Notes on Communication Complexity

Anil Ada

Random variables are denoted with boldface letters, not necessarily capital. If $\mathbf{x}$ is a random variable and μ a distribution, $\mathbf{x} \sim \mu$ means that $\mathbf{x}$ is distributed according to μ. The notation $\mathbf{E}[\cdot]$ and $\mathbf{Pr}[\cdot]$ is used for expectation and probability respectively. When the random variable(s) and the distribution(s) are clear from the context, the expectations and the probabilities do not have any subscripts, e.g. $\mathbf{E}[f(\mathbf{x})]$. If the distribution is clear but we would like to explicitly point out the random variables, we put the random variables as subscript, e.g. $\mathbf{E}_{\mathbf{x}}[f(\mathbf{x})]$. We also sometimes choose to make the distribution explicit in this notation, e.g. $\mathbf{E}_{\mathbf{x}\sim\mu}[f(\mathbf{x})]$. The uniform distribution is always denoted by U and the underlying set will always be clear from the context.

1 2 Player Deterministic Model

The most basic and fundamental model in communication complexity is the 2 player deterministic model (introduced in [Yao79]). The setting is as follows. We have a function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ and two players Alice and Bob. In these notes, we'll assume that $\mathcal{X} = \mathcal{Y} = \{0,1\}^n$ and $\mathcal{Z} = \{1,-1\}$. Alice gets $x \in \mathcal{X}$ and Bob gets $y \in \mathcal{Y}$. They want to collaboratively compute $F(x,y)$ by communicating with each other. Their communication consists of bits that are being transferred from one player to the other. They carry out this communication according to a **protocol** that they have agreed upon beforehand. More precisely, the protocol tells each player:

1. Whose turn it is to send a bit; the protocol determines this purely based on the communicated bits thus far, and we assume without loss of generality that Alice sends the first bit.

2. What bit to send; the protocol determines this based on the communicated bits thus far as well as the input of the player sending the bit.

The protocol also determines when communication stops and the value of the output based on the whole transcript of the communicated bits (which implies both players know the output at the end). The resource of interest is the number of communicated bits, or in other words, the length of the transcript. The goal is to compute the function with the shortest transcript possible. It is worth explicitly noting that we put no restriction on the computational capacities of Alice and Bob, and the sole interest is in the number of bits needed to communicate in order to compute the function.

Let P denote a protocol that correctly computes a function F. Denote by $\Pi_P(x, y)$ the **transcript** of protocol P for the input (x, y) (i.e. the sequence of communicated bits). The cost of P is

$$\mathrm{cost}(P) \overset{\text{def}}{=} \max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} |\Pi_P(x, y)|.$$

The **deterministic communication complexity** of F, denoted $\mathbf{D}(F)$, is the cost of the most efficient protocol that computes F correctly. That is,

$$\mathbf{D}(F) \overset{\text{def}}{=} \min_{\text{protocol } P \text{ that computes } F} \mathrm{cost}(P).$$

Unless explicitly stated otherwise (for example, we will do so in Chapter **??**), we deal with the standard setting of $\mathcal{X} = \mathcal{Y} = \{0, 1\}^n$ and $\mathcal{Z} = \{1, -1\}$, and we are interested in how fast $\mathbf{D}(F)$ grows as a function of n. Observe that every function can be trivially computed with $n + 1$ bits of communication: Alice sends x to Bob, Bob computes $F(x, y)$ and sends the result back to Alice. Hence for any F:

$$0 \leq \mathbf{D}(F) \leq n + 1.$$

In view of this, protocols of cost at most poly-log(n) are considered to be efficient and protocols of larger cost are deemed inefficient. As an example of an efficient protocol, suppose we want to determine if the majority of the bits in x *and* y is 1, i.e. is $|x| + |y| \geq n$? This function can be computed using $\lceil \log n \rceil + 1$ bits since Bob can compute the output if Alice sends him $|x|$. A canonical example of a hard function is the *equality* function which evaluates to -1 if and only if $x = y$. Intuitively one expects that for Alice and Bob to be sure that $x = y$, or detect a difference, they would have to compare x_i and y_i for all $i \in [n]$. That is, our intuition tells us that $\mathbf{D}(\text{EQUALITY}) \geq n$. But is this correct, and if it is, how do we formally prove it?

In order to prove lower bounds on communication complexity, we need to have a combinatorial understanding of what protocols do. To this end, we first observe that a protocol can be conveniently described with a binary tree as follows (see Figure 1). Each node v of the tree is labelled with the letter A or B (indicating whether the node belongs to Alice or Bob) and a function f_v. This function is of the form $f_v : \mathcal{X} \to \{0, 1\}$ if the label is A or it is of the form $f_v : \mathcal{Y} \to \{0, 1\}$ if the label is B, and it determines what bit the corresponding player communicates. Let us trace the behaviour of the protocol to understand the meaning of this tree. As always, Alice gets x and Bob gets y. First, without loss of generality, the root r is always labelled A, which means that Alice is the first to communicate a bit. Then the protocol determines what bit Alice will send by evaluating $f_r(x)$, i.e. Alice sends Bob $f_r(x)$. If $f_r(x)$ is 0, we move to the left child of the root and if $f_r(x) = 1$ we move to the right child. Without loss of generality let's assume we are at the right child, which we denote by v. If v is labelled with A, then it is again Alice's turn to speak. If it is labelled B, it is Bob's turn. And as before, the function f_v tells the player what bit to send. In this fashion we make our way down the tree until we reach a leaf node. Leaf nodes are special and they determine the output of the protocol.

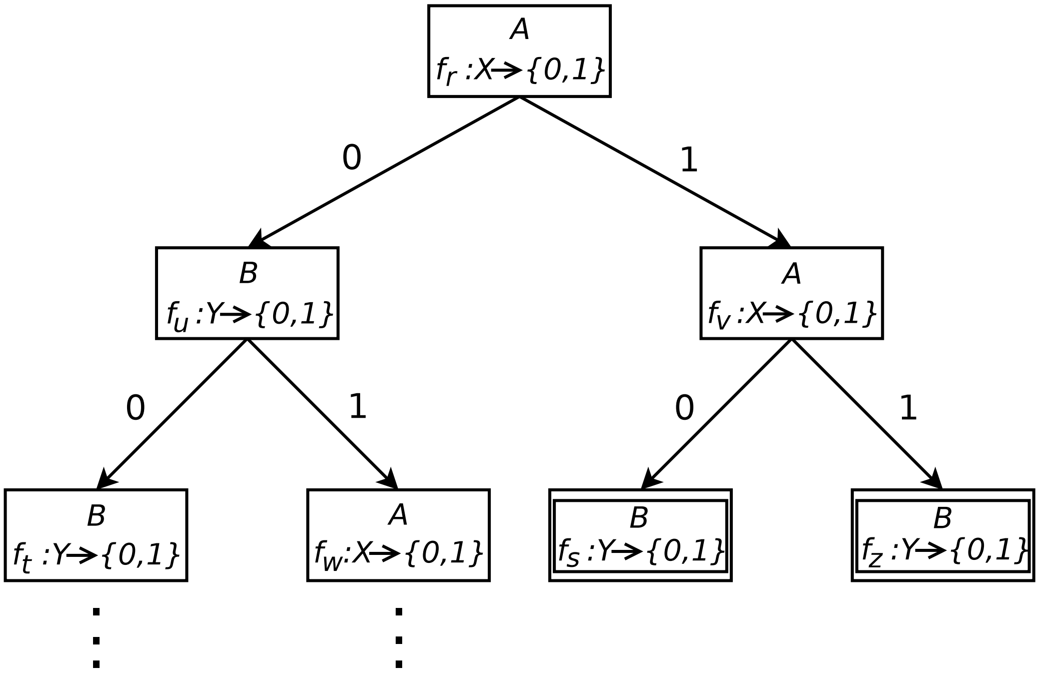
A
$f_r : X \rightarrow \{0,1\}$
0
1
B
$f_u : Y \rightarrow \{0,1\}$
A
$f_v : X \rightarrow \{0,1\}$
0
1
0
1
B
$f_t : Y \rightarrow \{0,1\}$
A
$f_w : X \rightarrow \{0,1\}$
B
$f_s : Y \rightarrow \{0,1\}$
B
$f_z : Y \rightarrow \{0,1\}$

Figure 1: A binary tree representing a protocol. Each node is labelled with A or B to indicate whose turn it is to speak. A function associated with a node tells the player what to send. Depending on whether 0 or 1 is sent, we move to the left or the right child of the node. The leaf nodes are indicated with double lines. The functions associated with them determine the output of the protocol.

