1. Introduction to communication complexity

Lecturer: Prahladh Harsha *Scribe: Sajin Koroth*

1.1 Yao's two-party communication model

The model consists of two parties [Yao79], *Alice* and *Bob*, holding inputs $x \in X$ and $y \in Y$ respectively. They exchange messages in order to compute a function $f : X \times Y \to \{0,1\}$ at (x, y); their goal is to do this with minimal amount of *interaction*, which is some measure of the communication between the two parties and will usually be the total number of bits exchanged by the parties. Figure 1 illustrates this model. The communication between the two parties is guided by a **protocol**, which specifies how each message sent depends on the input and the messages received previously. In the definition below, we assume that Alice (who has x) sends the first message.

Definition 1.1 (Protocol)**. Messages:** *A k-round protocol π in the two-party communication model is a sequence of functions $\langle M_1, M_2, M_3, \ldots, M_k \rangle$ where*

$$M_i : X \times (\{0,1\}^+)^{i-1} \to \{0,1\}^+ \quad \text{if } i \text{ is odd;}$$
$$M_i : Y \times (\{0,1\}^+)^{i-1} \to \{0,1\}^+ \quad \text{if } i \text{ is even.}$$

(Later in our discussion, we will also tacitly assume that the messages sent by the two parties to be prefix free.) On input x and y, the communication under the protocol proceeds as follows. In the first round, Alice sends $M_1(x)$; in the second Bob sends $M_2(y, M_1(x))$. In general, in round $i = 1, 2, \ldots, k$, if i is odd Alice sends $M_i(x, M_1, M_2, M_3, \ldots, M_{i-1})$, and if i is even, Bob sends $M_i(y, M_1, M_2, M_3, \ldots, M_{i-1})$. At the end, a referee declares the output by looking at the transcript alone.

Correctness: *The protocol is said to compute $f : X \times Y \to \{0,1\}$ if on all inputs $(x, y) \in X \times Y$, the output is $f(x, y)$.*

Transcript: *For a protocol π and an input (x, y), the **transcript** $\pi(x, y)$ is the sequence of messages $\langle M_1(x), M_2(y, M_1), \ldots, M_k(\cdots) \rangle$ exchanged by the two parties. The total number of bits exchanged by Alice and Bob, which we refer to as the length of the transcript, is $|\pi(x, y)| = \sum_{i=1}^{k} |M_i(\cdots)|$.*

1.1.1 The protocol tree

It will often be convenient to visualize the protocol as a binary decision tree. This tree is constructed as follows. Each internal node is labelled by a function of the form $f_A : X \to \{0,1\}$ or $g_B : Y \to \{0,1\}$. The leaves are labelled 0 or 1. The computation on this tree starts at the root, and proceeds down, as in decision trees, to a leaf. When at an internal

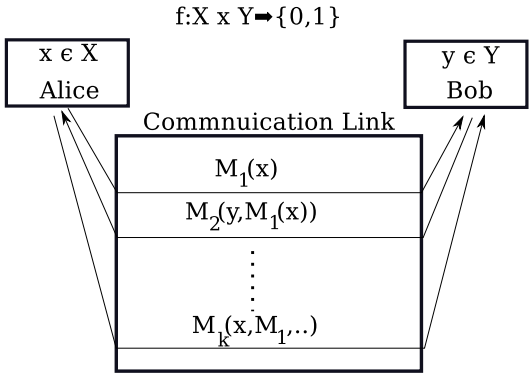

Figure 1: Yao's two party model of communication

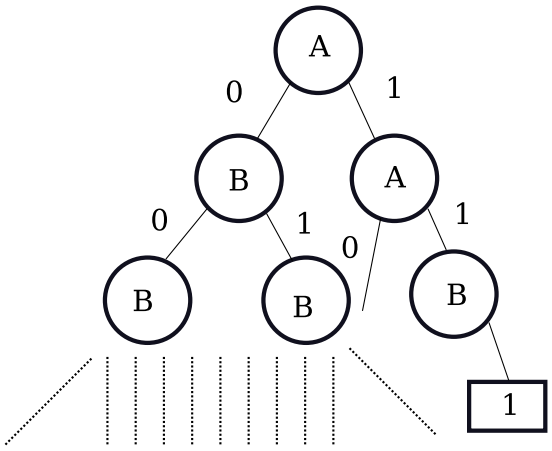

Figure 2: An example protocol tree

node v, the function labeling v is computed (by Alice if it is a function of x and by Bob if it is a function of y). Based on the result, the computation moves to the left or right child. The output of the computation is the label stored at the leaves. It is straightforward to translate a protocol as defined above into a tree, and also turn any such tree into a protocol. In particular, the transcript corresponds to the concatenation of function values computed at the intermediate internal nodes; $|\pi(x, y)|$ is the length of the path from the root to leaf reached at the end. Figure 1.1.1 illustrates an example protocol tree.

