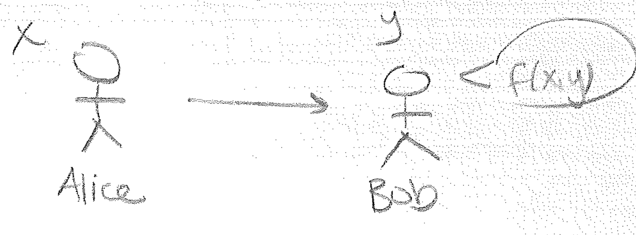
1

Many-Round Communication Complexity

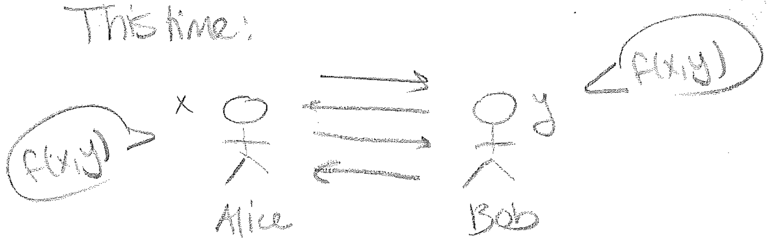
Agenda

1. Many Round CC: defs
2. Karchmer–Wigderson games + circuit applications
3. Techniques + Examples
   - tiling
   - fooling sets
   - rank lower bd
4. Open problems
   - log rank conjecture
   - direct sum conjecture.
5. ~~(If time) Intro to randomized many-round CC.~~

Last time: one-way:



This time:



Formally (somewhat), a protocol Π (deterministically) specifies the communication:

- who speaks next
- when to stop
- what to say next
- what to output

Π solves a relation R if $\forall x,y$, $(x,y,\Pi(x,y)) \in R$ (or $f: X \times Y \to Z$ if $\forall x,y$, $\Pi(x,y) = f(x,y)$)

The (deterministic) communication complexity $D(\Pi)$ is how many bits are transmitted by Π (in the worst case).

The communication complexity of R, $D(R)$ is

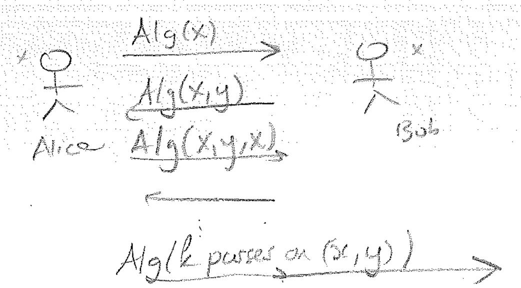
$$D(R) = \min_{\Pi \text{ solving } R} D(\Pi).$$

(2) Why study many rounds?

- Lower bounds for multiple-pass streaming algs

  eg, we'll see DISJ needs $\Omega(n)$ in many-round CC (randomized or deterministic)

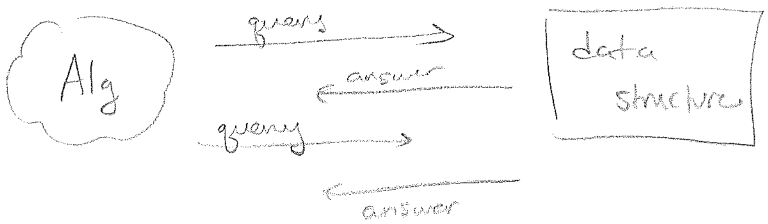
  This implies that a k-pass protocol for F_∞ needs $\Omega(n/k)$ space:



  if this appx F_∞ too well, then Bob knows whether x, y were disjoint

  If Alg has space $\le \frac{n}{Ck}$, then this has cost $\le \frac{2n}{C}$.

- Lower bounds for data structures in "cell probe" model.



  eg, can get lower bounds for nearest-neighbors data structures this way.

- Connection to circuits: Korchmer-Wigderson games.

  Given f, what is the circuit depth?

  Make a relation R_f so

  $(x,y,i) \in R_f \iff f(x)=f(y)$ or $f(x) \neq f(y)$ and $x_i \neq y_i$

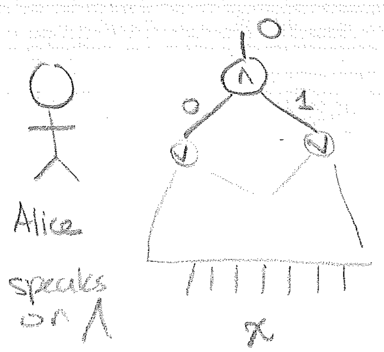
$\boxed{\text{Thm}}$ $\forall f$, depth$(f) = D(R_f)$.

