**6.841 Advanced Complexity Theory** Feb 28, 2007

# Lecture 7

*Lecturer: Madhu Sudan* *Scribe: Benjamin Rossman*

**Today:** Communication Complexity

- Definitions
- Connection to circuit complexity
- Basic results

The setting is the following. There are two players Alice and Bob, separated by physical distance. They want jointly to compute function $f(x, y)$ of inputs $x$ and $y$. Originally, only Alice knows $x$ and only Bob knows $y$. Alice and Bob communicate by sending each other bits, one at a time. Pictorially, we have something like

$$\begin{array}{ccccccc} x & \longrightarrow & \boxed{\text{Alice}} & \xrightarrow{b_1} & \boxed{\text{Bob}} & \longleftarrow & y \\ &&& \xrightarrow{b_2} &&& \\ &&& \xleftarrow{b_3} &&& \\ &&& \xrightarrow{b_3} &&& \\ &&& \vdots &&& \\ &&& \xleftarrow{b_k} &&& \end{array}$$

This goes on until Alice and Bob each have enough information to determine the value $f(x, y)$ (possibly without knowing the other party's complete input). At that point, the interaction ends. We think of such an interaction as a cooperative "game" between Alice and Bob, where the objective is for both parties to ascertain $f(x, y)$ in the fewest possible rounds.

We ask the question: how many bits must Alice and Bob exchange so that each of them can compute $f(x, y)$? For instance, Alice can transmit $x$ to Bob, then Bob can compute $f(x, y)$ and send the result back to Alice. Of course, we'd like to get away with transmitting fewer bits. We don't care how long Alice and Bob spend on their private computations; the only goal is minimizing the length of the interaction, i.e., the number of bits passed between Alice and Bob.

In a slightly more general framework, the combinatorial problem is given by a relation $R(x, y, z)$. Here $x, y$ are the inputs and $z$ is an admissible solution: given $x$ and $y$, we want $z$ such that $(x, y, z) \in R$. Usually, we think of inputs and outputs as sequences of bits, that is, $R \subseteq \{0, 1\}^n \times \{0, 1\}^n \times \{0, 1\}^m$. Also, it is common to consider $m \ll n$ ($m$ much smaller than $n$), since $z$ usually contains far less information than the pair $(x, y)$. (For all inputs $x, y$, we will assume there is at least one $z$ such that $(x, y, z) \in R$.)

Solutions to this combinatorial problem take the form of *procotols.* At each stage in an interaction (game), a protocol $\pi$ specifies:

1. whether the game ends, and
2. if the game continues who transmits the next bit and what that bit is.

At the $k$th stage, Alice's action under a protocol $\pi$ must be purely a function of the input $x$ and the history $b_1, \ldots, b_{k-1}$ of the interaction so far. (By "Alice's action" is meant either: "terminate with some output $z$", "send bit 0/1 to Bob", or "wait for Bob to send the next bit".) Similarly, Bob's action under $\pi$ must be a function of $y$ and $b_1, \ldots, b_{k-1}$. For the protocol $\pi$ to be correct, Alice and Bob should terminate at the same time (with both outputs in the set $\{z : f(x, y, z) \in R\}$), and only one of them should send a bit in each round so long as the game continues. This model was introduced by [Yao ∼1980].

**Definition 1** *The* communication complexity *of a protocol $\pi$ is defined by*

$$CC(\pi) = \max_{x,y}\{\# \text{ bits transmitted before } \pi \text{ terminates on inputs } x, y\}.$$

Communication complexity of relations and functions is defined as follows. For a relation $R$, let

$$CC(R) = \{CC(\pi) \mid \pi \text{ is a protocol computing } R\}.$$

For a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, let $CC(f) = CC(R)$ where $R$ is the relation $\{(x, y, f(x, y))\}$. We understand communication complexity far better for functions than for relations.

We are also interested in communication complexity of partial functions. A partial function is a function with range $\{0, 1, ?\}$ where ? denotes "don't care". This setup corresponds a *promise problems* in which Alice and Bob must jointly compute a function $f(x, y)$ with the promise that the input pair $(x, y)$ comes from a specified set (if the promise is violated, then any output is accepted). Communication complexity of a partial function $f : \{0,1\}^n \times \{0,1\}^n \to \{0, 1, ?\}$ is defined by $CC(f) = CC(R)$ where $R$ is the relation $\{(x, y, 0) : f(x, y) \in \{0, ?\}\} \cup \{(x, y, 1) : f(x, y) \in \{1, ?\}\}$. Communication complexity of partial functions is also very interesting: often lower bounds for promise problems are better than for any completion you can choose. Techniques for promise problems often don't extend to complete functions.

