



# **Introduction to Communication Complexity**

Maverick Woo

# Credit

Bernard Chazelle

- The Discrepancy Method, Chapter 10

Eyal Kushilevitz and Noam Nisan

- Communication Complexity

# Target Audience

This time I am very ambitious:

- you are supposed to understand every single slide,
- so there will be no advanced material.

# Communication Complexity

Started by Yao, STOC 1979

Some Complexity Questions Related To  
Distributed Computing

## Goal:

Isolate computational bottlenecks of certain problems and provide tools for resolving their complexity

# Yao's Model

- Only two parties
- Each party get a fixed part of the input
- Only concerned about communication
- Both party knows what to compute beforehand

More advanced models are out of our scope.



# The Classics

# Business Practice

We will only introduce formalism and lingo when they are needed.

For all the abstract notations, go see K&N.

# Inner Product Modulo Two

Alice and Bob (*again!?*) wants to compute

$$\langle x, y \rangle \stackrel{\text{def}}{=} \sum_{i=1}^n x_i y_i \pmod{2}.$$

(In this talk, all strings are binary and have length  $n$ .)

Alice knows  $x$  but not  $y$  and vice versa for Bob.

- How many bits must they exchange to know  $\langle x, y \rangle$  with full confidence?

# A Simple Solution

$(n + 1)$  bits suffice

# A Simple Solution

$(n + 1)$  bits suffice

- Alice sends Bob  $x$  .....  $n$  bits
- Bob sends Alice  $\langle x, y \rangle$  ..... 1 bit

Can we do better?

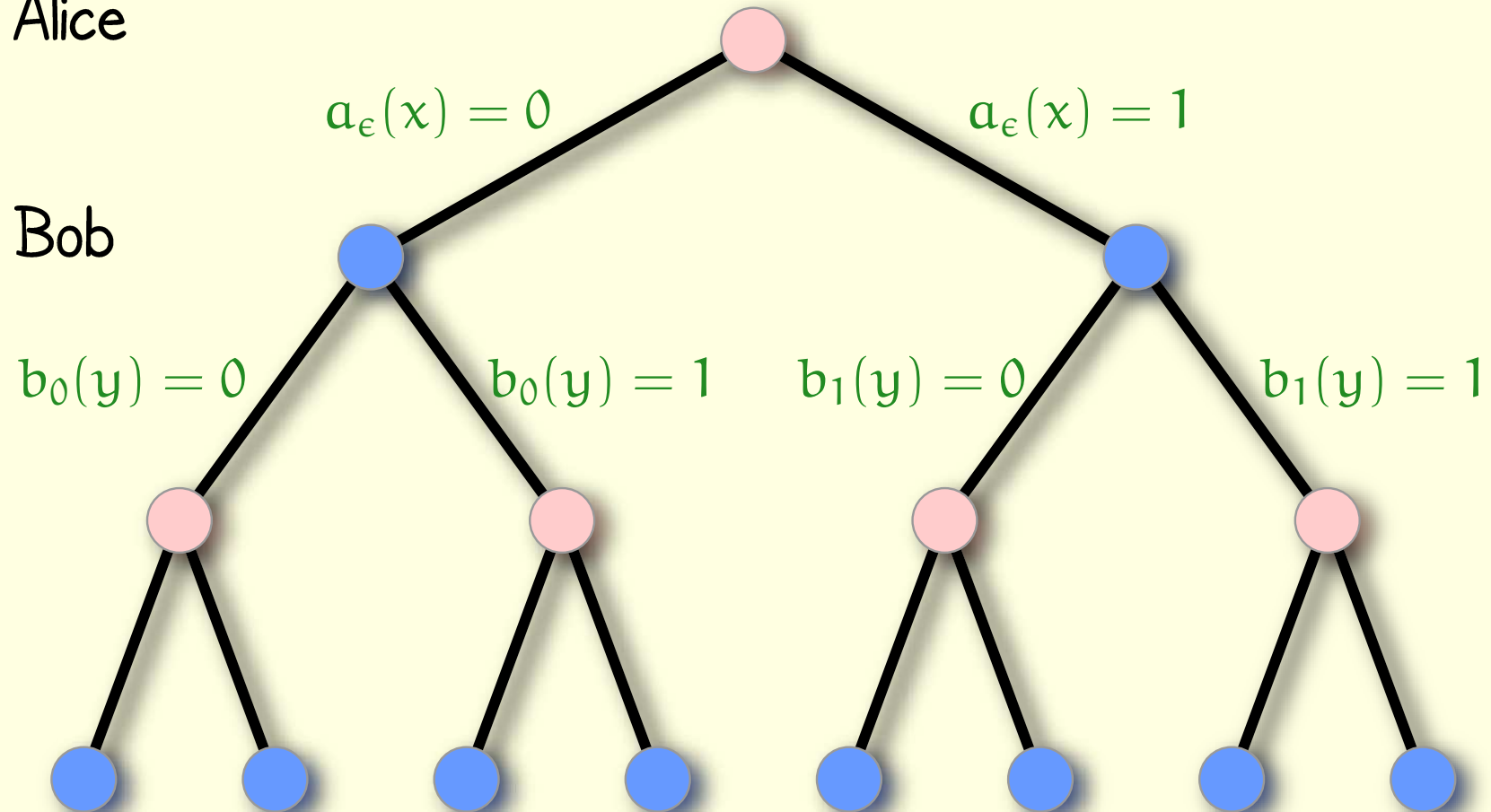
We need a precise model to prove lower bounds.