Observe that every protocol can be described with such a tree and this tree description is entirely consistent with the description we provided in the beginning. In particular, whose turn it is to speak is determined based only on the communicated bits thus far and what a player sends is determined by the communicated bits as well as the input of the player. Obviously the cost of the protocol is the height of the tree.

With this point of view, we will be able to gain a very good understanding of what a protocol does when computing a function F. First we represent F by a $|\mathcal{X}| \times |\mathcal{Y}|$ matrix M_F where the rows are labelled with $x \in \mathcal{X}$, columns are labelled with $y \in \mathcal{Y}$, and $M_F[x, y] = F(x, y)$. A submatrix $\mathcal{S} \times \mathcal{T}$ where $\mathcal{S} \subseteq \mathcal{X}$ and $\mathcal{T} \subseteq \mathcal{Y}$ is called a **rectangle**. The rectangle is said to be **monochromatic** if M_F restricted to $\mathcal{S} \times \mathcal{T}$ has the same value on all of its entries. We will now see that a protocol of cost c that computes F *partitions*[1] M_F into at most 2^c monochromatic rectangles. In fact, this is the most important property of a protocol and all lower bound techniques will be based on this observation.

Proposition 1.1. *Let P be a protocol that computes $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ with at most c bits of communication. Then P induces a partition of M_F into at most 2^c monochromatic rectangles.*

To see why this is the case, let's trace once again the behaviour of the protocol down the associated tree. We start at the root which is labelled with A. The root corresponds to the whole matrix $\mathcal{X} \times \mathcal{Y}$. The function f_r is boolean and therefore partitions $\mathcal{X}$ into two sets $\mathcal{X}_0$ and $\mathcal{X}_1$: for all $x \in \mathcal{X}_0$ Alice sends 0 to Bob, and for all $x \in \mathcal{X}_1$ she sends 1. Therefore the left child of r corresponds to the rectangle $\mathcal{X}_0 \times \mathcal{Y}$ and the right child corresponds to $\mathcal{X}_1 \times \mathcal{Y}$. In some sense, if we go to the left child, we eliminate (disregard) the inputs $\mathcal{X}_1 \times \mathcal{Y}$ and our new matrix is $\mathcal{X}_0 \times \mathcal{Y}$ (this is where the input (x, y) lives). If we go to the right child, we eliminate $\mathcal{X}_0 \times \mathcal{Y}$ and our new matrix is $\mathcal{X}_1 \times \mathcal{Y}$. Note that $\mathcal{X}_0 \times \mathcal{Y}$ and $\mathcal{X}_1 \times \mathcal{Y}$ are disjoint. This process inductively continues, so for each node of the tree, there corresponds a rectangle. If a node is the descendent of another, the rectangle of the descendent will be a subset of the other. Otherwise the rectangles are disjoint. Once we reach a monochromatic rectangle, there is no need to partition it further since we can safely declare $F(x, y)$ as the value of this rectangle. Hence each leaf node corresponds to a monochromatic rectangle. Suppose the height of the tree is c, i.e. the protocol has cost c. Then there are at most 2^c leaves. Thus, the protocol partitions M_F into at most 2^c monochromatic rectangles.

It is instructive to see a different proof of the above fact. The following gives an alternative definition of a rectangle.

Proposition 1.2. *A set $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$ is a rectangle if and only if for all $(x, y), (x', y') \in \mathcal{R}$, we have $(x, y') \in \mathcal{R}$.*

An important observation is that if a protocol produces the same transcript for (x, y) and (x', y'), i.e. $\Pi(x, y) = \Pi(x', y')$, then $\Pi(x, y) = \Pi(x', y') = \Pi(x, y')$. This

[1] The word *partition* here is important. The rectangles are mutually disjoint and together cover the whole matrix M_F.

implies that all the inputs that produce a particular transcript form a rectangle. There are at most 2^c different transcripts and therefore we have at most 2^c monochromatic rectangles that partition M_F.

Proposition 1.1 immediately suggests a lower bound strategy: to show a function F has high communication complexity, show that no matter how you partition M_F into monochromatic rectangles, you need many rectangles. Let's denote by $C^D(F)$ the minimum number of rectangles in any monochromatic disjoint cover of M_F. The lower bound strategy can be restated as follows.

Corollary 1.3.

$$\mathbf{D}(F) \geq \left\lceil \log C^D(F) \right\rceil .$$

With this tool, it is now easy to show $\mathbf{D}(\text{EQUALITY}) \geq n+1$. The matrix corresponding to the *equality* function is basically the identity matrix: the diagonal elements are -1 and the off-diagonal elements are 1. Observe that no monochromatic rectangle can contain more than one -1 since if a rectangle contains the entries (a,a) and (b,b), then it also has to contain (a,b), which corresponds to a 1 entry. This means that we need at least 2^n rectangles to cover the diagonal elements, plus we need at least one rectangle to cover the 1's in the matrix. So in total we need at least $1+2^n$ rectangles and hence $\mathbf{D}(\text{EQUALITY}) \geq \lceil \log(1+2^n) \rceil = n+1$.

Although every protocol that computes F induces a partition of M_F into monochromatic rectangles, simple examples show that the converse is not true. So if some monochromatic partitions do not correspond to any protocol, how tight is Corollary 1.3? The next theorem states that the gap is not very large.

Theorem 1.4.

$$\mathbf{D}(F) \leq O(\log^2 C^D(F)).$$

Let's reiterate that Proposition 1.1 and Corollary 1.3 are the basis for all lower bound techniques in communication complexity, including the randomized model which we will discuss in the next section. In most cases it is not easy to exactly determine $C^D(F)$ so all the various lower bound techniques try to find a suitable lower bound for $C^D(F)$. For instance one might try to upper bound the size of the largest monochromatic rectangle in M_F. If all monochromatic rectangles are small, then we can conclude that we need many rectangles to partition M_F. A more interesting lower bound technique uses the rank of M_F.

Proposition 1.5.

$$\mathbf{D}(F) \geq \log \operatorname{rank} M_F.$$

Proof. Suppose a protocol P of cost c computes F and denote by $\mathcal{S}_1 \times \mathcal{T}_1, \ldots, \mathcal{S}_t \times \mathcal{T}_t$ the t monochromatic rectangles that the protocol induces ($t \leq 2^c$). For each of these rectangles $\mathcal{S}_i \times \mathcal{T}_i$, define the $|\mathcal{X}| \times |\mathcal{Y}|$ matrix $M_{\mathcal{S}_i \times \mathcal{T}_i}$ by

$$M_{\mathcal{S}_i \times \mathcal{T}_i}[x,y] = \begin{cases} M_F[x,y] & \text{if } (x,y) \in \mathcal{S}_i \times \mathcal{T}_i \\ 0 & \text{otherwise} \end{cases}$$

These matrices are like the indicator matrices of the rectangles. Obviously we have $M_F = \sum_{i=1}^{t} M_{\mathcal{S}_i \times \mathcal{T}_i}$. By the subadditivity of the rank, we have $\operatorname{rank} M_F \leq \sum_{i=1}^{t} \operatorname{rank} M_{\mathcal{S}_i \times \mathcal{T}_i}$. Since each $M_{\mathcal{S}_i \times \mathcal{T}_i}$ has rank at most 1, we conclude that $\operatorname{rank} M_F$ is at most $t \leq 2^c$, i.e. $c \geq \log \operatorname{rank} M_F$. □

Arguably the most famous open problem in communication complexity is whether the rank lower bound is close to being tight.

Conjecture 1.6 (Log Rank Conjecture [LS88])**.** *There is some universal constant k such that*

$$\mathbf{D}(F) \leq O(\log^k \operatorname{rank} M_F).$$

Needless to say there are other lower bound techniques and each has its own advantages depending on the particular function we are dealing with.

One of the well-studied restrictions of the deterministic model is called the **simultaneous** model. Here, the players are not allowed to interact with each other. Upon receiving their inputs, the players send a message to an external referee. The referee, who does not see the players' inputs, determines the output based on these messages. The cost is the number of bits sent to the referee and we denote by $\mathbf{D}^{||}(F)$ the deterministic simultaneous communication complexity of F.

