Mathematical Methods in Computer Science
Lecture 6: Communication Complexity

Lecturer: Boaz Barak
Scribes: Aditya Bhaskara and Siddhartha Sen

1 Introduction

Communication complexity was introduced by Yao in 1979. In this model, there are two parties, Alice and Bob, who have access to strings x and y, respectively, where $x, y \in \{0,1\}^n$. The aim is for them to compute a function $f : \{0,1\}^n \to \{0,1\}$ with as little communication between them as possible. The minimum k for which there exists a protocol that uses at most k bits of communication to compute $f(x,y)$ for any x and y is called the communication complexity of f. We denote this quantity by $\mathsf{CC}(f)$. For a more formal definition we refer the reader to [2].

Trivially, for any f, $\mathsf{CC}(f)$ is at most n because one of the parties can send its entire string to the other, who can then compute f.[1] The main problem in communication complexity is to give better upper and lower bounds for $\mathsf{CC}(f)$, for specific functions f.

Communication complexity has been successful in showing lower bounds in several computational settings, such as data structures and streaming algorithms. In the following section, we will see some simple examples of lower bound proofs and an application that proves that simulating a two-tape Turing machine on a one-tape machine incurs a quadratic slowdown. Section 3 describes other techniques for proving upper and lower bounds on communication complexity. In Section 4 we extend some of these ideas to multi-party communication. We conclude in Section 5.

2 Warm-up: Lower bounds for some simple functions

Suppose Alice and Bob are given two n-bit strings x and y, respectively, and they wish to determine if $x = y$. That is, they wish to compute $EQ(x,y)$ which is 1 if $x = y$ and 0 otherwise.

Theorem 2.1. $\mathsf{CC}(EQ) \geq n$.

Proof. Assume for the sake of contradiction that $\mathsf{CC}(EQ) \leq n-1$. Then for every pair (x,y) there exists a communication 'transcript' $t \in \{0,1\}^{n-1}$ by which $EQ(x,y)$ can be computed. Now consider the pairs $(x,x), x \in 0,1^n$. Since there are only $2^n - 1$ possible transcripts, there must be two pairs (x,x) and (x',x') for which the communication transcripts are identical (call this t).

Now suppose Alice is given the string x and Bob is given x'. At any stage of the protocol (assuming it is deterministic), Alice cannot distinguish the current instance from the one where Bob is given x (and similarly for Bob). Thus the transcript is precisely the same, and they both conclude $x = x'$, which is a contradiction.

□

We will now look at an application of communication complexity to proving lower bounds in a different model of computation. Consider the language

$$\textsc{Pal} = \{s_1 s_2 \dots s_n s_n s_{n-1} \dots s_1 \ : s_i \in \{0,1\}\}.$$

[1] Technically, if we require both the parties to know the value of f, this protocol needs $n+1$ bits of communication, where the last bit is used to communicate the value of f to the other party.

The problem of deciding whether a string is in PAL is in P, and can in fact be solved in linear time on a Turing Machine with two tapes (the string is written on both tapes; the two heads start at opposite ends and check for equality as they move). But what is the time complexity of PAL on a 1-tape TM? We will show that this is in fact $\Omega(n^2)$. The proof will involve a simple application of Theorem 2.1.

Theorem 2.2. $\mathsf{TIME}(\text{PAL})_{\textit{1-tape TM}} \geq \Omega(n^2)$.

Proof. Suppose $n = 3k$ and consider the following string written on the TM's tape which is clearly in PAL:

$$S = \boxed{s_1 \dots s_k | 0 \dots 0 | s_k \dots s_1}$$

Consider a *recorder* R_j between tape positions j and $j+1$, for $k \leq j < 2k$. The recorder stores the state of the Turing machine whenever the tape head moves from j to $j+1$ or vice versa.

