Communication Complexity

Eyal Kushilevitz and Noam Nisan

CHAPTER 1

Basics

The general communication problem may be described in the following terms: A system must perform some task that depends on information distributed among the different parts of the system (called *processors*, *parties*, or *players*). The players thus need to communicate with each other in order to perform the task. Yao's model of communication complexity, which is the subject of this chapter, is the simplest scenario in which such a situation occurs. Yao's model makes the following simplifying assumptions:

- There are only two parts in the system.
- Each part of the system gets a fixed part of the input information.
- The only resource we care about is communication.
- The task is the computation of some prespecified function of the input.

These assumptions help us concentrate on the core issue of communication. Despite its apparent simplicity, this is a very rich model that exhibits a nice structure and in which issues such as randomization and nondeterminism, among others, can be studied. We can also translate our understanding of this model to many other scenarios in which communication is a key issue.

1.1. The Model

Let X, Y, Z be arbitrary finite sets and let $f : X \times Y \to Z$ be an arbitrary function. There are two players, Alice and Bob, who wish to evaluate $f(x, y)$, for some inputs $x \in X$ and $y \in Y$. The difficulty is that Alice only knows x and Bob only knows y. Thus, to evaluate the function, they will need to communicate with each other. The communication will be carried out according to some fixed protocol $\mathcal{P}$ (which depends only on f). The protocol consists of the players sending bits to each other until the value of f can be determined.

At each stage, the protocol $\mathcal{P}$ (for the function f) must determine whether the run terminates; if the run has terminated, the protocol must specify the answer given by the

protocol (that is, $f(x, y)$); and if the run has not terminated, the protocol must specify which player sends a bit of communication next. This information must depend solely on the bits communicated so far during this run of the protocol, because this is the only knowledge common to both Alice and Bob. In addition, if it is Alice's turn to speak (that is, to communicate a bit), the protocol must specify what she sends; this depends on the communication so far as well as on x, the input visible to Alice. Similarly, if it is Bob's turn to speak, the protocol must specify what he sends; this depends on the communication so far and on y, his input.

We are only interested in the amount of communication between Alice and Bob, and we wish to ignore the question of the internal computations each of them makes. Thus, we allow Bob and Alice to have unlimited computational power. The cost of a protocol $\mathcal{P}$ on input (x, y) is the number of bits communicated by $\mathcal{P}$ on input (x, y). The cost of a protocol $\mathcal{P}$ is the *worst* case (that is, maximal) cost of $\mathcal{P}$ over all inputs (x, y). The complexity of f is the minimum cost of a protocol that computes f.

To formalize this model from the players' point of view, we could define functions specifying who speaks at each point (for example, $NEXT: \{0,1\}^* \rightarrow \{Alice, Bob\}$), what they say, when they stop speaking, and what the answer is. Since we do not want to run the protocol but to analyze it, the following formalization, from the protocol designer's point of view, will be more convenient:

Definition 1.1: *A* protocol $\mathcal{P}$ *over domain* $X \times Y$ *with range* Z *is a binary tree where each internal node* v *is labeled either by a function* $a_v: X \rightarrow \{0,1\}$ *or by a function* $b_v: Y \rightarrow \{0,1\}$, *and each leaf is labeled with an element* $z \in Z$.

The value of the protocol $\mathcal{P}$ *on input* (x, y) *is the label of the leaf reached by starting from the root, and walking on the tree. At each internal node* v *labeled by* a_v *walking left if* $a_v(x) = 0$ *and right if* $a_v(x) = 1$, *and at each internal node labeled by* b_v *walking left if* $b_v(y) = 0$ *and right if* $b_v(y) = 1$. *The* cost of the protocol $\mathcal{P}$ on input (x, y) *is the length of the path taken on input* (x, y). *The* cost of the protocol $\mathcal{P}$ *is the height of the tree.*

Intuitively, each internal node v labeled by a function a_v corresponds to a bit sent by Alice (the bit being $a_v(x)$) and each internal node v labeled by b_v corresponds to a bit sent by Bob. Figure 1.1 shows a protocol tree for some (Boolean) function f defined on $X \times Y$ for $X = \{x, x', x'', x'''\}$ and $Y = \{y, y', y'', y'''\}$. The function f computed by this protocol appears in Figure 1.2. For example, on input (x'', y) the path followed by the protocol is the rightmost path of the tree. The value computed by the protocol is therefore 0. Also note that the value of the function a_4 on x and x' may be arbitrary. This is because $a_1(x) = a_1(x') = 0$ and therefore input pairs in which the value given to Alice is either x or x' take the left edge going from the root and hence a_4 will never be evaluated for such inputs.