③ Proof

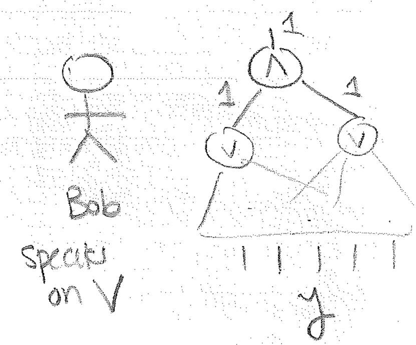
Show that $D(R_f) \le \text{depth}(f)$ by example.
Say C computes f; we'll give a protocol for R_f.

If $f(x)=f(y)$, they can output whatever they want

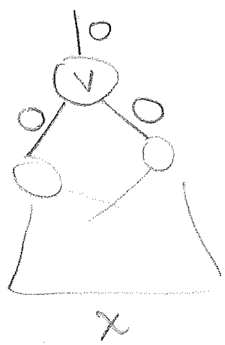
Say Alice has x and Bob has y. Say $f(x)=0$, $f(y)=1$, top gate $\wedge$.



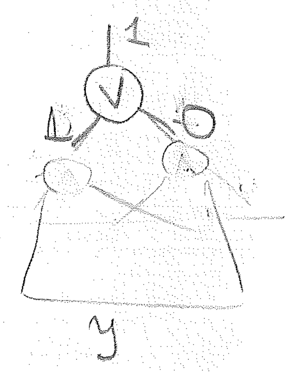
"0" ⟶
means left branch was a 0



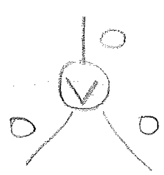
both players recurse left



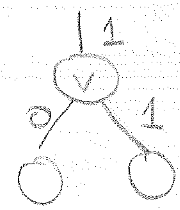
⟵ "0"
means left branch was a 1



recurse left



⟵ "1"
means right branch is a 1



etc.

eventually they find i s.t. $x_i \ne y_i$, and output that.

$\#\text{rounds} = \text{depth}(C) + O(1)$

↖ figure out that $f(x)=0$, $f(y)=1$

It's possible to reverse-engineer this and go the other way, which shows (2) $\text{depth}(f) \le D(R_f)$

④ <u>Notice:</u> It would thus be very interesting to find $f \in P$ w/ $D(R_f) \geq \log(n)\log\log(n)$

I can't do that, but here's an easy example w/ PARITY.

<u>Thm</u> depth(PARITY) $\geq 2\log(n)$.

<u>Pf</u> Say Alice has x w/ odd parity
Bob has y w/ even parity
They are tasked w/ finding i s.t. $x_i \neq y_i$.

CLAIM. Alice must send $\log_2(n)$ bits to Bob, and Bob must send $\log_2(n)$ bits to Alice.

pf. Say x has even parity, $y = x + e_j$ for some j.
So Bob needs at least $\lg(n)$ bits from Alice in order to be able to give n distinct outputs.
Same the other way. ∎

That was pretty easy. Other problems are less easy. For the rest of this lecture, we'll see some techniques.

Here are some example problems:

INDEX: $x \in \{0,1\}^n$, $y \in [n]$, $\text{INDEX}(x,y) = x_y$

EQ: $x, y \in \{0,1\}^n$, $\text{EQ}(x,y) = \mathbb{1}_{x=y}$

DISJ: $x, y \subseteq [n]$ $\text{DISJ}(x,y) = \begin{cases} 1 & x \cap y = \emptyset \\ 0 & \text{else} \end{cases}$

IP: $x, y \in \{0,1\}^n$, $\text{IP}(x,y) = \sum_{i=1}^n x_i y_i \mod 2$

Any guesses?

Last time $\vec{D}(\text{INDEX}), \vec{R}(\text{INDEX}) = \Omega(n)$

Now, $D(\text{INDEX}) = \log_2(n) + 1$. (Bob sends his index, Alice sends the answer; this is tight by pigeonhole)

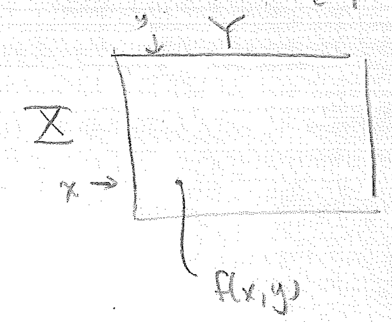
Did the other problems get easier?

