Some Results in Communication Complexity

MAK, Yan Kei

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Mathematics

The Chinese University of Hong Kong
October 2010

Thesis/Assessment Committee

Professor LEUNG Chi Wai (Chair)

Professor FENG Dejun (Thesis Supervisor)

Professor NG Kung Fu (Committee Member)

Professor Julien BARRAL (External Examiner)

Abstract

Communication complexity, introduced by Yao in 1979, deals with the minimization of the number of bits to be exchanged between two parties, Alice and Bob, for computing a Boolean function $f(x, y)$, where x and y are n-bit strings held by Alice and Bob respectively. It is assumed that the two parties have unbounded computational power, i.e. there is no restriction on the number of computation steps, or the size of the computer memory used; the only resource in concern is communication.

This survey paper gives a detailed discussion on different aspects of communication complexity including deterministic, nondeterministic and randomized protocols based on the existing foundations. We also present known techniques of obtaining upper and lower bounds in different models.

摘要

通信複雜性理論由姚期智教授於一九七九年創立，用以探討在處理分佈計算問題時各方如何達致最優協議，從而使要傳送的位元數目為最少。各方在計算步驟以至在所使用的計算機內存的大小等方面均沒有任何限制；整套理論唯一關注的是通信量。

本文將深入討論確定性、不確定性與隨機通信模型，並闡述在上述模型下為通信複雜度求上下界的一些已知技巧。

ACKNOWLEDGMENTS

I wish to express my gratitude to my supervisor Prof. Feng Dejun for his continual guidance in these two years. It is Prof. Feng who introduced me to this fascinating area of communication complexity. He is always kind and willing to spend his time on giving me valuable advice. He really has helped me a lot in preparing this thesis.

I would also like to thank my teachers and colleagues for their helpful discussions. In particular I express my heartfelt thanks to my thesis examiners Prof. Ng Kung Fu and Prof. Leung Chi Wai, my teachers Prof. Zhang Shengyu and Dr. Cheung Leung Fu, and my office roommates Kwok Tsz Chiu and Tsang Chi Shing Sidney.

Finally I would like to thank my family for their encouragement and support all the time.

The cliparts used in Chapter 1 (the light bulb and the coins) are taken from Wikimedia Commons. According to the licensing statement, their copyright holders have released them into the public domain.

Contents

Chapter 1

Introduction

This thesis is about the active research area of communication complexity. We begin with the basic deterministic communication model, followed by some techniques to obtain bounds on communication complexity. Variant models including the nondeterministic and randomized models are also discussed.

We first give the historical background in Section 1.1. Section 1.2 suggests several reasons why communication complexity is of research interest. The main ideas and results in this thesis are summarized in Section 1.3, and Section 1.4 mentions some of the recent development in this field. Lastly, Section 1.5 explains the structure of this thesis.

1.1 Historical background

Communication complexity was introduced by Yao [39] as a mathematical theory for studying communication processes. A first motivation is that deterministic communication complexity dominates the AT^2-complexity of VLSI chips [20]. Some of the early results are from Yao [39] (relating communication protocols to tiling of Boolean matrices), Mehlhorn and Schmidt [27] (introducing the rank lower bound) and Aho, Ullman and Yannakakis [2] (relating the deterministic

model to nondeterminism; the nondeterministic communication model was first introduced by Lipton and Sedgewick [21]).

For the randomized model, also defined in [39], the most important results are attributed to Newman [28] (comparing the private coin and public coin models) and Yao [40] (introducing distributional complexity).

The book by Kushilevitz and Nisan [17] is considered 'the reference' in this field. It includes all the above results, plus a number of interesting applications.

1.2 Why study communication complexity?

Communication complexity, though defined in an abstract manner, is by no means impractical. Consider the case where two parties are far away and communication is slow when compared to local computation. It is then wise to minimize the number of bits transmitted between them. This situation is captured exactly by the model of communication complexity.

Communication complexity is strongly related to many other fields in computer science in addition to VLSI chip design. The complexity of certain communication problems known as *Karchmer-Wigderson games* gives the minimum depth of Boolean circuits [12]. Other examples of applications include streaming complexity [3], instance complexity [25] and proof complexity [6].

In a mathematician's view, this research area is interesting and challenging. The study of communication complexity is a beautiful integration of mathematics and computer science. Different mathematical techiques, including Fourier analysis [33] and generalization of matrix ranks and norms [15, 38], have been extensively used to establish results in communication complexity. In the other direction, problems in communication complexity often have mathematical significance. The *log-rank conjecture*, for example, is closely related to chromatic numbers in graph theory [23]. On another paper devoted to the conjecture [29],

Nisan and Wigderson have proved that low rank of a Boolean matrix implies large discrepancy, i.e. every submatrix is 'highly unbalanced', a result independent of communication.

1.3 Main ideas and results

Deterministic communication complexity considers the scenario where two players, Alice and Bob, both with unlimited processing power, each holds an n-bit binary input string x and y respectively. Between them there is a channel which can be used for data transmission. They would like to cooperatively compute $f(x, y)$, where the function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is known to both beforehand. The goal is to minimize the number of bits needed to be transferred for the worst-case choice of x, y. This quantity is denoted as $D(f)$.

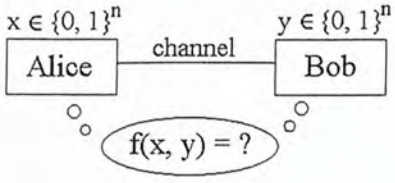

Figure 1.1: The deterministic communication model

We can associate the channel with several descriptive words:

- It is a *bit* channel, meaning that only traditional binary bits can be sent.
- It is a *bidirectional* channel, meaning that data transfer can go in both directions along the channel.
- It is a *noiseless* channel, meaning that no error is introduced by the channel.

Example 1.1. The very first example in Yao's paper is the $PARITY$ function:

$$PARITY(x, y) := (|x| + |y|) \bmod 2$$

It is easy to see that two bit transmissions suffice: Alice computes $s := |x| \bmod 2$ and sends s to Bob, then Bob can compute $PARITY(x, y) = (s + |y|) \bmod 2$. Hence,

$$D(PARITY) = O(1).$$

Remark 1.1. The above $PARITY$ is not exactly a function, but rather a *family* of functions on different input lengths. To be more rigorous we can write $PARITY_n$ for the function on n-bit strings, and the result in Example 1.1 would read:

$$D(PARITY_n) = O(1).$$

We omit the subscript just for simplicity of notation when no confusion occurs. This convention applies throughout the thesis.

Example 1.2. Let EQ be the *equality function*:

$$EQ(x, y) := \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

It can be shown that the *trivial protocol* of letting Alice send all her n bits to Bob is the best that we can do in this case. So,

$$D(EQ) = \Theta(n).$$

Yao [39] demonstrated that any deterministic protocol of f induces a monochromatic tiling of its matrix. Hence we have the ***tiling lower bound***:

Theorem 1.1. *$D(f) \geq \lceil \log_2 \chi(f) \rceil$, where $\chi(f)$ is the number of monochromatic rectangles in the optimal tiling of the matrix.*

Mehlhorn and Schmidt [27] observed that $\chi(f)$ can be lower-bounded by the rank of the matrix for f. This gives rise to the ***rank lower bound***:

Theorem 1.2. *For $f \not\equiv 0$, $D(f) \geq \lceil \log_2 rank(f) \rceil$, the logarithm of the rank of the corresponding matrix over the reals.*

Theorems 1.1 and 1.2 can both be used to obtain the result in Example 1.2.

Under the ***nondeterministc model***, instead of letting Alice and Bob communicate, an extra *nondeterministic input string* $z \in \{0,1\}^k$ is provided, which should be able to 'convince' the players in the case of $f(x,y) = 1$. The minimum value of k that makes it possible is called the *nondeterministic complexity* $N^1(f)$.

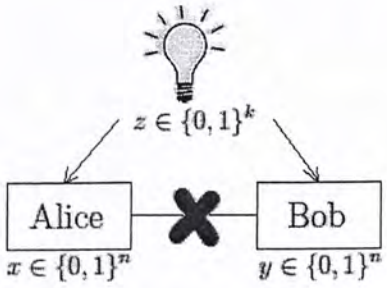

Figure 1.2: The nondeterministic model

A moment's thought reveals that nondeterministic protocols are related to monochromatic rectangle *covering* of the matrix for f. Indeed, the name of a 1-rectangle containing the (x,y)-entry serves as a proof that $f(x,y) = 1$. Hence:

Theorem 1.3. $N^1(f) = \lceil \log_2 C^1(f) \rceil$, *where* $C^1(f)$ *is the number of 1-rectangles in the optimal covering of the matrix.*

Aho, Ullman and Yannakakis [2] have shown a strong relationship between deterministic and nondeterministic complexities:

Theorem 1.4. $D(f) = O(N^1(f)N^1(\bar{f}))$, *where* $\bar{f}$ *is the negation of* f.

Under the ***(private coin) randomized model***, instead of just looking at his input string, a player may also *toss random coins* whenever he makes a move. Hence, for the same input pair (x,y), the execution of protocol may result in different paths because of the different results of coin tosses. We even allow the protocol to make errors with a small probability. The *(private coin) randomized communication complexity* of f is denoted by $R(f)$.

An alternative randomized model assumes that the two players share a common random string. The complexity measure in this ***public coin model*** is denoted by $R^{pub}(f)$.

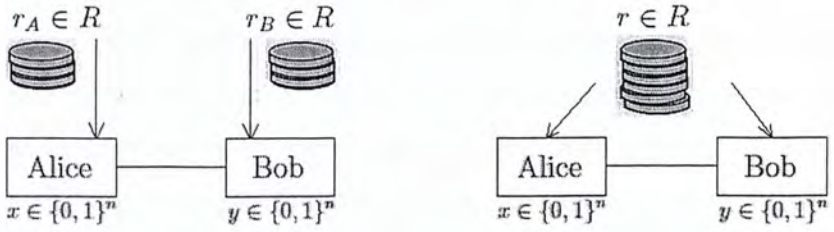

Figure 1.3: The private coin (left) and public coin (right) randomized models

Example 1.3. Rabin and Yao suggested a private coin protocol for EQ that takes $O(\log n)$ bit transmissions, using fingerprinting with random primes. This protocol turns out to be asymptotically optimal, which gives:

$$R(EQ) = \Theta(\log n).$$

In the public coin model we can do better using inner product. Only *two* bit transmissions are required, which implies:

$$R^{pub}(EQ) = \Theta(1).$$

Quite surprisingly, a result by Newman [28] shows that the two randomized models do not differ very much:

Theorem 1.5. $R(f) \leq R^{pub}(f) + O(\log n)$.

Yao also introduced the ***distributional model*** in [40]. Let μ be a probability distribution over the inputs, and ϵ be the probability of error over μ. Yao has shown a beautiful relationship between (μ, ϵ)-*distributional complexity* $D^{\mu}_{\epsilon}(f)$ and public coin randomized complexity $R^{pub}_{\epsilon}(f)$:

Theorem 1.6. $R^{pub}_{\epsilon}(f) = \max_{\mu} D^{\mu}_{\epsilon}(f)$.

1.4 Recent development

As mentioned in Section 1.2, exploration of new bounding methods and applications has always been a main concern in this field.

Another focus of research is the ***quantum communication model***, again introduced by Yao [41]. Here the bit channel is replaced by a quantum one, and the *quantum communication complexity* $Q(f)$ is defined accordingly. Some basic results have been established in [16]; for instance, $Q(f) \leq R(f)$, i.e. quantum protocols are at least as powerful as randomized ones. There are functions whose quantum complexity is significantly smaller than its randomized complexity [1]. For an introduction to the quantum models, please refer to [7] and [37].

More recent results can be found in the survey by Lee and Shraibman [19].

1.5 Structure of the thesis

In Chapter 2, we introduce the *deterministic model*. The notion of protocol tree is introduced, and methods for obtaining bounds are discussed.

In Chapter 3, we talk about the *nondeterministic model*, and prove the important theorem by Aho, Ullman and Yannakakis.