This is the broad setting for communication complexity. Let's look at upper and lower bounds. We know that $CC(R) \leq n + m$ by a trivial protocol: Alice sends $x$, Bob computes $z$ such that $R(x, y, z)$ and sends $z$. We know linear lower bounds for some functions, but few techniques for proving linear lower bounds in general.

## Karchmer-Wigderson games

Karchmer-Wigderson games involve a nonstandard way of associating a relation $R_f$ with a function $f$. Consider a function $f : \{0,1\}^n \to \{0,1\}$. The idea is look at pairs of inputs $x, y$ with the promise that $f(x) = 1$ and $f(y) = 0$. In particular, this implies $x \neq y$, so there exists a bit where $x$ and $y$ differ (i.e. $x_i \neq y_i$ for some $i \in \{1, \ldots, n\}$. This leads us to define a relation $R_f \subseteq \{0,1\}^n \times \{0,1\}^n \times \{1, \ldots, n\}$ by

$$R_f = \{(x,y,i) \mid f(x) = 1,\ f(y) = 0,\ x_i \neq y_i\} \cup \{(x,y,i) \mid f(x) = 0 \text{ or } f(y) = 1\}.$$

Note that $R_f$ contains all triples $(x,y,i)$ where the promise that $f(x) = 1$ and $f(y) = 0$ is violated. The relation $R_f$ correspond to a "game" in which Alice and Bob receive inputs $x$ and $y$ such that $f(x) = 1$ and $f(y) = 0$ and their objective is to find an index $i$ such that $x_i \neq y_i$.

We now prove a remarkable theorem relating a standard complexity measure of a function $f$ with the communication complexity of the relation $R_f$. It is hard to see where the original intuition came from, however the proof is nice and simple once you see it.

**Theorem 2** $CC(R_f) = \text{Depth}(f)$

Here $\text{Depth}(f)$ is the circuit depth of $f$, that is, the minimal depth of a circuit computing $f$ that has only 2-AND, 2-OR and NOT gates.

**Proof** We first show $CC(R_f) \leq \text{Depth}(f)$ (easy direction). This involves converting a circuit $\mathcal{C}$ computing $f$ into a protocols. Wlog, $\text{Depth}(\mathcal{C}) = \text{Depth}(f)$. Suppose the top (output) gate in $\mathcal{C}$ is an OR-gate with subcircuits $\mathcal{C}_0$ and $\mathcal{C}_1$ feeding into it. Let $f_0, f_1$ be the functions computed by $\mathcal{C}_0, \mathcal{C}_1$.

For $i = 0, 1$, we have $\text{Depth}(\mathcal{C}_i) \leq \text{Depth}(\mathcal{C}) - 1$, so that by induction there exists a protocol $\pi_i$ computing $f_i$ with $CC(\pi_i) \leq \text{Depth}(f) - 1$. We now derive a protocol $\pi$ for $f$. We may assume that Alice and Bob have inputs $x, y$ such that $f(x) = 1$ and $f(y) = 0$ (since otherwise any output is okay). Clearly $\max\{f_0(x), f_1(x)\} = 1$ while $f_0(y) = f_1(y) = 0$. In the first round of the protocol, Alice can therefore send Bob a bit $b \in \{0,1\}$ such that $f_b(x) = 1$ and $f_b(y) = 0$. Alice and Bob now proceed according to protocol $\pi_b$.

The induction step is symmetric when the top (output) gate in $\mathcal{C}$ is an AND-gate; this time Bob communicates a bit $b$ such that $f_b(x) = 1$ and $f_b(y) = 0$.

In other direction (just slightly more complicated), we convert protocols to circuits. This argument requires KM games for partial functions (in order to have a stronger induction hypothesis). We will show that $CC(R_f) \geq \text{Depth}(f)$ for every partial function $f : \{0,1\}^n \to \{0,1,?\}$. Recall that $(x,y,i) \in R_f$ iff either $\big[f(x) = 1 \text{ and } f(y) = 0 \text{ and } x_i \neq y_i\big]$ or $f(x) \neq 0$ or $f(y) \neq 1$.

