CS 2429 - Foundations of Communication Complexity

Lecture #1:

Lecturer: Toniann Pitassi

Scribe Notes by: Tasos Zouzias and Toniann Pitassi

1 The model

Let X, Y, Z be arbitrary finite sets and let $f : X \times Y \to Z$ be an arbitrary function. There are two players, Alice and Bob, who wish to compute the function $f(x, y)$. The main obstacle is that Alice *only* knows x and Bob *only* knows y. Thus, to compute the value $f(x, y)$, they will need to communicate with each other. We are assuming that they both follow a fixed protocol agreed upon beforehand. The protocol consists of the players sending bits to each other until the value of f can be determined.

We are only interested in the amount of communication between Alice and Bob, and we wish to ignore the question of the internal computations of each player. Thus, we assume that Alice and Bob have *unlimited* computational power. The cost of a protocol P is the *worst* case cost of P over all inputs (x, y). The complexity of f is the minimum cost of a protocol that computes f.

Formally how do we specify a protocol? In each step one of the players sends one bit of information to the other player. The bit depends on the input of the player who sends it, and all the previous bits communicated so far.

In every step, a protocol specifies:

1. Which player sends the next bit;
2. Value of this bit (as a function of that players' input, and history so far).

Usually we set $X = Y = \{0, 1\}^n$, and $Z = \{0, 1\}$. Without loss of generality, we can assume that the players always alternate and also the last bit sent is the value of the function $f(x, y)$.

Another view of a protocol which may be more convenient is the following:

Definition 1. *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 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.*

Next, we give some examples of functions that we will study in the up-coming lectures.

[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.

Example 1 (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 bit of b.*

Example 2 (Set disjointness). *$DISJ(x, y) = 1$ iff there exists i such that $x_i = y_i = 1$.*

Example 3 (Equality). *Equality function: $EQ(x, y) = 1$ iff $x = y$.*

Example 4 (Inner product). *The inner product function is defined as $IP(x, y) = \sum_{i=1}^{n} x_i y_i \pmod 2$.*

A simple general protocol : Let any function $f(x, y)$, with $|x| = |y| = n$. 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.

2 Randomized vs. Deterministic CC

In the deterministic model, the protocol specifies for all x, y a value $f(x, y)$. We say that a protocol P computes a function f if $\forall\ x, y,\ f(x, y) = P(x, y)$. Given a protocol P the communication complexity of a function f computed by P on inputs of length n is the maximum number of bits communicated in any run of the protocol, as we range over all inputs of length n.

Definition 2. *For a function $f : X \times Y \to Z$, the (deterministic) communication complexity of f, denoted by $D(f)$, is the minimum cost of P, over all protocols P that compute f.*

In the probabilistic case, players can toss random bits. There are two models depending if 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 has his/her own private random string.

Definition 3. *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 ε) 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 ε) if*

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

Definition 4. *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 ε.*
- *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 ε.*

Now, let's give an example for the above two protocols for the equality function.

Example 5 (Equality Revisited)**.** *Recall that $EQ(x,y) = 1$ iff $x = y$. Let's analyse the randomized communication complexity in the public and private coin protocol for the function EQ:*

Public Coin *Let $x \in X$, $y \in Y$, $X = Y = \{0,1\}^n$ be the input strings, and $r \in \{0,1\}$ 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*

$$P(x,y,r) = 1 \quad \text{iff} \quad \sum_{i=1}^n x_i r_i = \sum_{i=1}^n y_i r_i \pmod 2.$$

Note that the communication is only one bit! Now let's analyse this protocol. If $x = y$,then $\forall r$, 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 $1/2^c$.

Private Coin *In this setting, encode the input sets X, Y as $X = Y = \{1, 2, \ldots, 2^n\}$.*

The protocol is the following:

1. *Alice samples uniformly at random a prime from the set $\{1, 2, \ldots, 2n\}$.*
2. *Then she computes $v = x \pmod p$ and sends to Bob the prime p and the value v.*
3. *Bob sends 1 if and only if $v = y \pmod p$.*

Notice that p, v are integers from $[1, 2n]$, hence the above protocol has complexity $O(\log n)$.