2 Randomized Model

The previous section introduced the most basic communication complexity model. In this section we will introduce the randomized model which has a variety of interesting applications.

A natural way to extend the deterministic model to utilize randomness is to allow each player to privately flip coins and make decisions based on the outcomes of those coin flips. Normally, we have to allow some probability of error in computing the function correctly. To make this more concrete, let's say that Alice has access to a random binary string $\mathbf{r}_A$ and Bob has access to a random binary string $\mathbf{r}_B$. Then a randomized protocol computes F with ϵ error if

$$\forall (x,y) \in \mathcal{X} \times \mathcal{Y}, \quad \mathbf{Pr}\left[F(x,y) \neq P(x,y)\right] \leq \epsilon,$$

where $P(x,y)$ denotes the output of the protocol and the probability is over the random choices of $\mathbf{r}_A$ and $\mathbf{r}_B$. The cost of a randomized protocol is the maximum number of bits communicated, where the maximum is over all inputs and random strings. It is worth making it clear that the random strings being used by the players do not count towards the cost at all. We denote by $\mathbf{R}^{\epsilon}_{\text{pri}}(F)$ the **randomized communication complexity** of F with ϵ-error, i.e. the cost of the most efficient randomized protocol that computes F with ϵ-error (the subscript 'pri' will be clarified shortly). We are mainly interested in the case where $\epsilon < 1/2$ is some constant. The particular choice of the constant does not matter as it can be shown that it affects the communication complexity by only a constant factor.

Let us revisit the *equality* function and demonstrate the power of randomness. One might be tempted to think that even in the randomized model, the players are bound to check whether $x_i = y_i$ for most of the $i \in [n]$ to convince themselves that the two strings are equal or not (for instance the strings might differ in just one coordinate). On the contrary, by using a clever protocol, the players can compute $\mathrm{EQUALITY}(x, y)$ with high probability using only $O(\log n)$ bits of communication. We describe this protocol now.

To avoid confusion, for this protocol let's denote Alice's input by $a = a_0 a_1 \ldots a_{n-1}$ and Bob's input by $b = b_0 b_1 \ldots b_{n-1}$. The players fix some prime number $p \in [n^2, 2n^2]$. Alice views her input as the polynomial

$$q_A(x) = a_0 + a_1 x + a_2 x^2 + \cdots a_{n-1} x^{n-1} \mod p$$

over $\mathbb{Z}_p$, and Bob views his input as the polynomial

$$q_B(x) = b_0 + b_1 x + b_2 x^2 + \cdots b_{n-1} x^{n-1} \mod p.$$

Then Alice chooses uniformly at random an element $\mathbf{z} \in \mathbb{Z}_p$, and sends Bob $\mathbf{z}$ as well as $q_A(\mathbf{z})$. This requires $O(\log n)$ bits of communication. Bob computes $q_B(\mathbf{z})$, compares it to $q_A(\mathbf{z})$, and declares the output to be 1 if they are the same, 0 otherwise. It is easy to see that if $a = b$, then the protocol is always correct. On the other hand, if $a \neq b$, the players make a mistake if $q_A(\mathbf{z}) = q_B(\mathbf{z})$, i.e. $q_A - q_B(\mathbf{z}) = 0$. Note that $q_A - q_B$ is a polynomial of degree at most $n - 1$ and therefore has at most $n - 1$ roots. The players make an error if Alice accidentally picks one of the roots so the probability of error is at most $\frac{n-1}{p} \leq \frac{1}{n}$.

The model we have just introduced is called the "private-coin" model because each player has his/her own private random string. A perhaps less natural but more useful model is the "public-coin" model in which players share a common random string. It is clear that the public-coin model is stronger than the private-coin model and therefore a lower bound in the public-coin model immediately translates into a lower bound in the private-coin model (and we are mainly interested in lower bounds). Furthermore, it is well known that the two models are pretty much equivalent when the error probability is a constant: the communication complexity of a function in the private-coin model is at most $O(\log n)$ more than the communication complexity in the public-coin model [New91]. For these two reasons, and the fact that it is easier to reason about public-coin protocols, our discussion will be about the public-coin model only. Therefore, we drop the subscript 'pri' and denote by $\mathbf{R}^\epsilon(F)$ the randomized communication complexity of F in the public-coin model.

Going back to the *equality* example, let's show $\mathbf{R}^\epsilon(F) = O(1)$ for a constant error probability ϵ. Let $\mathbf{r} \in \{0,1\}^n$ denote the public random string. Alice sends Bob $\langle x, \mathbf{r} \rangle_2 \overset{\text{def}}{=} x_1 \mathbf{r}_1 + \cdots + x_n \mathbf{r}_n \mod 2$ and Bob compares this value to $\langle y, \mathbf{r} \rangle_2$. If they are the same, he declares the output to be 1, otherwise he outputs 0. If $x = y$ then this protocol never fails. If on the other hand $x \neq y$, then it is easy to see that the inner products will be equal with probability exactly $1/2$. So the error probability of the protocol is $1/2$. If we repeat this protocol k times with fresh random strings, it is easy to see that the error probability can be reduced to $1/2^k$.

Now that we have seen some interesting upper bounds, let's turn our attention to proving lower bounds. As mentioned earlier, one of the reasons for working with public-coin protocols rather than private-coin protocols is that public-coin protocols are easier to study and understand. A useful way of viewing a public-coin randomized protocol of cost c is as a probability distribution over deterministic protocols, each of cost at most c. Once the random string r is fixed, what the players do is totally deterministic. So the players follow a deterministic protocol P_r that corresponds to the random string r. The success criterion for a randomized protocol is equivalent to saying that for all inputs, at least $1 - \epsilon$ fraction of the deterministic protocols should produce the correct answer. Consider a matrix where the rows are labelled with all the possible P_r and the columns are labelled with the inputs (x, y). At the entry corresponding to a particular P_r and (x, y) we put a 1 if $P_r(x, y) = F(x, y)$, and 0 otherwise. The success criterion for the randomized protocol tells us that each column contains at least $1 - \epsilon$ fraction of 1's. So in total, the whole matrix has at least $1 - \epsilon$ fraction of 1's. This implies that there must be at least one row that contains at least $1 - \epsilon$ fraction of 1's. To sum up, if there is an ϵ-error randomized protocol for F of cost c, then there must be a deterministic protocol P^* of cost at most c such that

$$\mathbf{Pr}\left[F(\mathbf{x}, \mathbf{y}) \neq P^*(\mathbf{x}, \mathbf{y})\right] \leq \epsilon.$$

In fact, it is not difficult to see that the above statement is true for *any* probability distribution over the inputs (x, y). This property of a randomized protocol is the basis for all lower bound techniques because arguing against a deterministic protocol that makes some error is much easier than arguing directly against a randomized protocol. In particular, all the insight we have about deterministic protocols can be put to use in this setting.

Before moving forward, let's make the formal definition of the *distributional* communication complexity model that we have just motivated. Let μ be a distribution over $\mathcal{X} \times \mathcal{Y}$. The ϵ-error **distributional complexity** of F under μ is denoted by $\mathbf{D}_\mu^\epsilon(F)$ and is defined to be the minimum cost of a deterministic protocol P such that

$$\mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim\mu}\left[F(\mathbf{x}, \mathbf{y}) \neq P(\mathbf{x}, \mathbf{y})\right] \leq \epsilon.$$

We have already proved that for any μ, $\mathbf{R}^\epsilon(F) \geq \mathbf{D}_\mu^\epsilon(F)$. It turns out a much stronger relationship holds:

Proposition 2.1 ([Yao83])**.**

$$\mathbf{R}^\epsilon(F) = \max_\mu \mathbf{D}_\mu^\epsilon(F).$$

The proof easily follows from von Neumann's Minimax Theorem.