In Chapter 4, we discuss the *randomized model*. We will prove the famous *Yao's Minimax Principle* on distributional complexity.

In Chapter 5, we define *communication complexity classes*, analogous to those in traditional computational complexity theory.

In Chapter 6, we conclude our thesis with a list of *further topics*.

Chapter 2

Deterministic Communication Complexity

This chapter deals with the deterministic communication model. We give the necessary definitions in Section 2.1. The tiling lower bound, the fooling set bound and the rank lower bound are introduced in Sections 2.2, 2.3 and 2.4 respectively. In Section 2.5 we compare these bounds by their tightness and computational efficiency.

2.1 Definitions

As mentioned in Chapter 1, the deterministic model deals with the scenario where two players, Alice and Bob, have to cooperately compute $f(x, y)$, where $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is a function known before communication, and x, y are n-bit input strings held by Alice and Bob respectively. To give a formal definition, we introduce the notion of *protocol trees*:

Definition 2.1. A *deterministic communication protocol* is represented by a binary tree. Each intermediate vertex v is labelled with $p_v \in \{0,1\}$ indicating which player is responsible for this turn, and a function $c_v : \{0,1\}^n \to \{0,1\}$ indicating

what bit is to be sent according to that player's own input. Each leaf node l is labelled with a value $v_l \in \{0,1\}$. The *cost* of the protocol is the height of the tree.

Every input pair (x,y) induces a *path* in the protocol tree. We start with the root node r. The bit p_r indicates which player is to send the first bit. If Alice is to go first, she sends the bit $c_r(x)$ to Bob, and the execution of protocol proceeds with the left subtree if $c_r(x) = 0$, or the right otherwise. The situation is similar if p_r indicates that Bob is to go first, in which case he computes $c_r(y)$ and does the respective steps. This goes on until a leaf node l is reached, and v_l is the output of the protocol. The maximum number of bits exchanged over all possible (x,y) is thus the length of the longest path, i.e. the height of the tree.

Example 2.1. We draw in Figure 2.1 the tree representation of the protocol for *PARITY* that we proposed in Example 1.1. We label Alice's nodes with $p = 0$, and Bob's with $p = 1$. The function $c : \{0,1\}^n \to \{0,1\}$ is defined as:

$$c(x) := |x| \bmod 2.$$

Intermediate vertices are represented by rectangles, and leaf nodes are represented by circles.

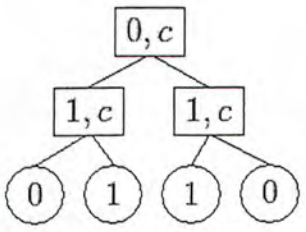

Figure 2.1: A protocol tree for *PARITY*

Remark 2.1. The definition we give above slightly differs from Yao's original one. First, in Yao's definition, each player is to send one bit *in turn*. Also, the

protocol can stop when any party (not necessarily both) knows the output. Such differences leave the number of bits transferred unchanged asymptotically.

Remark 2.2. Protocol trees of high cost can be complicated to draw. (See Example 2.2.) Nevertheless, the properties of trees are often useful when proving bounds in communication complexity, for example Theorem 2.2.

Remark 2.3. It is possible to define communication complexity without notions of protocol trees at all. Arora and Barak define in their recent book [4] a communication protocol as a sequence of functions. Their definition would be more complete if they mention the *prefix-freeness property*, i.e. messages exchanged should be self-delimiting, and no extra end-of-message symbol is required. (See [11, 32], for instance.)

Let $P(x, y)$ be the value v_l of the leaf node reached for input (x, y). A protocol defines a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, where $f(x, y) := P(x, y)$. As mentioned earlier, we are more interested in the opposite direction, i.e. we first have a function f, and then we find corresponding protocols.

Definition 2.2. A deterministic protocol P *computes* f if $P(x, y) = f(x, y)$ for every input pair (x, y). The *deterministic communication complexity* of f, denoted by $D(f)$, is the minimum cost over all deterministic protocols that compute f:

$$D(f) := \min_{P \text{ : protocol that computes } f} \{cost(P)\}$$

A usual way to obtain an upper bound for $D(f)$ is to give one possible protocol that computes f. Since $D(f)$ is defined as the *minimum* cost, it must be smaller than the cost of the proposed protocol.

Example 2.2. Consider the following ***trivial protocol*** of cost $n+1$ which works for any f: Alice sends all her n bits to Bob, and then with both x and y in hand, Bob computes $f(x, y)$ and sends the result to Alice. Thus we have

$$D(f) \leq n + 1 \text{ for all } f.$$

For completeness, we draw out the tree representation of the trivial protocol on $n = 1, 2, 3$ in Figure 2.2. As in Example 2.1, Alice's nodes are labelled with $p = 0$, and Bob's are labelled with $p = 1$. The functions $a_i, b_x : \{0,1\}^n \to \{0,1\}$ $(1 \leq i \leq n, x \in \{0,1\}^n)$ are defined as follows:

$$a_i(x) := x_i = \text{the } i\text{-th bit of } x, \text{ and}$$

$$b_x(y) := f(x, y).$$

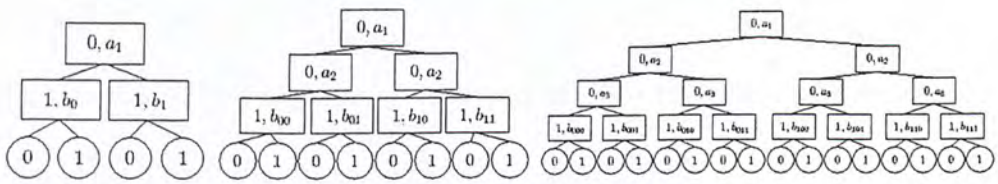

Figure 2.2: The trivial protocol for $n = 1, 2, 3$

We see that the number of nodes is roughly doubled when n increases by 1. That means the size of the tree grows exponentially in n.

Example 2.3. On the other hand, we have seen a two-bit protocol for the *PARITY* function in Example 1.1, which implies that

$$D(PARITY) = 2.$$

The $\geq$ direction follows from the fact that none of the players can deduce the function output alone, and therefore some message from the other side is needed.

2.2 Tiling lower bound

The results in this section, unless otherwise stated, are introduced by Yao [39].

Definition 2.3. Denote $N := 2^n$. The *communication matrix* M_f is the N-by-N Boolean matrix whose (x, y)-th entry is $f(x, y)$. Note that there is a one-to-one correspondence between Boolean functions f and communication matrices M_f.

A very first observation on communication protocols is that a protocol partitions M_f into disjoint monochromatic rectangles. Here by *rectangle* (or *combinatorial rectangle*) we mean a submatrix M of M_f corresponding to $S \times T$, where $S, T \subseteq \{0,1\}^n$. The rectangle is *monochromatic* if for all $x \in S$ and $y \in T$, $f(x,y)$ is the same.

Remark 2.4. For simplicity, we sometimes make an abuse of notation of treating M and $S \times T$ as the same thing. For instance, we say that $(x,y) \in M$ when we mean it belongs to $S \times T$.

Theorem 2.1. *A protocol for f partitions M_f into monochromatic rectangles.*

Proof. For a node v in the protocol tree, denote R_v as the set of inputs (x,y) that reach v while following their respective paths. We shall show that R_v corresponds to a submatrix M_v of M_f. Then we conclude that $\{M_l \mid l : \text{leaf}\}$ is the partition we want.

We use induction. Clearly, for the root r, R_r corresponds to the entire matrix M_f. Now for any other node $v \neq r$, suppose its parent w satisfies the proposition that R_w induces a rectangle, i.e. $R_w = S \times T$ for some $S, T \subseteq \{0,1\}^n$. Suppose further that v is the left child of w, and p_w indicates that it is Alice's turn. Then,

$$R_v = (S \times T) \cap \{(x,y) \mid c_w(x) = 0\} = (S \cap \{x \mid c_w(x) = 0\}) \times T$$

so R_v corresponds to a rectangle M_v. The other cases (right child, Bob's turn etc.) are similar.

Now M_l is monochromatic because each leaf l is labelled with a value $v_l \in \{0,1\}$. Each entry of M_f belongs to exactly one M_l since each input pair (x,y) has a unique path that leads to one of the leaves. □

Example 2.4. Let us illustrate the situation with $PARITY_2$, the $PARITY$ function with $n = 2$. The first round of the proposed protocol induces a partition of the matrix into two rectangles, corresponding to $\{00, 11\} \times \{0,1\}^n$ and

$\{01, 10\} \times \{0,1\}^n$ respectively. The second round further splits each rectangle into two. At the end of the protocol, we get a partition into four monochromatic rectangles. (See Figure 2.3.)

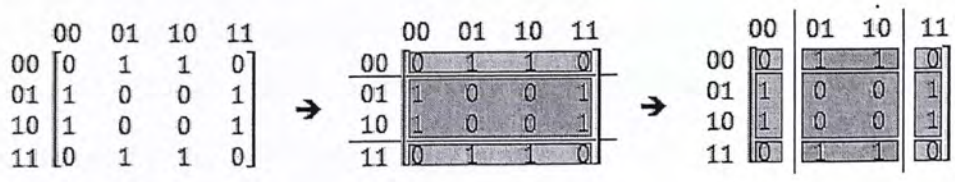

Figure 2.3: Monochromatic tiling for $PARITY_2$

Remark 2.5. It is important to note that the converse of Theorem 2.1 is not true. Figure 2.4 shows a communication matrix with an optimal tiling which does not correspond to any protocol. Indeed, any horizontal cut violates a 1-rectangle, and any vertical cut violates a 0-rectangle.

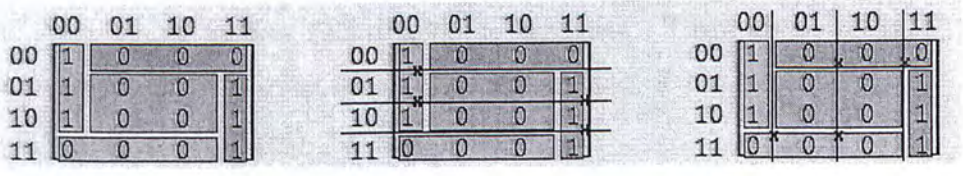

Figure 2.4: An optimal tiling which does not correspond to any protocol

Definition 2.4. We use $\chi_z(f)$ to denote the minimum number of disjoint z-rectangles needed to cover all z-entries in M_f, $z \in \{0,1\}$. Define also $\chi(f) := \chi_0(f) + \chi_1(f)$ to be the minimum number of rectangles in any monochromatic tiling of M_f.

With these notations, we state the ***tiling lower bound*** (also known as the ***partition bound***):

Theorem 2.2. $D(f) \geq \lceil \log_2 \chi(f) \rceil = \lceil \log_2(\chi_0(f) + \chi_1(f)) \rceil$.

Proof. Theorem 2.1 shows that the number of leaves of a protocol is bounded below by $\chi(f)$. The claim then follows from the fact that the height of a tree is at least the logarithm (base 2) of the number of leaves. □

The following theorem shows that the tiling lower bound is at least quadratically close to the deterministic communication complexity:

Theorem 2.3. *[2]* $D(f) \leq O(\log^2 \chi(f))$.

This is a corollary of Aho-Ullman-Yannakakis Theorem, which will be proved in the next chapter.

As an application of Theorem 2.2, we show that the trivial protocol given in Example 2.2 is asymptotically optimal for most functions:

Corollary 2.4. $D(f) = \Theta(n)$ *for almost all* f.