# Protocols

A protocol is a binary tree.

Alice

Bob



# Protocols

Some fine-prints:

- Each leaf is labelled either 0 or 1.
- A protocol is correct if for any  $x, y$ , the protocol arrives at a leaf labelled  $\langle x, y \rangle$  (or  $f(x, y)$  in general).
- The cost of the protocol is the length of its longest path. (Internal computation at each side is free.)

# Protocols

Some more fine-prints:

- It is not necessary for the two parties to strictly alternate.
- The tree is full but not necessarily complete.
- Different input pairs can arrive at the same leaf (thus via the same path).

# Randomized Protocols

Fix some  $\epsilon > 0$  and assume that  $x, y$  are independently drawn UAR from  $\{0, 1\}^n$ .

Let  $P$  be the property that a protocol computes  $f(n)$  for at least  $(\frac{1}{2} + \epsilon)$  of the time.

Define  $C_\epsilon(n)$  to be the minimum cost of all protocols satisfying  $P$ . This is the distributional communication complexity of  $f(n)$ .

# Lingo: Rectangle

If  $X$  and  $Y$  are subsets of  $\{0, 1\}^n$ , then  $X \times Y$  is called a (combinatorial) rectangle.

Each node in a protocol splits a rectangle into two. Each leaf  $l$  corresponds to the rectangle  $R_l$  and

$$\bigcup_l R_l = \{0, 1\}^n \times \{0, 1\}^n.$$

# Discrepancy

Define

$$H(x, y) = \begin{cases} 1 & \text{if } \langle x, y \rangle = 1 \\ -1 & \text{if } \langle x, y \rangle = 0. \end{cases}$$

Let  $D(X \times Y) = \left| \sum_{x \in X, y \in Y} H(x, y) \right|$  be the discrepancy of a rectangle.

The discrepancy  $D(n)$  of a protocol is the maximum discrepancy over all leaves.

# The Discrepancy Connection

**Lemma** For any  $0 < \epsilon \leq \frac{1}{2}$ ,

$$C_\epsilon(n) \geq 2n + \log(2\epsilon) - \log(D(n)).$$

**Proof** By definition of  $D(n)$ , within each leaf the difference between the number of pairs where the protocol succeeds and the number of those where it fails is bounded by  $D(n)$ .

# The Discrepancy Connection

**Lemma** For any  $0 < \epsilon \leq \frac{1}{2}$ ,

$$C_\epsilon(n) \geq 2n + \log(2\epsilon) - \log(D(n)).$$

**Proof** By definition of  $D(n)$ , within each leaf the difference between the number of pairs where the protocol succeeds and the number of those where it fails is bounded by  $D(n)$ . There are at most  $2^{C_\epsilon(n)}$  leaves. Summing over all leaves, we have an upper bound on the total difference,  $2^{C_\epsilon(n)} \times D(n)$ . (...)