In light of the relationship between randomized communication complexity and distributional communication complexity, we arrive at an obvious lower bound strategy: to prove lower bounds for $\mathbf{R}^{\epsilon}(F)$, pick your favorite distribution μ and prove a lower bound for $\mathbf{D}^{\epsilon}_{\mu}(F)$. As mentioned earlier, this is essentially how all lower bound arguments proceed. Given that a cost c deterministic protocol that computes F partitions M_F into at most 2^c monochromatic rectangles, a protocol that computes F with ϵ fraction of error partitions M_F into at most 2^c "almost" monochromatic rectangles, on average (not all rectangles that the protocol induces must be almost monochromatic but a good fraction must be). To rule out such a possibility with a small c, there are various tactics one can try. Arguably the most famous one is the so called *discrepancy method*. The idea is to show a lower bound for $\mathbf{D}^{\epsilon}_{\mu}(F)$ by showing that every (large enough) rectangle in M_F is balanced in the sense that there are roughly the same fraction of 1's and -1's.

Let's now mathematically formalize the discrepancy method. Let $\mathcal{S}\times\mathcal{T}$ be a rectangle, where $\mathcal{S}\subseteq\mathcal{X}$ and $\mathcal{T}\subseteq\mathcal{Y}$. For a distribution μ over $\mathcal{X}\times\mathcal{Y}$, define the discrepancy of the rectangle $\mathcal{S}\times\mathcal{T}$ with respect to F and μ as the absolute value of the difference between the weight of the 1's and the weight of the -1's in $\mathcal{S}\times\mathcal{T}$, i.e.

$$\begin{aligned}\mathrm{disc}_{\mu}(F,\mathcal{S}\times\mathcal{T}) \overset{\text{def}}{=} \big|&\mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim\mu}\left[F(\mathbf{x},\mathbf{y})=1 \text{ and } (\mathbf{x},\mathbf{y})\in\mathcal{S}\times\mathcal{T}\right] \\ &- \mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim\mu}\left[F(\mathbf{x},\mathbf{y})=-1 \text{ and } (\mathbf{x},\mathbf{y})\in\mathcal{S}\times\mathcal{T}\right]\big| \\ =& \left|\sum_{(x,y)\in\mathcal{S}\times\mathcal{T}} F(x,y)\mu(x,y)\right|.\end{aligned}$$

The **discrepancy** of F is the maximum discrepancy over all rectangles:

$$\mathrm{disc}_{\mu}(F) \overset{\text{def}}{=} \max_{\mathcal{S}\times\mathcal{T}} \mathrm{disc}_{\mu}(F,\mathcal{S}\times\mathcal{T}).$$

The discrepancy method (see e.g. [CG88]) says that to lower bound $\mathbf{D}^{\epsilon}_{\mu}(F)$, it suffices to upper bound the discrepancy $\mathrm{disc}_{\mu}(F)$.

Proposition 2.2 (Discrepancy Method)**.**

$$\mathbf{D}^{\epsilon}_{\mu}(F) \geq \log\left(\frac{1-2\epsilon}{\mathrm{disc}_{\mu}(F)}\right).$$

Proof. Let $\mathbf{D}^{\epsilon}_{\mu}(F)=c$, so there is a deterministic protocol P of cost c that computes F with ϵ error under μ. Let $\mathcal{S}_1\times\mathcal{T}_1,\ldots,\mathcal{S}_t\times\mathcal{T}_t$, $t\leq 2^c$, be the rectangles that P induces. We denote by $P(\mathcal{S}_i\times\mathcal{T}_i)$ the value the protocol outputs for the inputs $(x,y)\in\mathcal{S}_i\times\mathcal{T}_i$.

Then,

$$\begin{aligned}
1-2\epsilon &\leq \left|\mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim\mu}\left[F(\mathbf{x},\mathbf{y})=P(\mathbf{x},\mathbf{y})\right]-\mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim\mu}\left[F(\mathbf{x},\mathbf{y})\neq P(\mathbf{x},\mathbf{y})\right]\right| \\
&= \left|\sum_{(x,y)} F(x,y)P(x,y)\mu(x,y)\right| \\
&= \left|\sum_{i=1}^{t}\sum_{(x,y)\in\mathcal{S}_i\times\mathcal{T}_i} F(x,y)P(x,y)\mu(x,y)\right| \\
&= \left|\sum_{i=1}^{t} P(\mathcal{S}_i\times\mathcal{T}_i)\sum_{(x,y)\in\mathcal{S}_i\times\mathcal{T}_i} F(x,y)\mu(x,y)\right| \\
&\leq \sum_{i=1}^{t}\left|P(\mathcal{S}_i\times\mathcal{T}_i)\right|\left|\sum_{(x,y)\in\mathcal{S}_i\times\mathcal{T}_i} F(x,y)\mu(x,y)\right| \\
&= \sum_{i=1}^{t}\mathrm{disc}_\mu(F,\mathcal{S}_i\times\mathcal{T}_i) \\
&\leq t\cdot\mathrm{disc}_\mu(F)\leq 2^c\cdot\mathrm{disc}_\mu(F).
\end{aligned}$$

Rearranging, we get $2^c \geq \frac{1-2\epsilon}{\mathrm{disc}_\mu(F)}$. □

Let's see the discrepancy method in action by showing an exponentially small upper bound on the discrepancy of the *inner-product* function IP under the uniform distribution. The inner product function is defined as $\mathrm{IP}(x,y)=(-1)^{\sum_i x_i y_i}$. For a real valued matrix M, let $\|M\|$ denote its spectral norm, i.e. $\|M\| = \max_{u:\|u\|_2=1}\|Mu\|$. It turns out that it is easy to bound the discrepancy of a function under the uniform distribution in terms of the spectral norm of M_F.

Proposition 2.3.

$$\mathrm{disc}_U(F)\leq\frac{\|M_F\|}{2^n}.$$

Proof. Let $\mathcal{S}\times\mathcal{T}$ be a rectangle. Denote by $\mathbf{1}_\mathcal{S}$ the indicator vector for $\mathcal{S}$, i.e. the 2^n dimensional vector which has a 1 for positions corresponding to $\mathcal{S}$ and 0 everywhere else. Similarly for $\mathbf{1}_\mathcal{T}$. By the definition of discrepancy,

$$\mathrm{disc}_U(F,\mathcal{S}\times\mathcal{T})=\frac{1}{2^{2n}}\left|\sum_{(x,y)\in\mathcal{S}\times\mathcal{T}} F(x,y)\right|.$$

It is not hard to verify that the right hand side is equal to

$$\frac{1}{2^{2n}}|\mathbf{1}_\mathcal{S}^T\cdot M_F\cdot\mathbf{1}_\mathcal{T}|=\frac{1}{2^{2n}}|\langle M_F\mathbf{1}_\mathcal{T},\mathbf{1}_\mathcal{S}\rangle|.$$

Using the Cauchy-Schwarz inequality, we get $\frac{1}{2^{2n}}|\langle M_F\mathbf{1}_{\mathcal{T}}, \mathbf{1}_{\mathcal{S}}\rangle| \leq \frac{1}{2^{2n}}\|M_F\mathbf{1}_{\mathcal{T}}\| \cdot \|\mathbf{1}_{\mathcal{S}}\|$. Then by the definition of the spectral norm we conclude:

$$\begin{aligned}
\mathrm{disc}_U(F, \mathcal{S} \times \mathcal{T}) &\leq \frac{1}{2^{2n}}\|M_F\mathbf{1}_{\mathcal{T}}\|_2 \cdot \|\mathbf{1}_{\mathcal{S}}\|_2 \\
&\leq \frac{1}{2^{2n}}\|M_F\|\|\mathbf{1}_{\mathcal{T}}\|_2\|\mathbf{1}_{\mathcal{S}}\|_2 \\
&\leq \frac{1}{2^{2n}}\|M_F\|\sqrt{|\mathcal{T}|}\sqrt{|\mathcal{S}|} \\
&\leq \frac{\|M_F\|}{2^n}.
\end{aligned}$$

