Communication Complexity	August 30, 2011
Lecture 3	
Lecturer: Prahladh Harsha	*Scribe: Abhishek Bhrushundi*

Recap

Last time we saw how the depth of monotone formulas for a function can be lower bounded via communication complexity bounds. Specifically, we were able to show that $Depth^+(Match) = \Omega(n)$, where $Match : \{0,1\}^{\binom{n}{2}} \to \{0,1\}$ is a function that computes whether a given graph (represented by the characteristic vector of the edge set) has a perfect matching, and $Depth^+(f)$ denotes the depth of a monotone formula computing f. We shall see more such bounds later on in the course. For example, we shall see

Theorem 1 *Let* $STCONN(G,s,t) = \begin{cases} 1 & \textit{if there is a path from } s \textit{ to } t \textit{ in } G \\ 0 & \textit{otherwise} \end{cases}$.
Then, $Depth^+(STCONN) = \Omega(\log^2(n))$

It is known that $Depth(STCONN) = O(\log^2 n)$.

Nondeterministic communication complexity

In this model, Alice and Bob are allowed to make private guesses, and at any point of time, their message strings, apart from being dependant on previously sent messages, can also be dependant on the guesses. Note that, since we allow Alice and Bob to be computationally unbounded, their respective private guesses can be arbitrarily long. For a given function $f : X \times Y \to \{0,1\}$, we say P is a nondeterministic protocol if

1. For all (x,y) such that $f(x,y) = 1$, there is a guess (or rather a pair of private guesses for Alice and Bob) such that the protocol outputs 1.
2. For all (x,y) such that $f(x,y) = 0$, no matter what guesses Alice and Bob make, the protocol always outputs 0.

Example : Let us look at the function $\overline{EQ}_n = \begin{cases} 1 \text{ if } x \neq y \\ 0 \text{ otherwise} \end{cases}$. A straight-forward nondeterministic protocol for the function is the following:

1. Alice guesses a number between 1 and n (let's call it i), and sends it over to Bob.
2. Bob sends over the i^{th} bit of y to Alice.
3. Alice outputs 1, if $x_i \neq y_i$, otherwise she outputs 0.

Clearly, if $f(x,y) = 1$, then there is some position (and hence some guess that Alice can make) at which $x_i \neq y_i$, and the protocol outputs 1, otherwise no matter what position Alice choses, $x_i = y_i$, and the protocol outputs 0.
This yields $N(\overline{EQ}_n) \leq O(\log(n))$. Note that $N(f)$ denotes the nondeterministic communication complexity of f.

Nondeterminism and Covers

Recall that in case of a deterministic communication complexity protocol P, the protocol partitions M_f (the truth table) into monochromatic rectangles, and every monochromatic rectangle corresponds to a unique transcript or a unique leaf in the protocol tree. It is natural to ask whether such a notion of rectangles exists in the case of nondeterministic protocols.

Defining rectangles in the nondeterministic case : Let us try and define a suitable notion of rectangles in the case of nondeterministic protocols. Consider a protocol P and let (x, y) and (x', y') be two input pairs with $f(x, y) = f(x', y') = 1$ such that there exist nondeterministic guesses for Alice and Bob in both cases (say (z_1, z_1') for input (x, y), and (z_2, z_2') for input (x', y'), such that $P(x, y, z_1, z_1') = P(x', y', z_2, z_2') = 1$ and $\pi(x, y, z_1, z_1') = \pi(x', y', z_2, z_2')$. Here $P(x, y, z, z')$ denotes the output of the protocol on input (x, y) when Alice guesses z, and Bob guesses z'. In a similar spirit, $\pi(x, y, z, z')$ denotes the transcripts for these guesses. Clearly, when the inputs are (x, y') and (x', y), and the guesses are (z_1, z_2') and (z_2, z_1') respectively, we get the same transcipt and output (in this case 1). Thus, even (x, y') and (x', y) are points where f evaluates to 1, and we say that $(x, y), (x', y'), (x', y)$ and (x, y') are all on a 1-rectangle.

