

## Lecture 7: Communication Complexity and Lower Bounds

*Lecturer: Madhu Sudan**Scribe: David Chen*

This lecture gives an introduction to Communication Complexity. We go over the properties and examples of Communication Complexity, Karchmer-Wigderson games, a lower bound related to PARITY, and log-rank lower bounds.

## 1 Introduction to Communication Complexity

Communication complexity was introduced by Yao in 1979, and involves the following interaction between Alice and Bob. Alice knows  $x \in \{0,1\}^n$ , and Bob knows  $y \in \{0,1\}^n$ . Alice and Bob are allowed to send each other one bit at a time. The goal is to have both of them compute some  $z \in \{0,1\}^m$  such that  $z = f(x,y)$ . Note that the more general version of this interaction requires them to output  $z$  such that  $(x,y,z) \in R$  for some predetermined relation  $R \subseteq \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^m$ .

Given a relation  $R$ , then, we ask: how many bits are needed for Alice and Bob to output the same  $z$  that satisfies the relation? To address the problem, we **protocols**. A protocol  $\Pi$  specifies the following:

- Given a history of sent bits  $b_1, \dots, b_i$ , whether the interaction should stop (and Alice and Bob computes their output with the bits sent)
- How to calculate  $z$ . The protocol should specify functions  $f^A$  and  $f^B$  such that, upon stopping, Alice and Bob can calculate  $z = f^A(b_1, \dots, b_i, x) = f^B(b_1, \dots, b_i, y)$
- Who sends the next bit
- How to calculate the next bit sent. That is, if Alice is to send bit  $b_{i+1}$ , the protocol should specify a  $g$  such that  $b_{i+1} = g(b_1, \dots, b_i, x)$ .

A protocol  $\Pi$  solves a relation  $R$  if it halts and outputs a  $z$  such that  $(x, y, z) \in R$ .

## 2 Communication Complexity

We define the **Communication complexity** of a protocol to be  $CC(\Pi) =$  the number of bits transmitted with  $\Pi$  in the **worst case**. We consider  $R$  that are well-defined in the sense that for every pair  $(x, y)$ , there exists some  $z$  such that  $(x, y, z) \in R$ . Then, the communication complexity of a relation is

$$CC(R) = \min_{\Pi \text{ solving } R} \{CC(\Pi)\}$$

Using this model, we ignore all computation that Alice and Bob do on their own, and focus on the number of bits sent between the two. Trivially, an upper bound to this is  $2n$ . Alice and Bob can both send each other the entirety of their strings, and each can calculate  $f(x, y)$ . The question is how much better we can do for different relations.

### 3 Karchmer-Wigderson Games

Karchmer and Wigderson came up with the following question: given a  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , what is the circuit depth of  $f$ ? They address this by creating a relation  $R_f$  that is associated with  $f$ . Specifically, we define the relation to be such that  $(x, y, i) \in R$  if either:

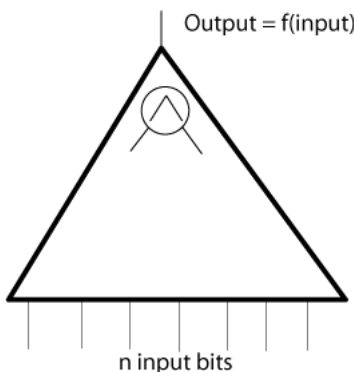
- $f(x) = f(y)$
- $f(x) \neq f(y)$  and  $x_i \neq y_i$

The second point is important because if  $f(x) \neq f(y)$ , then  $x$  and  $y$  must differ at some position  $i$ . This type of game brings us to the following theorem:

**Theorem 1** For every  $f$ ,  $depth(f) = \Theta(CC(R_f))$ .

#### Proof

We will first show that  $CC(R_f) \leq depth(f)$  by creating a protocol that solves  $R_f$ . Consider a circuit  $C$  that computes  $f$ . The circuit can be viewed as a tree of circuit elements, with a top element. Without loss of generality, assume that the last element of the circuit is an AND gate, that  $f(x) = 0$ , and that  $f(y) = 1$ . We will see that these assumptions will not change the structure of the proof.