Now consider two parties Alice and Bob who have access to the substring $S(1:j)$ and $S(j+1:n)$ respectively. Just knowing the recording of R_j, they can both decide if $S(1:k) = \mathsf{reverse}(S(2k:3k))$. Thus by Theorem 2.1 the length of the recorded message must be at least k. Thus the number of times the tape crossed R_j is at least $\frac{k}{\log |M|}$, where $|M|$ is the number of states in the Turing machine.

Since this can be said for all $k \leq j < 2k$, the total number of moves of the tape head (and therefore the time taken) is at least $\frac{k^2}{\log |M|} = \Omega(n^2)$.

□

Note that the function PAL is an example of a language that *needs* a quadratic slowdown when going from two-tape TMs to one-tape TMs. In terms of upper bounds, it can be shown that this is the worst possible, in particular, any computation that runs in time T on a two-tape TM can be simulated in time T^2 on a one-tape TM. In fact, for any fixed integer k, a computation that takes time T on a k-tape TM can be simulated on a two-tape TM in time $O(T \log T)$ (and hence on a one-tape TM in time $O(T^2 \log^2 T)$). The proofs of these facts are left as exercises.

In general there is no known super-quadratic lower bound on the time complexity of a (specific) language even on a 1-tape TM. So also, there is no known super linear lower bound on the time complexity of a language on a multi-tape TM. However it is known, for instance, that there exists a language $L \in \mathsf{TIME}(n^7) \setminus \mathsf{TIME}(n^3)$ (the Time Hierarchy theorem, see [2] for a general statement). The proof uses diagonalization and hence does not imply a lower bound for "natural" explicit functions like 3SAT. It is also known that for any $f(n) = o(\log^* n)$, there exists some function $g \in \mathsf{NTIME}(n) \setminus \mathsf{TIME}(nf(n))$; in other words, deterministic linear time is distinct from nondeterministic linear time [5]. This result was recently extended in [6].

3 Proving bounds on communication complexity

The known methods for proving lower bounds in communication complexity follow a general framework, summarized as follows. Given a class of functions f whose communication complexity we wish to bound, do the following:

1. Find a function $\Pi(f)$ such that $\Pi(f) \leq CC(f)$.
2. Show that $\Pi(f_0) \geq G(n)$ for some specific function f_0, where n is the size of the input string. Usually, $G(n)$ is something like $\Omega(n)$ or $n^{\Omega(1)}$.

In general it is always true that $\Pi(f)$ is large for a random f and sometimes true that $\Pi(f)$ takes exponential time to compute. The fooling set parameter used to prove Theorem 2.1 and the rank parameter described in Section 3.2 are computable in polynomial time, whereas the discrepancy parameter described in Section 3.3 is **NP**-hard to compute exactly. This begs the question of whether there are "natural" lower bounds in communication complexity that can be

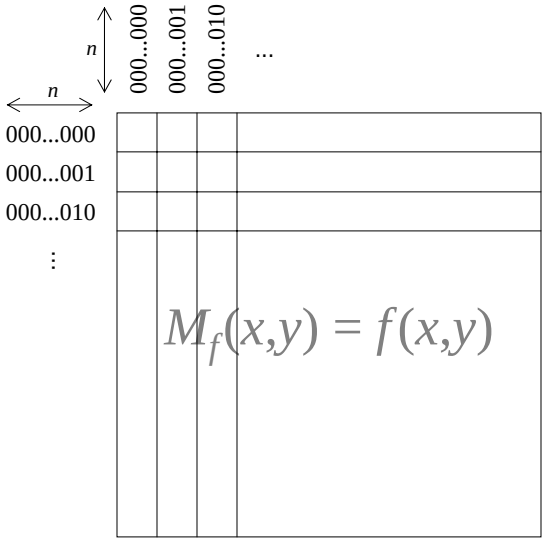

Figure 1: The $2^n \times 2^n$ matrix M_f for function $f(x, y)$.

characterized by the complexity of $\Pi(f)$. By "natural" here we mean that $\Pi(f)$ is constructive: that is, computable in time polynomial in the size of the truth table of the function. As we will see in Section 4, some lower bounds in communication complexity imply lower bounds in circuit classes for which pseudorandom functions exist; all of these lower bounds are in the multiparty model. This means that we do not expect to find natural lower bounds for these circuits, since such proofs immediately imply the ability to distinguish pseudorandom functions from random.