(5)

Techniques: tiling bounds.

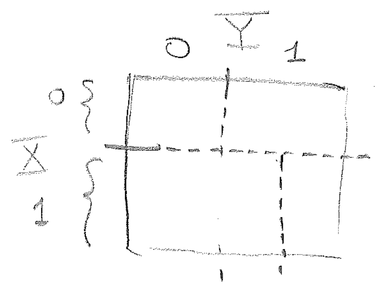
Consider a function $f: X \times Y \to \{0,1\}$ (for simplicity), and associate a matrix,

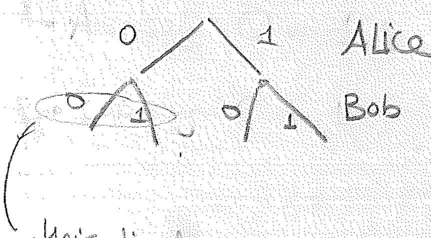
$M = M_f =$



$\in \{0,1\}^{|X| \times |Y|}$

What does a protocol do?





← this divides the set X in two, X_0, X_1

this divides $\{(x,y) : x \in X_0\}$ into $\{(x,y) : x \in X_0, y \in Y_0\}$ $\{(x,y) : x \in X_0, y \in Y_1\}$ etc.

So eventually it results in a TILING of M_f by <u>combinatorial rectangles</u>:

<u>Def</u> A <u>combinatorial rectangle</u> is a set $\{(x,y) : x \in I_1, y \in I_2\}$ for some $I_1 \subseteq X$, $I_2 \subseteq Y$.

A partition (or "tiling") of $M \in \{0,1\}^{X \times Y}$ into comb. rects. is <u>monochromatic</u> if every rectangle R in the tiling has

$$M_{xy} = 1 \quad \forall (x,y) \in R \qquad \text{or} \qquad M_{xy} = 0 \quad \forall (x,y) \in R$$

Observation: a successful protocol w/ k rounds results in a monochromatic tiling containing $\leq 2^k$ rectangles.

Turn this into a lower bound:

Suppose that every monochromatic tiling of M_f has $\geq t$ rectangles. Then $D(f) \geq \lg(t)$.

Example: EQ: $M_f = \underbrace{\begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix}}_{2^n} \Big\} 2^n$ Clearly need at least 2^n+1 rectangles to tile (one for each "1", one for the rest)

$$\Rightarrow D(EQ) \geq \lg_2(2^n+1)$$
$$D(EQ) \geq n+1 \quad \text{(since it's in } \mathbb{Z}\text{)}$$

Technique 2: fooling sets.

<u>Def</u> $F = \{(x_1, y_1), \ldots, (x_t, y_t)\}$ is a FOOLING SET for f if

(a) f is constant on F (say $f(x_i, y_i) = b \ \forall i$)

(b) $\forall i \neq j$, either $f(x_i, y_j) \neq b$ or $f(x_j, y_i) \neq b$

eg, if $b=0$, $\begin{array}{c|cc} & x_2 & y_2 \\ \hline x_1 & 0 & 0 \\ y_1 & 1 & 0 \end{array}$ or $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ or $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Notice: No monochromatic rectangle can contain more than one elt of F

$\Rightarrow$ <u>Lemma</u> $D(f) \geq \lg_2(|F|)$ for any fooling set F.

Example: DISJ. Let $F = \{(x, \mathbb{1} \oplus x) ; x \in \{0,1\}^n\}$. $DISJ(x,y) = 1 \ \forall (x,y) \in F$.
But at least one of
$$x \cap \mathbb{1} \oplus y \quad \text{or} \quad \mathbb{1} \oplus x \cap y \qquad \text{if } x \neq y$$
So F is a fooling set of size 2^n.

$$\Rightarrow D(DISJ_n) \geq n.$$

(7) Technique 3. Rank lower bound:

Lemma $D(f) \geq \lg_2(\mathrm{rank}(M_f))$ — rank is over any field you like (characteristic 0 will give best results)