**Figure 1:** A circuit  $C$  computes  $f$ , and is rooted at a single AND or OR gate

Because  $f(y) = 1$ , both inputs to the top AND must have been 1. .  $f(x) = 0$ , means that at least one of the inputs to the AND was a 0. If it was the left input, Alice will send 0 to Bob; if the right input, she sends a 1. But now, we have recursed to a lower level in the circuit. Because one of the subcircuits of the top AND will yield a different output for Alice and Bob, we have same problem as before, but on a smaller circuit. We recurse in this manner to either  $C_0$  and  $C_1$ , and repeat until we get to the lowest level of the circuit, where the input is simply one of the bits of  $x$  and  $y$ . Because this input is different for Alice and Bob, we have identified the index of difference between  $x$  and  $y$ . We send one message for every level of the circuit that we traverse down, so the number of messages is less than the depth of the circuit. We conclude that  $CC(R_f) \leq depth(f) + O(1)$ .

Note that this argument is also applicable when the top gate is an OR. In that case, because  $f(x) = 0$ , both inputs at the top level must have been 0, and, because  $f(y) = 1$ , at least one of the top inputs for Bob must have been a 1. So in this case, it is Bob who transmits the first bit, and we can once again recurse to a lower level of the tree.

To show the other direction, we make use of partial functions.

**Definition 2** A *partial function* is a mapping  $f : \{0, 1, ?\} \rightarrow \{0, 1, ?\}$  where the ‘?’ symbol represents a “don’t care”

In this direction, we make conversions from protocols to circuits. We create a KM game for relations on partial functions, and show that  $CC(R_f) \geq \text{depth}(f)$  for all partial functions  $f$ .

Say that  $\Pi$  is a protocol that solves  $R_f$ . Consider values of  $x$  and  $y$  such that  $f(x) = 0$ , and  $f(y) = 1$ . Say that under this protocol, Alice goes first, and sends a bit  $b \in \{0, 1\}$ . But then, we have partial functions  $f_0$  and  $f_1$  defined where

$$f_1(x) = \begin{cases} f(x) & \text{if } b(x) = 1, \\ ? & \text{otherwise.} \end{cases}$$

$f_0$  is defined analogously for a first bit of 0. After transmitting the first bit, however,  $\Pi$  will call one of either  $\Pi_0$  or  $\Pi_1$ , protocols that solve  $R_f$  after the first bit is sent and assumed to be 1 if we use  $\Pi_1$ , and 0 in the other case. Now, we are left with a protocol  $\Pi_b$  that is of communication complexity of one bit less than before. Also, the circuits  $C_0$  and  $C_1$  for  $f_0$  and  $f_1$  are of depth one less than before. We claim that  $C$  can be described by  $C_1 \text{OR} C_0$ . This is because when  $f_0 = 1$  or  $f_1 = 1$ ,  $f = 1$ . In addition, when  $f = 0$ , both  $f_1$  and  $f_0$  output 0 as well. Then,  $C = C_1 \text{OR} C_0$ , and  $\text{depth}(f) = \Theta(CC(R_f))$ . ■

## 4 Bounds for PARITY

It is known that PARITY has no  $o(n^2)$  sized formula using  $\{AND, OR, NOT\}$  gates. We will show the power of Communication Complexity by proving the following theorem, which is implied by the quadratic bound on PARITY:

**Theorem 3**  $\text{depth}(PARITY) \geq 2 \log n - O(1)$

**Proof** This is very easily shown with a communication game. Say that Alice has number  $x$  with odd parity, and Bob has  $y$  that has even parity, and that we want to figure out the index  $i$  where  $x \neq y$ . We argue that for any protocol  $\Pi$ , Alice must send at least  $\log n$  bits to Bob.

