = \Omega(n)$, we have $D_K(M) = \Omega(n)$ because the length of the reduction is linear.

Suppose that there are elements $a, b \in M \times M$ such that ab is idempotent but for all $s, t \in M \times M$ we have $sa^2t \neq ab$ and $sb^2t \neq ab$. Then we claim that $Part_k$ reduces to the word problem of $M \times M$. Again, our reduction produces n blocks of $k+2$ elements of $M \times M$. The first element of each block is always an a and the last one is always ab, while the $(i+1)$th element is b if the ith bit of the column is 1 and the identity $1_{M\times M}$ otherwise. Thus, if a column of A contains r 1's, the product of monoid elements in the corresponding block is ab^rab. The product of the n blocks is thus $(ab)^{2n} = ab$ if each column contains exactly one 1. If some column of A contains two or more 1's, then the corresponding block evaluates to $a(b^2)b^{r-2}ab$ and so the product of the n blocks can be written as xb^2y and cannot be ab. Similarly, if a column is all 0, the corresponding block evaluates to aab and the n blocks multiply out to some $xa^2y \neq ab$. Since $D_k(Part_k) = \omega(1)$, we get $D_k(M \times M) = \omega(1)$. Furthermore $D_k(M \times M)$ is at most $2 \cdot D_k(M)$ so we also get $D_k(M) = \omega(1)$. ∎

Proof of Lemma 22

We now prove:

Lemma 22 *For any constants k, c with $k \geq 2$ there exists an integer $N_k = N(k,c)$ such that every symmetric boolean function f that has a k-party simultaneous protocol of complexity c for the input partition in which players $X_1, \ldots, X_{k-1}$ each get N_k bits and player X_k gets the remaining $n-(k-1)N_k$ bits is (t_f, r_f, ρ_f)-periodic for some $t_f, r_f \leq (k-1)N_k$ and some $\rho_f \leq N_k$.*

Since we are only considering protocols with bounded communication cost, we can assume without loss of generality that the protocols are simultaneous, i.e. that each player sends a constant length message to a referee, who computes the answer solely based on the messages received. Furthermore since the function to be computed is symmetric, we can do the following normalization: for any partition of input bits to k-players, where the number of bits given to player i is n_i, consider any assignment of input bits where player i's forehead gets assignment x_i. We can assume without loss of generality that the message sent by player i is a function of $|x_1|, \ldots, |x_{i-1}|, |x_{i+1}|, \ldots, |x_k|$ since the players can simulate, at no additional cost, the original protocol on the normalized input $(x'_1, \ldots, x'_n)$ where $x'_i = 1^{|x_i|}0^{n_i-|x_i|}$.

Our proof of Lemma 22 will rely crucially on the following Ramsey-theoretical lemma.

Lemma 24 ([10]) *For any integers $r, k, m_1, \ldots, m_k > 0$, there is an integer $R = R(r, k, m_1, \ldots, m_k)$ such that for each r-coloring of $[R]^k$, there exist $x_1^0, \ldots, x_k^0, d < R$ such that all points of the set $\mathcal{P} = \{(x_1, \ldots, x_k) : x_i = x_i^0 + l_i.d, 0 \leq l_i < m_i\}$ have the same color.*

We are now ready to prove lemma 22.

Proof: (*Lemma 22)*

Let Π be a simultaneous $k+1$-player protocol of cost c that computes f under a partition of the following form. Players $P_1, \ldots, P_k$ each have $N_{k+1} = R(2^c, k, m_1, \ldots, m_k) - 1$ bits written on the forehead, where R is the number obtained from Lemma 24 with each $m_i = N_k$ for $i < k$ and $m_k = k \cdot N_k!$. Player P_{k+1} gets the remaining $n - kN_{k+1}$ bits. Because f is symmetric, we can view the player's task as evaluating a function from $\{0, \ldots, N_{k+1}\}^k \times \{0, \ldots, n - kN_{k+1}\} \rightarrow \{0,1\}$ and, as we noted earlier, the message sent by player P_{k+1} is a function of k integers in the range $\{0, \ldots, N_{k+1}\}$. In the remainder of the proof we use x_i to denote the *weight* of the input string on player i's forehead.

We color an element $(t_1, \ldots, t_k) \in [R]^k$ with the message sent by player P_{k+1} in the protocol Π when the input on player P_i's forehead has weight $t_i - 1$. Because the protocol has cost c, this is indeed a 2^c coloring of $[R]^k$ and so by Lemma 24 there is a set $\mathcal{P}$ of points in $\{0, \ldots, N_{k+1}\}^k$, such that player P_{k+1} sends the same message for every point in $\mathcal{P} = \{(x_1, \ldots, x_k) : x_i = x_i^0 + l_i.d, 0 \leq l_i < m_i\}$.