□

The spectral norm of M_{IP}, where IP denotes the *inner product* function, is easy to calculate. It is well known that the spectral norm of a matrix M is equal to the largest singular value of M, $\sigma_{\max}(M)$, which in return is equal to the square-root of the largest eigenvalue of $M^T M$. Using the definition of IP, one can easily check that $M_{\mathrm{IP}}^T M_{\mathrm{IP}} = 2^n I$, where I denotes the identity matrix. Therefore for all u, $M_{\mathrm{IP}}^T M_{\mathrm{IP}} u = 2^n u$. This implies $\lambda_{\max}(M_{\mathrm{IP}}^T M_{\mathrm{IP}}) = 2^n$, or in other words, $\|M_{\mathrm{IP}}\| = 2^{n/2}$. Using Proposition 2.3, we have $\mathrm{disc}_U(\mathrm{IP}) \leq 1/2^{n/2}$. Plugging this into the Discrepancy Method (Proposition 2.2), we conclude

$$\mathbf{R}^{\epsilon}(\mathrm{IP}) \geq \frac{n}{2} + \log(1 - 2\epsilon).$$

Is the discrepancy method the all powerful method that will give us tight lower bounds for any function? The answer is no and let's explain why. First note that for any function, achieving error probability 1/2 is trivial since we can just output a random bit. The discrepancy method is a very strong tool in the following sense. If one shows a lower bound of say $\Omega(n)$ on the randomized communication complexity of a function using the discrepancy method, then the lower bound applies to protocols that make $1/2 - 1/\exp(n)$ probability of error, i.e. error exponentially close to 1/2. For example, in the case of *inner product*, suppose we allow the protocol to make error $\epsilon = 1/2 - 1/2^{\alpha n}$ for some constant $\alpha < 1/2$. Then $\mathbf{R}^{\epsilon}(\mathrm{IP}) \geq n/2 + \log(1-2\epsilon) = n/2 - \alpha n = \Omega(n)$. When our primary interest is in constant probability of error, this is an overkill. There are many functions that require $\Omega(n)$ communication complexity when the error probability is a constant but has $O(1)$ communication complexity once we allow the error probability to be $1/2 - 1/\exp(n)$. In particular, it is well known that the discrepancy method cannot yield good lower bounds for any function with small *non-deterministic* communication complexity. A canonical example is the famous *disjointness* function and to handle such functions, one needs to develop more sophisticated tools. On this note, we end our discussion of the 2 party randomized communication complexity model and move on to the non-deterministic model.

3 Non-Deterministic Model

Non-determinism is a very important notion in computational complexity theory. At a high level, the motivation is to understand whether verifying a given solution to a problem is easier than finding a solution. The answer of course depends on which computational model we are dealing with. In communication complexity, non-determinism can be much more efficient and in this section, we will briefly go over non-deterministic communication complexity.

As usual, there are two equivalent ways to view the non-deterministic model. We can view it as a model in which players are allowed to take non-deterministic steps, or we can view it as a proof verification process. We prefer to use the latter version. As before, Alice gets $x \in \mathcal{X}$ and Bob gets $y \in \mathcal{Y}$. We also have a third player called God, who sees the input (x, y) and furnishes a proof string z which is then communicated to both Alice and Bob. Upon receiving z, Alice and Bob communicate with each other and agree on an output. If $F(x, y) = -1$, there must be at least one proof string z that leads Alice and Bob to output -1. On the other hand, if $F(x, y) = 1$, no matter what proof string Alice and Bob receive, they should output 1. We include in the cost the length of z. The **non-deterministic communication complexity** of F, denoted by $\mathbf{N}^{-1}(F)$, is the cost of the most efficient non-deterministic protocol that computes F as described above. The **co-non-deterministic communication complexity** of F is denoted by $\mathbf{N}^{1}(F)$ and is defined to be equal to $\mathbf{N}^{-1}(-F)$, the non-deterministic complexity of the negation of F.[2]

Recall the definition of the *disjointness* function. It is straightforward to see that $\mathbf{N}^{1}(\mathrm{DISJ}) \leq O(\log n)$. God provides an index $i \in [n]$ and Alice and Bob exchange x_i and y_i with each other in order to check if $x_i = y_i = 1$. If x and y are not disjoint, then there is an index i such that $x_i = y_i = 1$. If not, for no index we will have $x_i = y_i = 1$. A similar protocol also shows that $\mathbf{N}^{1}(\mathrm{EQUALITY}) \leq O(\log n)$. On the other hand, intuitively it seems unlikely that $\mathbf{N}^{-1}(\mathrm{EQUALITY})$ is small; how can God furnish a short proof that two strings are equal?

In Section 1, we defined $C^{D}(F)$ as the minimum number of disjoint monochromatic rectangles needed to partition M_F. Define $C^{z}(F)$ as the minimum number of possibly intersecting monochromatic rectangles needed to cover the z-entries of M_F. This quantity accurately characterizes the non-deterministic communication complexity of F.

Proposition 3.1.

$$\log C^{z}(F) \leq \mathbf{N}^{z}(F) \leq 2 + \log C^{z}(F).$$

We skip the proof of this proposition but remark that it is quite straightforward and uses the fact that once the proof string is fixed, Alice and Bob follow a deterministic protocol.

[2] Note that in the literature, $\mathbf{N}^{-1}(F)$ is almost always denoted by $\mathbf{N}^{1}(F)$ and $\mathbf{N}^{1}(F)$ is denoted by $\mathbf{N}^{0}(F)$. This is due to the range of the function F, which is often $\{0, 1\}$ as opposed to $\{1, -1\}$ as in here.

Needless to say, Proposition 3.1 is the backbone of all lower bound techniques for the non-deterministic model. Going back to the *equality* example, we see that a monochromatic rectangle can cover at most one -1 entry and therefore we need 2^n rectangles to cover all the -1 entries.

At the end of the previous section (Section 2), we mentioned that the discrepancy method fails to give good lower bounds on the randomized communication complexity of functions that have low non-deterministic communication complexity. Let us now make this formal.

Proposition 3.2 (see e.g. [Cha08] Lemma 6.17)**.** *Let F be such that* $\min\{\mathbf{N}^1(F), \mathbf{N}^{-1}(F)\} = c$. *Then, under any distribution μ over the inputs,*

$$\mathrm{disc}_\mu(F) \geq \Omega(1/2^c).$$

4 Multiparty Number on the Forehead Model

There are various ways one can extend the two player model to more players. Given $F : \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_k \to \mathcal{Z}$, the most natural generalization would be to distribute the input $(x_1, x_2, \ldots, x_k)$ so that Player i gets x_i. This is called the "number in the hand" multiparty model; it is an interesting model with nice applications. In this thesis however, we are interested in the so called "number on the forehead" multiparty model in which Player i sees all x_j with $j \neq i$. We visualize this scenario as x_i being written on the forehead of Player i. Once the input is distributed, the players once again follow a protocol in order to compute $F(x_1, \ldots, x_k)$. The description of a protocol is equivalent to the 2 player model and when a player communicates a bit, all the other players get to see it.

We can generalize the *equality* example seen in the 2 party setting to an arbitrary number of players in the obvious way: let $\mathrm{EQ}_k(x_1, x_2, \ldots, x_k) = 1$ if and only if $x_1 = \cdots = x_k$. When $k = 2$, we saw that the deterministic communication complexity of *equality* is $n+1$. On the other hand, when $k > 2$, it is easy to see that the communication complexity drops down to just 2 bits. Player 1 checks if $x_2 = x_3 = \cdots = x_k$ and Player 2 checks if $x_1 = x_3 = x_4 = \cdots = x_k$. If both equalities are confirmed, all the strings are equal, otherwise they are not. This example demonstrates the power of the multiparty number on the forehead model. The overlap of information among the players can be exploited to give efficient protocols.

We denote by $\mathbf{D}_k(F)$, $\mathbf{D}_k^{||}(F)$, $\mathbf{R}_k^\epsilon(F)$, $\mathbf{D}_{k,\mu}^\epsilon(F)$, and $\mathbf{N}_k^{-1}(F)$ the k-party deterministic, deterministic simultaneous, randomized, distributional and non-deterministic communication complexity of F respectively. In the 2 player setting, the single most important property of a protocol was the fact that it induced rectangles. For the k party model with $k \geq 3$, the appropriate generalization of the notion of a rectangle is called a cylinder intersection. A **cylinder** $\mathcal{C}_i$ in the ith direction is a subset of the input space $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ such that membership in $\mathcal{C}_i$ does not depend on the ith coordinate, i.e. if $(x_1, \ldots, x_i, \ldots, x_k) \in \mathcal{C}_i$ then $(x_1, \ldots, x_i', \ldots, x_k) \in \mathcal{C}_i$ for all $x_i' \in X_i$ (see Figure

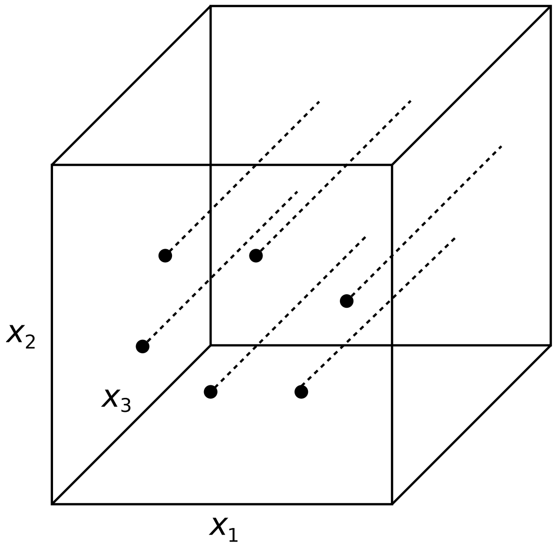

Figure 2: A cylinder in the 3rd direction. The bold dots represent a subset of $\mathcal{X}_1 \times \mathcal{X}_2$, which then completely determines the corresponding cylinder.

