CSE 291: Communication Complexity, Winter 2019
Deterministic protocols

Shachar Lovett

January 28, 2019

1 Overview

The model of communication complexity was introduce by Yao in the late 1970s [8] as an abstract way to capture communication bottlenecks in algorithms. In this model, there are several players, each with their own input. Their goal is to perform some distributed task based on their inputs, while minimizing communication. Importantly, we assume the players have *unbounded computational model*, and focus only on the communication. In this first part of the class, we focus on the simplest setting: deterministic protocols between two players, which are typically called Alice and Bob.

2 The model: deterministic protocols

Let Alice and Bob be two players, holding inputs $x \in X$ and $y \in Y$, respectively. A *deterministic communication protocol* is a protocol π, where in each turn one of the players send a message to the other one. Finally, the players need to determine an output $z \in Z$ for the protocol. We assume that each message is 1-bit long. Sometimes, it will be convenient to assume that the players alternate (that is, Alice sends a bit, then Bob, then Alice, and so on), which could lead to a factor of two slow-up in the protocol (which we typically not care about).

Given inputs $x \in X, y \in Y$, we call the *transcript* $\pi(x, y)$ of the protocol the sequence of bits sent between the players. The value of the protocol $z \in Z$ is some fixed function of the transcript $\text{val}(\pi) \in Z$. In the typical case where $Z = \{0, 1\}$, we can simply assume that the last bit in the protocol is its value. The *cost* of a protocol is the maximal number of bits sent, namely the length of the longest transcript.

A protocol π computes a (total) function $f : X \times Y \to Z$ if

$$\text{val}(\pi(x, y)) = f(x, y) \qquad \forall x \in X, y \in Y$$

The *deterministic communication complexity* of f is the *minimal cost* of a deterministic protocol which computes it. We denote it by $D(f)$.

Claim 2.1. *For any $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, $D(f) \leq n+1$.*

Proof. Alice sends her input x to Bob, who sends back $f(x,y)$. $\square$

Example 2.2 (Equality)**.** *The Equality function on n-bit inputs is $EQ : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ defined as*

$$EQ(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

We will later see that equality is maximally hard for deterministic protocols, namely $D(EQ) = n+1$.

Protocol tree. One way to model a deterministic protocol under these assumptions is via a *protocol tree.* The protocol is described by a binary tree, whose internal nodes correspond to players actions and whose leaves correspond to outputs. Specifically,

- Each internal node v belongs to either Alice or Bob.
- Every internal node v that belongs to Alice is associated with a function $A_v : X \to \{0,1\}$, which describes the bit sent by Alice when reaching this node, given her input x.
- Every internal node v that belongs to Bob is associated with a function $B_v : Y \to \{0,1\}$, which describes the bit sent by Bob when reaching this node, given his input y.
- Every leaf ℓ is associated with a value $\mathrm{val}(\ell) \in Z$.

Every input pair $x \in X, y \in Y$ defines a unique root-to-leaf path in the tree. If it reaches a leaf ℓ, the output of the protocol is $\mathrm{val}(\ell)$. The cost of a protocol is equivalent to the depth of the protocol tree; leaves are equivalent to transcripts.

A useful fact is that protocol trees can be balanced.

Lemma 2.3. *Any protocol tree with N leaves can be equivalently computed by a protocol tree with depth $O(\log N)$.*

Proof. Let T be a protocol tree computing some function. We denote by $|T|$ the number of leaves in T, where we assume $|T| = N$. Given a node v in T, denote by T_v the sub-tree of T rooted at v. First, we claim that there exists a node v such that

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

To see that, let u be a node such that $|T_u|$ is minimal conditioned on $|T_u| \geq 2N/3$. Let v', v'' be the children of u. Then $|T_u| = |T_{v'}| + |T_{v''}|$. Let $v \in \{v', v''\}$ be the one that maximizes $|T_v|$. Then $|T_v| < 2N/3$ by the maximality of u, and $|T_v| \geq |T_u|/2 \geq N/3$.