Proof. The upper bound is obvious due to the existence of the trivial protocol. The lower bound follows from a counting argument. There are at most $(2^N)^2$ rectangles in an N-by-N matrix. Thus the number of functions $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ with $\chi(f) \leq k$ is at most

$$((2^{(2^n)})^2 \cdot 2)^k = 2^{k(1+2^{n+1})}.$$

Take $k = 2^{n-2}$, then the fraction of functions with $\chi(f) > 2^{n-2}$ is at least

$$1 - \frac{2^{(2^{n-2})(1+2^{n+1})}}{2^{(2^n)^2}} = 1 - 2^{-2^{n-2}(2^{n+2}-2^{n+1}-1)} \geq 1 - 2^{-2^{n-2}}$$

which is very close to 1. Hence, for most of the functions,

$$D(f) \geq \lceil \log_2 2^{n-2} \rceil = n - 2.$$

□

Another simple corollary of the tiling bound is the ***rectangle size lower bound***. Let $mono(f)$ be the maximum of $\frac{|M|}{|M_f|}$ over all monochromatic rectangles M of M_f. ($|M|$ denotes the number of entries in M.)

Corollary 2.5. $D(f) \geq \lceil -\log_2 mono(f) \rceil$.

Proof. $\chi(f) \geq \frac{|M_f|}{mono(f) \cdot |M_f|} = \frac{1}{mono(f)}$. Then use Theorem 2.2. □

A slightly better lower bound can be obtained by considering the 0-entries and 1-entries of M_f separately, noting that entries of different values cannot be in the same monochromatic rectangle:

Corollary 2.6. $D(f) \geq \lceil \log_2(\frac{1}{mono_0(f)} + \frac{1}{mono_1(f)}) \rceil$, *where* $mono_z(f)$ *denotes the fraction of* z*-entries in* M_f *covered by the largest* z*-rectangle,* $z \in \{0, 1\}$.

These lower bounds may give very poor results:

Example 2.5. Consider the *greater-than function*:

$$GT(x, y) := \begin{cases} 1 & \text{if } x > y \text{ (treating them as binary integers)} \\ 0 & \text{otherwise} \end{cases}$$

Both $mono_0(GT)$ and $mono_1(GT)$ are at least $\frac{1}{4}$, so with the rectangle size technique there is no hope of getting a lower bound better than $D(GT) \geq \Omega(1)$. However, as we shall show in Example 2.9, $D(GT) = \Theta(n)$.

	00	01	10	11
00	0	0	0	0
01	1	0	0	0
10	1	1	0	0
11	1	1	1	0

	00	01	10	11
00	1	1	1	1
01	1	0	1	0
10	1	1	0	0
11	1	0	0	0

Figure 2.5: Matrices for GT_2 (left) and $DISJ_2$ (right)

In many other cases, however, (asymptotically) optimal bounds can be obtained:

Example 2.6. Let EQ be the equality function defined in Example 1.2. M_{EQ} is the identity matrix, whose largest 1-rectangle has size 1. Hence,

$$D(EQ) \geq \lceil \log_2(1 + 2^n) \rceil = n + 1.$$

Recall that the trivial protocol gives $D(EQ) \leq n+1$. Hence, we get the deterministic communication complexity of EQ exactly:

$$D(EQ) = n+1.$$

Example 2.7. Another example is the *disjointness function*:

$$DISJ(x,y) := \begin{cases} 0 & \text{if } x_i = y_i = 1 \text{ for some } 1 \leq i \leq n \\ 1 & \text{otherwise} \end{cases}$$

It is not hard to prove that the number of 1-entries in M_{DISJ} is 3^n, and the maximum size of a 1-rectangle is 2^n. Therefore,

$$D(DISJ) \geq \left\lceil \log_2 \left(1 + \frac{3^n}{2^n}\right) \right\rceil \geq \log_2 \frac{3}{2} \cdot n = \Omega(n).$$

An even weaker method uses the concept of *discrepancy*:

Definition 2.5. The *(uniform) discrepancy* of a Boolean function f is given by

$$disc(f) := \max_{M \,:\, \text{submatrix of } M_f} \frac{||M_0| - |M_1||}{|M_f|}$$

where M_z denotes the set of input pairs corresponding to the z-entries in M, $z \in \{0,1\}$.

Lemma 2.7. $disc(f) \geq mono(f)$.

Proof. The largest monochromatic rectangle in M_f has discrepancy $mono(f)$. □

Corollary 2.8. $D(f) \geq \lceil -\log_2 disc(f) \rceil$.

It turns out that discrepancy is a useful tool in the study of *randomized* communication complexity. More details are given in Chapter 4.

2.3 Fooling set lower bound

In this section we introduce the idea of fooling sets, again suggested in [39].

Definition 2.6. A *z-fooling set* for f is a set $S \subseteq \{0,1\}^n \times \{0,1\}^n$, satisfying:

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

The above definition requires that any two distinct pairs in a fooling set do not belong to the same monochromatic rectangle. Hence, the minimum number of rectangles $\chi(f)$ cannot be less than the size of any fooling set.

Definition 2.7. We use $fool_z(f)$ to denote the size of the largest z-fooling set, $z \in \{0,1\}$. Define $fool(f) := \max\{fool_0(f), fool_1(f)\}$ to be the maximum size of any fooling set for f.

We have arrived at the following ***fooling set lower bound***:

Corollary 2.9. $D(f) \geq \lceil \log_2(fool_0(f) + fool_1(f)) \rceil \geq \lceil \log_2 fool(f) \rceil$.

Example 2.8. M_{EQ} is the identity matrix, with the diagonal entries forming a 1-fooling set of size 2^n. Hence,

$$D(EQ) \geq \lceil \log_2(1+2^n) \rceil = n+1.$$

Again, with the trivial protocol, we get exactly

$$D(EQ) = n+1.$$

Example 2.9. For M_{GT}, the 0-fooling set induced by the diagonal entries and the 1-fooling set induced by the entries just below them gives $fool_0(GT) = 2^n$ and $fool_1(GT) = 2^n - 1$. (The $\leq$ direction holds because $fool_z(f)$ cannot exceed the number of rows in M_f containing z-entries.) Thus as in Example 2.8,

$$D(GT) = n+1.$$

Example 2.10. Similarly, the anti-diagonal entries of M_{DISJ} give a 0-fooling set with 2^n elements, so again we have

$$D(DISJ) = n + 1.$$

Example 2.11. Let RP denote the *relatively-prime function*:

$$RP(x,y) := \begin{cases} 1 & \text{if } x, y \text{ (treated as binary integers) are relatively prime} \\ 0 & \text{otherwise} \end{cases}$$

The diagonal elements $\{(x,x) \mid x : \text{prime}\}$ form a fooling set of size $\Omega\left(\frac{N}{\ln N}\right) \geq \Omega(\sqrt{N})$. Hence also

$$D(RP) \geq \Omega(\log \sqrt{N}) = \Omega(n).$$

The rectangle size bound and the fooling set bound can be considered as special cases of a more general theorem:

Theorem 2.10. *(See [17].) Let μ be a probability distribution over $\{0,1\}^n \times \{0,1\}^n$. If every monochromatic rectangle M has measure $\mu(M) \leq \delta$, then $D(f) \geq \lceil -\log_2 \delta \rceil$.*

Proof. There must be at least $\frac{1}{\delta}$ rectangles in any partition of M_f. Hence the tiling lower bound implies the claim. □

From Theorem 2.10 we get the rectangle size lower bound by taking μ as the uniform distribution over all input pairs. The fooling set lower bound is obtained as follows: given a 0-fooling set S_0 and a 1-fooling set S_1, let S be the disjoint union $S_0 \sqcup S_1$. Define μ as the uniform distribution over S (and $\mu(x,y) = 0$ for all other $(x,y) \notin S$). Since any monochromatic rectangle can contain at most one element in S, its weight cannot exceed $\frac{1}{|S|} = \frac{1}{|S_0|+|S_1|}$. Thus, $D(f) \geq \left\lceil -\log_2 \frac{1}{|S_0|+|S_1|} \right\rceil = \lceil \log_2(|S_0| + |S_1|) \rceil$, which gives our fooling set bound in Corollary 2.9.

2.4 Rank lower bound

Here we give an algebraic way to lower-bound the communication complexity of f. The following results are introduced by Mehlhorn and Schmidt [27].

Recall that the rank of a matrix is the dimension of its row space. We extend this notation and define the rank of a Boolean function:

Definition 2.8.

$$\begin{aligned} rank(f) &:= rank(M_f) \text{ over the reals, and} \\ rank_2(f) &:= rank(M_f) \text{ over } GF(2). \end{aligned}$$

Lemma 2.11. $\chi(f) \geq rank(f) \geq rank_2(f)$.

Proof. Suppose we have found an optimal monochromatic tiling for M_f. Let B_i be the N-by-N matrix whose 1-entries are exactly those in the i-th 1-rectangle in the tiling. Then we have

$$M_f = \sum_{i=1}^{\chi_1(f)} B_i.$$

Hence, by subadditivity of rank,

$$\begin{aligned} rank(M_f) &\leq \sum_{i=1}^{\chi_1(f)} rank(B_i) \\ &= \sum_{i=1}^{\chi_1(f)} 1 \\ &= \chi_1(f) \\ &\leq \chi(f). \end{aligned}$$

For the second part of inequality, let $r_1, \cdots, r_N$ denote the N rows of M_f. We first find a basis of $k := rank_2(f)$ vectors $S := \{r_{i_1}, \cdots, r_{i_k}\}$ for the row space over $GF(2)$ out of the vectors $\{r_1, \cdots, r_N\}$. This set is linearly independent not only over $GF(2)$, but also over the reals. Hence,

$$rank_2(f) = k \leq rank(f).$$

□

This lemma, together with Theorem 2.2, suggests the following ***rank lower bound***:

Theorem 2.12. $D(f) \geq \lceil \log_2 rank(f) \rceil \geq \lceil \log_2 rank_2(f) \rceil$ *for* $f \not\equiv 0$.

Note that $\log_2 rank(f)$ is well-defined only when $rank(f) > 0$, i.e. $f \not\equiv 0$.

A better bound is obtained by considering also the 0-rectangles:

Corollary 2.13. $D(f) \geq \lceil \log_2(2rank(f) - 1) \rceil$ *for* $f \not\equiv 0$.

Proof. Consider $\bar{f}$, the negation of f. $M_{\bar{f}} = J - M_f$, where J is the all-one matrix. This gives

$$rank(M_f) \leq 1 + rank(M_{\bar{f}}).$$

Similar argument as in the lemma shows that

$$\chi_0(f) \geq rank(M_{\bar{f}}).$$

Hence,

$$\begin{aligned} \chi(f) &= \chi_0(f) + \chi_1(f) \\ &\geq rank(M_{\bar{f}}) + rank(M_f) \\ &\geq (rank(M_f) - 1) + rank(M_f) \\ &\geq 2rank(M_f) - 1. \end{aligned}$$

Here, $\log_2(2rank(f) - 1)$ is only defined when $rank(f) > \frac{1}{2}$, i.e. again $f \not\equiv 0$. □

Example 2.12. We demonstrate the use of the above theorem with GT. The rank of M_{GT} is $2^n - 1$, so

$$D(GT) \geq \lceil \log_2(2(2^n - 1) - 1) \rceil = \Theta(n).$$

The theorem also gives similar conclusions for EQ and $DISJ$.

Remark 2.6. The rank lower bound gives yet another reason why the trivial protocol is asymptotically optimal for most functions. Komlós [14] has proved that most Boolean matrices have full rank, hence $D(f) \geq \lceil \log_2 rank(M_f) \rceil = \Omega(n)$ for most f.

Corollary 2.14. *If all the rows of M_f are distinct, then $D(f) \geq \lceil \log_2 n \rceil$.*

Proof. In view of the rank lower bound, it suffices to show that

$$rank_2(f) \geq n \text{ if all the rows of } M_f \text{ are distinct.}$$

Indeed, if there is a basis with less than n vectors for the row space of M_f over $GF(2)$, then the number of vectors in the row space (= the span of the basis) is less than 2^n, which is the number of distinct rows in M_f. This is a contradiction. □

We may also upper-bound the communication complexity using the rank:

Theorem 2.15. $D(f) \leq rank_2(f) + 1 \leq rank(f) + 1$.