# Proof Continued

(...) Now for a lower bound, the protocol is correct on at least  $(\frac{1}{2} + \epsilon)$  fraction of the possible inputs, and is incorrect on at most  $(\frac{1}{2} - \epsilon)$  fraction. So the difference must be at least  $2\epsilon$  fraction of the possible inputs, i.e.,  $2\epsilon(2^n \times 2^n)$ . Thus, we have

$$2\epsilon 2^{2n} \leq 2^{C_\epsilon(n)} \times D(n),$$

or

$$C_\epsilon(n) \geq 2n + \log(2\epsilon) - \log(D(n)). \blacksquare$$

# Bounding $D(n)$ for $\langle x, y \rangle$

**Theorem** For any  $0 < \epsilon \leq \frac{1}{2}$ , the distributional communication complexity of the inner product modulo 2 satisfies

$$C_\epsilon(n) \geq \frac{n}{2} + 1 - \log \frac{1}{\epsilon}.$$

**Proof** The  $(2^n \times 2^n)$  matrix  $M = (H(x, y))_{x, y}$  is the Hadamard matrix. (...)

# Proof Continued

(...) From linear algebra, we know  $M^T M = 2^n I$  and so for any vector  $v$ ,

$$\|Mv\|_2 = 2^{n/2} \|v\|_2.$$

Given a rectangle  $X \times Y$ , let  $v$  and  $w$  be the characteristic vectors of  $X$  and  $Y$  in  $\{0, 1\}^{2^n}$ . (...)

# Proof Continued

(...) By Cauchy-Schwarz,

$$\begin{aligned} D(X \times Y) &= |v^T H w| \\ &\leq \|v\|_2 \|H w\|_2 \\ &= 2^{n/2} \|v\|_2 \|w\|_2 \\ &\leq 2^{n/2} \sqrt{2^n} \sqrt{2^n}. \end{aligned}$$

Therefore  $C_\epsilon(n) \geq 2n + \log(2\epsilon) - 3n/2$ , which is

$$\frac{n}{2} + 1 - \log \frac{1}{\epsilon}. \blacksquare$$

# Deterministic Lower Bound

We use this to illustrate the Rank Method.

Let  $M = (\langle x, y \rangle)_{x,y}$ . Each leaf of the protocol is associated with a rectangle of maximum discrepancy (only 0's or only 1's).

Let  $l$  be a 1-leaf and let  $M_l$  be the matrix obtained by zeroing out everything outside of the rectangle. Thus,

$$M = \sum_l M_l.$$

# Rank Method

Over the reals, each  $M_l$  has rank 1. From linear algebra,

$$\text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B).$$

Thus,

$$\text{rank}(M) \leq \sum_l \text{rank}(M_l) \leq \text{number of 1-leaves}.$$

By considering the 0-leaves instead, we have another matrix  $M'$  and

$$\text{rank}(M') \leq \text{number of 0-leaves}.$$

# Rank Method Continued

Notice that  $M + M' = U$ , the rank-1 matrix with 1 everywhere. Thus,

$$\text{rank}(M) \leq \text{rank}(-M') + \text{rank}(U)$$

and now

$$\text{rank}(M') \geq \text{rank}(M) - 1.$$

Since a protocol is a binary tree, its cost is at least  $\log(2 \text{rank}(M) - 1)$ . But how do we use this?

# Computing $\text{rank}(M)$ for $\langle x, y \rangle$

Let  $P = M^2$ , i.e.,

$$P(x, y) = \sum_z \langle x, z \rangle \langle z, y \rangle.$$

There are 3 cases.

- If either  $x$  or  $y$  is  $0$ , then  $P(x, y) = 0$ .
- If  $x = y \neq 0$ , then half of the  $z$ 's will give  $\langle x, z \rangle = 1$ . Thus,  $P(x, y) = 2^{n-1}$ .
- If  $x \neq y \neq 0$ , then  $P(x, y) = 2^{n-2}$ .

