# CS 2401 - Introduction to Complexity Theory

# Lecture #10: Fall, 2015

**Lecturer: Toniann Pitassi**

**Scribe Notes by: Assimakis Kattis**

## 1 Deterministic Communication Complexity

### 1.1 Motivation and Definitions

The notion of communication complexity can be understood in the context of two parties, hereby referred to as Alice and Bob, who are attempting to compute a function $f : X \times Y \to Z$, for $X, Y, Z$ given finite sets. In doing so, the two players only have access to partial inputs $x \in X$ and $y \in Y$ respectively and are computing $f(x, y)$ according to a fixed protocol $P$, given beforehand. The aim of the protocol is to allow the two parties to arrive at the *same* final answer for $f(x, y)$, while communicating the smallest amount of bits about their inputs to each other.

This notion is fundamentally information theoretic in its construction. That is, we are only interested in the amount of communication between Alice and Bob, ignoring the computational requirements that each might face in constructing the messages they send to each other. Thus, we assume that both players have *unlimited* computational power but only a *limited* understanding of the other's input. This leads us to define the cost of a protocol $P$ as the *worst* case cost of $P$ over all inputs $(x, y) \in X \times Y$, where cost is defined as the total number of bits that need to be sent by Alice and Bob while executing $P$. We define the *complexity* of $f$ as the minimum cost of a protocol computing $f$ for all $(x, y)$:

**Definition** For a function $f : X \times Y \to Z$, the (deterministic) communication complexity of $f$, denoted by $c(f)$, is the minimum cost of $P$, over all protocols $P$ that compute $f$. $c(f)$ is a function of $n$, the length of $x$ and $y$.

More specifically, we can formally define a protocol based on who sends the next bit, and what its value - and thus size - is going to be. Without loss of generality, we can assume that the players always alternate, and that the last bit sent is the value $P(x, y)$ outputted by the protocol. This can be assumed since it only increases the length of the protocol by a constant factor, thus not impacting its complexity. It is generally customary to set $X = Y = \{0, 1\}^n$, and $Z = \{0, 1\}$. In this case, the cost of a protocol $P$, $C_P(n)$, is the worst case cost of $P$ over all inputs of length $n$, and the communication complexity of $f$ is the minimum cost $C_P(n)$ over all $P$ computing $f$. An equivalent definition of this notion is also given below:

**Definition** A protocol $P$ over $X \times Y$ with range $Z$ is a binary tree where each internal node $v$ is labelled either by a function $a_v : X \to \{0, 1\}$ or by a function $b_v : Y \to \{0, 1\}$[1], and each leaf is labelled with an element of $Z$. The value of the protocol $P$ on input $(x, y)$ is the label of the leaf

[1] If a node is labelled by $a_v$ intuitively means that Alice is sending a bit at this point, similarly for $b_v$ and Bob.

reached by starting from the root, and traversing the tree. The cost of the protocol $P$ on input $(x, y)$ is the length of the path on input $(x, y)$. The cost of the protocol $P$ is the height of the tree.

### 1.2 Examples

**Example** [Parity] The parity function of $(x, y)$ has value 1 if $x, y$ have the same parity. A simple protocol is the following: Alice sends the parity of $x$ (1 if the number of 1's in $x$ is odd, and 0 otherwise). Then Bob replies 1 if and only if the parity of $y$ is equal to the parity of $x$.

**Example** [Set disjointness] $DISJ(x, y) = 1$ iff there exists $i$ such that $x_i = y_i = 1$.

**Example** [Equality] Equality function: $EQ(x, y) = 1$ iff $x = y$.

**Example** [Inner product] The inner product function is defined as $IP(x, y) = \sum_{i=1}^{n} x_i y_i \pmod 2$.

A trivial upper bound on the communication complexity of any boolean function can be constructed by sending all input bits to the other party:

**Proposition 1** *For any boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, there exists a protocol $P$ solving it with at most $n + 1$ bits of communication.*

**Proof** Alice sends $x$. Bob sends $f(x, y)$. The total communication is $n + 1$ bits.

Therefore, the (deterministic) communication complexity of any boolean function is at most $n+1$. However, for many functions, we can develop much more efficient protocols, i.e., protocols with poly-logarithmic communication bits for specific functions.

### 1.3 Rectangles

In proving lower bounds on the communication complexity of different functions, it is convenient to construct a *combinatorial* view of protocols. Indeed, we can view protocols as a way to partition the space of all possible input pairs, $X \times Y$, into special sets called combinatorial rectangles. Let $P$ be a protocol and $v$ be a node of the protocol tree. We denote by $R_v$ the set of inputs $(x, y)$ that reach node $v$. Let $L$ be the set of leaves of the protocol $P$. It is easy to see that the set $\{R_l\}_{l \in L}$ is a *partition* of $X \times Y$. This motivates the following definition:

**Definition** [Rectangle] A rectangle in $X \times Y$ is a subset $R \subseteq X \times Y$ such that $R = A \times B$ for some $A \subseteq X$ and $B \subseteq Y$.