Proof. We explicitly give a protocol for f with cost $rank_2(f) + 1$. Before communication, Alice and Bob agree on any basis $\{v_1, \cdots, v_k\}$ for the row space of M_f over $GF(2)$. To determine the message to be transmitted, Alice first finds her row r_x in M_f. There is a unique way to write

$$r_x = \sum_{i=1}^{k} a_i v_i$$

where $a_i \in \{0, 1\}$ and the addition is done in $GF(2)$. Alice then sends the a_i's to Bob. On receiving the a_i's, Bob can recover r_x and hence determine the value of $f(x, y)$ by looking at the corresponding entry. The total number of bits transmitted by Alice is $k = rank_2(f)$. □

It is an open question whether there is a constant $c > 1$ such that $D(f) \leq O(\log^c rank(f))$ for every function f. This is commonly known as the ***log-rank conjecture***. (See, for instance, [29, 34].)

2.5 Comparison of the bounds

Among all the above-mentioned bounds, the tiling bound is the tightest, since all the other bounds are derived from it. The comparison between the fooling set bound and the rank lower bound is more interesting. There are cases where the fooling set bound works better, but there are also functions where the rank gives much tighter bounds.

Example 2.13. Define the ALL function as follows:

$$ALL(x,y) = \begin{cases} 1 & \text{if } x = y = 1^n \\ 0 & \text{otherwise} \end{cases}$$

It has a 0-fooling set of size 2 and a 1-fooling set of size 1, so the fooling set bound says $D(ALL) \geq 2$. (And indeed, $D(ALL) = 2$.) However, the rank of M_{ALL} is only 1, so the rank lower bound gives $D(ALL) \geq 0$, which is completely meaningless.

Dietzfelbinger, Hromkovič and Schnitger [9] have made a thorough investigation of the two lower-bounding techniques. They have shown that the lower bound obtained by the rank method, ignoring the constant factors, is always at least as strong as the fooling set bound. It is not difficult (but a bit lengthy) to show the following inequality:

Lemma 2.16. $fool_1(f) \leq rank_2(M_f \otimes {M_f}^T) = (rank_2(f))^2$, *where* $\otimes$ *denotes the Kronecker product.*

Applying the lemma to $\bar{f}$ gives

$$fool_0(f) \leq (rank_2(\bar{f}))^2 \leq (rank_2(f) + 1)^2.$$

Hence, we conclude that

$$fool_0(f) + fool_1(f) \leq 2(rank_2(f) + 1)^2$$

i.e.

$$\lceil \log_2(rank_2(f)) \rceil \geq \Omega(\log(fool_0(f) + fool_1(f))).$$

On the other hand, there are cases where the rank lower bound gives exponentially better results than the fooling set method:

Example 2.14. Define the *inner-product function* as follows:

$$IP(x, y) = \sum x_i y_i \bmod 2$$

It is known that $rank(IP) = 2^n - 1$, hence $D(IP) = \Theta(n)$. On the other hand, $rank_2(IP) = n$, which implies that $fool_0(IP) + fool_1(IP) \leq 2(n+1)^2$. Thus, the rank method gives an optimal bound for IP, while the bound given by the fooling set technique is exponentially weaker.

Another measure for comparison is the difficulty of computing the related quantities. The rank of a matrix can be efficiently computed using Gaussian elimination. For the other methods, however, no polynomial-time algorithm is known. Even the computation of $mono(f)$, though seemingly simple, is NP-complete [26].

Chapter 3

Nondeterministic Communication Complexity

In this chapter we study the nondeterministic communication model. Section 3.1 contains the formal definitions. The largest possible gap between the deterministic and nondeterministic complexities is explored in Section 3.2. Finally, the famous Aho-Ullman-Yannakakis Theorem, which relates the two models, is proved in Section 3.3.

3.1 Definitions

The materials presented in this section can be found in [19].

Definition 3.1. The *nondeterministic communication complexity* of f, denoted by $N^1(f)$, is defined by:

$$\begin{aligned} N^1(f) \;:=\; & \min\{k \mid \exists A, B : \{0,1\}^n \times \{0,1\}^k \to \{0,1\} \text{ s.t.} \\ & f(x,y) = 1 \Leftrightarrow \exists z \in \{0,1\}^k \text{ s.t. } A(x,z) = B(y,z) = 1\} \end{aligned}$$

z is often referred to as a *witness*, *proof*, or *certificate*. Intuitively, it gives sufficient information to the players to conclude independently, with their own inputs, that $f(x,y) = 1$.

Definition 3.2. The *co-nondeterministic communication complexity* of f, denoted by $N^0(f)$, is defined by:

$$N^0(f) := N^1(\bar{f})$$

As mentioned in Chapter 1, nondeterministic communication complexity is related to *rectangle covers* (not partitions), where overlaps are allowed:

Definition 3.3. C^z is defined as the minimum number of z-rectangles required to cover all z-entries of M_f, $z \in \{0,1\}$.

Theorem 3.1. $N^z(f) = \lceil \log_2 C^z(f) \rceil$.

Proof. It suffices to show that

$$N^1(f) = \lceil \log_2 C^1(f) \rceil$$

The case $z = 0$ then follows by considering $\bar{f}$.

We first prove the $\leq$ direction. Let $k := C^1(f)$. Alice and Bob first agree on an optimal rectangle covering of M_f. Each of the 1-rectangles in the cover is given a unique index from $\{0, 1, ..., k-1\}$, which takes $\lceil \log_2 k \rceil$ bits. The witness z can then be the index of a 1-rectangle that contains (x, y). Let A return 1 if and only if the z-th 1-rectangle intersects with row x, and similarly let B return 1 iff that rectangle intersects with column y. Note that if $f(x,y) = 1$, we can always find a z such that $A(x,z) = B(y,z) = 1$. However, if $f(x,y) = 0$, then (x,y) is not contained in any 1-rectangle, so for every z either $A(x,z) = 0$ or $B(y,z) = 0$. Hence, $N^1(f) \leq \lceil \log_2 k \rceil$.

For the opposite direction, we write $k := N^1(f)$ and try to find a 1-covering with at most 2^k rectangles. Now for $z \in \{0,1\}^k$, take

$$\begin{aligned} R_z &:= \{(x,y) \mid A(x,z) = B(y,z) = 1\} \\ &= \{x \mid A(x,z) = 1\} \times \{y \mid B(y,z) = 1\} \end{aligned}$$

Each R_z induces a rectangle M_z. The collection $\{M_z \mid z \in \{0,1\}^k\}$ covers all 1-entries by definition of N^1, and this covering has size $\leq 2^k$. □

Corollary 3.2. $D(f) \geq \max\{N^0(f), N^1(f)\}$.

Proof. We just need to note that a partition can be viewed as a special case of covering (which has no overlapping), which implies

$$C^z(f) \leq \chi_z(f).$$

Hence,

$$\begin{aligned} N^z(f) &= \lceil \log_2 C^z(f) \rceil \\ &\leq \lceil \log_2 \chi_z(f) \rceil \\ &\leq \lceil \log_2 \chi(f) \rceil \\ &\leq D(f) \end{aligned}$$

□

Remark 3.1. As the name suggests, we could have defined nondeterministic communication complexity as the lowest cost of protocols that can take nondeterministic steps, and have at least one accepting path for every input pair satisfying $f(x, y) = 1$. However, the quantity derived in this way may not always be equal to our $N^1(f)$. For the ALL function defined in Example 2.13, $C^1(ALL) = 1$, so $N^1(ALL) = \lceil \log_2 1 \rceil = 0$. But certainly, being a non-constant function, ALL cannot be computed by a protocol that involves no communication at all. Nonetheless, the values under the two definitions differ only by at most one.

3.2 Gaps between $N^0(f)$, $N^1(f)$ and $D(f)$

This section deals with the difference of nondeterministic and deterministic complexities. The results are mentioned in [17].

We observe that the proof of Theorem 2.10 (concerning measures of monochromatic rectangles) works not only for rectangle partition, but also for covers. This

means that the familiar techniques of fooling set and rectangle size lower bounds still work under the nondeterministic model.

Example 3.1. Recall the GT function defined in Example 2.5. As mentioned in Example 2.9, it has a 0-fooling set of size 2^n, and a 1-fooling set of size $2^n - 1$. That implies

$$\begin{aligned} C^0(GT) &\geq 2^n \\ C^1(GT) &\geq 2^n - 1 \end{aligned}$$

Hence,

$$N^0(GT) \geq \lceil \log_2 2^n \rceil = n$$

$$N^1(GT) \geq \lceil \log_2 (2^n - 1) \rceil = \Theta(n).$$

Example 3.2. Recall the EQ function defined in Example 1.2. As shown in Example 2.8, $fool_1(EQ) = 2^n$, so $C^1(EQ) = 2^n$, i.e.

$$N^1(EQ) = \lceil \log_2 2^n \rceil = n.$$

For the co-nondeterministic communication complexity, an obvious choice of witness z for $f(x, y) = 0$ is a position i, $1 \leq i \leq n$, followed by a bit indicating whether ($x_i = 0$ and $y_i = 1$) or ($x_i = 1$ and $y_i = 0$). In terms of rectangle cover, each of the two cases gives n 0-rectangles. (See Figure 3.1.) Hence $C^0(EQ) \leq 2n$, i.e.

$$N^0(EQ) \leq \lceil \log_2 2n \rceil = \lceil \log_2 n \rceil + 1.$$

We see that there is an exponential gap between $N^0(EQ)$ and $N^1(EQ)$ (and also $D(EQ)$).

Example 3.3. Next we look at the $DISJ$ function defined in Example 2.7. Again the fooling set argument shows that

$$N^1(DISJ) = n.$$

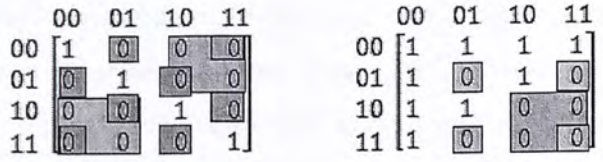

Figure 3.1: Communication matrix for EQ_2 (left) and $DISJ_2$ (right)

For the 0-entries, we take the witness z to be a position i, $1 \leq i \leq n$, such that $x_i = y_i = 1$. In the language of covering, this corresponds to a 0-covering of n rectangles. Hence $C^0(DISJ) \leq n$, i.e.

$$N^0(DISJ) \leq \lceil \log_2 n \rceil.$$

We have an exponential gap between $N^0(DISJ)$ and $N^1(DISJ)$ (and $D(DISJ)$).

The gaps in Examples 3.2 and 3.3 are essentially the largest that we can get for any function, as the following theorem shows:

Theorem 3.3. $D(f) \leq 2^{N^z(f)} + 1$, $z \in \{0,1\}$.

Proof. Again it suffices to prove the claim for $z = 1$. Write $k := N^1(f)$. For each $z \in \{0,1\}^k$, Alice sends the bit $A(x,z)$ to Bob. After that, Bob computes the $B(y,z)$'s and announces that $f(x,y) = 1$ iff for some z both $A(x,z) = B(y,z) = 1$. Correctness of this protocol follows from the definition of $N^1(f)$. The number of bits sent by Alice is $2^k = 2^{N^1(f)}$. □

3.3 Aho-Ullman-Yannakakis Theorem

This section is devoted to the important theorem by Aho, Ullman and Yannakakis [2], which states that the deterministic communication complexity of f is dominated asymptotically by the product of the nondeterministic and co-nondeterministic communication complexities:

Theorem 3.4. ***(Aho-Ullman-Yannakakis Thm.)*** $D(f) = O(N^0(f)N^1(f))$.

Proof. The main insight is that a 0-rectangle cannot intersect with a 1-rectangle both in rows and in columns. (See Figure 3.3.) Hence, given any set of 1-rectangles, a 0-rectangle either intersects in rows with $\leq \frac{1}{2}$ of them, or intersects in columns with $\leq \frac{1}{2}$ of them (or both).

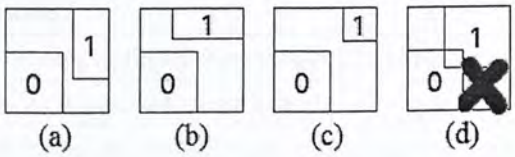

Figure 3.2: Intersecting (a) in rows, (b) in columns, and (c) no intersection (note that (d) is impossible)

Figure 3.4 shows a protocol for Alice and Bob. Before communication, they first agree on an optimal rectangle cover, and give each of the 0-rectangles an index.