The node v corresponds to a rectangle R_v of all the inputs x, y that would pass through it. In the new protocol π', the players first check if their input $x, y \in R_v$ or not, by sending one bit to each other. If it is, they continue in $|T_v|$, otherwise they continue in $T \setminus T_v$. In either case, the new protocol tree has $\leq 2N/3$ leaves. Thus we obtain that

$$depth(\pi') \leq 2 \log_{3/2} N = O(\log N).$$

□

Partial functions. We would sometime need to study protocols for *partial functions.* Let $D \subset X \times Y$ be a subset of the inputs. A partial function is $f : D \to Z$. A protocol π computes f if

$$\mathrm{val}(\pi(x, y)) = f(x, y) \qquad \forall (x, y) \in D$$

Note that while the protocol is defined for all $x \in X, y \in Y$, we do not care what it computes for inputs $(x, y) \notin D$.

Example 2.4 (Gap Hamming Distance). *Let $x, y \in \{0, 1\}^n$. Their hamming distance is $dist(x, y) = |\{i : x_i \neq y_i\}|$. The Gap Hamming Distance function on n-bit inputs is the following partial function:*

$$GHD(x, y) = \begin{cases} 1 & \text{if } dist(x, y) \geq n/2 + \sqrt{n} \\ 0 & \text{if } dist(x, y) \leq n/2 - \sqrt{n} \\ * & \text{otherwise} \end{cases}$$

Relations. A *relation* is a function which can obtain multiple possible values. It can be modeled by $R \subset X \times Y \times Z$. A protocol π computes a relation R if

$$(x, y, \mathrm{val}(\pi(x, y))) \in R \qquad \forall x \in X, y \in Y.$$

One can also define *partial relations* in the obvious way.

Example 2.5 (Different bit). *Let $X, Y \subset \{0, 1\}^n$ be disjoint sets. The* different bit *problem is to find, given $x \in X, y \in Y$, an index $i \in [n]$ such that $x_i \neq y_i$. Equivalently, it is defined by the relation $R = \{(x, y, i) : x \in X, y \in Y, x_i \neq y_i\}$. We will later see that this relation has tight connections to formula lower bounds in complexity theory.*

3 Example: Clique vs Independent Set

The Clique vs Independent Set (CL-IS) problem was introduced by Yannakakis [7] as part of his program to prove lower bounds on how linear programs can solve NP-hard problems.

Definition 3.1 (Clique vs Independent Set). *The problem is defined by a graph $G = (V, E)$ on n nodes, which is known in advance. The inputs to the players are as follows. Alice receives a clique $C \subset V$ in the G, and Bob receives an independent set $I \subset V$ in G. Their goal is to check whether C, I are disjoint or not. Note that it always holds that $|C \cap I| \in \{0, 1\}$.*

A naive protocol has cost $O(n)$, where one of the players send their input to the other. However, we can do much better.

Lemma 3.2. *$D(\mathit{CL\text{-}IS}) = O(\log^2 n)$.*

Proof. We design a deterministic protocol with cost $O(\log^2 n)$. The protocol operates in rounds, where each round has cost $O(\log n)$ and there are $O(\log n)$ rounds.

Before the i-th round starts, the players identified a set V_i of nodes such that $C \cap I \subset V_i$. At the end of the i-th round, the players construct V_{i+1} with the same promise, such that $|V_{i+1}| \le |V_i|/2$, or they halt and output an answer. After $\log n$ rounds we have $|V_{\log n}| = 1$ in which case they can determine the answer in $O(1)$ cost. We will show how each round can be implemented using $O(\log n)$ communication.

We next describe the i-th round. Let $d_i(v)$ denote the degree of v in V_i. First, Alice checks if there exists a node $v \in C$ such that $d_i(v) \le |V_i|/2$. If she finds such a node, she sends it to Bob. If $v \in I$ then Bob terminates and answers 1. Otherwise, the players set

$$V_{i+1} = \{u \in V_i : (u, v) \in E\}.$$

Note that $C \cap I \subset V_{i+1}$ and that by design $|V_{i+1}| \le |V_i|/2$.