Analysis: If $x = y$, then for every p, $x \pmod p = y \pmod p$, so our protocol is sound.

In the case where $x \neq y$: Let $p_1, p_2, \ldots, p_k$ be the set of "bad" primes with respect to x, y, i.e.,

$$x \pmod{p_i} = y \pmod{p_i}, \quad i = 1, 2, \ldots, k.$$

If $p_1, p_2, \ldots, p_k$ satisfy the above equation, then

$$x = y \pmod P,$$

where $P = p_1 \cdot p_2 \ldots p_k$. Since $p_i \geq 2$, $P \geq 2^k$. However, $P < 2^n$ since $x \neq y$, which implies that $2^k < 2^n$. It follows that $k < n$, which means that our protocol is correct with probability greater than $1/2$.

3 Nondeterministic/co-Nondeterministic CC

In the non-deterministic model of communication complexity, players share *nondeterministic* bits z. Now a protocol is a function of x, y, z, and we say that a protocol P computes a function f if for all x, y:

$$\begin{aligned} f(x,y) = 1 &\implies \exists z\ P(x,y,z) = 1 \\ f(x,y) = 0 &\implies \forall z\ P(x,y,z) = 0. \end{aligned}$$

The communication complexity in this model is defined as the maximum length of z plus the number of bits exchanged over all x, y. Similarly, exchanging the position of the existential and for all quantifier we can define co-nondeterministic CC. The basic example is set disjointness (see Example **??**). If the players share $\log n$ nondeterministic bits, they guess i and check if $x_i = y_i = 1$.

4 Communication Complexity Classes

We can define the following communication complexity classes:

$$P^{CC},\ RP^{CC}, BPP^{CC}, NP^{CC},\ coNP^{CC}$$

as the set of functions $f(x, y)$ that can be solved with poly-logarithmic communication complexity.

Remark: Unlike the computational complexity classes where NO non-trivial relationship is known, here we know almost everything, i.e.,

$$P^{CC} \subsetneq RP^{CC} \subsetneq BPP^{CC} \subsetneq NP^{CC} \subsetneq coNP^{CC}.$$

5 Applications

Communication complexity arguments have found numerous application to a large number of different areas. Applications include the following.

1. Bisection width of networks
2. VLSI
3. Decision tree lower bounds
4. Data structures – cell probe model and dynamic data structures
5. Boolean circuit complexity. This includes Depth 2 threshold circuits, and the circuit class ACC.
6. Turing machine time-space trade-offs
7. Streaming algorithms
8. Game theory (truthfulness vs. accuracy)
9. Differential privacy
10. Proof complexity

6 Course Outline

The main topics that we will discuss in this course can be summarized as follows:

- 2 player protocol, upper bounds, lower bounds methods (Fooling set method, rank method, discrepancy method);
- Public/private coin protocols;
- The number-on-forehead model (NOF). Upper bounds and Lower bounds.
- Applications.

7 Basics

The success in proving good lower bounds on the communication complexity comes from the *combinatorial* view of protocols. The idea is to view protocols as a way to partition the space of all possible input pairs, $X \times Y$, into sets. Let P be a protocol and v be a node of the protocol tree. We denote by R_v is 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 discussion leads to the following fundamental element in the combinatorics of protocols.

Definition 5 (Rectangle)**.** *A rectangle in $X \times Y$ is a subset $R \subseteq X \times Y$ such that $R = A \times B$ for some $A \subseteq B$ and $B \subseteq Y$.*

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

Proposition 1. *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, i.e., monochromatic.

Definition 6 (f-monochromatic)**.** *A subset $R \subseteq X \times Y$ is f-monochromatic if f is fixed*[2] *on R.*

The following two statement are immediate from the above definitions.

Fact 2. *Any protocol P for f induces a partition of $X \times Y$ into f-monochromatic rectangles. The number of (f-monochromatic) rectangles equals the number of leaves of P.*

Fact 3. *If any partition of $X \times Y$ into f-monochromatic rectangles requires at least t rectangles, then $D(f) \geq \log_2 t$.*

[2]There exists $z \in \{0, 1\}$ such that for all $(x, y) \in R$, $f(x, y) = z$.