Definition 2 (1-rectangle) *A 1-rectangle (say R) for a nondeterministic protocol P computing f consists of all inputs (x, y) such that:*

1. $f(x, y) = 1$
2. *There is a transcript T such that $\forall\ (x, y) \in R$, there exist nondeterministic guesses z, z' such that $\pi(x, y, z, z') = T$, and $P(x, y, z, z') = 1$.*

Notice that 1-rectangles in the nondeterministic case could be possibly overlapping. Together, all the 1-rectangles of a valid nondeterministic protocol for f must *cover* the 1's in M_f, leading to what we call a 1-cover.

Can we define 0-rectangles in a similar manner? : Unfortunately, the above argument fails when we try to look at points (x, y) and (x', y') where f evaluates to 0. The same argument as above goes through till the point we prove that (x, y') and (x', y), given the guesses (z_1, z_2') and (z_1', z_2) respectively, have the same transcript and output (0 in this case). Note, however, this does not say anything above the value of f at these points: there could be some other nondeterministic guess for which the protocol, on input (x, y'), could output 1 (Similar things might happpen on input (x', y)), which would imply that $f(x, y')$ (and possiblly $f(x', y)$) is 1. Thus, the notion of 0-rectangles, defined in the above spirit, does not make much sense. Note that points on which f is zero are never contained in 1-rectangles.

Covers and nondeterministic communication complexity : Let $C^1(f)$ denote the size of the smallest cover needed to cover all the 1s in M_f, where a cover may consist of possibly overlapping rectangles. We claim the following:

Lemma 3 $N(f) = \log(C^1(f))$

Proof We first claim that $N(f) \leq \log(C^1(f))$ i.e. if Alice and Bob have a 1-cover consisting of $C^1(f)$ rectangles, then by exchanging $\log(C^1(f))$ bits, they can compute $f(x, y)$. Following is the protocol:

1. A priori, both Alice and Bob have a 1-cover with all the rectangles numbered.
2. Alice nondeterministically chooses a rectangle from the cover which intersects the row corresponding to input x. Let's say the index of this rectangle is k. If there is no such rectangle, Alice outputs 0.

3. Alice sends k over to Bob. This takes $\log(C^1(f))$ bits.

4. Bob checks if his input lies in the rectangle, and outputs 1 if it does, and 0 otherwise. (We'll ignore this step when counting the total number of bits communicated. In case we do include this bit, we should claim $N(f) \in \Theta(\log(C^1(f)))$

Clearly, when $f(x, y) = 1$, there is some rectangle in the 1-cover, the index of which Alice can guess, leading to a correct computation. On the other hand, if $f(x, y) = 0$, no rectangle will ever contain (x, y).

We now show $N(f) \geq \log(C^1(f))$, or equivalently, given a protocol P which uses $N(f)$ bits of communication, we can construct of a 1-cover of size at most $2^{N(f)}$. As argued above, two inputs for which f evaluates to 1 lie in the same 1-rectangle if for each of the inputs there is some guess that leads to the same transcript and output equal to 1. Clearly, in case of a valid protocol, all points with $f(x, y) = 1$ are covered by some 1-rectangle. Given the fact that there are at most $2^{N(f)}$ transcripts, we can always construct a 1-cover induced by P is of size at most $2^{N(f)}$. Let's see why this is true. Suppose there were $2^{N(f)} + 1$ 1-rectangles in a 1-cover. By the piegon-hole principle, there are two distinct 1-rectangles, say R_1 and R_2, which have the same transcript T (a transcript which has the final output as 1). Now, for all points $(x, y) \in R_1 \cup R_2$, there is some guess for which the protocol follows transcript T. But this is exactly the definition of a 1-rectangle, and it follows that there is a 1-rectangle which contains $R_1 \cup R_2$. Thus, we can replace R_1 and R_2 by this rectangle, reducing the number of rectangles, yet maintaining the covering property. ■