If this failed, Bob checks if there exists a node $v \in I$ such that $d_i(v) \ge |V_i|/2$. If he finds such a node, he sends it to Alice. If $v \in C$ then Alice terminates and answers 1. Otherwise, the players set

$$V_{i+1} = \{u \in V_i : (u, v) \notin E\}.$$

Note that $C \cap I \subset V_{i+1}$ and that by design $|V_{i+1}| \le |V_i|/2$.

Finally, if both Alice and Bob failed, then they know that $d_i(v) > |V_i|/2$ for all $v \in C$ and that $d_i(v) < |V_i|/2$ for all $v \in I$. This implies that C, I must be disjoint, in which case the players terminate and output 0.

Note that each round can be implemented using $O(\log n)$ communication, which leads to the overall $O(\log^2 n)$ communication cost. ☐

An open problem since the work of Yannakakis is whether this $O(\log^2 n)$ is tight or not. Only in recent years this was proven to be true. First, Göös [1] proved a lower bound of $\Omega(\log^{1.128} n)$. This was improved in [2] to a near tight $\Omega(\log^2 n / \log\log^{O(1)} n)$ lower bound. These proofs go via *lifting theorems*, which we will discuss later in this course.

4 Lower bound techniques

In this section we develop some general techniques to prove lower bounds for deterministic protocols. Let $f : X \times Y \to Z$. We can associate it with a matrix M_f, which is a $X \times Y$ matrix defined as $(M_f)_{x,y} = f(x, y)$. The lower bound techniques are all based on studying M_f, in particular monochromatic rectangles in M_f.

4.1 Rectangles

Definition 4.1 (Rectangles)**.** *A rectangle in $X \times Y$ is a set $R = A \times B$ with $A \subset X, B \subset Y$.*

Definition 4.2 (Monochromatic rectangles)**.** *Given $f : X \times Y \to Z$ and $z \in Z$, a rectangle R is monochromatic if $f(x, y) = z$ for all $(x, y) \in R$. In this case we say R is z-monochromatic.*

Claim 4.3. *Let $f : X \times Y \to Z$. If $D(f) = c$ then there is a partition of M_f into 2^c monochromatic rectangles.*

Proof. Every bit in the protocol partitions the matrix into two parts, either by partitioning the rows (if Alice sends the bit) or the columns (if Bob sends the bits). After c bits, we get a partition of M_f into 2^c rectangles. As the protocol computes a value $z \in Z$ for each transcript deterministically, each of these rectangles must be monochromatic. □

The *partition number* $P(f)$ is the minimal number of disjoint monochromatic rectangles that M_f can be partitioned into. As we just saw

$$D(f) \geq \log P(f),$$

where all logarithms are in base 2.

Corollary 4.4. *Let $f : X \times Y \to Z$. Assume that any monochromatic rectangle R in M_f has size $|R| \leq \varepsilon |X \times Y|$. Then $D(f) \geq \log(1/\varepsilon)$.*

Recall that $\text{EQ} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined as $\text{EQ}(x, y) = 1$ if $x = y$. We prove it has maximal deterministic communication complexity.

Claim 4.5. $D(EQ) = n + 1$.

Proof. Let $M = M_{\text{EQ}}$. Then M is a diagonal $2^n \times 2^n$ matrix, with 1 on the diagonal and 0 outside the diagonal. In particular, every 1-monochromatic rectangle contains exactly one diagonal element. So we need 2^n 1-monochromatic rectangles to cover all the 1s. We need at least one more to cover the 0s. So

$$P(\text{EQ}) \geq 2^n + 1.$$

Thus

$$D(\text{EQ}) \geq \log_2(2^n + 1) > n.$$

As $D(\text{EQ})$ is an integer, $D(\text{EQ}) \geq n + 1$. On the other hand, $D(f) \leq n + 1$ for any boolean function on n bits. □

An example for a promise problem is the Deutsch-Jozsa problem, which is a promise version of the Equality function. The inputs are $x, y \in \{0,1\}^n$ where n is even. The goal is to decide if $x = y$ or if they are maximally separated, namely if $\text{dist}(x, y) = n/2$. Formally, it is defined as

$$DJ(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } \text{dist}(x, y) = n/2 \\ * & \text{otherwise.} \end{cases}$$

This problem has a randomized protocol (a model which we will define in later classes) with $O(1)$ cost; and even a quantum protocol with cost $O(1)$ and zero error. However, it cannot be solved by efficient deterministic protocols.