Here we give a detailed explanation of the protocol. $\mathcal{A}$ is for 'alive', meaning that only the rectangles in $\mathcal{A}$ may contain (x, y). $\mathcal{A}$ initially contains all 1-rectangles in the cover, and is updated only in Lines 11 and 18. For Line 11, since the R found by Alice intersects with row x, those rectangles not intersecting in rows with R cannot intersect with row x and therefore can be safely removed. For similar reasons, the removal of rectangles in Line 18 is also safe.

Next we shall analyse its correctness, i.e. all values returned are correct. The only value-returning instructions are in Lines 4 and 20. On reaching Line 4, $\mathcal{A}$ is empty, meaning that (x, y) is not contained in any 1-rectangle. So, it is correct to deduce that $f(x, y) = 0$. On reaching Line 20, both Alice and Bob fail to find an R, which means (by the 'main insight') that no 0-rectangle intersects with row x and column y at the same time. Thus, $f(x, y) = 1$.

Now, we need to show that the protocol indeed stops and returns an output. We observe that every rectangle-removal instruction (Lines 11 and 18) reduces the size of $\mathcal{A}$ by $\geq \frac{1}{2}$. Hence, the process must stop in $\lceil \log_2 C^1(f) \rceil + 1 = N^1(f) + 1$

Line no.	Instruction
1	Initialize $\mathcal{A}$:= the set of all 1-rectangles in the cover
2	Repeat lines 3-20:
3	If $\mathcal{A}$ is empty,
4	Conclude that $f(x, y) = 0$ and the protocol is finished
5	Otherwise,
6	Alice tries to find a 0-rectangle R in the cover such that:
7	It intersects with row x, and
8	It intersects in rows with $\leq$ ½ of the 1-rectangles in $\mathcal{A}$.
9	If such an R is found,
10	Alice sends the index of R to Bob, and
11	Rectangles NOT intersecting in rows with R are removed from $\mathcal{A}$.
12	Otherwise,
13	Alice asks Bob to find a 0-rectangle R in the cover such that:
14	It intersects with column y, and
15	It intersects in columns with $\leq$ ½ of the 1-rectangles in $\mathcal{A}$.
16	If found,
17	Bob sends the index of R to Alice, and
18	Rectangles NOT intersecting in columns with R are removed from $\mathcal{A}$.
19	Otherwise,
20	Conclude that $f(x, y) = 1$ and the process is finished.

Figure 3.3: The protocol for Aho-Ullman-Yannakakis Theorem

rounds.

Finally, we determine the cost of the protocol. Within each round, it takes 1 bit for each player to indicate whether he or she succeeds in finding an R, and another $N^0(f)$ bits for the index of R. Hence the cost is bounded above by

$$(N^0(f) + 2)(N^1(f) + 1) = O(N^0(f)N^1(f)).$$

□

Corollary 3.5. $D(f) \leq O(\log^2 \chi(f))$.

Proof. We simply note that $N^z(f) \leq \lceil \log_2 \chi(f) \rceil$. (See Corollary 3.2.) □

We conclude this chapter by showing that any one of the $N^z(f)$ in the above theorem can be replaced by the logarithm of the rank:

Theorem 3.6. *[22]* $D(f) = O(N^z(f) \log rank(f)), z \in \{0, 1\}$.

Proof. Once again we only prove it for $z = 1$. The key observation this time is as follows: suppose we have a Boolean matrix R of rank t. Suppose M is a 0-rectangle corresponding to $S \times T$, and let M_A be the submatrix of R consisting of only the rows S, and M_B be the submatrix consisting of only the columns T. (See Figure 3.4.) It is easy to see that $rank(M_A) + rank(M_B) \leq t$. This implies that either $rank(M_A) \leq \frac{t}{2}$ or $rank(M_B) \leq \frac{t}{2}$.

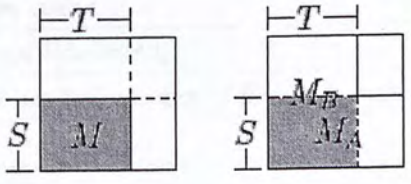

Figure 3.4: Meaning of M_A and M_B in the proof of Theorem 3.6

A protocol for Alice and Bob is presented in Figure 3.6. Note that it is essentially the same as that for Theorem 3.4, only with the set of rectangles $\mathcal{A}$ replaced by t. Analysis as in Theorem 3.4 proves our claim.

□

Remark 3.2. Theorems 3.4 and 3.6 can also be proved by solving recurrence relations on the number of leaf nodes. See [17] for more details.

Line no.	Instruction
1	Initialize $t := rank(f)$.
2	Repeat lines 3-20:
3	If $t = 0$,
4	Conclude that $f(x, y) = 0$ and the protocol is finished
5	Otherwise,
6	Alice tries to find a 0-rectangle R in the cover such that:
7	It intersects with row x, and
8	R_A has rank $\leq \frac{1}{2} t$.
9	If such an R is found,
10	Alice sends the index of R to Bob, and
11	Set $t := rank(R_A)$.
12	Otherwise,
13	Alice asks Bob to find a 0-rectangle R in the cover such that:
14	It intersects with column y, and
15	R_B has rank $\leq \frac{1}{2} t$.
16	If found,
17	Bob sends the index of R to Alice, and
18	Set $t := rank(R_B)$.
19	Otherwise,
20	Conclude that $f(x, y) = 1$ and the process is finished.

Figure 3.5: The protocol for Theorem 3.6

Chapter 4

Randomized Communication Complexity

We turn our attention to the randomized communication model, where the players are allowed to 'toss random coins'. Two useful theorems are stated in Section 4.1 as preliminaries. The rigorous definition of randomized protocols is given in Section 4.2. Section 4.3 shows that the exact value of the error constant in the definition is not important, and Section 4.4 talks about the largest possible gap between the deterministic and randomized complexities. In Section 4.5 the public coin communication model is introduced, and its relationship with distributional complexity is explained in Section 4.6.

4.1 Preliminaries

We first state two famous theorems that will be useful in later sections.

Theorem 4.1. *(Chernoff bound) Let $X_1, X_2, \cdots, X_t$ be independent Poisson trials with $\Pr[X_i = 1] = p_i$ and $\Pr[X_i = 0] = 1 - p_i$. Then for $X := \sum X_i$ and $\mu := E[X]$, for any $\delta > 0$,*

$$\Pr[X > (1+\delta)\mu] < \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}.$$

Chernoff bound gives exponential falloff of the tail probability with distance from the mean. For our purpose, we shall use the following simplified version:

Corollary 4.2. *With the assumptions in Theorem 4.1, for any* $\delta < 2e - 1$,

$$\Pr[X > (1+\delta)\mu] < \exp(-\mu\delta^2/4).$$

The next theorem is one of the most important results in game theory:

Theorem 4.3. ***(Von Neumann's Minimax Theorem)*** *For any finite two-person zero-sum game, there exists a unique value* V *and a pair of mixed strategies* μ *and* ν *such that:*

1. *Player 1 can guarantee that the expected payoff is at least* V *by applying* μ, *no matter which strategy Player 2 uses, and*
2. *Player 2 can guarantee that the expected payoff is not more than* V *by applying* ν, *no matter which strategy Player 1 uses.*

The proof of Chernoff bound can be found in [10]. For an introduction to game theory, see [31].

4.2 Definitions

Instead of tossing coins *as they go* as mentioned in Chapter 1, we may also let the players make all necessary coin tosses *before communication.* Having said that, we have implicitly assumed that the number of random bits involved in a randomized protocol is finite. This assumption is reasonable because we expect a randomized protocol to have a cost no larger than the trivial protocol, so we should not need too many random bits.

Here we give a definition of randomized protocol trees analogous to Definition 2.1 of deterministic protocols:

Definition 4.1. Let R be the (finite) set of random strings, and ν_A, ν_B be probability distributions over R. A *randomized communication protocol* is represented by a binary tree. Each intermediate vertex v is labelled with $p_v \in \{0,1\}$ indicating which player is responsible for this turn, followed by a function $c_v : \{0,1\}^n \times R \to \{0,1\}$ indicating what bit is to be sent according to the player's input and his result of coin tosses. Each leaf node l is labelled with a value $v_l \in \{0,1\}$. The *cost* of the protocol is the height of the tree.

The execution of randomized protocols resembles that of deterministic ones, except that now random strings are involved. We use r_A and r_B to denote the random strings for Alice and Bob respectively. r_A is drawn from R according to ν_A, and r_B is drawn according to ν_B. These are done independently before the players communicate. The action to be taken at each node v is determined by the value of p_v, and correspondingly $c_v(x, r_A)$ or $c_v(y, r_B)$.

Definition 4.2. A randomized protocol *computes f with error ϵ* if for every input pair (x, y), it returns $f(x, y)$ with probability $\geq 1 - \epsilon$. Denote

$$R_\epsilon(f) := \min_{\mathcal{P} \,:\, \text{randomized protocol}} \{cost(\mathcal{P}) \mid \mathcal{P} \text{ computes } f \text{ with error } \epsilon\}$$

The *randomized communication complexity* of f is then defined as

$$R(f) := R_{1/3}(f).$$

Example 4.1. Recall that the EQ function defined in Example 1.2 satisfies $D(EQ) = n + 1$. We now give a randomized protocol due to Rabin and Yao whose cost is $O(\log n)$.

Treat x, y as binary integers. Alice picks one of the first n^2 primes p uniformly randomly. She sends it to Bob, followed by the value of $x \bmod p$. Bob then concludes that $x = y$ iff $x \bmod p = y \bmod p$.

The protocol errs only if $x \neq y$ and $x \bmod p = y \bmod p$, i.e. $|x - y|$ is a non-zero multiple of the randomly chosen p. Since the number of prime factors

of $|x-y|$ cannot exceed $\log_2 |x-y| < n$, the probability of error of the protocol is less than $\frac{n}{n^2} = \frac{1}{n}$. For the cost of the protocol, we recall the well-known fact that the number of primes less than N is $\Omega\left(\frac{N}{\ln N}\right)$, hence the n^2-th prime is upper-bounded by, say, $O(n^3)$. That means its binary representation has $O(\log n)$ bits. The value of $x \bmod p$ is less than p, so it also takes $O(\log n)$ bits. Over all, we have shown that

$$R_{1/n}(EQ) \leq O(\log n) + O(\log n) + 1 = O(\log n).$$

Example 4.2. We also have

$$R(GT) = O(\log n).$$

Here we show a weaker upper bound of $O(\log^2 n)$. We do binary search on the most significant bit that x and y differs. For each phase of the search, Alice and Bob run the randomized EQ protocol on the first half of their input bits. If it returns 0 (i.e. not equal), they go to the next phase with the first half of the bits; otherwise, they proceed with the second half. In each phase the number of bits involved is halved. The process stops when only ≤ 8 bits are left, and the two players then exchange those remaining bits to determine their final output. The cost of this protocol is

$$O(\log n) \cdot \lceil \log n \rceil = O(\log^2 n).$$

The probability of error comes from the error of the EQ protocol. It is bounded above by

$$\frac{1}{n} + \frac{1}{n/2} + \cdots + \frac{1}{8} \leq \frac{1+2+\cdots+n/8}{n} \leq \frac{n/4}{n} = \frac{1}{4}.$$

4.3 Error reduction

A first-time reader may feel strange why we take $\epsilon = \frac{1}{3}$ but not other values in the definition of randomized communication complexity. Here we show that the

actual value of ϵ is not important, as long as $0 < \epsilon < \frac{1}{2}$. This result is mentioned in [17]:

Theorem 4.4. *For* $0 < \epsilon' < \epsilon < \frac{1}{2}$, $R_{\epsilon'}(f) = O\left(R_\epsilon(f) \cdot \log \frac{1}{\epsilon'}\right)$.