In the remainder of this section, we discuss different methods for proving lower bounds on communication complexity that fall in the general framework described above.

3.1 Tiling

In what follows, we view the function $f(x, y)$ as a matrix M_f that represents all possible input pairs (x, y) as shown in Figure 1. M is a $2^n \times 2^n$ matrix whose (x, y)'th entry is the value $f(x, y) \in \{0, 1\}$. Define a *combinatorial rectangle* to be any submatrix of M, and call this rectangle *monochromatic* if all values in the rectangle are identical. A monochromatic tiling of M_f is a partition of M_f into disjoint monochromatic rectangles. Let $\chi(f)$ be the minimum number of rectangles in any monochromatic tiling of M_f. $\chi(f)$ is the function we will use to lower bound $CC(f)$, following our general framework. In particular, we can show the following:

Theorem 3.1. $\lg(\chi(f)) \leq CC(f)$.

Proof. Assume without loss of generality that the first bit of the protocol is sent by Alice to Bob. This bit partitions M_f into at most two rectangles, one corresponding to the subset of Alice's inputs corresponding to the first bit being 0, and the other corresponding to the first bit being 1. Similarly, the next bit communicated by the protocol partitions each of these rectangles into two smaller rectangles. After k bits have been communicated, M_f is partitioned into at most 2^k rectangles. If $CC(f) = k$, then the tiling created by these partitions must be monochromatic since the value of f is uniquely determined. Thus it follows that $2^k \geq \chi(f)$ (the minimum tiling).

□

We can also use $\chi(f)$ to upper bound the communication complexity, as follows:

Theorem 3.2. $CC(f) \leq O(\lg^2(\chi(f)))$.

Proof. We give a protocol for Alice and Bob with communication complexity $O(\lg^2(\chi(f)))$. Suppose Alice and Bob fix a tiling τ with $\chi(f)$ (disjoint) rectangles (they can do this because both know M_f). The protocol is now as follows.

Alice selects a rectangle $R \in \tau$ that contains row x (which one to choose if there are many will be described below) and communicates it to Bob. This can be done using $\lg(\chi(f))$ bits. Now Bob knows $x \in R$ so he eliminates all rectangles that are row-disjoint with R. Of the remaining, Bob picks a rectangle C which contains y and communicates it to Alice. This forms a 'round' of communication. Then Alice eliminates rectangles that are column-disjoint with C, selects a rectangle R' among the remaining rectangles that contains x, and the process continues.

Suppose Alice selects R such that Bob ends up eliminating as many rectangles as possible, and Bob does the same while choosing C. We now show that after a round at most 2/3 fraction of the rectangles remain under consideration (by Alice). Suppose not. So every rectangle in τ shares a row with at least 2/3 of the rectangles in τ (else Alice can choose this rectangle and we would be done). So also we can conclude that every rectangle in τ shares a column with 2/3 of the rectangles in τ (else Bob can choose this rectangle). Thus there has to be a pair of rectangles that share both a row and column, a contradiction since τ consists of disjoint rectangles.

Thus there are at most $O(\lg(\chi(f)))$ rounds, and each round takes $\lg(\chi(f))$ communication. □

The tiling method is the strongest lower bound technique: the fooling set (Section 2), rank (Section 3.2), and discrepancy (Section 3.3) methods imply a bound on $\chi(f)$, and hence can never prove better lower bounds than tiling.