Lemma 4.6. *$D(DJ) = \Omega(n)$.*

Proof. Assume DJ has a deterministic protocol with cost c. It induces a partition of $M = M_{DJ}$ into 2^c monochromatic rectangles. We will prove that each 1-monochromatic rectangle R contains at most $2^{(1-\varepsilon)n}$ diagonal elements in M. Thus we need at least $2^{\varepsilon n}$ rectangles, which gives $c \geq \varepsilon n$.

We will use the famous Frankl-Rödl theorem from combinatorics: for any $\alpha > 0$ there exists $\varepsilon > 0$ such that the following holds. Let $A, B \subset \{0,1\}^n$ be sets of size $|A|, |B| \geq 2^{(1-\varepsilon)n}$. Let $k = \lceil \alpha n \rceil$. Then

$$\{k, k+1, \ldots, n-k\} \subset \{d(a,b) : a \in A, b \in B\}.$$

To see how to use this, let $R = A \times B$ be a 1-monochromatic rectangle. Applying the Frankl-Rödl theorem for $\alpha = 1/2$ gives an ε, such that if $|A|, |B| \geq 2^{(1-\varepsilon)n}$ then there exists $a \in A, b \in B$ for which $d(a,b) = n/2$. Thus the protocol makes a mistake on R. So, in every 1-monochromatic rectangle we have $|A|, |B| \leq 2^{(1-\varepsilon)n}$, which means that R hits at most $2^{(1-\varepsilon)n}$ elements on the diagonal. □

4.2 Fooling sets

The argument for equality can be generalized as follows. Let $f : X \times Y \to Z$. A *fooling set* is a subset $S \subset X \times Y$ such that the following holds. There exists a value $z \in Z$ for which:

(i) $f(x,y) = z$ for all $(x,y) \in S$.

(ii) If $(x_1, y_1), (x_2, y_2) \in S$ then either $f(x_1, y_2) \neq z$ or $f(x_2, y_1) \neq z$.

Claim 4.7. *Assume f has a fooling set S. Then $D(f) \geq \log |S|$.*

This technique can be extended to handle partial functions. Let $f : D \to Z$ be a partial function for $D \subset X \times Y$. Then $S \subset D$ is a fooling set if there exists $z \in Z$ for which

(i) $f(x,y) = z$ for all $(x,y) \in S$.

(ii) If $(x_1, y_1), (x_2, y_2) \in S$ then either $(x_1, y_2) \in D$ and $f(x_1, y_2) \neq z$ or $(x_2, y_1) \in D$ and $f(x_2, y_1) \neq z$.

4.3 Rank

Let $f : X \times Y \to \{0,1\}$ be a total boolean function.If $D(f) = c$ then M_f can be partitioned into 2^c monochromatic rectangles $R_1, \ldots, R_{2^c}$. In particular, $M_f = \sum R_i$, and each R_i is a rank one matrix. In particular, the rank of M_f (over any field) is at most 2^c. This gives the *log rank lower bound.*

Claim 4.8. *Let $f : X \times Y \to \{0,1\}$. Then $D(f) \geq \log \mathit{rank}(M_f)$, where the rank can be computed over any field.*

We can take the rank over any field. However, the rank can be very different. A good example is the inner-product function. The inner product function $IP : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined as follows:

$$IP(x,y) = \langle x, y \rangle \mod 2.$$

It can be shown that:

- Over $\mathbb{F}_2$, the rank of M_{IP} is n.
- Over the reals (or any field of characteristics other than 2), the rank of M_{IP} is 2^n.

In particular, we get that $D(IP) \geq n$.

Another important example is the set-disjointness problem. The inputs are $x, y \in \{0,1\}^n$, which are interpreted as subsets $x, y \subseteq [n]$. The goal is to check if they are disjoint,

$$\text{DISJ}(x,y) = \begin{cases} 1 & \text{if } x, y \text{ are disjoint} \\ 0 & \text{otherwise} \end{cases}$$

The matrix M_{DISJ} is better known as the Kneser graph, which is known to have full rank. We will give a simpler proof for a slightly weaker bound.