In light of the above discussion, it is easy to see that the fooling set argument can be used to give a lower bound for EQ_n even in the nondeterministic case. For $f = EQ_n$, $D(f) = \Omega(n)$, $N(f) = \Omega(n)$, and $N(\bar{f}) = \Omega(\log(n))$. A natural question to ask is how does deterministic communication complexity of a function f connect to its nondeterministic counterparts? Can it happen that both $N(f)$ and $N(\bar{f})$ are small, but $D(f)$ is large? We shall now addresses this question.

Relating $D(f), N(f)$ and $N(\bar{f})$

We prove the following theorem:

Theorem 4 *Let $f : S \times T \to \{0, 1\}$ be a function, then $D(f) \leq O(N(f)N(\bar{f}))$.*

We present both, an algorithmic proof, and a combinatorial proof for the above.

Algorithmic proof for theorem 4 : Suppose Alice and Bob have the 1-cover corresponding to a nondeterministic protocol for f, and a 0-cover corresponding to a protocol for $\bar{f}$, and they want to compute $f(x, y)$. As in the proof of *Lemma* 3, they want to determine which rectangle their input lies in (a 0-rectangle or a 1-rectangle?) to find out $f(x, y)$. The idea is that, in each round, Alice and Bob communicate $O(\log(C^1(f)))$ bits and kill the number of potential 0-rectangles in which their input might lie by half. This easily gives that the number bits of communication required is at most $O(\log(C^1(f)) \log(C^0(f))) = O(N(f)N(\bar{f}))$.
Following is the protocol:

1. We begin by assuming that all 0-rectangles are "alive".

2. Alice considers all the 0-rectangles are alive (this is done by updating the list suitably, if she has recieved a new rectangle name from Bob). If there is no 0-rectangle alive that intersects row x, Alice outputs 1.

3. In case, there is at least one 0-rectangle alive intersecting row x, Alice tries to find a 1-rectangle that intersects row x and is disjoint from at least half of the alive 0-rectangles on the rows. If such a 1-rectangle is found, Alice sends its number to Bob, otherwise Alice sends $FAIL$ to Bob. The control now passes to Bob.

4. Let us look at the case when Bob actually recieves an index of a rectangle from Alice (call the rectangle R). In this case, Bob eliminates all the 0-rectangles that are disjoint from R. If there is no 0-rectangle alive intersecting the column y, Bob outputs 1. Otherwise Bob finds a 1-rectangle that intersects column y, and is disjoint from more than half of the alive 0-rectangles on the columns. If such a rectangle is found, Bob sends its index to Alice, otherwise Bob send $FAIL$ to Alice, and passes control to her.

Why does the above protocol terminate? Even if it does, why should it output the correct value? A few observations are in order to shed some light on the correctness of the protocol:

1. If $f(x, y) = 0$, the 0-rectangle containing (x, y) is alive throughout the protocol. This is because no matter what 1-rectangle intersecting x or y you pick at any stage, it will never be disjoint from this 0-rectangle.

2. If $f(x, y) = 1$, then the 1-rectangle containing (x, y) is disjoint with at least half of the 0-rectangles on either the rows or the columns. This follows from the fact that a 1-rectangle is completely disjoint from *all* 0-rectangles.

Observation 2 implies that there can never be two consecutive $FAIL$ communications. Thus, in the worst case, every second communication is an index of some rectangle, which leads to the number of 0-rectangles being slashed by half, thus bounding the number of rounds of communication by $O(\log(C^0(f)))$. Since the amount of communication sent in every round is at most $O(\log(C^1(f)))$, we have that $D(f) \leq O(\log(C^0(f)) \log(C^1(f)))$.