3.2 Rank

The rank method is an algebraic method used to lower bound $\chi(f)$ and hence the communication complexity of f. Recall the definition of rank of a square matrix M, which is the maximum number of linearly independent rows (or columns) in M. Another way to view this quantity is to decompose M into a set of rank 1 matrices of the same size which add up to yield M. The size of the smallest such set is exactly the rank of M. This leads us to the following claim:

Theorem 3.3. $\lg(\mathsf{rank}(M_f)) \leq \lg(\chi(f))$.

Proof. Consider a monochromatic tiling of M with $\chi(f)$ rectangles. Each rectangle in this tiling can be viewed as a matrix of rank at most 1 by filling all entries outside of the rectangle with 0's. Since the rectangles partition M, the resulting matrices constitute a set of rank 1 matrices of the form described above. The rank of M is the size of the smallest such set, so it must be that $\mathsf{rank}(M_f) \leq \chi(f)$. □

Although the tiling method is a stronger lower bound technique than the rank method, it is conjectured that rank captures the communication complexity of f to within a polynomial factor. This is:

Conjecture 3.4. *(log rank conjecture)*
There is a constant $c > 1$ such that $CC(f) = O(\lg^c(\mathsf{rank}(M_f)))$ for all f and input sizes n.

If the log-rank conjecture is true, then every communication complexity lower bound can be "naturalized" in the sense that we can can efficiently approximate $CC(f)$ for any f (by just computing the rank). The log-rank conjecture is informally related to matrix rigidity because both problems conflate an algebraic property of the matrix (its rank) with its combinatorial properties. In the case of the log-rank conjecture the combinatorial property is how low the communication complexity of the matrix is; in the case of matrix rigidity it is how close the matrix is to being sparse. In some sense the two problems derive their difficulty from this interaction of algebra and combinatorics. The primary difference between the problems is that they operate in different regimes: in the log-rank conjecture we typically think of low rank as polylogarithmic, while in matrix rigidity we talk about slightly sub-linear rank (e.g., $n/\log\log n$).

3.3 Discrepancy

In earlier lectures, we have encountered discrepancy of matrices. We now define discrepancy for a function $f : \{0,1\}^n \times \{0,1\}^n \to \mathbb{R}$ in the natural way, i.e., as the discrepancy of the matrix M_f with $M_f(x,y) = f(x,y)$. Formally, (for convenience let us write $N = 2^n$)

Definition 3.5. Suppose $f : \{0,1\}^n \times \{0,1\}^n \to \mathbb{R}$. Its discrepancy

$$\mathsf{Disc}(f) = \max_{A,B \subseteq \{0,1\}^n} \frac{1}{N^2} \left| \sum_{x \in A, y \in B} f(x,y) \right|.$$

It is simple to see that $\chi(f)$ defined earlier satisfies $\chi(f) \geq \frac{1}{\mathsf{Disc}(f)}$. This is because if $\chi(f)$ monochromatic rectangles cover the entire rectangle (of size N^2), there must be one of size at least $\frac{N^2}{\chi(f)}$ and using rows and columns of this for A, B as in Defn.3.5 gives the result. Thus if $\chi(f)$ is small, the discrepancy is large. But the other direction is not necessarily true – in particular we can have huge discrepancy (≈ 1) but $\chi(f)$ can be 2^n, as can be seen from the 'equality' function EQ we have considered earlier.

The observation above along with what we have seen earlier implies that obtaining a good upper bound on $\mathsf{Disc}(f)$ implies a good lower bound on $\mathsf{CC}(f)$. In the following lemma, we bound the discrepancy in terms of the spectral norm of the matrix M_f defined as above. Recall that the rows and columns of M_f are indexed by elements of $\{0,1\}^n$ and $M_f(x,y) = f(x,y)$. Also, the spectral norm of a matrix M is

$$\|M\| = \max_{u \in \mathbb{R}^N, \|u\|=1} \|Mu\| = \max_{u,v \in \mathbb{R}^N, \|u\|=\|v\|=1} u^T M v.$$

The equality of the last two terms is easy to see.