Claim 4.9. *$D(DISJ) \geq n - O(\log n)$.*

Proof. Assume for simplicity that n is even. Restrict the disjointness problem to sets x, y of size exactly $n/2$. Then checking if x, y are disjoint is equivalent to checking if $x = [n] \setminus y$, which means that the corresponding matrix is a permuted identity matrix of size $\binom{n}{n/2}$. Thus $\text{rank}(M_{\text{DISJ}}) \geq \binom{n}{n/2} = \Theta(2^n/\sqrt{n})$. □

The log rank conjecture. How tight is the log-rank lower bound? The log-rank conjecture of Lovász and Saks from 1988 [4] speculates that it is tight, up to polynomial factors.

Conjecture 4.10 (Log rank conjecture)**.** *Let $f : X \times Y \to \{0,1\}$ with $\mathit{rank}(M_f) = r$, where the rank is computed over the reals. Then $D(f) \leq \log^{O(1)} r$.*

It is easy to prove an exponentially worse bound.

Claim 4.11. *Let $f : X \times Y \to \{0,1\}$ be a function with $\mathit{rank}(M_f) = r$. Then $D(f) \leq r + 1$.*

Proof. We may assume by rearranging the rows, that the first r rows of M_f span the other rows in the matrix. This means that the first r elements in a column determine the column. This gives the following protocol: Bob sends these r bits, at which point Alice can determine the column and compute $f(x,y)$ and send it. □

We are currently very far from settling the log rank conjecture. The best known bounds are:

- **Lower bounds:** There exists f with $\text{rank}(M_f) = r$ and $D(f) = \Omega(\log^2 r)$ [2].
- **Upper bounds:** For any f with $\text{rank}(M_f) = r$ we have $D(f) = O(\sqrt{r} \log r)$ [5].

5 Application: Time-space tradeoffs for Turing machines

We show how to get lower bounds on the time-space tradeoff for Turing machines using deterministic communication complexity. To be concrete, we consider multi-tape Turing machine with a read-only input tape and a constant number of read/write work tapes.

Theorem 5.1. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Assume that M is a multi-tape Turing machine that accepts the language*

$$L_f = \{x2^n y : x, y \in \{0,1\}^n, f(x,y) = 1\}.$$

Assume that M runs on inputs of length m in time $T(m)$ and space $S(m)$. Then

$$D(f) = O(T(3n)S(3n)).$$

Proof. Let M be such a Turing machine. We will see how to use it to simulate computing $f(x,y)$. Its input head can be in on of three regions: Alice's input x, Bob's input y, or the boundary area 2^n. Let $z = x2^n y$. Alice and Bob jointly simulate running $M(z)$ as follows. Each player simulates running $M(z)$ until the input head moves to the other player's input region (so for example, Alice simulates $M(z)$ as long as the input head is in her input's region, or the boundary region, until it crosses to Bob's input region). When such a crossover happens, the player sends the content of the work tapes and the internal state of the Turing machine to the other player, which then continues the simulation. Note that if running $M(z)$ takes T steps and uses S memory, then the control is passed from one player to the next at most T/n times, and in each case, one player sends $O(S)$ bits to the other player. So together they compute $f(x,y)$ by a deterministic protocol with cost $O(TS/n)$. □

Corollary 5.2. *Consider the language*

$$L = \{x2^n x : x \in \{0,1\}^n, n \geq 1\}.$$

Then

1. *A multi-tape Turing machine which computes L requires $T(n)S(n) = \Omega(n^2)$.*
2. *A single-tape Turing machine which computes L requires $T(n) = \Omega(n^2)$.*

Proof. The language L corresponds to $f = \text{EQ}$, which has $D(\text{EQ}) = n + 1$. This proves the bound for the multi-tape machine. The bound for single-tape machine follows as $S(n) = O(1)$ for such machines. □

6 Application: Formula depth lower bounds

Karchmer-Wigderson games [3] are a surprising connection between communication complexity and formula lower bounds.