Combinatorial proof for theorem 4 : Let $C^P(g)$ denote the least number of leaves in a protocol (deterministic) tree for a boolean function g or, equivalently, the number of rectangles in the smallest partition for M_g given by any protocol for g. Let $L(k, l)$ denote the maximum of $C^P(g)$ over all boolean functions g such that $C^0(g) \leq l$, $C^1(g) \leq k$.
Consider an optimal cover for f. Let R be a specific 0-rectangle in the cover. We assume that greater than $\frac{k}{2}$ 1-rectangles share no row with $R = S \times T$ (if this is not the case, greater than $\frac{k}{2}$ 1-rectangles share no column with R, and the proof that follows can be suitably modified). Now following is a protocol for f:

1. Alice checks if $x \in S$. If this is the case, the search space now becomes $S \times Y$ (assuming $f : X \times Y \to \{0, 1\}$). This would mean that the number of 1-rectangles has reduced by half.

2. Otherwise, $x \notin S$, in which case the search space becomes $\bar{S} \times Y$, and the number of 0-rectangles goes down by 1.

The two disjoint cases give us $L(k, l) \leq L(\frac{k}{2}, l) + L(k, l-1)$. It is not hard to see that the recurrence solves to $L(k, l) = \binom{l+\log(k)}{\log(k)} \leq (l + 1)^{\log k}$. Thus, we can conclude that the number of leaves in a protocol tree for f is at most $(l + 1)^{\log k} \approx exp(\log l, \log k)$. The only issue at hand is that the protocol tree might not be a balanced one, making it hard for us to relate the number of leaves to the depth. It is easy to show that any protocol can be converted into a "balanced" protocol (consult assignment 1), for which the depth of the protocol tree is logarithmic in the number of leaves. This tell us that $D(f) \leq O(\log l \log k) = O(N(f)N(\bar{f}))$.

The function $DISJ_{k,n}$

Are there any tight examples for the inequality proved in the previous section? We now demonstrate a function for which $D(f) \geq \Omega(N(f)N(\bar{f}))$.
Let us look at the following situation: Alice has a set $X \subseteq [n]$, and Bob has a set $Y \subseteq [n]$, such that $|X| = |Y| = k << n$. As in the disjointness function, we assume that Alice and Bob have have the characteristic vectors for their respective sets. We then define:

$$DISJ_{k,n} = \begin{cases} 1, \; if \; X \cap Y = \phi \\ 0 \text{ otherwise} \end{cases}$$

We will show that

1. $D(DISJ_{k,n}) = \Omega(k \log n)$
2. $N(DISJ_{k,n}) = O(k + \log\log n)$
3. $\forall \; k, \; N(\overline{DISJ_{k,n}}) = O(\log n)$

3 has an easy protocol: Alice guesses an element in $[n]$ that is present in both X, Y, and sends its index over to Bob, who checks if the element is in his set. If the sets are not disjoint, there is some guess which will work, otherwise there is no such guess.
Plugging $k = \log n$ in the above, we get $D(f) \geq \Omega(N(f)N(\bar{f}))$.

Showing $N(DISJ_{k,n}) = O(k + \log\log n)$: A trivial protol gives $N(DISJ_{k,n}) \leq O(\log \binom{n}{k}) \approx O(k \log n)$. We use hash functions to reduce the nondeterministic communication complexity. Suppose there is a small family of hash functions $H = \{h : [n] \to \{0,1\}\}$ such that for all disjoint subsets X, Y of $[n]$ such that $|X| = |Y| = k$, there exists a hash function $h \in H$ such that $h(X) = \{0\}$ and $h(Y) = \{1\}$ (Note that X and Y denote the sets of Alice and Bob respectively).
Let us say the size of such a family H is t. Then, following is a nondeterministic protocol for $DISJ_{k,n}$:

1. Alice and Bob both know the family H, and index the functions in H.
2. Among all the functions $h \in H$ such that $h(X) = \{0\}$, Alice nondeterministically chooses one (say h', and sends its index to Bob. If there is no such function, Alice outputs 0.
3. Bob computes $h'(Y)$, and outputs 1 if $h'(Y) = \{1\}$, and 0 otherwise.