1.2 Communication complexity

The communication complexity of a function is a measure of its hardness when the inputs are distributed among Alice and Bob. This measure focusses only on the cost associated with exchanging the information and disregards the computational effort Alice and Bob need to invest in computing their messages.

Definition 1.2 (Deterministic communication complexity)**.** *The deterministic communi-*

cation complexity of a function $f : X \times Y \to \{0,1\}$ *is*

$$\mathrm{D}(f) = \min_{\pi} \max_{(x,y)} |\pi(x,y)|,$$

where π *ranges over all protocols that computer* f*, and* (x,y) *ranges over* $X \times Y$*.*

Equality: The n-bit *equality* function, $\mathsf{EQ}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, takes the value 1 on input (x,y) iff $x = y$.

It is easy to see that $\mathrm{D}(\mathsf{EQ}_n) \leq n+1$: Alice sends x over to Bob, who then tell Alice the result. In fact, all functions on n-bit inputs can be computed in this manner in two rounds with $n+1$ bits of communication. Can we do significantly better for EQ_n?

Theorem 1.3. $\mathrm{D}(\mathsf{EQ}_n) \geq n$

Proof. Fix a protocol for EQ_n and consider its protocol tree.

Claim 1.4. *The tree has at least* 2^n *leaves.*

Note that the theorem follows immediately from this claim, because a binary tree with at least 2^n leaves must have depth at least n.

To justify the claim, we will show a set of 2^n inputs, on no two of which can the computation reach the same leaf. Consider,

$$F = \{(x,x) : \{0,1\}^n\}.$$

Every input in F leads to a leaf labelled 1. The key observation is that if (x_1,y_1) and (x_2,y_2) lead to a leaf ℓ, then (x_1,y_2) and (x_2,y_1) also lead to ℓ. This is because the function evaluated at each node in the protocol tree depends on one input (either x or y) at a time. Now, if $x \neq y$, the input (x,y) must lead to a leaf labelled 0. Thus, if $x_1 \neq x_2$, then (x_1,x_1) and (x_2,x_2) lead to different leaves. This justifies the claim. □

Disjointness: The function $\mathsf{DISJ}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined as follows. We view the input strings x and y as characteristic vectors of sets X and Y. The function $\mathsf{DISJ}_n(x,y) = 1$ iff $X \cap Y = \emptyset$.

Theorem 1.5. $\mathrm{D}(\mathsf{DISJ}_n) \geq n$.

Proof. The proof is similar to the one above. We will show that the protocol tree has at least 2^n leaves by showing 2^n inputs all of which must lead to distinct leaves. Consider

$$F = \{(x,\overline{x}) : x \in \{0,1\}^n\},$$

where $\overline{x}$ is the bit-wise complement of x. Note that if x is the characteristic vector of x, then $\overline{x}$ is the characteristic vector of the complement of X, which we denote by $\overline{X}$. We claim that no two elements of F lead to the same leaf. Let $(x_1,\overline{x}_1)$ and $(x_2,\overline{x}_2)$ both lead to the same leaf ℓ. Clearly, ℓ must be labelled 1. Then, as argued above $(x_1,\overline{x}_2)$ and $(x_2,\overline{x}_1)$ lead also lead to ℓ. Let x_1 be the characteristic vector of X_1 and x_2 be the characteristic vector of X_2. Then $X_1 \cap \overline{X}_2 = \emptyset$, implying $X_1 \subseteq X_2$. Similarly, $X_2 \subseteq X_1$. Thus, $X_1 = X_2$, implying $x_1 = x_2$. It follows that distinct elements of F lead to distinct leaves. □