Let  $x \in \{0, 1\}^n$  be a random string with parity 0, and let  $y = x + e_j$ , where  $e_j$  is a string of all 0’s, except in position  $j$ . Then,  $y$  has parity 1. In order to figure out which bit is the different one, Bob must receive at least  $\log_2 n$  bits from Alice (from the Pigeonhole Principle). Similarly, Alice must receive at least  $\log_2 n$  bits from Bob. We need  $\geq 2 \log n - O(1)$  bits in all to solve this. From the previous theorem, we conclude that the circuit complexity is also  $\geq 2 \log n - O(1)$ . ■

It is interesting to note that this circuit bound depends heavily on the basis of gates used. Such bounds would be different, for example, if we had a basis of  $\{AND, PARITY\}$  gates.

## 5 Lower bounds for Tiling

Consider a function  $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ . It is useful to visualize this function as a  $n \times n$  matrix with entry  $M_{x,y} = f(x, y)$ . Say that Alice and Bob transmit bits to each other, and have a transcript  $\bar{b} = (b_1, b_2, \dots, b_i)$ . Define  $S_{\bar{b}}$  to be the cells of  $M$  that give rise to this type of interaction. That is, define  $S_{\bar{b}} = \{(x, y) \mid \Pi(x, y) \text{ produces interaction } \bar{b}\}$ .

**Claim 4** *The sets  $S_{\bar{b}}$  are generalized “rectangles” in the sense that they can be written as a cartesian product  $S_a \times S_b$ , where  $S_a, S_b \subseteq \{0, 1\}^n$ .*

**Proof** Define  $S_a$  to be the projection of  $S_{\bar{b}}$  onto the first coordinate, and  $S_b$  similarly for the second coordinate. Now, suppose we have  $(x_1, y_1), (x_2, y_2) \in S_{\bar{b}}$ . The points  $x_i$  and  $y_i$  are the inputs to Alice and Bob that would have brought them this sequence of interaction. Consider the point of view of Alice. She receives the same bits of information from Bob, regardless of whether he had  $y_1$  or  $y_2$ ; they are indistinguishable to her. Thus, the points  $(x_1, y_2)$  and  $(x_2, y_1)$  must also be in  $S_{\bar{b}}$ . ■

In these generalized rectangle, all entries of the matrix have the same value. If we communicate  $k$  bits for a protocol, then  $2^k$  rectangles will tile the matrix. Thus, the number of tiles to cover a matrix is  $\leq 2^{CC(f)}$ . This observation can be applied to many applications. If we define  $N_0 =$  number of rectangles to tile  $f^{-1}(0)$ , and  $N_1$  analogously for 1, then  $CC(f) \geq \min(\log N_0, \log N_1)$ . This is helpful when  $f(x, y) = 1$  if  $x = y$ . In this case, the matrix is simply the  $2^n \times 2^n$  identity matrix. We need  $2^n$  tilings to cover this matrix, so we know that  $CC(f) \geq n$  in this case.

The following theorem is attributed to Yao, and relates circuit complexity to the rank of the matrix.

**Theorem 5**  $\log \text{rank}(M_f) \leq CC(f)$ .

**Proof** Consider a matrix with all 0s and a rectangle of 1s. It has rank 1 (because all the rows of 1 are linearly dependent), and is covered by one rectangle. But then, our matrix  $M_f$  can be decomposed into the sum of zero matrices with all the rectangles that tile it. This is written  $M_f = M_{R1} + \dots + M_{Rk}$ . We use the fact that, given matrices  $A$  and  $B$ ,  $\text{rank}(A+B) \leq \text{rank}(A) + \text{rank}(B)$ . But then, we have  $\text{rank}(M_f) \leq k \leq 2^{CC(f)}$  and so  $\log \text{rank}(M_f) \leq CC(f)$  as stated. ■