The connection between rectangles and protocols is implicit in the following proposition.

**Proposition 2** *For all $l \in L$, the set $R_l$ is a rectangle.*

**Proof** By induction on the depth of the protocol tree.

Moreover, by the definition of the protocol in the above rectangles the function $f$ has a fixed value, which we designate as monochromatic by the following definition:

**Definition** [$f$-monochromatic] A subset $R \subseteq X \times Y$ is $f$-monochromatic if $\exists z \in \{0,1\}$ such that for all $(x,y) \in R$, $f(x,y) = z$ on $R$.

The following two statements are immediate from the above definitions:

**Fact** Any protocol $P$ for $f$ induces a partition of $X \times Y$ into $f$-monochromatic rectangles. The number of such rectangles equals the number of leaves of $P$.

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

# 2 Deterministic Lower Bounds

We can use the intuition provided by the combinatorial approach to communication complexity in order to achieve deterministic lower bounds. Indeed, the key insight here is that, if there exists a protocol $P$ with complexity $c(f)$ for a function $f : X \times Y \to \{0,1\}$, then it induces a partition of $R_f = X \times Y$ into at most $2^{c(f)}$ monochromatic rectangles, which is the maximum number of leaves in the corresponding tree. $R_f$ can be thought of as illustrating all possible inputs to $f$ from both $X$ and $Y$. This motivates the approach below.

## 2.1 The Fooling Set Argument

Using the above insight, we can construct a method for proving deterministic CC lower bounds. We preface this with an example. Consider the following $2^n \times 2^n$ matrix $R_{EQ}$ associated with the equality function $EQ(x,y)$, where $x, y \in \{0,1\}^n$.

$$R_{EQ} := \begin{bmatrix} 1 & 0 & & \dots & 0 \\ 0 & 1 & & \ddots & \vdots \\ \vdots & \ddots & & \ddots & 0 \\ 0 & \dots & & 0 & 1 \end{bmatrix}$$

Each "1" has to be in its own 1-monochromatic rectangle. Thus the number of monochromatic rectangles is greater than $2^n$. This observation motivates the following definition of a "fooling set".

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

1. $\forall\ (x,y) \in S$, $f(x,y) = z$;

2. for any two distinct $(x_1,y_1)$, $(x_2,y_2) \in S$, either $f(x_1,y_2) \neq z$ or $f(x_2,y_1) \neq z$.

**Lemma 3** *If $f$ has a fooling set $S$, then $c(f) \geq \log_2 |S|$.*

**Proof** We know that $R_f$ has at most $2^{c(f)}$ monochromatic rectangles. However, by the definition of the fooling set we need a distinct $f$-monochromatic rectangle for each element in $S$, meaning $2^c \geq |S|$. This implies the above result.

This method can be used to show that the equality function actually takes the maximal number of bits to compute.

**Example** [Hardness of equality] For $EQ : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, $c(EQ) \geq n$

**Proof** Define $S = \{(x,x) : x \in \{0,1\}^n\}$. It can be easily verified that this is a fooling set for $EQ$. Since $|S| = 2^n$, this suffices for the proof.

## 2.2 The Rank Lower Bound Method

Given any boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ we can associate a $2^n \times 2^n$ matrix $M_f$, where $M_f(x,y) = f(x,y)$. In words, $M_f$ specifies the values of the function $f$ on any input $(x,y) \in X \times Y$. The rank lower bound method is an algebraic method to give lower bounds on $c(f)$ by computing the rank of $M_f$.

**Definition** For any function $f$, rank$(f)$ is the linear rank of $M_f$ over $\mathbb{R}$.

The following lemma gives a lower bound on the deterministic communication complexity of $f$ through the rank of $M_f$:

**Lemma 4** *For any function $f$, $c(f) \geq \log_2 rank(f)$.*

**Proof** Let $L_1$ be the set of leaves of any protocol tree that gives output 1. For each $l \in L_1$, let $M_l$ be a $2^n \times 2^n$ matrix which is 1 on all $(x,y) \in R_l$ and 0 otherwise. It is clear that

$$M_f = \sum_{l \in L_1} M_l.$$

We know that the rank function is sub-additive, meaning that $\text{rank}(A+B) \leq \text{rank}(A) + \text{rank}(B)$ for any matrices $A, B$. Therefore:

$$\text{rank}(M_f) \leq \sum_{l \in L_1} \text{rank}(M_l).$$

Notice that $\text{rank}(M_l) = 1$ for any $l \in L_1$ since $M_l$ can be expressed as an outer-product of two vectors[2]. Therefore $\text{rank}(M_f) \leq |L_1| \leq |L| \leq 2^{c(f)}$, which implies that:

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

### 2.2.1 The Log-Rank Conjecture

The above fact shows that communication complexity lower bounds can be proven from rank lower bounds. It is a longstanding open question whether or not there is a converse relationship. The best known upper bound is $c(f) \leq rank(f) + 1$, and thus the gap between these bounds is enormous. To this end, the log rank conjecture states that the deterministic communication complexity of any two party function $f$ is equal to the log of the rank of $M_f$, up to polynomial factors. (An early paper by Lovasz and Saks [3] is attributed to the conjecture.)

[2] These vectors are the characteristic vectors for the rectangle that reaches $l$.

**Claim 5 (Log-Rank Conjecture)** *For any function $f$, $c(f) = (\log(rank(f)))^{O(1)}$*

Despite much research, very little is known about this conjecture. Until recently the best upper bound known [2] was:

$$c(f) \leq \log_2(4/3) \cdot rank(f)$$

.

A recent paper [1] established a connection between the log-rank conjecture and a number theoretic conjecture known as the Freiman-Ruzsa conjecture:

**Theorem 6** *Assuming the Freiman-Ruzsa conjecture over $F_2^n$, for any boolean $f$*

$$c(f) \leq O(rank(f)/\log(rank(f))).$$

Very recently, Lovett [4] proved the following unconditional result, based on the discrepancy of low rank matrices.:

**Theorem 7**

$$c(f) \leq O(\sqrt{rank(f)}\log(rank(f))).$$

## 3 Randomized Communication Complexity

In the probabilistic case, players are permitted to toss random bits, represented by a random bit string $r$. There are two models depending on whether the coin tosses are public or private. In the public random string model the players share a common random string, while in the private model each player does not have access to the other's string. This yields the following definitions:

**Definition** Let $P$ be a randomized protocol.

**Zero-sided error:** $P$ computes a function $f$ with zero-sided error if for every $(x, y)$,

$$\Pr[P(x,y) \;=\; f(x,y)] = 1.$$

Notice that in this case, the number of bits communicated is a random variable.

**One-sided error:** $P$ computes a function $f$ (with one sided error $\varepsilon$) if for every $(x, y)$ such that $f(x, y) = 0$,

$$\Pr[P(x,y) \;=\; 0] \;=\; 1,$$

and for every $(x, y)$ such that $f(x, y) = 1$,

$$\Pr[P(x,y) \;=\; 1] \;\geq\; 1-\varepsilon,$$

**Two-sided error:** $P$ computes a function $f$ (with error $\varepsilon$) if

$$\forall\, x \in X,\; y \in Y, \quad \Pr\left[P(x,y) = f(x,y)\right] \geq 1-\varepsilon,$$

Using the above, we are now in a position to define the complexity of a randomized protocol.

**Definition** Let $f : X \times Y \to \{0,1\}$ be a function. We consider the following complexity measures for $f$:

- $R_0(f)$ is the minimum average case cost of a randomized protocol that computes $f$ with zero error.
- For $0 < \varepsilon < 1/2$, $R_\varepsilon(f)$ is the minimum worst case cost of a randomized protocol that computes $f$ with error $\varepsilon$.
- For $0 < \varepsilon < 1/2$, $R^1_\varepsilon(f)$ is the minimum worst case cost of a randomized protocol that computes $f$ with one-sided error $\varepsilon$. We define $R^1(f) = R^1_{1/2}(f)$.

Below, we look at a randomized protocol for the equality function. This will be useful in illustrating the additional power that randomization yields, as the equality function was previously shown to be deterministically hard.

**Example** [Equality Revisited] Recall that $EQ(x,y) = 1$ iff $x = y$. Here we analyse the randomized communication complexity in the public coin protocol for the function $EQ$:

> Let $x \in X$, $y \in Y$, $X = Y = \{0,1\}^n$ be the input strings, and let $r \in \{0,1\}^n$ be the public coin tosses. The protocol is the following: Alice computes the bit $a = (\sum_{i=1}^n x_i r_i) \pmod 2$ and sends it to Bob. Then Bob computes $b = (\sum_{i=1}^n y_i r_i) \pmod 2$. The value of the protocol is
>
> $$P(x,y,r) = 1 \quad \text{iff} \quad \sum_{i=1}^n x_i r_i = \sum_{i=1}^n y_i r_i \pmod 2.$$
>
> This requires communication of only two bits. If $x = y$, then for all $r$, the protocol is correct, i.e., $P(x,y,r) = 1$. If $x \neq y$, then with probability $1/2$ (over the public coin tosses) $P(x,y,r) = 1$, i.e., our protocol is wrong. If we repeat the above random experiment $c$ times independently, then the probability that our protocol is wrong on all of the executions is $1/2^c$. This protocol has $O(1)$ probabilistic communication complexity when the error $\epsilon$ is a constant.