Definition 1.2: *For a function* $f: X \times Y \rightarrow Z$, *the* (deterministic) communication complexity *of* f *is the minimum cost of* $\mathcal{P}$, *over all protocols* $\mathcal{P}$ *that compute* f. *We denote the (deterministic) communication complexity of* f *by* $D(f)$.

Sometimes we consider slight variations of this model. For example, in our model the value of the protocol must be evident solely from the communication exchanged

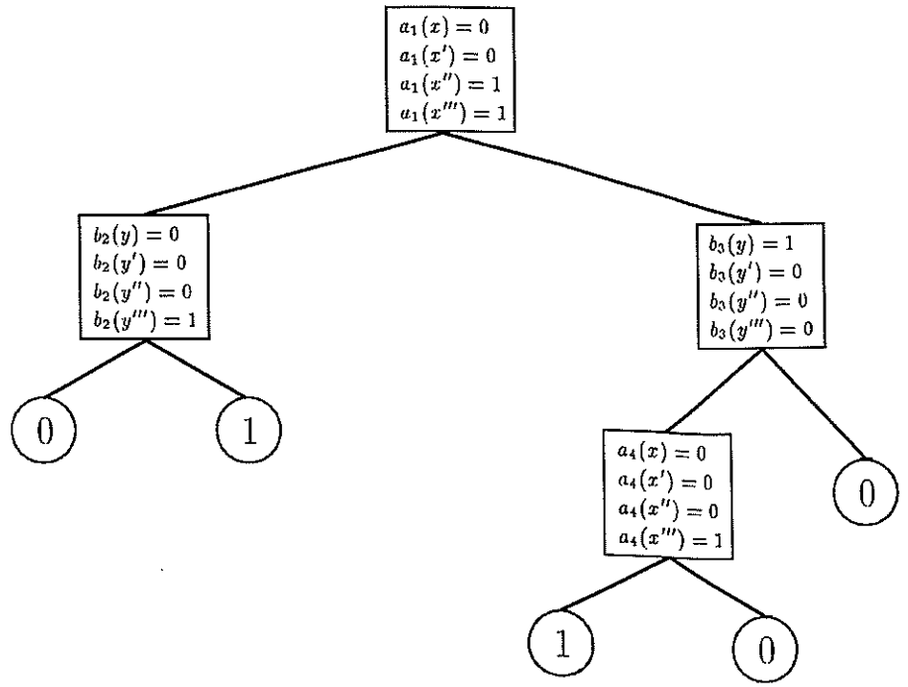

Figure 1.1: A protocol tree

	y	y'	y''	y'''
x	0	0	0	1
x'	0	0	0	1
x''	0	0	0	0
x'''	0	1	1	1

Figure 1.2: The function f computed by the protocol of Figure 1.1

by the parties (this is implicit in the definition by the requirement that the protocol terminates only at leaves where the output is uniquely defined). We may relax this by requiring that only one of the parties knows the answer. This changes the complexity by at most $\log_2 |Z|$. In our model, the order of communication between the two parties is arbitrary. We may require that Alice and Bob each send one bit in her/his turn. This changes the complexity by at most a factor of two.

The simplest way for Alice and Bob to evaluate a function f is for one of the players, say Alice, to send all her input to Bob (requiring $\log_2 |X|$ bits using an appropriate encoding), for Bob to compute $f(x, y)$ privately (with his unlimited computational power), and then for Bob to send the answer back to Alice ($\log_2 |Z|$ more bits). We thus have:

Proposition 1.3: *For every function* $f : X \times Y \to Z$,

$$D(f) \leq \log_2 |X| + \log_2 |Z|.$$

► **Example 1.4:** Alice and Bob hold subsets $x, y \subseteq \{1, \ldots, n\}$ respectively and they wish to compute MAX(x, y), the maximal number in $x \cup y$. For this, Alice sends to Bob the maximal number in x ($\log n$ bits). Bob compares this value with the maximal number in y and sends the larger of them as the output ($\log n$ bits). Therefore, $D(\text{MAX}) \leq 2 \log n$.