2). A **cylinder intersection** $\mathcal{C}$ is just an intersection of k cylinders, one in each direction, i.e. $\mathcal{C} = \cap_{i=1}^{k} \mathcal{C}_i$ where $\mathcal{C}_i$ is a cylinder in the ith direction. It is important to take a moment and observe that when $k = 2$, this definition corresponds to the notion of a rectangle (see Figure 3).

In Proposition 1.2, we gave an alternative definition of a rectangle. The same characterization holds also for cylinder intersections. A set of k points

$$(x'_1, x_2, \ldots, x_k), (x_1, x'_2, \ldots, x_k), \ldots, (x_1, x_2, \ldots, x'_k)$$

in $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ is called a **star** if $x'_i \neq x_i$ for all $i \in [k]$. The point $(x_1, x_2, \ldots, x_k)$ is called the center of the star.

Proposition 4.1. *A set $\mathcal{C} \subseteq \mathcal{X}_1 \times \cdots \mathcal{X}_k$ is a cylinder intersection if and only if for every star in $\mathcal{C}$, its center is also contained in $\mathcal{C}$.*

Now it is easy to see that a multiparty protocol induces a partition of M_F into monochromatic cylinder intersections. Here M_F denotes the k-dimensional matrix (often called a tensor) such that $M_F[x_1, \ldots, x_k] = F(x_1, \ldots, x_k)$ for all $(x_1, \ldots, x_k) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$.

Proposition 4.2. *Let P be a deterministic protocol that computes $F : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathcal{Z}$ with at most c bits of communication. Then P induces a partition of M_F into at most 2^c monochromatic cylinder intersections.*

Proof. As in the 2 player case, it is easy to see that if the protocol produces the same transcript for all the elements of a star, then the protocol must produce the same

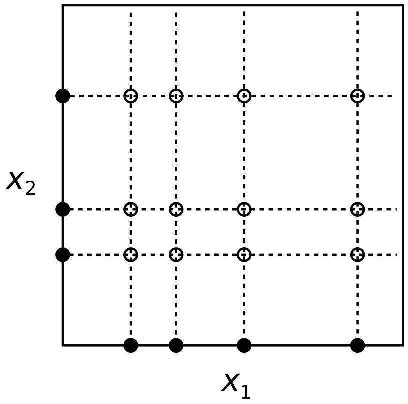

Figure 3: A rectangle as the intersection of two cylinders.

transcript for the center of the star as well. Therefore, the set of all points corresponding to a particular transcript forms a cylinder intersection. There are at most 2^c possible transcripts and the statement follows. □

The definition of discrepancy naturally generalizes to cylinder intersections. We will now make the formal definition with respect to complex valued functions $F : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathbb{C}$ since the definition does not call for a restriction on F to be boolean. Furthermore, there are situations where one can be interested in the discrepancy of complex valued functions.

Given a cylinder intersection $\mathcal{C} = \cap_{i=1}^k \mathcal{C}_i$, let ϕ_i denote the characteristic function of $\mathcal{C}_i$, i.e. $\phi_i(x_1, \ldots, x_k) = 1$ if $(x_1, \ldots, x_k) \in \mathcal{C}_i$ and $\phi_i(x_1, \ldots, x_k) = 0$ otherwise. Then $\phi \stackrel{\text{def}}{=} \prod_{i=1}^k \phi_i$ is the characteristic function of $\mathcal{C}$. For a distribution μ over $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$, and a cylinder intersection $\mathcal{C}$, the discrepancy of F with respect to μ and $\mathcal{C}$ is

$$\begin{aligned}\operatorname{disc}_\mu(F, \mathcal{C}) &\stackrel{\text{def}}{=} \left| \sum_{(x_1,\ldots,x_k)\in\mathcal{C}} F(x_1, \ldots, x_k)\mu(x_1, \ldots, x_k) \right| \\ &= \left| \mathbf{E}_{(\mathbf{x}_1,\ldots,\mathbf{x}_k)\sim\mu} \left[F(\mathbf{x}_1, \ldots, \mathbf{x}_k)\phi(\mathbf{x}_1, \ldots, \mathbf{x}_k) \right] \right| . \qquad (1)\end{aligned}$$

The **discrepancy** of F is the maximum discrepancy over all cylinder intersections:

$$\operatorname{disc}_\mu(F) \stackrel{\text{def}}{=} \max_{\mathcal{C}} \operatorname{disc}_\mu(F, \mathcal{C}).$$

The discrepancy method generalizes to the multiparty setting with the same proof [BNS92].

Proposition 4.3 (Discrepancy Method). *Let $F : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \{1, -1\}$, and μ a distribution over $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$. Then,*

$$\mathbf{D}^\epsilon_{k,\mu}(F) \geq \log\left(\frac{1 - 2\epsilon}{\operatorname{disc}_\mu(F)}\right).$$

In the two party setting, we saw how to upper bound the discrepancy of F using the spectral norm of the matrix M_F. This linear algebraic technique does not work in the multiparty setting because M_F is no longer a matrix and a cylinder intersection is a relatively complicated combinatorial object. There is still however a famous trick one can resort to in order to deal with cylinder intersections: repeatedly apply Cauchy-Schwarz inequality to get rid of the cylinder intersection.

Lemma 4.4 ([CT93, Raz00])**.** *Let $F : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathbb{C}$ and let μ_i be a distribution over $\mathcal{X}_i$. Define the distribution μ as the product of the μ_i, that is $\mu(x_1, \ldots, x_k) = \mu_1(x_1) \cdots \mu_k(x_k)$. Then,*

$$(\mathrm{disc}_\mu(F))^{2^k} \leq \mathbf{E}_{\substack{\mathbf{x}_1^0,\ldots,\mathbf{x}_k^0 \\ \mathbf{x}_1^1,\ldots,\mathbf{x}_k^1}} \left[\prod_{u \in \{0,1\}^k} \mathfrak{C}^{u_1+\cdots+u_k}(F(\mathbf{x}_1^{u_1}, \ldots, \mathbf{x}_k^{u_k})) \right], \tag{2}$$

where in the expectation, $(\mathbf{x}_i^0, \mathbf{x}_i^1)$ are distributed according to the product distribution $\mu_i \times \mu_i$.