Lemma 3.6. $\mathsf{Disc}(f) \leq \frac{\|M_f\|}{N}$.

Proof. Suppose $A, B \subseteq \{0,1\}^n$ are such that $\mathsf{Disc}(f) = \frac{1}{N^2} \left| \sum_{x \in A, y \in B} f(x,y) \right|$. Equivalently,

$$\begin{aligned} \mathsf{Disc}(f) = \frac{1}{N^2} |\mathbf{1}_B^T M_f \mathbf{1}_A^T| &\leq \frac{1}{N^2} \sqrt{|A||B|}\, \|M_f\| \\ &\leq \frac{\|M_f\|}{N}. \end{aligned}$$

□

As an example, consider the Hadamard matrix with entries $H(x,y) = (-1)^{x \cdot y}$. The spectral norm $\|H\| = \sqrt{N}$, because any unit vector is mapped by H to a vector of length precisely $\sqrt{N}$ (H is just the Fourier transform, scaled by $\sqrt{N}$). Thus we get a lower bound of $\log\left(\frac{N}{\|H\|}\right) = \Omega(n)$ on the communication complexity of the dot product.

Given a matrix M, computing the discrepancy is a well-studied problem. Note that it is not clear if this can be done in time polynomial in the size of the matrix, because the maximum is over all A, B subsets of the rows and columns. Alon and Naor [1] give a constant factor approximation algorithm using a clever rounding of the natural SDP relaxation for the problem. Also, by a simple reduction from Max-Cut, it follows that the problem is Max-SNP hard (i.e., it does not have a polynomial time approximation scheme unless $\mathbf{P} = \mathbf{NP}$).

Another way of looking at the matrix M_f in our case is as the adjacency matrix of a bipartite graph. The rows and columns represent the two sets of vertices and there is an edge between x, y iff $M_f(x,y) = 1$. To find the discrepancy, we are interested in subsets of vertices A, B such that $\left|E(A,B) - \frac{1}{2}|A||B|\right|$ is as large as possible. Note that $\frac{1}{2}|A||B|$ is the expected number of edges in a random bipartite graph on A, B. Thus the discrepancy measures in some sense the 'distance from random' (it is un-normalized, as we defined it).

Another quantity that is typically used as a measure of 'distance from random' is the number of 4-cycles as compared to the random graph. It turns out that having the right number of 4-cycles implies many other 'quasirandom' properties (see [3]). Also, this quantity can be generalized naturally to the case of k-party communication for $k > 2$.

Given an f, we define

$$\mathcal{E}(f) = \mathbb{E}_{\substack{x,x' \\ y,y'}}[f(x,y)f(x,y')f(x',y)f(x',y')].$$

We now obtain an upper bound on the discrepancy in terms of $\mathcal{E}(f)$.

Lemma 3.7. $\mathsf{Disc}(f) \leq (\mathcal{E}(f))^{1/4}$

Proof. Fix some A, B. Denote by $a(x), b(x)$ the characteristic functions of A and B respectively. Consider the quantity

$$\begin{aligned} D &= |\mathbb{E}_{x,y}[f(x,y)a(x)b(y)]| \\ &\leq \mathbb{E}_y\big[b(y)\,|\mathbb{E}_x[f(x,y)a(x)]|\,\big] \end{aligned}$$

Thus

$$\begin{aligned} D^2 &\leq \mathbb{E}_y\big(\mathbb{E}_x[f(x,y)a(x)]\big)^2 \\ &= \mathbb{E}_y\big[\mathbb{E}_{x,x'}[f(x,y)f(x',y)a(x)a(x')]\big] \\ &= \mathbb{E}_{x,x'}\big[a(x)a(x')\mathbb{E}_y[f(x,y)f(x',y)]\big] \end{aligned}$$