Exercise 1.5: Alice and Bob hold subsets $x, y \subseteq \{1, \ldots, n\}$, respectively, and they wish to compute AVG(x, y), which is defined as the average number in the multiset $x \cup y$. Prove that $D(\text{AVG}) = O(\log n)$. (Note that the average need not be an integer.)

In many cases more clever protocols can be designed.

► **Example 1.6:** Alice and Bob hold subsets $x, y \subseteq \{1, \ldots, n\}$, respectively. MED$(x, y)$ is defined to be the median of the multiset $x \cup y$ (if $x \cup y$ contains an even number $t = 2k$ of (not necessarily distinct) elements, then the median is defined as (say) the k-th smallest element). By Proposition 1.3, $D(\text{MED}) \leq n + \log_2 n$ (observe that a subset of $\{1, \ldots, n\}$ can be represented using an n-bit string).

A better protocol for MED may proceed using binary search as follows: At each stage Alice and Bob have an interval $[i, j]$ in which the median may still lie. They halve the interval by deciding whether the median is above or below $k = (i + j)/2$. This is done by Alice sending to Bob the number of elements in x that lie above k and the number that lie below k ($O(\log n)$ bits). Bob can now decide whether the median lies above or below k, and he sends this information to Alice (1 bit). This protocol has $O(\log n)$ stages and each requires $O(\log n)$ bits of communication, so $D(\text{MED}) = O(\log^2 n)$.

Exercise 1.7*: Give an $O(\log n)$ bit protocol for MED.

Exercise 1.8*: Given any graph G on n vertices, we define the following "clique vs. independent set" problem with respect to G: Alice receives as an input C, which is a clique in G (a set of vertices with an edge between any two of them). Bob receives as an input I, which is an independent set in G (a set of vertices with no edges between them). The function $\text{CIS}_G(C, I)$ is defined as the size of $C \cap I$ (observe that this size is either 0 or 1). Prove that $D(\text{CIS}_G) = O(\log^2 n)$, for all G. (A lower bound better than $\Omega(\log n)$ is not known for any G.)

As these examples show, Yao's communication complexity model is quite powerful and allows the design of "clever" protocols. Our main concern will be proving lower bounds on communication complexity: Given a function f, we would like to prove that in any protocol that computes f at least a certain number of bits must be exchanged. For functions with a large range, the following trivial lower bound is often useful.

Exercise 1.9: Show that for every function $f : X \times Y \to Z$, $D(f) \geq \log_2 |\text{Range}(f)|$, where Range$(f)$ is the set of all $z \in Z$ for which there exists a pair $(x, y) \in X \times Y$ such

that $f(x, y) = z$. Conclude that for the function MED the upper bound of Exercise 1.7 is tight (up to a constant).

For the case we are mostly concerned with, that is *Boolean* functions $f : X \times Y \to \{0, 1\}$, this bound only says that $D(f) \geq 1$ and is therefore useless.

From this point on, unless explicitly stated, we always assume $Z = \{0, 1\}$. Most of the techniques we present easily extend to the nonboolean case. In other cases, bounds can be obtained by considering the functions $f_i(x, y)$, the i-th bit of $f(x, y)$, instead of considering f. These functions are Boolean, hence our techniques can be applied. Also, in the Boolean case, we will often not insist that the output be clear from the communication because with a $+1$ increase in the communication complexity this property can be achieved.

1.2. Rectangles

The success in proving good lower bounds on the communication complexity of various functions comes from the *combinatorial* view we take on protocols. The idea is to view protocols as a way to partition the space of all possible input pairs, $X \times Y$, into sets such that for all input pairs in the same set the same communication is sent during the execution of the protocol. In the terminology of protocol trees this means that the inputs in each set are those inputs that reach a certain leaf. Then, we show that these sets of inputs are restricted to have a very special structure. This restriction is imposed by the fact that Alice sees only x and Bob sees only y. This leads to the introduction of the most fundamental element in the combinatorics of protocols – the (*combinatorial*) *rectangle*.

Definition 1.10: *Let $\mathcal{P}$ be a protocol and v be a node of the protocol tree. R_v is the set of inputs (x, y) that reach node v.*

It immediately follows that:

Proposition 1.11: *If L is the set of leaves of a protocol $\mathcal{P}$, then $\{R_\ell\}_{\ell \in L}$ is a partition of $X \times Y$.*

The structure of the sets R_ℓ (and more generally the sets R_v) is not at all arbitrary. The study of this structure is at the center of our approach to communication complexity.