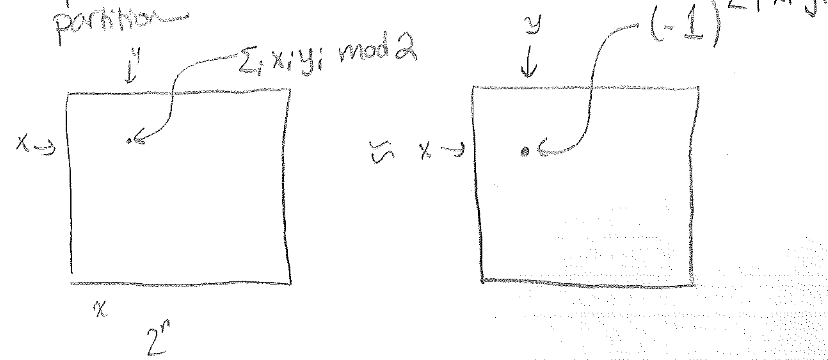
Pf. Write M_f as a sum of the "1" combinatorial rectangles in an optimal partition.

$$M_f = \mathbb{1}_{I_1}\mathbb{1}_{J_1}^T + \mathbb{1}_{I_2}\mathbb{1}_{J_2}^T + \cdots + \mathbb{1}_{I_t}\mathbb{1}_{J_t}^T$$

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$$

so $\mathrm{rank}(M_f) \leq \underset{\text{in optimal partition}}{\#\text{rectangles}} \leq 2^{D(f)}$. ∎

Example. $M_{IP} =$



This is the $2^n \times 2^n$ Hadamard matrix, which has rank 2^n over $\mathbb{R}$.

(indeed, any 2 distinct rows are $\perp$:

$$\sum_{y \in \mathbb{F}_2^n} (-1)^{\langle x, y\rangle}(-1)^{\langle x', y\rangle} = \sum_{y \in \mathbb{F}_2^n} (-1)^{\langle x+x', y\rangle}$$

$$= \begin{cases} 2^n & x + x' = 0 \bmod 2 \text{ aka } x = x' \\ 0 & \text{else} \end{cases}$$

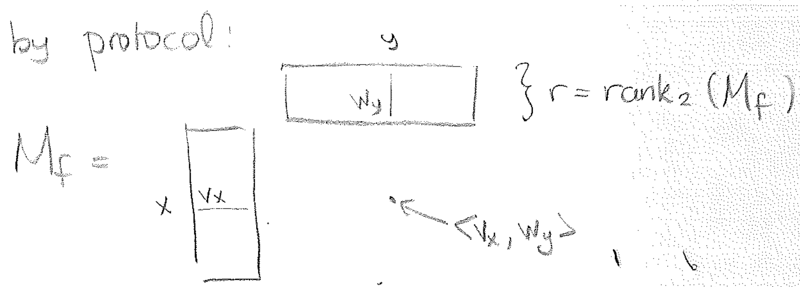
$\Rightarrow D(IP_n) \geq n.$

8) How tight is this rank bound?

Lemma: $D(f) \leq \mathrm{rank}_2(M_f) + 1$ (for a Boolean f)

rank over $\mathbb{F}_2$

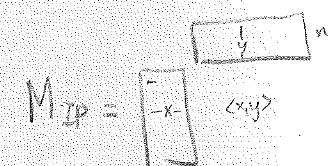
Proof by protocol:



Alice sends v_x, Bob computes $\langle v_x, w_y \rangle = f(x,y)$ and sends that back to Alice.

Hence, $\lg_2(\mathrm{rank}(M_f)) \leq D(f) \leq \mathrm{rank}_2(M_f) + 1$

Notice that this is tight for IP: $\mathrm{rank}_2(M_{IP_n}) = n$, since



<u>Log-rank conjecture</u> (Lovasz, Saks '88)

$$D(f) = (\log \mathrm{rank}(M_f))^{O(1)}$$

rank over $\mathbb{R}$

Known (Nisan-Wigderson, Kushilevitz '94) If it's true, the exponent is ≥ 1.63 $(= \log_3(6))$

Thm [Lovett '13] $D(f) \leq O(\sqrt{\mathrm{rank}(f)} \log \mathrm{rank}(f))$

(Basically) equiv. to $\log \chi(G) = (\log \mathrm{rank}(M_G))^{O(1)}$ $\forall$ undirected G

adjacency

(9) Direct sum conjecture.

Say Alice + Bob want to compute f on k unrelated inputs.

- $f^k((x_1,y_1),(x_2,y_2),\ldots,(x_k,y_k)) = (f(x_1,y_1),\ldots,f(x_k,y_k))$
- Clearly $D(f^k) \le k \cdot D(f)$ (just do it k times).

Conjecture $\forall f$, $D(f^k) = \Omega(k \cdot D(f))$.

(This would imply circuit lower bds [Karchmer-Raz-Wigderson].)