Proof. Let $\mathcal{P}$ be a randomized protocol for f of cost $k = R_\epsilon(f)$ with error ϵ. The idea is to run the protocol $t = O\left(\log \frac{1}{\epsilon'}\right)$ times, and take the majority result to be the final output.

Let X_i be the indicator of the event that the i-th trial returns a wrong answer, $1 \leq i \leq t$. Take $\delta := \frac{1}{2\epsilon} - 1$, and $t := \frac{4 \ln \frac{1}{\epsilon'}}{\epsilon \delta^2}$, and apply Chernoff bound to upper-bound the error of our scheme:

$$\begin{aligned} \Pr\left[\frac{1}{t}\sum X_i > \frac{1}{2}\right] &= \Pr\left[\sum X_i > (1+\delta)\epsilon t\right] \\ &< \exp(-\epsilon t \delta^2/4) \\ &= \exp\left(-\epsilon \cdot \frac{4 \ln \frac{1}{\epsilon'}}{\epsilon \delta^2} \cdot \delta^2/4\right) \\ &= \epsilon' \end{aligned}$$

The cost of our scheme is at most

$$kt \leq R_\epsilon(f) \cdot \frac{4 \ln \frac{1}{\epsilon'}}{\epsilon \left(\frac{1}{2\epsilon} - 1\right)^2} = O\left(R_\epsilon(f) \cdot \log \frac{1}{\epsilon'}\right).$$

□

Corollary 4.5. *For* $0 < \epsilon < \frac{1}{2}$, $R_\epsilon(f) = O\left(R(f) \cdot \log \frac{1}{\epsilon}\right)$.

The following corollary can be deduced by replacing ϵ' by $\frac{1}{n}$ in the proof of Theorem 4.4:

Corollary 4.6. $R_{1/n}(f) = O\left(R(f) \cdot \log n\right)$.

4.4 Exponential gap with $D(f)$

Obviously, $R(f) \leq D(f)$ for all f. In Example 4.1 we see that the gap between $R(EQ)$ and $D(EQ)$ is at least exponential. The following theorem from [17] shows that the gap cannot be larger than exponential:

Theorem 4.7. $R(f) = \Omega(\log D(f))$.

Proof. Let $\mathcal{P}$ be a randomized protocol of cost $k = R(f)$ that computes f with error $\epsilon = \frac{1}{3}$. We conduct a *deterministic simulation* of $\mathcal{P}$ with cost $O(2^k)$, hence proving the claim.

For each leaf node l of $\mathcal{P}$, Alice computes the probability of reaching it *from her side*: Alice locates all the intermediate nodes that correspond to her turns within the path leading to l, finds for each of those nodes the probability (over the random strings, with respect to her input x) of staying in that path, and computes their product. The value she finds is

$$p_l^A := \Pr_{r_A \sim \nu_A} [\exists y \in \{0,1\}^n, r_B \in R \text{ s.t. the leaf } l \text{ is reached} \mid x].$$

Bob can similarly compute the probability p_l^B *from his side*:

$$p_l^B := \Pr_{r_B \sim \nu_B} [\exists x \in \{0,1\}^n, r_A \in R \text{ s.t. the leaf } l \text{ is reached} \mid y].$$

The probability of reaching l for their input pair (x, y) is then

$$p_l = p_l^A \cdot p_l^B.$$

By definition of $\mathcal{P}$, the value $z \in \{0,1\}$ for which the sum of p_l's over the z-labelled leaves is greater than $\frac{1}{2}$ is the correct value of $f(x, y)$.

The only problem now is that we cannot send the exact value of real numbers which may take infinitely many bits. The remedy is to send only $k + 3$ bits of accuracy. The rounding error for p_l is bounded above by the rounding error of p_l^A, which is at most $2^{-(k+3)} = \frac{1}{8} \cdot 2^{-k}$. The number of leaves is at most 2^k, so the total rounding error for the sum of probabilities is at most $2^k \cdot \left(\frac{1}{8} \cdot 2^{-k}\right) = \frac{1}{8}$. This value, added to the error of $\mathcal{P}$ of $\epsilon = \frac{1}{3}$, is still less than $\frac{1}{2}$, so Bob can get the final output correctly. The total number of bits transmitted by Alice is bounded above by

$$2^k \cdot (k + 3) = O(2^k) = O\left(2^{R(f)}\right).$$

□

Example 4.3. We have shown in Example 4.1 that $R(EQ) = O(\log n)$. Together with the above theorem, we conclude that this bound is asymptotically tight:

$$R(EQ) = \Theta(\log n).$$

4.5 The public coin model

The discussion above assumes that Alice and Bob have their own *private* random coins, and they cannot see the other's result of coin tosses. Here we study the *public coin model*, in which the two players share a common random string.

A neat way to study public coin protocols is to treat them as *collections of deterministic protocols*, and the public random string tells the players which one of them they are to use. Since the number of random bits is finite, the size of the collection is also finite.

Definition 4.3. A *public coin protocol* $\mathcal{P}$ is a collection of deterministic protocols $\{P_1, \cdots, P_k\}$ associated with a probability distribution ν. The *cost* of $\mathcal{P}$ is defined by

$$cost(\mathcal{P}) := \max_i \{cost(P_i)\}.$$

We say that $\mathcal{P}$ *computes* f *with error* ϵ if for all (x, y),

$$err_f(\mathcal{P}, x, y) := \Pr_{P_i \sim \nu}[P_i(x, y) \neq f(x, y)] \leq \epsilon.$$

Definition 4.4.

$$R_\epsilon^{pub}(f) := \min_{\mathcal{P} \text{ : public coin protocol}} \{cost(\mathcal{P}) \mid err_f(\mathcal{P}, x, y) \leq \epsilon \text{ for all } (x, y)\}$$

The *randomized communication complexity* of f *under the public coin model* is then

$$R^{pub}(f) := R_{1/3}^{pub}(f).$$

It is straightforward to check that these definitions correspond exactly to those of the private coin model, except that the players now jointly use one random string. We note that $R_\epsilon^{pub}(f) \leq R_\epsilon(f)$, because a private coin protocol can be turned into a public coin one by simply taking the concatenation of r_A and r_B as the public string.

Example 4.4. Consider the following randomized protocol for EQ with a public random string z drawn from $\{0,1\}^n$ uniformly: Alice computes the inner product $\langle x, z\rangle$ over $GF(2)$ and sends this one-bit result to Bob. Bob then announces $x = y$ if and only if $\langle x, z\rangle = \langle y, z\rangle$.

This protocol fails only when $x \neq y$ and $\langle x, z\rangle = \langle y, z\rangle$. Note that

$$\Pr_{z\sim\text{uniform}}(\langle x, z\rangle = \langle y, z\rangle \mid x \neq y) = \frac{1}{2}.$$

Repeating the protocol with a different z reduces the error probability to $\left(\frac{1}{2}\right)^2 = \frac{1}{4}$. Hence, we conclude that

$$R^{pub}(EQ) = O(1).$$

As expected, EQ (once again) demonstrates the largest possible gap between the two models of randomized communication. The following important theorem by Newman [28] says that we can turn a public coin protocol into a private coin one using only $O(\log n)$ extra bits:

Theorem 4.8. *For $0 < \epsilon' < \epsilon < \frac{1}{2}$ with $\epsilon + \epsilon' < \frac{1}{2}$,*

$$R_{\epsilon+\epsilon'}(f) \leq R_\epsilon^{pub}(f) + O\left(\log n + \log\frac{1}{\epsilon'}\right).$$

Proof. Let $\mathcal{P}$ be a public-coin protocol with cost $R_\epsilon^{pub}(f)$. For each deterministic protocol in the corresponding set $\{P_1, \cdots, P_k\}$, for each (x, y), define $X_i(x, y)$ as the *error indicator*:

$$X_i(x,y) := \begin{cases} 1 & \text{if } P_i(x,y) \neq f(x,y) \\ 0 & \text{otherwise} \end{cases}$$

The idea is to pick t of the P_i's randomly (possibly with repetition), and construct a private coin protocol with the resulting collection $\{P_{i_1}, \cdots, P_{i_t}\}$: Alice draws an integer j from $\{1, \cdots, t\}$ uniformly randomly, sends it to Bob, and then they run P_{i_j} on (x, y). We need to show that there is some choice of $t = 2^{O(\log n + \log \frac{1}{\epsilon'})}$ and $P_{i_1}, \cdots, P_{i_t}$ such that the probability of error for the corresponding construction is at most $\epsilon + \epsilon'$.

Write $\delta := \frac{\epsilon'}{\epsilon}$. Take $t := \frac{8\epsilon n \cdot \ln 2}{\delta^2}$. Then for any (x, y),

$$\begin{aligned}
\Pr\left[\frac{1}{t}X_i(x,y) > \epsilon + \epsilon'\right] &= \Pr\left[\frac{1}{t}X_i > (1+\delta)\epsilon\right] \\
&< \exp(-\epsilon t \delta^2/4) \\
&= \exp\left(-\epsilon \cdot \frac{8\epsilon n \cdot \ln 2}{\delta^2} \cdot \delta^2/4\right) \\
&= \exp(-2n \ln 2) \\
&= 2^{-2n} \\
\Pr[\exists (x,y) \text{ s.t. prob. of error } > \epsilon + \epsilon'] &\leq \sum_{x,y} \Pr\left[\frac{1}{t}X_i(x,y) > \epsilon + \epsilon'\right] \\
&< \sum_{x,y} 2^{-2n} \\
&= 2^n \cdot 2^n \cdot 2^{-2n} \\
&= 1
\end{aligned}$$

This implies there is a non-zero probability that a random construction has the desired properties, so there must exist at least one satisfying construction. □

Corollary 4.9. $R(f) \leq R^{pub}(f) + O(\log n)$.

4.6 Distributional complexity

We turn our attention to the distributional model, where randomness is introduced in the input.

Definition 4.5. For a probability distribution μ over the inputs, for $\epsilon > 0$, define

the (μ, ϵ)-*distributional communication complexity* as

$$D^{\mu}_{\epsilon}(f) := \min_{P \,:\, \text{deterministic protocol}} \{cost(P) \mid \mu(\{P(x,y) \neq f(x,y)\}) \leq \epsilon\}.$$

The celebrated ***Yao's Minimax Principle*** [40] demonstrates a beautiful relationship between distributional complexity and randomized complexity:

Theorem 4.10. $R^{pub}_{\epsilon}(f) = \max_{\mu} D^{\mu}_{\epsilon}(f)$.

Proof. The $\geq$ *direction*: Let $k := R^{pub}_{\epsilon}(f)$, then there exists a randomized protocol $\mathcal{P}$ with cost k such that for all (x,y), the probabilitiy of error $\leq \epsilon$. Then for all μ, the expected error over all (x,y) is at most $\max_{x,y} err_f(\mathcal{P}, x, y) \leq \epsilon$. That means we can find a deterministic protocol P_i in its collection such that the probability of error over μ is $\leq \epsilon$. This P_i has cost $\leq cost(\mathcal{P}) = k$.

The $\leq$ *direction*: Write $k := \max_{\mu} D^{\mu}_{\epsilon}(f)$. This implies for all μ, there exists deterministic protocol P of cost $\leq k$ such that the probability of error according to μ is at most ϵ. Consider a two-person game where Player 1 is to pick an input pair (x,y) and Player 2 is to pick a deterministic protocol P of cost $\leq k$. Player 1 gets a payoff of 1 from Player 2 if $P(x,y) \neq f(x,y)$, and gets 0 otherwise. Now we can apply von Neumann's Minimax Theorem (Theorem 4.3) to this finite two-person zero-sum game and conclude that there exists a unique value V, and some distributions μ, ν, such that

1. If Player 1 picks his (x,y) according to μ , he can guarantee that the expected payoff is at least V, no matter what P is, and

2. If Player 2 picks his P according to ν , he can guarantee that the expected payoff is at most V, no matter what (x,y) is.