Definition 1.12: *A* combinatorial rectangle (*in short, a rectangle*) *in $X \times Y$ is a subset $R \subseteq X \times Y$ such that $R = A \times B$ for some $A \subseteq X$ and $B \subseteq Y$.*

An equivalent definition is given by the following proposition.

Proposition 1.13: *$R \subseteq X \times Y$ is a rectangle if and only if*

$$(x_1, y_1) \in R \text{ and } (x_2, y_2) \in R \Rightarrow (x_1, y_2) \in R.$$

PROOF:

Only if: Assume R is a rectangle, that is $R = A \times B$. If $(x_1, y_1) \in R$, then $x_1 \in A$. Similarly, because $(x_2, y_2) \in R$, then $y_2 \in B$. It follows that $(x_1, y_2) \in A \times B = R$.

If: Define the sets

$$A = \{x \mid \text{exists } y \text{ such that } (x, y) \in R\}$$

and

$$B = \{y \mid \text{exists } x \text{ such that } (x, y) \in R\}.$$

We claim that $R = A \times B$. The inclusion $R \subseteq A \times B$ is clear from the definition of A and B ($(x, y) \in R$ implies $x \in A$ and $y \in B$ hence $(x, y) \in A \times B$). To show that $A \times B \subseteq R$, consider $(x, y) \in A \times B$. Since $x \in A$ there exists y' such that $(x, y') \in R$. Similarly, because $y \in B$ there exists x' such that $(x', y) \in R$. Using the assumption this implies $(x, y) \in R$. □

The connection between rectangles and communication complexity is given by the following proposition:

Proposition 1.14: *For every protocol $\mathcal{P}$ and leaf ℓ in it, R_ℓ is a rectangle.*

PROOF: We will prove by induction on the depth of v that R_v is a rectangle. For the root, it is clear that $R_{root} = X \times Y$, which is a rectangle. Otherwise, let w be the parent of v and assume, without loss of generality, that v is the left son of w and that in w Alice speaks (that is, w is labeled with a function $a_w : X \to \{0, 1\}$). Then

$$R_v = R_w \cap \{(x, y) \mid a_w(x) = 0\}.$$

By the induction hypothesis, $R_w = A_w \times B_w$, and thus

$$R_v = (A_w \cap \{x \mid a_w(x) = 0\}) \times B_w$$

which is a rectangle. □

Note that the proof shows that all the sets R_v are rectangles and not only those corresponding to leaves of the tree. It may be instructive to see another proof of this proposition using the second definition of a rectangle, given by Proposition 1.13.

PROOF: Assume $(x_1, y_1) \in R_\ell$ and $(x_2, y_2) \in R_\ell$, we will show that also $(x_1, y_2) \in R_\ell$, that is, we will show that on input (x_1, y_2) the protocol will behave the same as on (x_1, y_1) and (x_2, y_2). We will follow the communication performed (that is, the path taken) on input (x_1, y_2) and show that it never deviates from the path to ℓ. If we have reached a node v on the path in which Alice speaks, then because she cannot distinguish between (x_1, y_2) and (x_1, y_1) (in both cases, she evaluates $a_v(x_1)$) she will behave the same on both inputs. Thus, on (x_1, y_2) we will also move toward ℓ. If we have reached a node v in which Bob speaks then because Bob cannot distinguish between (x_1, y_2) and (x_2, y_2) he will behave the same on both inputs, that is, again we will move toward ℓ. □

	000	001	010	011	100	101	110	111
000	0	1	1	0	1	0	0	0
001	1	0	0	0	0	0	0	1
010	1	0	0	0	1	0	0	0
011	0	0	1	0	0	0	0	1
100	1	0	0	0	1	0	0	1
101	1	1	1	0	0	0	1	1
110	0	0	0	0	1	0	0	0
111	0	1	1	0	1	1	0	1

Figure 1.3: A 0-monochromatic rectangle

In both proofs the reason R_ℓ is a rectangle is that Alice and Bob each know only one of the inputs x and y. The second proof is in fact a "cut-and-paste" argument: We take the way Alice behaves out of the communication on (x_1, y_1), and the way Bob behaves out of the communication on (x_2, y_2), and we put them together to get the communication on (x_1, y_2). This kind of argument is found in many other proofs, especially those that deal with crossing sequences. Indeed, as is shown in Part III of this book, in many cases we can reprove results that were initially proved with crossing sequences using communication complexity. The new proofs do not repeat the cut-and-paste argument, thus abstracting away these types of arguments.