# Ta! Da!

Stripping the non-interesting first row and first column,  $P$  has  $2^{n-1}$  on the diagonal and  $2^{n-2}$  elsewhere. This is a circulant matrix and has non-zero determinant. Thus,  $\text{rank}(P) = 2^n - 1$  and  $\text{rank}(M) \geq 2^n - 1$ .

By the Rank Method, we have

$$C(n) \geq \log(2^{n+1} - 3)$$

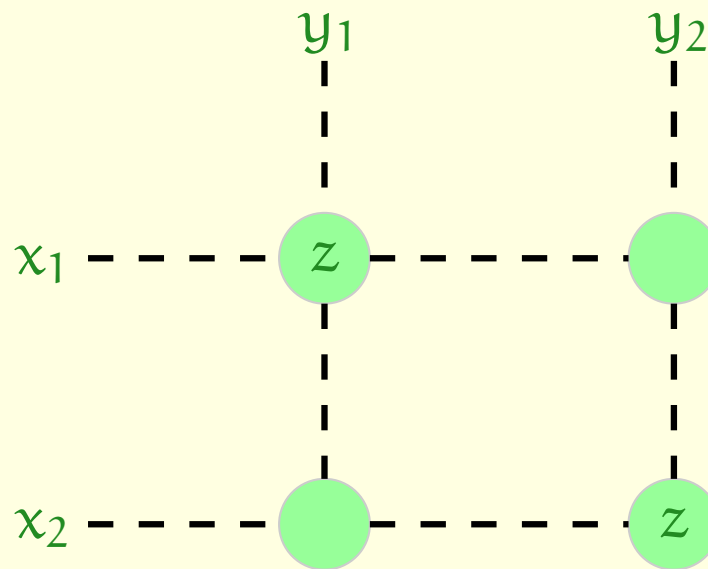
. Since  $C(n) \leq n + 1$ , we have  $C(n) = n + 1$  (exactly).

# Back to Rectangles

Here is an alternative definition:

$R \subseteq X \times Y$  is a mono-chromatic rectangle iff

$$(x_1, y_1) \in R \wedge (x_2, y_2) \in R \Rightarrow (x_1, y_2) \in R$$



# Copy and Paste Argument

It is crucial that Alice doesn't know  $y$  and Bob doesn't know  $x$ .

Copy: the way Alice behaves out of the communication on  $(x_1, y_1)$  and the way Bob behaves out of the communication on  $(x_2, y_2)$

Paste: interleave together to get the communication of  $(x_1, y_2)$

# Fooling Set

Consider  $(x_1, y_1)$  and  $(x_2, y_2)$ . Suppose  $f$  has the same value on them, say  $z$ .

If we can show that the value of  $f$  is *not*  $z$  in at least one of  $(x_1, y_2)$  and  $(x_2, y_1)$ , then  $(x_1, y_1)$  and  $(x_2, y_2)$  cannot possibly be in the same leaf.

- Hence a lower bound on the number of leaves

# Checking Equality

Alice and Bob each hold an  $n$ -bit string. How to check if they are equal?

As always, we can send one string over and send the result back. Thus,  $n + 1$  bits suffice.

- Can we do better?

# Using Fooling Set

Consider this set of size  $2^n$ :

$$S = \{(\alpha, \alpha) \mid \alpha \in \{0, 1\}^n\}.$$

Verify that this is a fooling set.

# Using Fooling Set

Consider this set of size  $2^n$ :

$$S = \{(\alpha, \alpha) \mid \alpha \in \{0, 1\}^n\}.$$

Verify that this is a fooling set.

Therefore we have at least  $2^n$  1-leaves. But then, we still need to have a 0-leaf, thus we need at least  $n + 1$  bits to distinguish them all.

Since we already know an  $(n + 1)$ -bit solution, this bound is tight.

# A Little Remark

How about the Rank Method?

Define  $M = (x == y)_{x,y}$ . This matrix has full rank.