Clearly, if X and Y are disjoint, there is guess such that $h'(X) = 0$ and $h'(Y) = 1$, and the communication complexity is $\log t$.
We will use probabilistic method to show the existence of a family of size $t = \lceil 2^{2k} 2k \log n \rceil$, which completes the proof.
Let us pick $H = \{h_1, h_2, \ldots h_t\}$ at random, and fix two disjoint sets X and Y of size k. Now,

$$Pr_h[\text{h separates } X \text{ and } Y] = \frac{1}{2^k}\frac{1}{2^k} = \frac{1}{2^{2k}}$$

$$Pr_h[\text{h does separates } X \text{ and } Y] = 1 - \frac{1}{2^{2k}}$$

$$Pr_{h_1,h_2,\ldots,h_t}[\forall i, \; h_i \text{ does separates } X \text{ and } Y] = (1 - \frac{1}{2^{2k}})^t$$

$$Pr_{h_1,h_2,\ldots,h_t}[\exists \text{ disjoint } X, Y, \; \forall h \in H, \; h \text{ does separates } X \text{ and } Y] = \binom{n}{k}^2 (1 - \frac{1}{2^{2k}})^t$$

$$\leq \quad n^{2k} exp(\frac{-t}{2^{2k}})$$

It is not hard to see that choosing $t = \lceil 2^{2k} 2k \log n \rceil$ makes the above probability strictly less than 1. This gives $N(DISJ_{k,n}) \leq O(k + \log\log n)$.

It only remains to show that $D(DISJ_{k,n}) \geq \Omega(\log \binom{n}{k})$. For this we appeal to the technique of rank lowers bounds.

Rank Lower Bounds

Recall, that for matrices A and B over some field $\mathbb{F}$, we have that $rank(A+B) \leq rank(A)+rank(B)$. Let us look at the matrix M_f as a matrix with entries coming from $\mathbb{R}$. Consider a partitioning of M_f into monochromatic rectangles, and let $R_1, R_2, \ldots R_m$ be the 1-rectangles in the partition. Also, let us define M_i, for $1 \leq i \leq m$, to be the matrix which is of the same dimension as M_f, and is zero everywhere except in R_i. It is clear that $M_f = \sum_i M_i$. This tells us that $rank(M_f) \leq \sum_i rank(M_i)$. Clearly, the rank of all M_is is 1, and m is essentially the number of 1-rectangles in the partition. This gives us

Observation 5 $rank(M_f) \leq$ *# 1-rectangles in any partition of* M_f

We will now try to relate the 0-rectangles in a partition to $rank(M_f)$. We can do this by replacing M_f in the above argument with $M_{\bar{f}}$. This will give $rank(M_{\bar{f}}) \leq$ # 0-rectangles in any partition of M_f. Notice that $M_{\bar{f}} = J - M_f$, where J is the all 1s matrix. This gives $rank(M_f) \leq 1 + rank(M_{\bar{f}})$, which in turn implies

Observation 6 $rank(M_f) - 1 \leq$ *# 0-rectangles in any partition of* M_f

Combining the two observations, and using the fact that $D(f) \geq$ minimum # monochromatic rectangles required to tile M_f, we get the important lemma

Lemma 7 $D(f) \geq \log_2(2 \times rank(M_f) - 1)$

The following claim is left as an exercise:

Claim 8 $r(M_{DISJ_n}) = 2^n$, $r(M_{DISJ_{k,n}}) = \binom{n}{k}$

This immediately gives us what we were trying to prove in the last section: $D(DISJ_{k,n}) = \Omega(k \log n)$. Again, one could ask if the above inequality is tight up a constant? It turns out this is not the case. However, it is conjecture that there is a c such that $D(f) \leq (\log(rank(M_f)))^c$, for all functions f. This is known as the **polylog rank conjecture**.