If a protocol $\mathcal{P}$ computes a function f then for every leaf ℓ of $\mathcal{P}$, all inputs $(x, y) \in R_\ell$ must have the same value of f, the value with which ℓ is labeled.

Definition 1.15: *A subset $R \subseteq X \times Y$ is called* f-monochromatic (*in short, monochromatic*) *if f is fixed on R.*

Figure 1.3 shows an example of a monochromatic rectangle, where the rows correspond to $X = \{0, 1\}^3$, the columns correspond to $Y = \{0, 1\}^3$, and the rectangle $R = A \times B$ is defined by $A = \{001, 010, 100, 110\}$ and $B = \{001, 010, 011, 101, 110\}$. We emphasize that, although in many figures it will be convenient to draw the rectangles as having adjacent rows and columns, the definition does *not* require this.
This section can now be summarized by:

Lemma 1.16: *Any protocol $\mathcal{P}$ for a function f induces a partition of $X \times Y$ into f-monochromatic rectangles. The number of rectangles is the number of leaves of $\mathcal{P}$.*

Figure 1.4 shows the partition of the space of inputs $X \times Y$ by the protocol of Figure 1.1 (for the function f given in Figure 1.2).

Corollary 1.17: *If any partition of $X \times Y$ into f-monochromatic rectangles requires at least t rectangles, then $D(f) \geq \log_2 t$.*

PROOF: By Lemma 1.16, the leaves of any protocol for f induce a partition of $X \times Y$ into f-monochromatic rectangles. Hence, by the assumption, any such protocol must

	y	y'	y''	y'''
x	0	0	0	1
x'	0	0	0	1
x''	0	0	0	0
x'''	0	1	1	1

Figure 1.4: The function f computed by the protocol of Figure 1.1

have at least t leaves and thus the depth of its tree (since the tree is binary) is at least $\log_2 t$. □

This corollary gives a strategy for proving lower bounds on the communication complexity of a function f: prove lower bounds on the number of rectangles in any partition of $X \times Y$ into f-monochromatic rectangles. In the next sections we present several techniques for doing this.

Exercise 1.18: Let $X = Y = \{1, \ldots, n\}$. A geometric rectangle is a set of the form $\{(x,y) \mid x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}\}$, for some values $x_{\min}, x_{\max}, y_{\min}$, and $y_{\max}$ in $\{1, \ldots, n\}$. A *comparison protocol* is a one in which at each node v, if Alice needs to transmit a bit then this bit is the result of comparing her input x with some value θ_v (that is, $a_v(x)$ is 0 if $x < \theta_v$ and 1 if $x \geq \theta_v$). Similarly, at each node v where Bob speaks he sends the result of comparing his input y with some value θ_v. Prove that every comparison protocol for computing a function f partitions the space $X \times Y$ into f-monochromatic geometric rectangles.

1.3. Fooling Sets and Rectangle Size

Our first lower bound technique is called the "fooling set" technique. It says that if we exhibit a large set of input pairs such that no two of them can be in a single monochromatic rectangle, this implies that the number of monochromatic rectangles is large. The idea is to use the fact that each protocol partitions the space $X \times Y$ into monochromatic rectangles (Lemma 1.16), together with the property of rectangles given by Proposition 1.13 (see Figure 1.5). These together imply that if two input pairs (x_1, y_1) and (x_2, y_2) are in the same monochromatic rectangle induced by a given protocol $\mathcal{P}$, then the value of f on both of them is some z, and that the two input pairs (x_1, y_2)

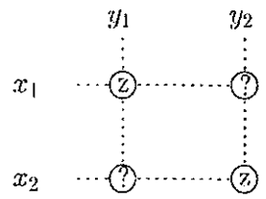

Figure 1.5: The rectangle's property

and (x_2, y_1) must also be in the same monochromatic rectangle. In particular, f has the same value z on both of (x_1, y_2) and (x_2, y_1). In other words, if the value of f on either (x_1, y_2) or (x_2, y_1) is not z, then (x_1, y_1) and (x_2, y_2) cannot be in the same rectangle. This is formalized by the following definition and lemma.