On inputs where the components held by the first k parties form a point of $\mathcal{P}$, player P_{k+1} is useless and the remaining k players are effectively computing a symmetric function in the sufficiently large set $\mathcal{P}$. More precisely, for each possible input x to player P_{k+1}, define a function $\overline{f_x} : \{0, \ldots, (k-1)(N_k) + k(N_k!)\} \rightarrow \{0,1\}$, where $\overline{f_x}(u) = \overline{f}(x + \sum_{i=1}^k x_i^0 + u \cdot d)$.

We build a simultaneous k-party protocol Π_x of cost c for the symmetric function f_x for a partition in which players $P_1, \ldots, P_{k-1}$ get N_k bits each and the remaining $k(N_k!)$ bits are held by P_k: let the input to $P_1, \ldots, P_k$ be respectively $y_1, \ldots, y_k$. By definition we have

$$\overline{f_x}(y_1, \ldots, y_k) = \overline{f}(x_1^0 + y_1 \cdot d, \ldots, x_k^0 + y_k \cdot d, x)$$

and the players will therefore simulate the protocol Π on input $(x_1^0 + y_1 \cdot d, \ldots, x_k^0 + y_k \cdot d, x)$ to compute f_x: Player P_i, who sees the $(k-1)$-tuple $(y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_k)$, sends to the referee the same message that he would have sent using protocol Π on seeing the k-tuple $(x_1^0 + y_1 \cdot d, \ldots, x_{i-1}^0 + y_{i-1} \cdot d, x_{i+1}^0 + y_{i+1} \cdot d, \ldots, x_k^0 + y_k \cdot d, x)$. This is possible because x_i^0, x and d are constants known beforehand. The referee can complete the simulation because the message sent by player X_{k+1} is the same for all of these inputs. Thus Π_x correctly computes f_x with cost at most c. Applying our inductive hypothesis, we get that for each $x \leq n - kN_{k+1}$ there exist constants $t_x, r_x \leq (k-1)N_k$ and $\rho_x \leq N_k$ such that f_x is (t_x, r_x, ρ_x)-periodic i.e. $\overline{f_x}(u) = \overline{f_x}(u + \rho_x)$ for all $t_x \leq u \leq k(N_k!) - r_x$.

Let $t_f = (\sum_{i=1}^k x_i^0) + (k-1) \cdot N_k \cdot d$, let $r_f = kN_{k+1} - t_f$ and let $\rho_f = d \cdot N_k!$. Consider any w such that $t_f \leq w \leq n - r_f$ and let $u = w - t_f$. Then, $0 \leq u \leq n - kN_{k+1}$. Using results obtained above, $\overline{f}(w) = \overline{f_u}((k-1)N_k) = \overline{f_u}((k-1)N_k + j\rho_u)$ for every j such that $j\rho_u < k(N_k!) - (k-1)N_k$. Hence, $\overline{f}(w) = \overline{f_u}((k-1)N_k + N_k!) = \overline{f}(w + d.(N_k!) = \overline{f}(w + \rho_f)$. One can easily verify that $t_f, r_f \leq kN_{k+1}$ and $\rho_f \leq N_{k+1}$. This completes the induction. ∎

Theorem 21 now follows as well as:

Corollary 25 *If $f : \Sigma^n \rightarrow \{0,1\}$ is symmetric and has bounded k-party communication complexity then in fact f has bounded two-party complexity.*

Proof: Let $\Sigma = \{a_1, \ldots, a_t\}$. For any $\Sigma^0 \subseteq \Sigma$ and any word w in $(\Sigma - \Sigma^0)^*$, we denote as $f_w^{\Sigma_0}$ the symmetric function over alphabet Σ^0 defined by $f_w^{\Sigma_0}(x) = f(wx)$. We now argue by induction on t the cardinality of Σ. Our base case is Theorem 21. If $t \geq 3$ then let $\Sigma_0 = \{a_1, a_2\}$ Since f has bounded k-party complexity then so does $f_w^{\Sigma^0}$ for any w. Applying our result for binary alphabets

we get that for any w we get that $f_w^{1,2}$ is (t, r, ρ)-periodic for $t = r = (k-1)N_k$ and $\rho = N_k!$. In particular this means that the function $f_x^{\Sigma-\Sigma_0}$ is determined by the numbers $|x|_{a_1}$ and $|x|_{a_2}$ up to the thresholds t, r and modulo ρ. This can be computed at constant cost by two players and since $f_x^{\Sigma-\Sigma_0}$ is a symmetric with bounded k-party communication complexity over an alphabet of cardinality smaller than t it can be evaluated at constant bounded two-party cost by our induction hypothesis.

∎