The rank lower bound

$$\log(2 \times 2^n - 1)$$

follows immediately.

# Exercises

By inspection, tell me the communication complexity of  
 $f(x, y) = x > y$ .

# Exercises

By inspection, tell me the communication complexity of  
 $f(x, y) = x > y$ .

- $n + 1$

# Exercises

By inspection, tell me the communication complexity of  $f(x, y) = x > y$ .

■  $n + 1$

How about disjointness  $f(x, y) = (x \cap y == \phi)$ ?

Assume  $x, y$  are characteristic vectors from the universe  $\{1, 2, \dots, n\}$ .

# Exercises

By inspection, tell me the communication complexity of  $f(x, y) = x > y$ .

■  $n + 1$

How about disjointness  $f(x, y) = (x \cap y == \phi)$ ?

Assume  $x, y$  are characteristic vectors from the universe  $\{1, 2, \dots, n\}$ .

■  $n + 1$



# Hot Topic: Searching in a Finite Universe

# The Database Game

Alice (the querier) has a set  $L$  of potential queries and Bob (the responder) has a set  $P$  of potential key-sets.

An instance is  $(l, p) \in L \times P$  with one or multiple answers.

# An Example

## Predecessor Searching

- $l$  is an integer
- $p$  is a collection of integers
- The solution is the largest element in  $p$  that does not exceed  $l$ .

# Another Example

## Half-plane Range Detection

- $l$  is a line in  $\mathbb{R}^2$
- $p$  is a set of points on the plane
- The solution is  $0$  if all points are on one side of the line and  $1$  otherwise.

# Yet Another Example

Approximate Nearest Neighbor On The Hamming Cube

- $\mathbf{l}$  is a point on the cube  $\{0, 1\}^d$
- $\mathbf{p}$  is a collection of points on that cube
- The solution is any point of  $\mathbf{p}$  whose  $L^1$  distance to  $\mathbf{l}$  does not exceed  $\delta \geq 1$  times the shortest distance between  $\mathbf{l}$  and any point in  $\mathbf{p}$ .

# Cell Probe Model

Consider a table  $T$  of  $n^c$  cells, each  $w$ -bits wide, where  $n = |p|$ ,  $c$  is a fixed arbitrarily large constant and  $w$  is a problem-dependent parameter.

An algorithm in this model has two parts:

- A table assignment strategy: given  $p$ , how would you like  $T$  to be filled?
- An infinite sequence of functions  $f_1, f_2, \dots$

# Algorithm Execution

Presented with a query  $\mathcal{L}$ , the algorithm will:

1. Evaluates the index  $f_1(\mathcal{L})$ .
2. Lookup  $T[f_1(\mathcal{L})]$ .
3. If  $T[f_1(\mathcal{L})]$  provides a solution, the algorithm terminates.
4. Otherwise, evaluate the index  $f_2(\mathcal{L}, T[f_1(\mathcal{L})])$ .
5. Lookup  $T[f_2(\mathcal{L}, T[f_1(\mathcal{L})])]$ .
6. etc.

# Lower Bounds

Given an algorithm for Predecessor Searching, there exist a key-set and a query that require  $\Omega(\log b / \log \log b)$  probes to answer, where  $b$  is the number of bits needed to encode the integers used to specify the query and the keys. This holds for any word size  $w = b^{O(1)}$ .

# Lower Bounds

Given an algorithm for Half-plane Range Detection, there exist a key-set and a query line that require  $\Omega(\log b / \log \log b)$  probes to answer, where  $b$  is the number of bits needed to encode the rational coefficients of the line and the coordinates in the point set. This holds for any word size  $w = b^{O(1)}$ .

# Lower Bounds

Given an algorithm for Approximate Nearest Neighbor Searching On The Hamming Cube, there exist a key-set and a query that require  $\Omega(\log \log d / \log \log \log d)$  probes to answer, where  $d$  is the dimension of the cube. This holds for any approximation factor  $\delta < 2^{(\log d)^{1-\epsilon}}$ , with any fixed  $\epsilon$  and for any word size  $w = d^{O(1)}$ .

# Take Home Exercise



# The End