Squaring again and using the same trick with y, we have

$$D^4 \leq \mathbb{E}_{x,x',y,y'}\big[f(x,y)f(x',y)f(x,y')f(x',y')\big] = \mathcal{E}(f)$$

Since this holds for all A, B, we have $\mathsf{Disc}(f)^4 \leq \mathcal{E}(f)$. □

4 Multi-party Communication

So far we have dealt with two-party communication complexity where two parties have access to strings x, y respectively and they want to compute some function $f(x,y)$. We will now generalize this to the case of more than two parties. What we look at is the so called *number on the forehead* model.

Suppose there are k people in a room, with person P_i having a string x_i on his/her forehead. In other words, P_i knows x_j for all $j \neq i$. The aim is to compute some function $f(x_1, \ldots, x_k)$ using the least possible amount of communication (everyone can 'hear' what everyone else says). We now define a k-party analogue of *combinatorial rectangles*. For simplicity, write $\Sigma = \{0,1\}^n$.

Definition 4.1. A set $S \subseteq \Sigma^k$ is said to be a *cylinder in dimension i* if for all $(x_1, \ldots, x_k) \in S$, it is also true that $(x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_k) \in S$ (for all x_i'). A *cylinder intersection* is a set of form $\cap_{i=1}^k S_i$ where S_i is a cylinder in dimension i.

Given an $f : \Sigma^k \to \{\pm 1\}$, we can define a k-dimensional matrix M_f as before, with M_f having $(x_1, \ldots, x_k)$th entry equal to $f(x_1, \ldots, x_k)$. Now suppose every partition of Σ^k into monochromatic cylinder intersections requires at least R parts. Then an argument analogous to Theorem 3.2 gives that the k-party communication complexity is at least $\lg R$.

The discrepancy is now defined to be

$$\mathsf{Disc}(f) = \frac{1}{2^{nk}} \max_C \left| \sum_{\mathbf{x} \in C} f(\mathbf{x}) \right|$$

where the maximum ranges over all cylinder intersections C. Let us now define a k-party version of $\mathcal{E}(f)$.

Definition 4.2. Suppose $f : \Sigma^k \to \{1, -1\}$.

$$\mathcal{E}(f) = \mathbb{E}_{\substack{x_1,\ldots,x_k \\ x'_1,\ldots,x'_k \in \Sigma}} \prod_{\mathbf{x}} f(\mathbf{x})$$

where the product inside ranges over $\mathbf{x}$ that are of the form $(y_1, \ldots, y_k)$, with each $y_i = x_i$ or $y_i = x'_i$.

An analogue of Lemma 3.7 now holds (apply Cauchy-Schwarz k times). We refer to [2] for the details of the proof.

Lemma 4.3. $\mathsf{Disc}(f) \leq \mathcal{E}(f)^{1/2^k}$

4.1 Generalized Inner Product: A lower bound

We have seen a lower bound of $\Omega(n)$ for the communication complexity of the inner product function. In this section we will see a generalized definition of the inner product, and we will prove a lower bound on the k-party communication complexity.

Given k vectors $x_1, \ldots, x_k \in \Sigma$, define the *generalized inner product* (GIP) to be $\oplus_{i=1}^{n} \wedge_{j=1}^{k} x_{j,i}$, where $\oplus$ denotes addition modulo 2. Since we want a function with range $\{-1, 1\}$, we can write $\text{GIP} = \prod_{i=1}^{n} (-1)^{x_{1i} x_{2i} \ldots x_{ki}}$.

Theorem 4.4. *(Babai, Nisan, Szegedy)* $\mathsf{CC}_k(\text{GIP}) \geq \Omega\left(\frac{n}{4^k}\right)$

Proof. From the results of the previous section, it suffices to prove that $\mathcal{E}(\text{GIP}) \leq 2^{-\Omega\left(\frac{n}{2^k}\right)}$.