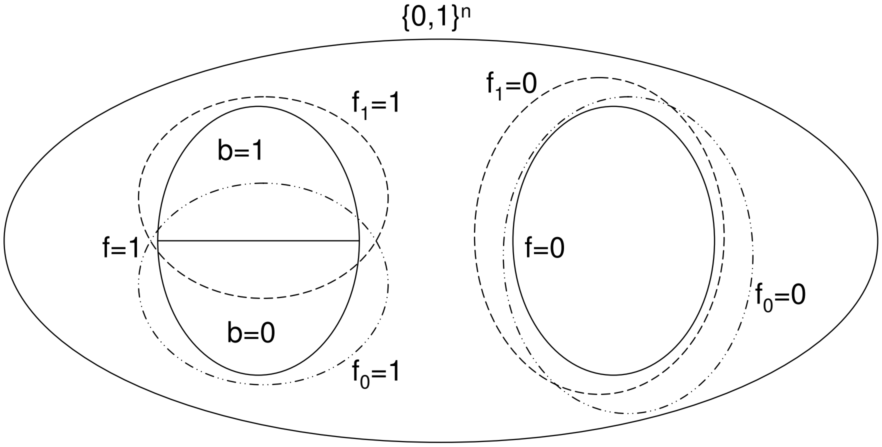
Let $\pi$ be a protocol computing $f$ (i.e. computing $R_f$), and consider inputs $x, y \in \{0,1\}^n$ such that $f(x) = 1$ and $f(y) = 0$. Suppose Alice sends the first bit under $\pi$. That is, she computes $b(x) \in \{0,1\}$ and sends it to Bob. We claim there is a circuit $\mathcal{C}$ computing $f$ of the form $\mathcal{C}_0$ OR $\mathcal{C}_1$ where

$\max\{\text{depth}(\mathcal{C}_0), \text{depth}(\mathcal{C}_1)\} \leq CC(R_f) - 1$. (Had Bob communicated the first bit, then we would have $\mathcal{C}_0$ AND $\mathcal{C}_1$ instead.)

So what are circuits $\mathcal{C}_0$ and $\mathcal{C}_1$? In order to use the induction hypothesis, we define partial functions $f_0, f_1 : \{0,1\}^n \to \{0,1,?\}$ by

$$f_i(x) = \begin{cases} 1 & \text{if } f(x) = 1 \text{ and } b(x) = i, \\ 0 & \text{if } f(x) = 0, \\ ? & \text{otherwise.} \end{cases}$$

For $i = 0, 1$, protocol $\pi$ induces a protocol $\pi_i$ for computing $f_i$: the protocol $\pi_i$ simply dictates what happens after Alice sends the first bit $b(x) = i$. The longest communication under $\pi_0$ or $\pi_1$ is precisely 1 less than the longest communication under $\pi$, that is, $\max\{CC(\pi_0), CC(\pi_1)\} = CC(\pi) - 1$. We now apply the induction hypothesis to obtain circuits $\mathcal{C}_i$ of depth $CC(\pi_i)$ computing $f_i$ for $i = 0, 1$. Let $\mathcal{C}$ be the circuit $\mathcal{C}_0$ OR $\mathcal{C}_1$. We claim that $\mathcal{C}$ computes $f$. This is easily seen from the picture:



The region where either $f_0 = 1$ or $f_1 = 1$ contains the region where $f = 1$, while the region where $f = 0$ is contained by both regions $f_0 = 0$ and $f_1 = 0$.

In the case where Bob sends the first bit, by a similar argument we get a circuit $\mathcal{C}$ with an AND-gate on top. ■

### Application: Circuit Depth of PARITY

Let's see an application of this theorem. It is known that the smallest circuit computing PARITY using only 2-AND, 2-OR and NOT gates has size $\geq n^2$. From this it follows that $\text{depth}(\oplus) \geq 2\log_2(n)$. However, we can prove this directly using Theorem 1. It suffices to show that $CC(R_\oplus) \geq 2\log_2(n)$.

We will argue that, under any protocol computing PARITY, Alice must send Bob at least $\log_2(n)$ bits and vice-versa, so that at least $2\log_2(n)$ bits are transmitted in total. Why must Alice send Bob at least $\log_2(n)$ bits? Let $x$

be uniformly distributed among inputs in $\{0,1\}^n$ such that $\oplus(x) = 1$, and let $y = x + e_i$ where $i$ is uniform in $\{1, \ldots, n\}$ and $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ with 1 in the $i$th place. Clearly $\oplus(y) = 0$. By the pigeonhole principle, Bob must receive $\geq \log n$ bits from Alice in order to determine the unique coordinate $i$ where $x$ and $y$ different, since for $k < \log_2(n)$ there are more possibilities for $i \in \{1, \ldots, n\}$ than possible sequences of bits $b_1, \ldots, b_k$ received from Alice.

This lower bound is almost the best known. Any lower bound greater than $\Omega(\log_2(n))$ for $CC(R_\oplus)$ would be a real breakthrough.

## Communication Complexity of Functions

Communication complexity of functions is better understand than for relations or partial functions. For a function $f : \{0,1\}^n \rightarrow \{0,1\}$, we clearly have $CC(f) \leq n+1$. In fact, $\Pr_f[CC(f) < n] \rightarrow 0$ as $n \rightarrow \infty$, so a random function is very likely to have nearly the maximum possible communication complexity. Explicit example of functions $f$ with $CC(f) \geq n$ are known. We will describe a few.