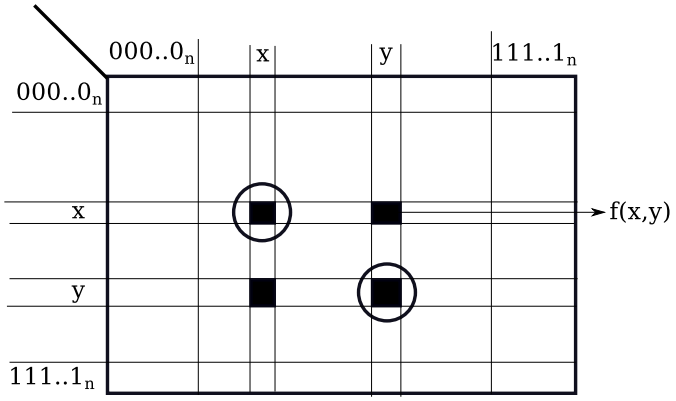

Figure 3: Viewing $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ as a $2^n \times 2^n$ matrix

Decomposition into monochromatic rectangles: The argument used above is rather general and can be applied to other settings. Associate with the function f a $2^n \times 2^n$ truth-table matrix, M_f where $M_f[x,y] = f(x,y)$ (see Figure 1.2). Fix a protocol π for f. Observe that if the transcripts $\pi(x,y)$ and $\pi(x',y')$ are the same, say T then the transcript $\pi(x,y')$ and $\pi(x',y)$ ought to equal T. Hence for each transcript T of π (corresponding to a leaf ℓ in the protocol tree), the inputs $R_T = \{(x,y)|\pi(x,y) = T\}$ form a monochromatic(the function value is same throughout the points in the rectangle) combinatorial rectangle $R_T = A_T \times B_T$ where

$$\begin{aligned} A_T &= \{x|\exists y, \pi(x,y) = T\} \\ B_T &= \{y|\exists x, \pi(x,y) = T\}\,. \end{aligned}$$

Hence if at least t monochromatic combinatorial rectangles are needed to tile the truth-table matrix M_f, then the protocol must produce at least t distinct transcripts. The corresponding protocol tree must then have at least t distinct leaves, and depth at least $\log t$.

1.3 Randomized Communication Complexity

We have seen that both EQ_n and DISJ_n has a very high deterministic communication complexity. Can randomness help? In randomized communication protocols, the two parties in addition to their inputs can use a random string to determine their messages. There are two models of randomized communication complexity, based on whether or not the string R_A used by Alice and the string R_B used by Bob are the same.

Public coin model: In this model R_A and R_B, and we refer to them by just R. We may think of R being generated by shared coin tosses whose outcomes are visible to both parties.

Private coin model: In this mode R_A and R_B are generated independently, and neither Alice nor Bob is aware of the outcome of the other's coin tosses.

This changes our model as follows. Alice and Bob receive their inputs, x and y; they also receive their random strings R_A and R_B (identical or independently generated). Then, in

any computation where Alice was to use x, she may now use (x, R_A); similarly, Bob's input y is now replaced by (y, R_B). In particular, the transcript is now a function of (x, y, R_A, R_B); the referee who declares the output based on the transcript is allowed access to R_A and R_B. Furthermore, we say that a protocol computes a function f, if for all x and y

$$\Pr_{R_A,R_B}[\pi(x, y; R) \text{ computes } f \text{ correctly}] \geq \frac{3}{4}$$

Note that $\frac{3}{4}$ was selected arbitrarily, any constant greater than $\frac{1}{2}$ would lead to a similar randomized complexity measure. Also note that the coins R, R_A and R_B could follow any distribution fixed by the protocol as we allow Alice and Bob to be computationally "unbounded"; if their distribution is left unspecified, we will assume that these strings are chosen uniformly from their domain, often of the form $\{0, 1\}^r$.

Definition 1.6 (Randomized communication complexity)**.** *We have two measures based on the model for randomness.*

Public coin protocols: $\mathrm{R}^{pub}_{1/4}(f) = \min_{\pi} \max_{(x,y,R)} |\pi(x, y; R)|$*, where* π *ranges over public coin protocols for* f *with shared random string* R*.*

Private coin protocols: $\mathrm{R}^{prv}_{1/4}(f) = \min_{\pi} \max_{(x,y,R)} |\pi(x, y; R_A, R_B)|$*, where* π *ranges over private coin protocols for* f *with respective random strings* R_A *and* R_B *for Alice and Bob.*

Let us re-examine the communication complexity of EQ_n and DISJ_n in this model.

Theorem 1.7. $\mathrm{R}^{pub}_{1/4}(\mathsf{EQ}_n) = O(1)$.