Indeed, we will prove that.

$$\mathcal{E}(\text{GIP}_{k,n}) = \left(1 - \frac{1}{2^k}\right)^n.$$

Say we fix $x_1, \ldots, x_k, x'_1, \ldots, x'_k \in \Sigma$. Denote by H the set of vectors $\mathbf{x}$ of form $(y_1, \ldots, y_k)$ with $y_i = x_i$ or $y_i = x'_i$ for all i. Note that

$$\begin{aligned} \prod_{\mathbf{x} \in H} f(\mathbf{x}) &= \prod_{\mathbf{x} \in H} \prod_{i=1}^{n} (-1)^{y_i} \quad \text{where } \mathbf{x} = (y_1, \ldots, y_k) \\ &= \prod_{i=1}^{n} (-1)^{(x_{1,i} + x'_{1,i})(x_{2,i} + x'_{2,i}) \ldots (x_{k,i} + x'_{k,i})} \end{aligned}$$

Now $\mathcal{E}(\text{GIP}_{k,n})$ is the average of the above quantity over all choices of $x_1, \ldots, x_k, x'_1, \ldots, x'_k \in \Sigma$ (which is $\{0,1\}^n$). We will fix $x_1, \ldots, x_k$ and show that the expectation over x'_i is precisely the quantity we want.

$$\begin{aligned} & \mathbb{E}_{x'_1,\ldots,x'_k \in \Sigma} \prod_{i=1}^{n} (-1)^{(x_{1,i} + x'_{1,i})(x_{2,i} + x'_{2,i}) \ldots (x_{k,i} + x'_{k,i})} \\ = \; & \frac{1}{2^{nk}} \prod_{i=1}^{n} \sum_{x'_{j,i} \in \{0,1\}} (-1)^{(x_{1,i} + x'_{1,i})(x_{2,i} + x'_{2,i}) \ldots (x_{k,i} + x'_{k,i})} \\ = \; & \frac{1}{2^{nk}} \prod_{i=1}^{n} \left(2^k - 1\right) = \left(1 - \frac{1}{2^k}\right)^n \end{aligned}$$

The last equality is because for a fixed i, the product in the exponent is even for all choices of $x'_{j,i}$ except the unique choice that makes it equal to 1. □

Note that the above lower bound is trivial when $k = \text{polylog}(n)$. Finding functions f for which there exist non-trivial lower bounds on $\mathsf{CC}_k(f)$ for these values of k is one of the important open problems in multiparty communication complexity. We will see shortly that proving such lower bounds imply interesting lower bounds in *circuit* complexity.

4.2 Application to Circuit lower bounds

We have looked at these in previous lectures, but let us summarize a few basic notions regarding circuit complexity. A function $f : \{0,1\}^* \to \{0,1\}$ is said to be computable by a family of circuits $\{C(n) : n \in Z\}$ such that for every n, $C(n)$ takes a string x in $\{0,1\}^n$ as input and outputs $f(x)$ (i.e., the circuit may vary drastically with n but given n, the circuit should output $f(x)$ for every $x \in \{0,1\}^n$).

Now let us recall that ACC is the set of functions computable by poly-sized, constant depth circuits with MOD_p gates (of unbounded fan-in) for a constant number of different p.[2]

The following result due to Yao will help in the connection between showing a certain function is not in ACC and communication complexity. The result is analogous to that used in Razborov's proof of $\mathsf{Parity} \notin \mathbf{AC^0}$. For a polynomial f with integer coefficients, let us denote by $\|f\|$ the logarithm of the sum of the absolute values of the coefficients (basically the number of bits needed to 'write down' the polynomial). Also, by $MOD(x; n_1, n_2, \ldots, n_k)$, we will mean the quantity

$$(\ldots((x \mod n_1) \mod n_2)\ldots \mod n_k).$$

Yao [4] proved that

Theorem 4.5. *Suppose f can be computed on a polynomial sized circuit of depth k with MOD_{p_i} gates at depth i, for $1 \le i \le k$. Then there exist*

1. *Polynomials F_n with integer coefficients with $\|F_n\| = polylog(n)$, $degree(F_n) = polylog(n)$, and*
2. *Integers $m_{n,i}$, $1 \le i \le k$ with $\log m_{n,i} = polylog(n)$*

such that $f(x) = MOD(F_n(x); p_1^{m_{n,1}}, p_2^{m_{n,2}}, \ldots, p_k^{m_{n,k}})$.