The first point says that there exists a distribution μ such that for all deterministic protocols P of cost $\leq k$, the probability of error is at least V. Hence, by definition of k, we must have $\epsilon \geq V$. Now the second point states that there

exists a distribution ν on the collection of all deterministic protocols of cost $\leq k$, so:

$$k \geq R_V^{pub}(f) \geq R_\epsilon^{pub}(f).$$

□

An important result from this theorem is a tight bound of the randomized communication complexity of the $DISJ$ function defined in Example 2.7. Razborov [35] has explicitly given a distribution μ for which $D_\epsilon^\mu(DISJ) = \Omega(n)$ for sufficiently small ϵ. This implies

$$R(DISJ) = \Theta(n).$$

It is interesting to note that the *discrepancy method* introduced in Chapter 2 is related to $D_\epsilon^\mu(f)$.

Definition 4.6. Let M be a submatrix of M_f. Denote

$$disc^\mu(M, f) := |\mu(M_0) - \mu(M_1)|$$

where M_z is the set of input pairs corresponding to the z-entries in M, $z \in \{0, 1\}$. The *discrepancy of f according to μ* is

$$disc^\mu(f) := \max_{M \,:\, \text{submatrix of } M_f} disc^\mu(M, f).$$

It is clear that the uniform discrepancy $disc(f)$ defined in Definition 2.5 corresponds to $disc^{\text{uniform}}(f)$. The following theorem, also proved in [17], shows that discrepancy provides a way to bound distributional complexity:

Theorem 4.11. *For every distribution μ and $0 < \epsilon < \frac{1}{2}$,*

$$D^\mu_{\frac{1}{2}-\epsilon}(f) \geq \left\lceil \log_2 \left(\frac{2\epsilon}{disc^\mu(f)} \right) \right\rceil.$$

Proof. We recall the basic fact that a deterministic protocol for f partitions M_f into rectangles. Here since we allow errors, the resulting rectangles M_l of a leaf l

may not be monochromatic. We use R_l to denote the set of all inputs that reach l, as in the proof of Theorem 2.1.

Let P be a deterministic protocol with cost $k = D^{\mu}_{\frac{1}{2}-\epsilon}(f)$ and computes f with probability of error (according to μ) at most $\frac{1}{2} - \epsilon$. We have

$$\begin{aligned}
2\epsilon &= \left(\frac{1}{2} + \epsilon\right) - \left(\frac{1}{2} - \epsilon\right) \\
&\leq \mu\{P(x,y) = f(x,y)\} - \mu\{P(x,y) \neq f(x,y)\} \\
&= \sum_{l \,:\, \text{leaf}} (\mu\{P(x,y) = f(x,y), (x,y) \in R_l\} - \mu\{P(x,y) \neq f(x,y), (x,y) \in R_l\}) \\
&\leq \sum_{l \,:\, \text{leaf}} |\mu\{f(x,y) = P(x,y), (x,y) \in R_l\} - \mu\{f(x,y) \neq P(x,y), (x,y) \in R_l\}| \\
&= \sum_{l \,:\, \text{leaf}} disc^{\mu}(M_l, f) \\
&\leq \sum_{l \,:\, \text{leaf}} disc^{\mu}(f) \\
&\leq 2^k \cdot disc^{\mu}(f)
\end{aligned}$$

Hence,

$$D^{\mu}_{\frac{1}{2}-\epsilon}(f) = k \geq \left\lceil \log_2 \left(\frac{2\epsilon}{disc^{\mu}(f)} \right) \right\rceil .$$

□

We end this chapter with a proof of

$$R(IP) = \Theta(n)$$

where the inner-product function IP is defined in Example 2.14.

Example 4.5. [8] We shall show that $disc^{\text{uniform}}(IP) \leq 2^{-n/2}$, and therefore

$$D^{\text{uniform}}_{\frac{1}{2}-\epsilon}(IP) \geq \left\lceil \log_2 \left(\frac{2\epsilon}{2^{-n/2}} \right) \right\rceil = \left\lceil \frac{n}{2} - \log_2 \frac{1}{\epsilon} + 1 \right\rceil .$$

Construct a sign matrix H from M_{IP} through replacing each 0-entry by -1. This H is the famous Hadamard matrix under Sylvester's construction, which is

known to satisfy a number of nice properties. In particular, $HH^T = 2^n \cdot I_{2^n}$, whose only eigenvalue is 2^n, implying that $\|H\|_2 = \sqrt{2^n}$. Now for any submatrix M corresponding to $S \times T, S, T \subseteq \{0,1\}^n$, let $\chi_S, \chi_T \in \{0,1\}^N$ be the characteristic (row) vectors of S and T respectively. Then,

$$\begin{aligned} disc^{\text{uniform}}(M, IP) &= 2^{-2n} \cdot \left| \sum_{x \in S, y \in T} H(x,y) \right| \\ &= 2^{-2n} \cdot |\chi_S \cdot H \cdot \chi_T^T| \\ &\leq 2^{-2n} \cdot \|\chi_S\|_2 \, \|H\|_2 \, \|\chi_T\|_2 \\ &= 2^{-2n} \cdot \sqrt{|S|} \sqrt{2^n} \sqrt{|T|} \\ &\leq 2^{-2n} \cdot \sqrt{2^n} \cdot \sqrt{2^n} \cdot \sqrt{2^n} \\ &= 2^{-n/2} \end{aligned}$$

Hence,

$$disc^{\text{uniform}}(IP) = \max_M disc^{\text{uniform}}(M, IP) \leq 2^{-n/2}.$$

Chapter 5

Communication Complexity Classes

In this chapter we categorize communication problems into different complexity classes. In Section 5.1 we define the classes, and in Section 5.2 we introduce the polynomial-time hierarchy in communication complexity. Section 5.3 gives the notions of reducibility and completeness.

The theorems stated in this chapter, unless otherwise specified, are due to Babai, Frankl and Simon [5].

5.1 Basic classes

We first define classes containing the functions 'efficiently solvable' in different models. We consider complexities of $polylog(n)$ (i.e. $O(\log^c n)$ for some constant c), which is substantially smaller than the linear complexity of the trivial protocol, as efficient.

Definition 5.1.

$$\begin{aligned} P^{cc} &:= \{f : D(f) = polylog(n)\} \\ NP^{cc} &:= \{f : N^1(f) = polylog(n)\} \\ coNP^{cc} &:= \{f : N^0(f) = polylog(n)\} \\ BPP^{cc} &:= \{f : R(f) = polylog(n)\} \end{aligned}$$

We see that Aho-Ullman-Yannakakis Theorem (Theorem 3.4) implies that:

$$P^{cc} = NP^{cc} \cap coNP^{cc}.$$

The three functions EQ, $DISJ$ and GT, with their negations, show that $P^{cc}, NP^{cc}, coNP^{cc}$ and BPP^{cc} are all different.

5.2 Polynomial-time hierarchy

We can also define analogs of the polynomial-time hierarchy. The motivation comes from the definition of $N^1(f)$ (Definition 3.1):

$$\begin{aligned} N^1(f) := \min\{k \mid \exists A, B : \{0,1\}^n \times \{0,1\}^k \to \{0,1\} \text{ s.t.} \\ f(x,y) = 1 \Leftrightarrow \exists z \in \{0,1\}^k \text{ s.t. } A(x,z) = B(y,z) = 1\} \end{aligned}$$

In other words, $f \in NP^{cc}$ if and only if $\exists k = polylog(n)$, $\exists A, B : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}$ such that

$$f(x,y) = 1 \Leftrightarrow \exists z \in \{0,1\}^k \text{ s.t. } A(x,z) = B(y,z) = 1$$

We extend this definition and say that a function f is in Σ_i^{cc} if $\exists k_1, \cdots, k_i = polylog(n)$, $\exists A, B : \{0,1\}^n \times \{0,1\}^{k_1} \times \cdots \times \{0,1\}^{k_i} \to \{0,1\}$ such that

$$\begin{aligned} f(x,y) = 1 \Leftrightarrow \exists z_1 \in \{0,1\}^{k_1}, \forall z_2 \in \{0,1\}^{k_2}, \exists z_3 \in \{0,1\}^{k_3}, \cdots \\ \text{s.t. } A(x, z_1, \cdots, z_i) = B(y, z_1, \cdots, z_i) = 1. \end{aligned}$$

Define Π_i^{cc} as $\text{co}\Sigma_i^{cc}$, i.e. the negations $\bar{f}$ of functions f in Σ_i^{cc}. The Σ_i^{cc}'s and Π_i^{cc}'s form the *polynomial-time hierarchy* in communication complexity, sometimes denoted by PH^{cc}.

Here we give a formal definition using notations of communication matrix based on the above discussion:

Definition 5.2. Let Σ_0^{cc} be the set of functions that are 0 on some rectangle M and 1 everywhere else, and $\Pi_0^{cc} := co\Sigma_0^{cc}$ be the collection of their negations. Now define:

$$\Sigma_i^{cc} := \left\{ f : f = \bigvee_{j=1}^{2^{polylog(n)}} f_j, f_j \in \Pi_{i-1}^{cc} \right\}, \text{ and}$$

$$\Pi_i^{cc} := \left\{ f : f = \bigwedge_{j=1}^{2^{polylog(n)}} f_j, f_j \in \Sigma_{i-1}^{cc} \right\}.$$

The following is immediate from the definition (as expected):

$$\Sigma_1^{cc} = NP^{cc}, \text{ and } \Pi_i^{cc} = co\Sigma_i^{cc} \text{ for any } i.$$

It is interesting to note that the idea of the proof of Gács-Sipser Theorem (Theorem 7.15 in [4]) still works here. Hence we have the following result that relates BPP^{cc} to the polynomial-time hierarchy:

Theorem 5.1. $BPP^{cc} \subseteq \Sigma_2^{cc} \cap \Pi_2^{cc}$.

It is an open question whether $\Sigma_2^{cc} = \Pi_2^{cc}$. It is also not known whether IP is in PH^{cc}. Figure 5.1 shows our current 'world picture'.

5.3 Reducibility and completeness

Now we give the concepts of reducibility and completeness, which are useful in the study of the 'relative hardness' of different functions:

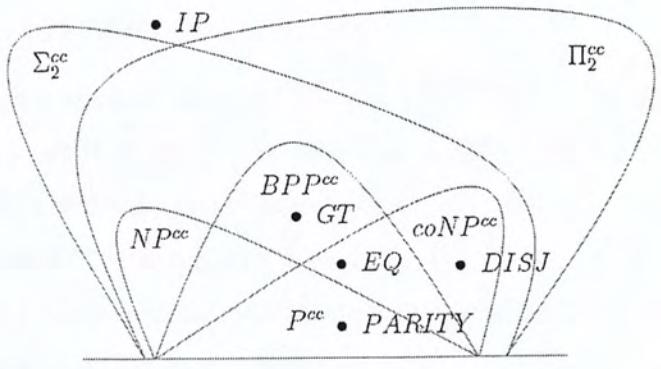

Figure 5.1: The current 'world picture' in communication complexity

Definition 5.3. f is *reducible to* g (denote $f \leq g$) if there exists $m = polylog(n)$ and $h_x, h_y : \{0,1\}^n \to \{0,1\}^{2^m}$ such that

$$f(x,y) = 1 \Leftrightarrow g(h_x(x), h_y(y)) = 1.$$

The following properties are easily verified:

Lemma 5.2. *If $f \leq g$ and $g \leq h$, then $f \leq h$.*

Lemma 5.3. *If $f \leq g$ and $g \in P^{cc}$, then $f \in P^{cc}$ also.*

Proof. The claims follow directly from the fact that

$$m(\log_2 2^{k(\log_2 n)^c})^d = mk^d(\log_2 n)^{cd} = polylog(n).$$

□

Remark 5.1. We may replace P^{cc} with NP^{cc} etc. in Lemma 5.3, and the result still holds.

Definition 5.4. For a class $\mathcal{C}$, g is *$\mathcal{C}$-complete* if $g \in \mathcal{C}$ and

$$f \leq g \text{ for all } f \in \mathcal{C}.$$

Intuitively, $\mathcal{C}$-complete functions are the 'hardest' among all functions in $\mathcal{C}$.