Proof. The proof is constructive. We will provide a randomized protocol for computing EQ_n. The protocol is as follows:

1. The public coin is tossed n times to get a bit string of length n, say r, which is available to both Alice and Bob.
2. Alice sends $\sum_{i=1}^{n} x_i r_i \bmod 2$ to Bob. and sends the
3. Bob sends back $\sum_{i=1}^{n} y_i r_i \bmod 2$. The output is 1 if Alice's and Bob's bits are the same and is 0 otherwise.

If $x = y$, then the protocol makes no error. If $x \neq y$, then the protocol errs if $\sum_{i=1}^{n} x_i r_i = \sum_{i=1}^{n} y_i r_i \pmod 2$, that is, if $\sum_{i=1}^{n}(x_i - y_i) r_i = 0 \pmod 2$. Since r is chosen at random, this can happen with probability at most $\frac{1}{2}$. Running the protocol for two different r's would ensure that the probability of error is at most $(\frac{1}{2})^2 = \frac{1}{4}$.

Clearly, only four bits are exchanged. □

Theorem 1.8. $\mathrm{R}^{prv}_{1/4}(\mathsf{EQ}_n) = O(\log n)$

Proof. Alice and Bob use the following matrix.

$$\begin{pmatrix} 1^0 & 1^1 & 1^2 & \cdots & 1^{n-1} \\ 2^0 & 2^1 & 2^2 & \cdots & 2^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ L^0 & L^1 & L^2 & \cdots & L^{n-1} \end{pmatrix},$$

where $L = 4n$, and they also agree in advance on a prime number p such that $L \leq p \leq 2L$. Note that no two rows of the matrix are equal modulo p. The protocol is the following

1. Alice picks a random number $i, 1 \leq i \leq n$ and selects the i-th row $(i^0, i^1, i^2, \ldots, i^{n-1})$ from the matrix.

2. Alice then sends Bob the value i and the dot product of the i-th row with x, that is, $\sum_{j=0}^{n-1} x_j i^j \bmod p$.

3. Bob computes $\sum_{j=0}^{n-1} x_j i^j \bmod p$ and tells Alice if it agrees with what Alice sent. The bit sent by Bob is the outcome of the protocol.

When $x = y$, there can be no error. Suppose $x \neq y$. We need to estimate the probability that

$$\sum_{j=0}^{n-1} x_j i^j = \sum_{j=0}^{n-1} x_j i^j \pmod{p}$$

for i chosen uniformly at random from $[L]$, that is, i is the root of the polynomial $P(z) = \sum_{j=0}^{n-1}(x_j - y_j)z^j$ over $\mathbb{F}_p$. Now, $P(z)$ is a non-zero polynomial (for at least one value of j we have $x_j - y_j \neq 0 \pmod{p}$) of degree at most $n-1$. It can have at most $n-1$ roots in $\mathbb{F}_p$. Hence,

$$\Pr[\text{Error}] = \Pr_{z \in [L]}[P(z) = 0] \leq \frac{n-1}{4n} \leq \frac{1}{4}$$

Note that Alice sends $O(\log n)$ bits to Bob and Bob sends one bit to Alice. □

Exercise: $\mathrm{R}^{\mathrm{prv}}_{1/4}(\mathsf{EQ}_n) = \Omega(\log n)$.

1.3.1 Randomized protocols for DISJ_n

We saw earlier that both EQ_n and DISJ_n have deterministic communication complexity of $\Omega(n)$, and we just saw that randomness (with a little error) helps for EQ_n. The following deep result of Kalyanasundaram and Schnitger shows that such savings are not possible for the disjointness function.

Theorem 1.9 (Kalyanasundaram-Schnitger [KS92], Razborov [Raz92])**.**

$$\mathrm{R}^{pub}_{1/4}(\mathsf{DISJ}_n) = \Theta(n).$$

We will prove this theorem later in the course.

References

[KS92] Bala Kalyanasundaram and Georg Schnitger. *The probabilistic communication complexity of set intersection.* SIAM J. Discrete Math., 5(4):545–557, 1992. (Preliminary version in *2nd Structure in Complexity Theory Conference*, 1987). doi:10.1137/0405044.

[Raz92] Alexander A. Razborov. *On the distributional complexity of disjointness.* Theoretical Comp. Science, 106(2):385–390, 1992. doi:10.1016/0304-3975(92)90260-M.

[Yao79] Andrew Chi-Chih Yao. *Some complexity questions related to distributive computing (preliminary report).* In *Proc. 11th ACM Symp. on Theory of Computing (STOC)*, pages 209–213. 1979. doi:10.1145/800135.804414.