A *formula* is a way to compute a boolean function. It is defined by a binary tree, whose leaves are labeled by variables or their negations, and whose internal nodes are labeled by either OR or AND. Given a boolean function $f : \{0,1\}^n \to \{0,1\}$, we can measure the optimal *size* and optimal *depth* of a formula that computes f.

Given f, define the following communication game, which is the Karchmer-Wigderson game associated with f. Let $X = f^{-1}(0), Y = f^{-1}(1)$. The goal of the players is to find $i \in [n]$ for which $x_i \neq y_i$. Namely, to compute the different-bit relation

$$KW_f = \{(x, y, i) : f(x) = 0, f(y) = 1, x_i \neq y_i\}.$$

Theorem 6.1. *For any boolean function f, $D(KW_f) = depth(f)$.*

To prove Theorem 6.1 we need to extend the definitions of formulas and Karchmer-Wigderson games to partial functions: formulas are only needed to compute f correctly on inputs where f is defined; and the Karchmer-Wigderson game assumes that f is defined on x, y (and $f(x) = 0, f(y) = 1$).

Theorem 6.2. *For any partial boolean function f, $D(KW_f) = depth(f)$.*

Proof. Let $f : D \to \{0,1\}$ for some $D \subset \{0,1\}^n$. We denote the inputs to f by $z \in D$, where the inputs in the KW-game are denoted $x \in f^{-1}(0), y \in f^{-1}(1)$.

We first prove that $D(KW_f) \leq depth(f)$. Assume f has a formula F of depth d which is known to Alice and Bob. By assumption, $F : \{0,1\}^n \to \{0,1\}$ and $F(z) = f(z)$ for all $z \in D$. Each node v of the formula computes a boolean function $F_v : \{0,1\}^n \to \{0,1\}$. Let r denote the root of the formula. Let u, v denote the children of the root r.

Assume first the root r is an OR gate. As $f(x) = 0, f(y) = 1$ we have

$$F_u(x) = 0 \text{ and } F_v(x) = 0, \qquad F_u(y) = 1 \text{ or } F_v(y) = 1.$$

Bob, who knows y, can find $w \in \{u, v\}$ such that $F_w(y) = 1$. He sends one bit to Alice to specify this. The players then continue to the sub-formula F_w, for which it still holds that $F_w(x) = 0, F_w(y) = 1$. The other case is similar. If the root r is an AND gate, then

$$F_u(x) = 0 \text{ or } F_v(x) = 0, \qquad F_u(y) = 1 \text{ and } F_v(y) = 1.$$

In this case, Alice can determine $w \in \{u, v\}$ for which $F_w(x) = 0$, and they continue in the same way. Finally, when they reach a leaf labeled by z_i or $\neg z_i$. Thus, it must hold that $x_i \neq y_i$, and hence they output i.

Next, we prove that $depth(f) \leq D(KW_f)$. Fix a deterministic protocol π for f with cost c. We prove by induction on c. If $c = 0$ then there must exist $i \in [n]$ for which $x_i \neq y_i$ for all $x \in f^{-1}(0), y \in f^{-1}(1)$. Thus, there must exist $b \in \{0,1\}$ such that $x_i = b, y_i = 1 - b$ for all $x \in f^{-1}(0), y \in f^{-1}(1)$, and hence $depth(f) = 0$. So assume $c \geq 1$.

Assume first that the first player to speak is Alice, which sends a message $m(x) \in \{0,1\}$. Let $X = f^{-1}(0)$ and $Y = f^{-1}(1)$. Note that $X \cup Y = D$. We can partition $X = X_0 \cup X_1$ where

$$X_b = \{x \in X : m(x) = b\} \qquad b = 0, 1.$$

Define also $D_b = X_b \cup Y$ and $f_b : D_b \to \{0,1\}$ to be the restriction of f to D_b. We have $D(KW_{f_b}) \le c - 1$, and thus by induction $\mathit{depth}(f_b) \le c - 1$. To conclude, note that

$$f(z) = f_0(z) \wedge f_1(z) \qquad \forall z \in D.$$

Thus, we can construct a formula for f of depth d, by composing the two formulas for f_0, f_1 with an AND gate at the top. The case where Bob speaks first is analogous, except that now we partition Y and compose the two formulas with an OR gate at the top. $\square$