Proof. We prove the lemma by induction on k and in order to reduce clutter, we will prove it for real valued functions as opposed to complex valued functions. The proofs are identical. Our induction hypothesis is that the lemma is true for every function with $k-1$ players. Let $\mathcal{C} = \cap_{1 \leq i \leq k} \mathcal{C}_i$ be an arbitrary cylinder intersection with the characteristic function $\phi(x_1, \ldots, x_k) = \phi_1(x_1, \ldots, x_k) \cdots \phi_k(x_1, \ldots, x_k)$. Recall that ϕ_i does not depend on x_i. Then, writing the discrepancy as in (1), we have

$$\begin{aligned}
\mathrm{disc}_\mu(F, \mathcal{C}) &= \left| \mathbf{E} \left[F(\mathbf{x}_1, \ldots, \mathbf{x}_k) \prod_{i=1}^{k} \phi_i(\mathbf{x}_1, \ldots, \mathbf{x}_k) \right] \right| \\
&\leq \mathbf{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_{k-1}} \left[\left| \phi_k(\mathbf{x}_1, \ldots, \mathbf{x}_k) \mathbf{E}_{\mathbf{x}_k} \left[F(\mathbf{x}_1, \ldots, \mathbf{x}_k) \prod_{i=1}^{k-1} \phi_i(\mathbf{x}_1, \ldots, \mathbf{x}_k) \right] \right| \right].
\end{aligned}$$

Squaring both sides and using the consequence $\mathbf{E}[\mathbf{Z}]^2 \leq \mathbf{E}[\mathbf{Z}^2]$ of Cauchy-Schwarz inequality, we obtain

$$\begin{aligned}
&\mathrm{disc}_\mu(F, \mathcal{C})^2 \\
&\leq \mathbf{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_{k-1}} \left[\phi_k(\mathbf{x}_1, \ldots, \mathbf{x}_k)^2 \mathbf{E}_{\mathbf{x}_k} \left[F(\mathbf{x}_1, \ldots, \mathbf{x}_k) \prod_{i=1}^{k-1} \phi_i(\mathbf{x}_1, \ldots, \mathbf{x}_k) \right]^2 \right] \\
&= \mathbf{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_{k-1}} \left[\mathbf{E}_{\mathbf{x}_k} \left[F(\mathbf{x}_1, \ldots, \mathbf{x}_k) \prod_{i=1}^{k-1} \phi_i(\mathbf{x}_1, \ldots, \mathbf{x}_k) \right]^2 \right].
\end{aligned} \tag{3}$$

If we let

$$F^{x_k^0, x_k^1}(x_1, \ldots, x_{k-1}) \overset{\mathrm{def}}{=} F(x_1, \ldots, x_{k-1}, x_k^0) F(x_1, \ldots, x_{k-1}, x_k^1),$$

and also let

$$\phi_i^{x_k^0,x_k^1}(x_1,\dots,x_{k-1}) \stackrel{\mathrm{def}}{=} \phi_i(x_1,\dots,x_{k-1},x_k^0)\phi_i(x_1,\dots,x_{k-1},x_k^1)$$

for each $i \in \{1,\dots,k-1\}$, then we can rewrite (3) as

$$\begin{aligned}
&\mathrm{disc}_\mu(F,\mathcal{C})^2 \\
&\le \mathbf{E}_{\mathbf{x}_1,\dots,\mathbf{x}_{k-1}}\left[\mathbf{E}_{\mathbf{x}_k^0,\mathbf{x}_k^1}\left[F^{\mathbf{x}_k^0,\mathbf{x}_k^1}(\mathbf{x}_1,\dots,\mathbf{x}_{k-1})\prod_{i=1}^{k-1}\phi_i^{\mathbf{x}_k^0,\mathbf{x}_k^1}(\mathbf{x}_1,\dots,\mathbf{x}_{k-1})\right]\right] \\
&\le \mathbf{E}_{\mathbf{x}_k^0,\mathbf{x}_k^1}\left[\left|\mathbf{E}_{\mathbf{x}_1,\dots,\mathbf{x}_{k-1}}\left[F^{\mathbf{x}_k^0,\mathbf{x}_k^1}(\mathbf{x}_1,\dots,\mathbf{x}_{k-1})\prod_{i=1}^{k-1}\phi_i^{\mathbf{x}_k^0,\mathbf{x}_k^1}(\mathbf{x}_1,\dots,\mathbf{x}_{k-1})\right]\right|\right] \\
&= \mathbf{E}_{\mathbf{x}_k^0,\mathbf{x}_k^1}\left[\mathrm{disc}_{\mu'}(F^{\mathbf{x}_k^0,\mathbf{x}_k^1},\mathcal{C}')\right]. \qquad (4)
\end{aligned}$$

Above, μ' is the product of μ_1 up to μ_{k-1} and $\mathcal{C}'$ is the cylinder intersection defined by $\prod_{i=1}^{k-1}\phi_i^{\mathbf{x}_k^0,\mathbf{x}_k^1}(\mathbf{x}_1,\dots,\mathbf{x}_{k-1})$. Raising both sides of equation (4) to the power of 2^{k-1}, we get

$$\mathrm{disc}_\mu(F,\mathcal{C})^{2^k} \le \mathbf{E}_{\mathbf{x}_k^0,\mathbf{x}_k^1}\left[\mathrm{disc}_{\mu'}(F^{\mathbf{x}_k^0,\mathbf{x}_k^1},C')\right]^{2^{k-1}}.$$

A repeated application of the Cauchy-Schwarz inequality implies $\mathbf{E}\left[\mathbf{Z}\right]^{2^{k-1}} \le \mathbf{E}\left[\mathbf{Z}^{2^{k-1}}\right]$. Hence,

$$\mathrm{disc}_\mu(F,\mathcal{C})^{2^k} \le \mathbf{E}_{\mathbf{x}_k^0,\mathbf{x}_k^1}\left[\mathrm{disc}_{\mu'}(F^{\mathbf{x}_k^0,\mathbf{x}_k^1},\mathcal{C}')^{2^{k-1}}\right].$$

Now applying the induction hypothesis to $\mathrm{disc}_{\mu'}(F^{x_k^0,x_k^1},\mathcal{C}')^{2^{k-1}}$, we get the desired result. □

The RHS of Inequality 2 is important and deserves a name. Let μ be a product distribution over $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$, i.e. $\mu(x_1,\dots,x_k) = \mu_1(x_1)\cdots\mu_k(x_k)$, where μ_i is a distribution over $\mathcal{X}_i$. We define the **cube measure** of a complex valued function $F : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathbb{C}$ under μ as

$$\mathcal{E}_\mu(F) = \mathbf{E}_{\substack{\mathbf{x}_1^0,\dots,\mathbf{x}_k^0\\ \mathbf{x}_1^1,\dots,\mathbf{x}_k^1}}\left[\prod_{u\in\{0,1\}^k}\mathfrak{C}^{u_1+\cdots+u_k}(F(\mathbf{x}_1^{u_1},\dots,\mathbf{x}_k^{u_k}))\right].$$

The cube measure is always a non-negative real number. In fact, the quantity $(\mathcal{E}_U(F))^{1/2^k}$, where U is the uniform distribution, is known as the *hypergraph uniformity norm* and is a measure of "quasirandomness" of F. When $F = f \circ \textsc{xor}$, the hypergraph uniformity norm of F corresponds to Gowers uniformity norm of f over $\mathbb{F}_2^n$ (see e.g. [Gow10, Section 2.4] and references therein). Lemma 4.4 can now be restated as

$$\mathrm{disc}_\mu(F) \le (\mathcal{E}_\mu(F))^{1/2^k}.$$

Let us see the above inequality in action and show an exponentially small upper bound on the *generalized-inner-product* function GIP. This function is defined as $\mathrm{GIP}(x_1,x_2,\dots,x_k) = (-1)^{\sum_i x_{1,i}x_{2,i}\cdots x_{k,i}}$.

Theorem 4.5.

$$\mathrm{disc}_U(\mathrm{GIP}) \leq \exp\left(-\frac{n}{4^k}\right).$$

Proof. Using Lemma 4.4, our task is to upper bound the cube measure $\mathcal{E}_U(\mathrm{GIP})$. Since we can decompose GIP as PARITY ∘ AND, and PARITY is just multiplication over ±1 valued variables, we have

$$\begin{aligned}\mathcal{E}_U(\mathrm{GIP}) &= \mathbf{E}\left[\prod_{u\in\{0,1\}^k} \mathrm{GIP}(\mathbf{x}_1^{u_1}, \ldots, \mathbf{x}_k^{u_k})\right] \\ &= \mathbf{E}\left[\prod_{u\in\{0,1\}^k} \prod_{i=1}^{n} (-1)^{\textsc{and}(\mathbf{x}_{1,i}^{u_1}, \ldots, \mathbf{x}_{k,i}^{u_k})}\right].\end{aligned}$$

Using independence, we can move the inside product outside to obtain

$$\begin{aligned}\mathcal{E}_U(\mathrm{GIP}) &= \prod_{i=1}^{n} \mathbf{E}\left[\prod_{u\in\{0,1\}^k} (-1)^{\textsc{and}(\mathbf{x}_{1,i}^{u_1}, \ldots, \mathbf{x}_{k,i}^{u_k})}\right] \\ &= \left(\mathcal{E}_U(\textsc{and})\right)^n.\end{aligned}$$