### Lower Bounds by Tiling

Tiling is a technique for proving lower bounds on the communication complexity of functions. Say we have a function $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$ (it is useful to think of the $2^n \times 2^n$ 0-1 matrix $M_f$) and inputs $x, y \in \{0,1\}^n$. Consider some exchange of bits between Alice and Bob under a fixed protocol $\pi$ computing $f$, say $A \xrightarrow{b_1} B$, $A \xrightarrow{b_2} B$, $B \xrightarrow{b_3} A$, $\ldots$, $B \xrightarrow{b_k} A$ (as in the earlier picture).

Let $\bar{b} = (b_1, b_2, b_3, \ldots, b_k)$ be the transcript of this interaction, and let $S_{\bar{b}} = \{(x,y) : \pi(x,y) \text{ has interaction } \bar{b}\}$. What sets $S_{\bar{b}} \subseteq \{0,1\}^n \times \{0,1\}^n$ can arise in this way?

**Claim 3** *For every $\bar{b}$, the set $S_{\bar{b}}$ is a "rectangle". That is, there exist subsets $S_A \subseteq \{0,1\}^n$ and $S_B \subseteq \{0,1\}^n$ such that $S_{\bar{b}} = S_A \times S_B$.*

**Sketch of Proof** Consider any two pairs $(x_1, y_1), (x_2, y_2) \in S_{\bar{b}}$. Clearly, it suffices to show that $(x_1, y_2) \in S_{\bar{b}}$. To see this, suppose Alice has input $x_1$ and Bob input $y_2$. We ask: assuming $(x_1, y_2) \notin S_{\bar{b}}$, i.e. if $(x_1, y_2)$ leads to some different interaction $\bar{b}'$, then when is the earliest stage where $\bar{b}$ and $\bar{b}'$ can differ? Imagine that the first $k-1$ stages of $\bar{b}$ and $\bar{b}'$ are identical, but the $k$th stage is different. We immediately see that this is impossible! Indeed, up until the $k$th stage, everything Alice sees is consistent with the possibility that Bob has $y_1$ (since $(x_1, y_1) \in S_{\bar{b}}$), and similarly, everything that Bob sees is consistent with the possibility that Alice has $x_2$ (since $(x_2, y_2) \in S_{\bar{b}}$). Because a protocol is, by definition, a deterministic function of Alice and Bob's state of knowledge at each stage, it follows that the $k$th stage of $\bar{b}'$ (i.e. who transmit the $k$th bit and what that bit is) must the same as the $k$th stage of $\bar{b}$. Therefore, $\bar{b}$ and $\bar{b}'$ cannot be different, so we conclude $(x_1, y_2) \in S_{\bar{b}}$. (This type of argument is called a *crossing sequence argument*: we show the interaction histories $\bar{b}$ and $\bar{b}'$ can never cross.) ■

Note that a protocol $\pi$ computing $f$ can terminate after an interaction $\bar{b}$ if, and only if, both and Alice and Bob can be certain that $f$ is *constant* on the sets $S_{\bar{b}}$. So the goal of a protocol is, in some sense, to decompose the matrix $M_f$ into rectangles on which $f$ is constant. This is the idea behind the result (proof omitted).

**Lemma 4 (Tiling Lemma)** *Let $N_0$ (resp. $N_1$)be the minimal number of rectangles needed to partition the set of zero entries (resp. one entries) in $M_f$. Then $\max\{\log N_0, \log N_1\} \leq CC(f)$.*

As an application, consider the equality function $EQ(x, y)$ defined as 1 if $x = y$ and 0 if $x \neq y$. $M_{EQ}$ is just the $2^n \times 2^n$ identity matrix. Clearly $N_1 = 2^n$. So by the Tiling Lemma, we have $CC(EQ) \geq n$.

The next lemma, due to Yao, achieves a similar lower bound by means of rank.

**Lemma 5 (Rank Lower Bound)** $\log \operatorname{rank}(M_f) \leq CC(f)$ *(where rank can be taken over any field).*

**Proof Idea** $M_f$ is unique expressed as a sum of $N_1$ matrices having 1s at entries lying in some "rectangle" in $\{1, \ldots, 2^n\} \times \{1, \ldots, 2^n\}$ and 0s everywhere else, for example

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

(corresponding to the rectangle $\{1, 3, 4\} \times \{1, 4\}$ in the case $n = 2$). Every matrix of this form has rank 1 (over any field). The result follows from the Tiling Lemma and the elementary linear algebra inequality $\operatorname{rank}(A + B) \leq \operatorname{rank}(A) + \operatorname{rank}(B)$. ■

As an application, we get that the inner product function $IP(x, y) = \sum_{i=1}^{n} x_i y_i \mod 2$ has communication complexity $\geq n$. Indeed, this follows from the easily checked fact that $M_{IP}$ has rank $\geq 2^n - 1$. Thus, $CC(IP) \geq \lceil \log 2^n - 1 \rceil = n$ by Lemma 5.