Theorem 5.4. *$DISJ$ is $coNP^{cc}$-complete.*

Proof. We have proved in Example 2.3 that $N^0(DISJ) \leq \lceil \log_2 n \rceil$. It remains to show that every $f \in coNP^{cc}$ is reducible to $DISJ$. Write $m := N^0(f) = polylog(n)$. By definition of co-nondeterministic complexity, we can cover all 0-entries of M_f with 2^m 0-rectangles. Now define $h_x : \{0,1\}^n \to \{0,1\}^{2^m}$ such that the i-th bit of $h_x(x)$ is 1 if and only if row x intersects with the i-th 0-rectangle. Define h_y similarly on columns. Then,

$$\begin{aligned} DISJ(h_x(x), h_y(y)) = 1 &\Leftrightarrow \nexists i \text{ s.t. the } i\text{-th bits of } h_x(x) \text{ and } h_y(y) \text{ are both 1} \\ &\Leftrightarrow \text{no 0-rectangle intersects with both row } x \text{ and column } y \\ &\Leftrightarrow (x,y) \text{ is not contained in any 0-rectangle} \\ &\Leftrightarrow f(x,y) = 1 \end{aligned}$$

□

Lemma 5.5. *$DISJ$ is not reducible to EQ.*

Proof. Suppose otherwise, i.e. we can find h_x, h_y such that

$$DISJ(x,y) = 1 \Leftrightarrow h_x(x) = h_y(y).$$

Note that $DISJ(0^n, y) = 1$ for all y, hence

$$h_x(0^n) = h_y(y) \text{ for all } y.$$

By symmetry, we also have

$$h_x(x) = h_y(0^n) \text{ for all } x.$$

These imply, in particular,

$$h_x(1^n) = h_y(0^n) = h_x(0^n) = h_y(1^n)$$

which is incorrect since $DISJ(1^n, 1^n) \neq 1$. □

We have shown that $DISJ$ is $coNP^{cc}$-complete but EQ is not; in this sense $DISJ$ is 'relatively harder' than EQ.

Chapter 6

Further topics

This concluding chapter lists some topics in communication complexity which the author finds interesting.

6.1 Quantum communication complexity

In this model, the players are allowed to transfer qubits instead of classical bits. The *quantum communication complexity* $Q(f)$ is defined accordingly with error probability just as before.

It has been proved in [16] that $Q(f) \leq R(f)$, and for most functions $Q(f) = \Theta(n)$. (This implies $R(f)$ is also linear for almost all functions.) In particular, $Q(IP) = \Theta(n)$, so quantumness does not help for the inner-product function.

Nevertheless, there are many interesting cases where quantum protocols outperform their randomized counterparts significantly. It has been recently shown in [1] that $Q(DISJ) = \Theta(\sqrt{n})$, which is quadratically better than $R(DISJ) = \Theta(n)$.

The following table shows the asymptotic behavior of the communication complexities of EQ, $DISJ$ and GT under different models:

f	EQ	$DISJ$	GT
$D(f)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
$N^1(f)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
$N^0(f)$	$\Theta(\log n)$	$\Theta(\log n)$	$\Theta(n)$
$R(f)$	$\Theta(\log n)$	$\Theta(n)$	$\Theta(\log n)$
$Q(f)$	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(\log n)$

Table 6.1: Asymptotic behaviour of the complexities of EQ, $DISJ$ and GT

6.2 More techniques for bounds

Various mathematical tools have been used in the study of communication complexity. In [33], Raz has suggested using Fourier analysis to give bounds in randomized communication complexity. Klauck [13] has shown that similar techniques give lower bounds for the quantum model as well. Moreover, Shi and Zhang [36] have proved recently, using Fourier analysis, that $D(f) = \Theta(\log rank(M_f))$ for all *symmetric XOR functions*. These include the *Hamming distance function* $HAM_d(x, y)$ which gives 1 if and only if $|x \oplus y| > d$. This result implies that the class of symmetric XOR functions satisfies the log-rank conjecture.

Another common technique is generalizing matrix rank and norms. To name a few examples, Yannakakis [38] has shown that the logarithm of nonnegative rank is an upper bound on nondeterministic communication complexity, and Krause [15] has proved that approximate rank can be used to lower-bound randomized complexity.

6.3 Complexity of communication complexity

We finish this thesis with a discussion about the following decision problem:

Input: An $N \times N\{0,1\}$-matrix M_f, and a non-negative integer K.
Output: Is $D(f) \leq K$?

It is not difficult to see that this decision problem is in NP: the proof can be a protocol of cost $t \leq \min\{K, n+1\}$, whose representation takes $O(N^2)$ bits. The verifier needs only to check whether it is indeed a valid protocol that returns $f(x,y)$ correctly for every (x,y). This takes at most $O(N^4)$ steps.

It is somewhat contrary to the common impression to note that the problem of whether this *communication complexity computation problem* is NP-hard is actually still open. It has been proved, however, that assuming the intractability of factoring, there is no polynomial-time algorithm to approximate the deterministic communication complexity within a certain factor [18].

On the other hand, we know that computing the *nondeterministic* communication complexity on a given communication matrix is NP-complete. (This is related to the *biclique cover problem* [30]. See Figure 6.1.) It has also been proved that even approximating $N^1(f)$ is hard unless $P = NP$ [24].

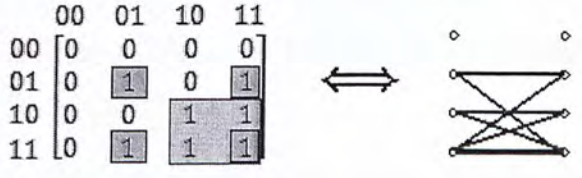

Figure 6.1: Relating nondeterministic complexity to biclique cover

Bibliography

[1] S. Aaronson and A. Ambainis, *Quantum search of spatial regions*, Proc. 44th IEEE Symp. Foundations of Computer Science, pp.200-209, 2003.

[2] A. Aho and J.D. Ullman and M. Yannakakis, *On notions of information transfer in VLSI circuits*, Proc. 15th ACM Symp. Theory of Computing, pp.133-138, 1983.

[3] N. Alon and Y. Matias, and M. Szegedy, *The space complexity of approximating the frequency moments*, Proc. 28nd ACM Symp. Theory of Computing, pp.20-29, 1996.

[4] S. Arora and B. Barak, *Complexity Theory: A Modern Approach*, Cambridge University Press, 2009.

[5] L. Babai and P. Frankl and J. Simon, *Complexity classes in communication complexity theory*, Proc. 27th IEEE Symp. Foundations of Computer Science, pp.337-347, 1985.

[6] P. Beame and T. Huynh and T. Pitassi, *Hardness Amplification in Proof Complexity*, Proc. 42nd ACM Symp. Theory of Computing, pp.87-96, 2010.

[7] G. Brassard, *Quantum communication complexity (a survey)*, quant-ph/0101005, 2001.

[8] B. Chor and O. Goldreich, *Unbiased Bits From Sources of Weak Randomness and Probabilistic Communication Complexity*, SIAM J. Comput., v.17(2), pp.230-261, 1988.

[9] M. Dietzfelbinger and J. Hromkovič and G. Schnitger, *A comparison of two lower bound methods for communication complexity*, Proc. 19th MFCS '94, Lecture Notes in Computer Science 841, Springer-Verlag, pp.423-432, 1994.

[10] T. Hagerup and C. Rüb, *A Guided Tour of Chernoff Bounds*, Infomation Processing Letters, v.33, pp.305-308, 1990.

[11] J. Hromkovič, *Communication Complexity and Parallel Computing*, Springer-Verlag, 1997.

[12] M. Karchmer and A. Wigderson, *Monotone circuits for connectivity require super-logarithmic depth*, Proc. 20th ACM Symp. Theory of Computing, pp.539-550, 1988.

[13] H. Klauck, *Lower bounds for quantum communication complexity*, Proc. 42th IEEE Symp. Foundations of Computer Science, pp.288-291, 2001.

[14] J. Komlós, *On the determinant of (0,1)-matrices*, Studia Sci. Math Hungar, v.2, pp.7-21, 1965.

[15] M. Krause, *Geometric arguments yield better bounds for threshold circuits and distributed computing*, Theoretical Computer Science, v.156, pp.99-117, 1996.

[16] I. Kremer, *Quantum Communication*, Master's Thesis, Computer Science Department, Hebrew University, 1995.

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

[18] E. Kushilevitz and E. Weinreb, *On the complexity of communication complexity*, Proc. 41st ACM Symp. Theory of Computing, pp.465-474, 2009.

[19] T. Lee and A. Shraibman, *Lower bounds on communication complexity*, Foundations and Trends in Theoretical Computer Science, v.3(4), pp.263-399, 2007.

[20] T. Lengauer, *VLSI Theory*, in Handbook of Theoretical Computer Science, Vol.A, Elsevier, pp.835-868, 1990.

[21] R.J. Lipton and R. Sedgewick, *Lower bounds for VLSI*, Proc. 13th ACM Symp. Theory of Computing, pp.300-307, 1981.

[22] L. Lovász, *Communication complexity: a survey*, in Path, Flows and VLSI-Layout, Springer-Verlag, pp.235-265, 1990.

[23] L. Lovász and M. Saks, *Möbius functions and communication complexity*, Proc. 29th IEEE Symp. Foundations of Computer Science, pp.81-90, 1988.

[24] C. Lund and M. Yannakakis, *On the hardness of approximating minimization problems*, Journal of the ACM, v.41(5), pp.960-981, 1994.

[25] A. Matos and A. Teixeira and A. Souto, *Non-deterministic communication complexity and instance complexity*, Proc. 3rd Computability in Europe 2007, pp.274-282, 2007.

[26] A. Matos and A. Teixeira and A. Souto, *On the largest monochromatic combinatorial rectangles with an application to Communication Complexity*, Accepted in Proc. of Computability in Europe 2010.

[27] K. Mehlhorn and E.M. Schmidt, *Las Vegas is better than determinism in VLSI and distributed computing*, Proc. 14th ACM Symp. Theory of Computing, pp.330-337, 1982.

[28] I. Newman, *Private vs. common random bits in communication complexity*, Infomation Processing Letters, v.39, pp.67-71, 1991.

[29] N. Nisan and A. Wigderson, *On Rank vs. Communication Complexity*, Combinatorica, v.15, pp.557-566, 1995.

[30] J. Orlin, *Contentment in graph theory: Covering graphs with cliques*, Indag. Math, v.80(5), pp.406-424, 1977.

[31] G. Owen, *Game Theory*, 2nd edition, Academic Press, 1982.

[32] C.H. Papadimitriou and M. Sipser, *Communication complexity*, Proc. 14th ACM Symp. Theory of Computing, pp.196-200, 1982.

[33] R. Raz, *Fourier analysis for probabilistic communication complexity*, Computational Complexity, v.5, pp.205-221, 1995.

[34] R. Raz and B. Spieker, *On the "log rank"-Conjecture in Communication Complexity*, Proc. 34th IEEE Symp. Foundations of Computer Science, pp.168-176, 1993.

[35] A. Razborov, *On the distributional complexity of disjointness*, Proc. ICALP, pp.249-253, 1990.

[36] Y. Shi and Z. Zhang, *Communication complexities of symmetric XOR functions*, Quantum Inf. Comput., v.9, pp.255-263, 2009.

[37] R. de Wolf, *Quantum communication and complexity*, Theoretical Computer Science, v.287(1), pp.337-353, 2002.

[38] M. Yannakakis, *Expressing combinatorial optimization problems by linear programs*, Journal of Computer and System Sciences, v.43(3), pp.441-466, 1991.

[39] A.C. Yao, *Some complexity questions related to distributive computing*, Proc. 11th ACM Symp. on Theory of Computing, pp.209-213, 1979.

[40] A.C. Yao, *Lower Bounds by Probabilistic Arguments*, Proc. 24th IEEE Symp. Foundations of Computer Science, pp.420-428, 1983.

[41] A.C. Yao, *Quantum Circuit Complexity*, Proc. 34th IEEE Symp. Foundations of Computer Science, pp.352-361, 1993.