Thus, all we need to do is bound the cube measure of the AND function on k variables. It is not difficult to see that if for all $j \in \{1, \ldots, k\}$, $\mathbf{x}_{j,i}^0 \neq \mathbf{x}_{j,i}^1$, then the expectation is -1. This happens with probability $1/2^k$. On the other hand, if there is some j such that $\mathbf{x}_{j,i}^0 = \mathbf{x}_{j,i}^1$, the product evaluates to 1. Therefore,

$$\mathcal{E}_U(\textsc{and}) = \left(1 - \frac{1}{2^k}\right) - \frac{1}{2^k} = 1 - \frac{1}{2^{k-1}}.$$

So $\mathcal{E}_U(\mathrm{GIP}) = (1 - 1/2^{k-1})^n \leq \exp(-n/2^{k-1})$, and the result follows from Lemma 4.4. □

Corollary 4.6.

$$\mathbf{R}_k^\epsilon(\mathrm{GIP}) \geq \frac{n}{4^k} + \log(1 - 2\epsilon).$$

Note that the above lower bound collapses once k reaches $\log n$. This is an unavoidable consequence of Lemma 4.4 where we used the Cauchy-Schwarz inequality repeatedly in order to get rid of the cylinder intersection. As all lower bounds in the NOF model use this trick, they all suffer the exponential loss in the number of players. As mentioned in the introduction, proving lower bounds in the NOF model for $\log n$ players is an outstanding open problem.

5 Communication Complexity Classes

In computational complexity theory we try to classify problems in terms of the resources required to compute their solution. An important part of this classification requires well defined complexity classes, like P, NP, and BPP, which correspond to problems with efficient deterministic, non-deterministic and randomized solutions respectively. In communication complexity, we can define ([BFS86]) analogous complexity classes once we agree on the meaning of "efficient". Conventionally, protocols of cost at most poly-log(n) are considered to be efficient. This naturally leads to the following communication complexity classes corresponding to the different communication complexity models:

Complexity class	P_k^{cc}	NP_k^{cc}	$co\mathsf{NP}_k^{cc}$	BPP_k^{cc}
Complexity measure	$\mathbf{D}_k$	$\mathbf{N}_k^{-1}$	$\mathbf{N}_k^{1}$	$\mathbf{R}_k$

Unlike the Turing Machine world, we have a reasonably good understanding of the relationships between the communication complexity classes since we can actually prove strong lower bounds. For instance, the two player *non-equality* function is in BPP_2^{cc} and NP_2^{cc} but not in P_2^{cc}. Therefore we know that $\mathsf{P}_2^{cc} \neq \mathsf{NP}_2^{cc}$ and $\mathsf{P}^{cc} \neq \mathsf{BPP}_2^{cc}$. We also know that $\mathsf{NP}_2^{cc} \neq \mathsf{BPP}_2^{cc}$ via the *disjointness* function.

6 Information Complexity

The techniques we have seen so far are some of the highlights of the first generation methods in communication complexity. In recent years, a new method based on information theory, introduced in the seminal paper [CWYS01], has flourished and contributed significantly to the advancement of the field. We will now very briefly touch upon this second generation technique. Our discussion will be limited to the 2 party model since these techniques currently do not extend to the multiparty NOF model.

In a nutshell, information theory methods in communication complexity try to measure how much information Alice and Bob reveal about their inputs to a third party or each other when they follow a communication protocol. There are several ways to measure this quantity but we will for now refer to it informally as *information complexity.* This information is measured in bits and therefore it serves as a lower bound on the communication complexity of a function: if a protocol has cost c, it cannot reveal more than c bits of information. One can then obtain lower bounds on communication complexity by lower bounding the information complexity of a function. This approach puts powerful and intuitive tools from information theory at our disposal.

Let μ be a distribution over the input space $\mathcal{X} \times \mathcal{Y}$, and let P be a protocol that computes a function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. Recall that $\Pi_P(x, y)$ denotes the transcript that the protocol produces when the input is (x, y). The **external information cost** of a protocol with respect to μ is defined as

$$\mathrm{IC}_\mu^{\mathrm{ext}}(P) \overset{\mathrm{def}}{=} I(\mathbf{x}, \mathbf{y} : \Pi_P(\mathbf{x}, \mathbf{y})),$$

where $(\mathbf{x}, \mathbf{y})$ has distribution μ. This intuitively measures how much information a third party learns about Alice's and Bob's inputs by looking at the transcript of the protocol. Another useful measure is the **internal information cost**, which is defined to be

$$\mathrm{IC}^{\mathrm{int}}_{\mu}(P) \stackrel{\mathrm{def}}{=} I(\mathbf{y} : \Pi_P(\mathbf{x}, \mathbf{y})|\mathbf{x}) + I(\mathbf{x} : \Pi_P(\mathbf{x}, \mathbf{y})|\mathbf{y}).$$

This measures how much information Alice learns about Bob's input plus how much information Bob learns about Alice's input.

Let us restrict our discussion to external information cost. The ϵ-error **information complexity** of a function F with respect to a distribution μ is denoted by $\mathrm{IC}_{\mu,\epsilon}(F)$ and is defined to be the minimum $\mathrm{IC}^{\mathrm{ext}}_{\mu}(P)$ among all randomized protocols P that compute F with ϵ error. It is straightforward to see that for any distribution μ, $\mathbf{R}^{\epsilon}(F) \geq \mathrm{IC}_{\mu,\epsilon}(F)$.

To illustrate how this can be used to prove communication complexity lower bounds, let's give a very high level and vague sketch of the lower bound for *disjointness*. As we have seen before, *disjointness* has the composed structure $\mathrm{DISJ} = \textsc{or} \circ \textsc{and}$. Intuitively one expects that any protocol that solves *disjointness* with good accuracy must implicitly solve each of the n instances of the AND function. Suppose ν is a distribution over the inputs of a two bit AND function and define μ to be the n-fold product of ν, i.e. $\mu = \nu^n$. Then one can hope to show $\mathrm{IC}_{\mu,\epsilon}(\mathrm{DISJ}) = n \cdot \mathrm{IC}_{\nu,\epsilon}(\textsc{and})$. Unfortunately this may not be true in general, for example when ν is not a product distribution over $\{0, 1\} \times \{0, 1\}$. And in the case of *disjointness*, it is essential that ν is not a product distribution. To get around this problem, one defines an appropriate random variable so that conditioned on it, the "direct sum" property that we hoped for holds. This then reduces our task of showing an $\Omega(1)$ lower bound on the information complexity of the AND function on 2 bits. With some effort, this can be proved by elementary means. The details can be found in [BYJKS04].

References

[BFS86] László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *Proceedings of Symposium on Foundations of Computer Science*, pages 337–347, 1986.

[BNS92] László Babai, Noam Nisan, and Mario Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992.

[BYJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, June 2004.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, April 1988.

[Cha08] Arkadev Chattopadhyay. *Circuits, Communication and Polynomials.* PhD thesis, McGill University, 2008.

[CT93] Fan R.K. Chung and Prasad Tetali. Communication complexity and quasi randomness. *SIAM Journal on Discrete Mathematics*, 6(1):110–123, 1993.

[CWYS01] Amit Chakrabarti, Anthony Wirth, Andrew Yao, and Yaoyun Shi. Informational complexity and the direct sum problem for simultaneous message complexity. *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 270–278, 2001.

[Gow10] W. Timothy Gowers. Decompositions, approximate structure, transference, and the Hahn-Banach theorem. *Bulletin of the London Mathematical Society*, 42:573–606, 2010.

[LS88] László Lovász and Michael Saks. Lattices, Möbius functions and communication complexity. In *29th Annual Symposium on Foundations of Computer Science*, pages 81–90. IEEE Computer Society, 1988.

[New91] Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, July 1991.

[Raz00] Ran Raz. The BNS-Chung criterion for multi-party communication complexity. *Computational Complexity*, 9(2):113–122, 2000.

[Yao79] Andrew C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of ACM Symposium on Theory of Computing*, pages 209–213, 1979.

[Yao83] Andrew C. Yao. Lower bounds by probabilistic arguments. In *Proceedings of Symposium on Foundations of Computer Science*, pages 420–428, 1983.