7.1 Fooling Set

Consider the following $2^n \times 2^n$ matrix associated with equality function $EQ(x, y)$, $|x| = |y| = n$.

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

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 7. *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*

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

(ii) 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 4. *If f has a fooling set S of size t, then $D(f) \geq \log_2 t$.*

7.2 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 $D(f)$ by computing the rank of M_f.

Definition 8. *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 5. *Let a function f . Then $D(f) \geq \log_2$ 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

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

Fact : The rank function is a sub-additive function, i.e., $\text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B)$ for any matrix 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[3]. Therefore $\text{rank}(M_f) \leq |L_1| \leq |L|$, which implies that

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

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

□

7.3 Covers

Basic definition:

Definition 9. *Let $f : X \times Y \to \{0,1\}$ be a function:*

1. $C^P(f)$ = *minimum number of leaves in a* protocol tree *for f.*
2. $C^D(f)$ = *minimum number of monochromatic rectangles in a (rectangular disjoint)* partition *of $X \times Y$.*
3. $C(f)$ = *minimum number of monochromatic rectangles that* covers $X \times Y$.
4. $C^z(f)$ = *minimum number of monochromatic rectangles needed to cover the z-inputs*[4] *of f.*

Proposition 6. *For all $f : X \times Y \to \{0,1\}$:*

- $C(f) \leq C^D(f) \leq C^P(f) \leq 2^{D(f)}$.
- $C(f) = C^0(f) + C^1(f)$.

Lemma 7 (Balancing Protocols)**.** *Let f a function. Then*

$$\log C^P(f) \leq D(f) \leq 2 \log_{3/2} C^P(f)$$

Proof. The lower bound on $D(f)$ is immediate. For the upper bound, it suffices to show that given any deterministic protocol P that computes f with s leaves, we are able to create a new protocol P' for f with deterministic communication complexity $O(\log s)$.

By hypothesis, we know that the protocol tree T has s leaves. The proof relies heavily on the following claim.

Claim 8. *For any tree T with s leaves, $|T| = s$, there exists a node v of T, such that for the sub-tree T_v rooted at v the following inequality holds,*

$$\frac{s}{3} \leq |T_v| \leq \frac{2s}{3}.$$

The input of the new protocol P' is the protocol tree T, the input pair (x, y), and the number of leaves s.

1. Alice and Bob determine a node v such that $\frac{s}{2} \leq |T_v| \leq \frac{2s}{3}$.
2. Both decide if $(x, y) \in R_v$[5], by sending one bit each of them, in total 2 bits.
3. If yes, recurse on the rectangle R_v.

[4] The z-inputs of a function f is the set $\{(x, y) \mid f(x, y) = z\}$.

[5] R_v is the rectangle that corresponds to the sub-tree T_v

4. If no, recurse on the tree T_{new}, where T_{new} is the same tree as T, except that the sub-tree T_v is replaced by a single node/leave with value 0.

Let's analyse the above protocol. Let $Q(s)$ be the number of bits that are communicated by the above protocol when the input tree has s leaves. It is easy to see that the following recursion on $Q(s)$ holds,

$$Q(s) \leq 2 + Q(\frac{2s}{3}),$$

where 2 is the bits that are communicated at the current step and $Q(2s/3)$ the number of bits that will be communicated in the next (recursive) step in worst case. Also note that $Q(1) = 0$. Applying the above inequality repeatedly we get

$$\begin{aligned} Q(s) &\leq \underbrace{2 + 2 + \cdots + 2}_{i} + Q(\frac{2^i s}{3^i}), \quad \text{by setting } i = \log_{3/2} s, \\ &= 2 \log_{3/2} s. \end{aligned}$$

Setting $s = C^P(f)$, i.e., the minimum number of leaves for a protocol that computes f, and notice that $D(f) = Q(C^P(f))$, gives the lemma. □