## 3.1 Newman's Theorem

The above example gives a brief illustration of the power that randomization provides. However, what kind of difference can we expect between the public and private coin protocols? To this end, the following theorem due to Newman provides a link between the two, since any public coin protocol can be transformed into a private coin protocol with a small penality in the error and a small additive penalty in the communication complexity. Before we show this, a relevant result from probability is given below:

**Fact** [Chernoff Bound] For a given random variable $X$, we can show that $\forall t > 0$:

$$Pr(X \geq a) \leq \frac{E[e^{tX}]}{e^{t \cdot a}}$$

**Theorem 8** *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a function. For every $\delta > 0$ and every $\epsilon > 0$,*

$$R^{priv}_{\epsilon+\delta}(f) \leq R^{pub}_{\epsilon}(f) + O(\log n + \log \delta^{-1})$$

**Proof** We will prove that any public coin protocol $P$ with error $\epsilon$ can be transformed into another public coin protocol, $P'$ such that:

i The communication complexity of $P'$ is the same as that of $P$.

ii $P'$ uses only $O(\log n + \log \delta^{-1})$ random bits.

iii The error of $P'$ is at most $\epsilon + \delta$.

The theorem follows since $P'$ can be easily converted to a private coin protocol with the desired parameters: first Alice will privately flip that many random coins, then send them to Bob, and then they proceed to follow the protocol $P'$.

Let $Z(x, y, r)$ be a random variable that is equal to 1 if $P(x, y, r)$ outputs an incorrect answer, and 0 otherwise. Since $P$ has error $\epsilon$, $E_r[Z(x, y, r)] \leq \epsilon$ for every $x, y$. Let $r_1, \ldots, r_t$ be random strings, where we will soon set $t = O(n/\delta^2)$, and define $P_{r_1,\ldots,r_t}(x, y)$ as follows: Alice and Bob choose $i \leq t$ at random, and then run $P(x, y, r_i)$. We will prove that there exist strings $r_1, \ldots, r_t$ such that $E_i[Z(x, y, r_i)] \leq \epsilon + \delta$ for all $(x, y)$. For this choice of strings, the protocol $P_{r_1,\ldots,r_t}$ will be our desired protocol, $P'$.

We use the probabilistic method to show the existence of $r_1, \ldots, r_t$. Choose $r_1, \ldots, r_t$ at random, and consider a particular input pair $(x, y)$. The probability that $E_i[Z(x, y, r_i)] > \epsilon+\delta$ is exactly the probability that $1/t \sum_{i=1}^{t} Z(x, y, r_i) > \epsilon+\delta$. Applying the Chernoff bound, since $E_r[Z(x, y, r)] \leq \epsilon$,

$$Pr_{r_1,\ldots,r_t}\left[\left(1/t\sum_{i=1}^{t} Z(x, y, r_i) - \epsilon\right) > \delta\right] \leq 2e^{-2\delta^2 t}.$$

By choosing $t = O(n/\delta^2)$, this is smaller than $2^{-2n}$. Thus for a random choice of $r_1, \ldots, r_t$, the probabilty that there exists a bad $(x, y)$ such that $E_i[Z(x, y, r_i)] > \epsilon+\delta$ is smaller than $2^{2n}2^{-2n} = 1$ (by the union bound). Thus there exists $r_1, \ldots, r_t$ such that for every $(x, y)$ the error of $P_{r_1,\ldots,r_t}$ on $(x, y)$ is at most $\epsilon + \delta$. It is easy to check that the number of random bits used by the protocol $P_{r_1,\ldots,r_t}$ is $\log t = O(\log n+\log \delta^{-1})$, and that the communication complexity of $P_{r_1,\ldots,r_t}$ is the same as that of $P$.

## References

[1] E. Ben-Sasson, S. Lovett, and N. Ron-Zewi, *An additive combinatorics approach relating rank to communication complexity*, Journal of the ACM (JACM), 61 (2014), p. 22.

[2] A. Kotlov, *Rank and chromatic number of a graph*, Journal of Graph Theory, 26 (1997), pp. 1–8.

[3] L. Lovász and M. Saks, *Lattices, mobius functions and communications complexity*, (1988).

[4] S. Lovett, *Communication is bounded by root of rank*, in Proceedings of the 46th Annual ACM Symposium on Theory of Computing, ACM, 2014, pp. 842–846.