We demonstrate the power of this technique by proving a tight $2 \log n$ formula depth lower bound for the parity of n bits. Let $PARITY : \{0,1\}^n \to \{0,1\}$ be defined as

$$PARITY(x) = \sum x_i \mod 2.$$

Claim 6.3. *Assume n is a power of two. Then $\mathit{depth}(PARITY) = 2 \log n$.*

Proof. The upper bound is via induction. We can compute the parity of two bits by a formula of depth 2. To compute the parity of n bits, compute the parity of the first $n/2$ bits, the parity of the last $n/2$ bits, and then compute their parity.

We next move to the lower bound. Let KW_{PARITY} be the associated KW game. The players have inputs $x \in \{0,1\}^n$ with even parity and $y \in \{0,1\}^n$ with odd parity. Then need to find i with $x_i \ne y_i$. Consider the following partial function f, where we further assume that $\text{dist}(x, y) = 1$, namely that there is exactly one bit where x, y differ.

Assume that Alice sends to Bob c_1 bits. Bob, given his input y and these bits, should be able to compute the unique index i where $x_i \ne y_i$. Consider a fixed value of y. Alice's input is $x = y \oplus e_i$ where $i \in [n]$ is arbitrary. However, the number of outputs of Bob is at most 2^{c_1}. So we must have $c_1 \ge \log n$. Similarly, Bob must send to Alice at least $c_2 \ge \log n$ bits. So together they need to send at least $c_1 + c_2 \ge 2 \log n$ bits. $\square$

Open problem 6.4. *Prove super-logarithmic lower bounds on formula depth of any explicit function f. Equivalently, prove a super-logarithmic lower bound on $D(KW_f)$.*

6.1 Monotone Karchmer-Wigderson games

A function $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if $f(x) \le f(y)$ whenever $x \le y$ in the partial order of sets. Equivalently, the set $f^{-1}(0)$ is a down-set. A *monotone formula* is a formula where the leaves are labelled by variables (no negations are allowed). Monotone formulas compute monotone functions. Given a monotone function f define by $\mathit{mondepth}(f)$ the minimal depth of a monotone formula computing it.

Corresponding to monotone formula depth are monotone Karchmer-Wigderson games. The setup is similar. Given a monotone function $f : \{0,1\}^n \to \{0,1\}$, the players get inputs $x \in f^{-1}(0), y \in f^{-1}(1)$. The difference is that now they need to output an index $i \in [n]$ for which $x_i = 0, y_i = 1$. We denote this problem as $monKW(f)$.

We can adapt the proof of Theorem 6.1 and prove the following.

Theorem 6.5. *For any monotone boolean function f, $D(monKW_f) = mondepth(f)$.*

Surprisingly, there are techniques to prove super-logarithmic lower bounds on the monotone depth of Karchmer-Wigderson games! Karchmer-Wigderson [3] used this to prove that checking if a graph contains a long path requires monotone depth $\Omega(\log^2 n)$. Raz-Wigderson [6] optimized this technique, and showed that checking if a bipartite graph contains a bipartite matching requires monotone depth $\Omega(n)$.

References

[1] M. Göös. Lower bounds for clique vs. independent set. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1066–1076. IEEE, 2015.

[2] M. Göös, T. Pitassi, and T. Watson. Deterministic communication vs. partition number. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1077–1088. IEEE, 2015.

[3] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, 1990.

[4] L. Lovász and M. Saks. Lattices, mobius functions and communications complexity. In *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pages 81–90. IEEE, 1988.

[5] S. Lovett. Communication is bounded by root of rank. *Journal of the ACM (JACM)*, 63(1):1, 2016.

[6] R. Raz and A. Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM (JACM)*, 39(3):736–744, 1992.

[7] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991.

[8] A. C.-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, pages 222–227. IEEE, 1977.