Definition 1.19: *Let $f : X \times Y \to \{0, 1\}$. A set $S \subset X \times Y$ is called a* fooling set *(for f) if there exists a value $z \in \{0, 1\}$ such that*

- *For every $(x, y) \in S$, $f(x, y) = z$.*
- *For every two distinct pairs (x_1, y_1) and (x_2, y_2) in S, either $f(x_1, y_2) \neq z$ or $f(x_2, y_1) \neq z$.*

Lemma 1.20: *If f has a fooling set S of size t, then $D(f) \geq \log_2 t$.*

PROOF: It is enough to prove that no monochromatic rectangle contains more than one element of S. Assume that a rectangle R contains two distinct pairs (x_1, y_1) and (x_2, y_2) that belong to S. By Proposition 1.13, it must also contain (x_1, y_2) and (x_2, y_1). However, since S is a fooling set, the value of f on both (x_1, y_1) and (x_2, y_2) is z, whereas on at least one of (x_1, y_2) and (x_2, y_1) the value of f is different than z. It follows that R is not monochromatic. Therefore, at least t rectangles are needed to cover S, and the lemma follows by Corollary 1.17. □

The above bound is obtained by considering a fooling set that contains only points of $X \times Y$ whose f-value is some $z \in \{0, 1\}$. In fact, we get a lower bound on the number of z-rectangles. To improve the lower bound, we can obtain such a bound for both the 0-rectangles and the 1-rectangles. The total number of rectangles needed in a partition is at least the sum of these two numbers. Despite the simplicity of the fooling set technique, it gives tight bounds for several interesting functions.

▶ **Example 1.21:** Alice and Bob each hold an n-bit string, $x, y \in \{0, 1\}^n$. The equality function, $\text{EQ}(x, y)$, is defined to be 1 if $x = y$ and 0 otherwise. A fooling set of size 2^n is

$$S = \{(\alpha, \alpha) \mid \alpha \in \{0, 1\}^n\}$$

(because for every α, $\text{EQ}(\alpha, \alpha) = 1$, whereas for every $\alpha \neq \beta$, $\text{EQ}(\alpha, \beta) = 0$). It follows that $D(\text{EQ}) \geq n$. By also counting 0-rectangles, we conclude $D(\text{EQ}) \geq n + 1$. Finally, recall that $D(f) \leq n + 1$ for every function $f : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ (by Proposition 1.3). Therefore, $D(\text{EQ}) = n + 1$.

Exercise 1.22: Alice and Bob each hold an n-bit integer $0 \leq x, y < 2^n$. The "greater than" function, $\text{GT}(x, y)$, is defined to be 1 iff $x > y$. Prove that $D(\text{GT}) = n + 1$.

▶ **Example 1.23:** Alice and Bob each hold a subset of $\{1, \ldots, n\}$ (x and y, respectively). The disjointness function, $\text{DISJ}(x, y)$, is defined to be 1 iff $x \cap y = \emptyset$. A fooling set of size 2^n is given by

$$S = \{(A, \bar{A}) \mid A \subseteq \{1, \ldots, n\}\}$$

(without loss of generality, for $A \neq B$ there exists an element a such that $a \in A, a \notin B$; hence $A \cap \bar{B} \neq \emptyset$). It follows that $D(\text{DISJ}) \geq n$. Once again, by also counting 0-rectangles we conclude $D(\text{DISJ}) = n + 1$.

The fooling set technique is a special case of a more general technique for proving lower bounds on the communication complexity. The idea is to prove that the "size" of every monochromatic rectangle is small, implying that many monochromatic rectangles are needed in any partition of $X \times Y$. Naturally, the "size" measure can be chosen to our advantage, and in general we can use any probability distribution as a size measure.