Note that this theorem easily implies the following

Theorem 4.6. *Suppose for some function f we can prove that $\mathsf{CC}_{polylog(n)}(f) \ge \log^{\omega(1)} n$. Then $f \notin \mathsf{ACC}$.*

To see this, suppose we consider some n, and consider the polynomial F_n as guaranteed by Theorem 4.5. Suppose its degree is d. Now consider $d+1$ parties and the number on the forehead model. All the parties know F_n and the integers $m_{n,i}$. Since the polynomial is of degree d, every monomial has at most d different x_i's involved, thus at least one of the $d+1$ parties can 'see' the values of all of them (since it is the number on the forehead model) and can thus compute the monomial. Every party computes the monomials he/she can compute and finally they all add up the results (if two parties can compute a monomial we can assume there is some initial agreement on who is supposed to compute it). Also, the bounds on $\|F_n\|$ and the degree imply that the amount of communication is polylog(n). Thus there exists a $d+1$ party protocol with communication complexity polylog(n), a contradiction since $\mathsf{CC}_{\mathrm{polylog}(n)}(f) \ge \log^{\omega(1)} n$.

The rough idea in the proof of Theorem 4.5 is to inductively compute polynomials for the output of the MOD_p gates starting at the lowest level and moving up, and show that the polynomials do not 'blow up' in size. We do not go into the details of the construction (see [4]), but the following sequence of polynomials play an important role. Let $S_1(x) = 3x^2 - 2x^3$, and inductively define $S_i(x) = S_1(S_{i-1}(x))$. A property of these polynomials is that for any integers t, m and N, if $N \mod m \in \{0,1\}$, then $S_t(N) \mod m^{2^t} = N \mod m$. This can be proved easily by induction on t, and this plays a key role in the proof. We refer to [4] for the details.

[2] A MOD_p gate has some m inputs $x_1, \ldots, x_m$ and the ouput is 1 if $\sum_i x_i \equiv 0 \pmod p$ and 0 otherwise

5 Conclusion

We tried to present a few introductory results in Communication complexity and some applications to other areas like circuit complexity. The tiling and discrepancy arguments have been strengthened in many ways, as we will see in the coming lectures.

For any proofs that have been omitted or just sketched, we refer the reader to the chapter on Communication complexity in [2], or to the papers cited.

References

[1] N. Alon, A. Naor. *Approximating the cut-norm via Grothendieck's inequality.* In Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing, 2004 (72–80).

[2] S. Arora, B. Barak. *Computational Complexity: A modern approach.* Chapter on Communication Complexity.

[3] F. Chung, R. L. Graham, R. M. Wilson. *Quasi-random graphs.* In Combinatorica **9** (1989), 345–362.

[4] A. C. Yao. *On ACC and Threshold Circuits.* In Proceedings of the Thirty-first IEEE Symposium on Foundations of Computer Science, October 1990 (619-627).

[5] W. J. Paul, N. Pippenger, E. Szemerédi, and W. T. Trotter. *On determinism versus nondeterminism and related problems.* In Proceedings of Twenty-fourth IEEE Symposium on Foundations of Computer Science, November 1983 (429-438).

[6] R. Santhanam. *On Separators, Segregators and Time versus Space.* In Proceedings of the Sixteenth Annual IEEE Conference on Computational Complexity, June 2001 (286-294).