Proposition 1.24: *Let μ be a probability distribution of $X \times Y$. If any f-monochromatic rectangle R has measure $\mu(R) \leq \delta$, then $D(f) \geq \log_2 1/\delta$.*

PROOF: Since $\mu(X \times Y) = 1$, there must be at least $1/\delta$ rectangles in any f-monochromatic partition of $X \times Y$. Thus, the bound follows from Corollary 1.17. □

To see that the fooling set technique is indeed a special case of Proposition 1.24, we consider any fooling set S of size t and define a probability distribution μ as follows: $\mu(x, y) = 0$ for $(x, y) \notin S$ and $\mu(x, y) = 1/t$ for $(x, y) \in S$. What we have shown in the proof of Lemma 1.20 is that every monochromatic rectangle R can contain at most one element of the fooling set and thus has measure $\mu(R) \leq 1/t$. Another special case is obtained by looking at the actual size of the f-monochromatic rectangles. If we can prove that all monochromatic rectangles are of size smaller than k, then the number of rectangles is at least $|X||Y|/k$ and hence $D(f) \geq \log|X| + \log|Y| - \log k$. To see that this is also a special case of Proposition 1.24, consider the uniform distribution $\mu(x, y) = 1/|X||Y|$. The above property guarantees that every monochromatic rectangle R has measure $\mu(R) \leq k/|X||Y|$. A slight variant of this argument can be obtained by considering only the inputs for which $f(x, y) = 0$. This is done in Example 1.25.

► **Example 1.25:** Alice and Bob each hold an n-bit string, $x, y \in \{0, 1\}^n$. The inner-product function, IP, is defined by $\text{IP}(x, y) = \sum_{i=1}^{n} x_i y_i \ (mod\ 2)$. That is, $\text{IP}(x, y)$ is just the inner product $\langle x, y \rangle$ modulo 2. We will show that any 0-monochromatic rectangle covers at most 2^n of the input pairs (out of about $2^{2n}/2$ 0s). Thus, if μ is the uniform distribution on the 0s of the function then, for all rectangles R, $\mu(R) \leq 2^{-(n-1)}$. This implies, by Proposition 1.24, that $D(\text{IP}) \geq n - 1$.

Let $R = A \times B$ be any 0-rectangle. First, we replace A by $A' = span(A)$ and B by $B' = span(B)$, where $span(C)$ denotes the linear span, over the vector space Z_2^n, of vectors in the set C. The extended rectangle $A' \times B'$ may have larger area but since the inner product satisfies

$$\langle a + a', b + b' \rangle = \langle a, b \rangle + \langle a, b' \rangle + \langle a', b \rangle + \langle a', b' \rangle$$

then $A' \times B'$ is still 0-monochromatic. Finally, since A' and B' are orthogonal subspaces

of Z_2^n, then by linear algebra the sum of $dim(A')$ and $dim(B')$ is at most n, the dimension of the whole space. Therefore, the size of the rectangle is bounded by $|A'||B'| = 2^{dim(A')}2^{dim(B')} \leq 2^n$.

Exercise 1.26: Prove that the size of any 1-monochromatic rectangle of the DISJ function (Example 1.23) is at most 2^n. Conclude that $D(\text{DISJ}) = \Omega(n)$.

1.4. Rank Lower Bound

In this section we present a very different lower bound technique. This technique also gives a lower bound on communication complexity by giving a lower bound on the number of monochromatic rectangles in any partition of $X \times Y$, but it does so in an *algebraic* way. This allows us later to use algebraic tools for proving communication complexity lower bounds. To this end, we associate with every function $f : X \times Y \to \{0, 1\}$ a matrix M_f of dimensions $|X| \times |Y|$. The rows of M_f are indexed by the elements of X and the columns by the elements of Y. The (x, y) entry of M_f is simply defined as $f(x, y)$. For example, Figure 1.6 shows the matrix M_{EQ} corresponding to the equality function EQ (Example 1.21). This matrix is simply the identity matrix. The following algebraic property of the matrix M_f is useful for proving lower bounds on the communication complexity of f.

Definition 1.27: *rank(f) is the linear rank of the matrix M_f over the field of reals.*

Lemma 1.28: *For any function $f : X \times Y \to \{0, 1\}$,*

$$D(f) \geq \log_2 \text{rank}(f).$$

PROOF: Let $\mathcal{P}$ be a protocol for f and let L_1 be the set of leaves of $\mathcal{P}$ in which the output is 1. For each leaf $\ell \in L_1$, define a matrix M_ℓ by $M_\ell(x, y) = 1$ for $(x, y) \in R_\ell$ and $M_\ell(x, y) = 0$ for $(x, y) \notin R_\ell$, where R_ℓ is the rectangle of all inputs reaching the

	000	001	010	011	100	101	110	111
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	0	0
101	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	1	0
111	0	0	0	0	0	0	0	1

Figure 1.6: The matrix M_{EQ}