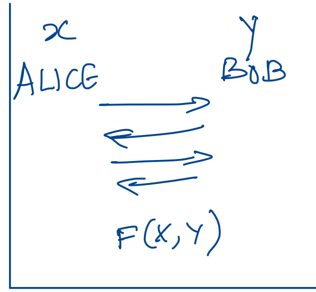
# 15-252 Spring 2021, Lecture 5

## COMMUNICATION COMPLEXITY

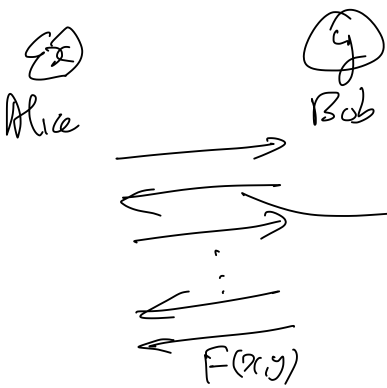



Most basic, fundamental setting:
2 party communication protocols
(deterministic)

Two players Alice & Bob
Trying to compute a function $F : X \times Y \to Z$
($X, Y, Z$ - finite domains,
usually $Z = \{0,1\}$
$X, Y = \{0,1\}^n$ )

Alice holds $x \in X$, Bob holds $y \in Y$
They want to know $F(x,y)$

Would like to do this by communicating back and forth, leading to both of them learning $F(x,y)$ at the end




exchange bits in a comm. protocol.

Alice only knows $x$,
Bob only knows $y$.
$F(x,y)$: computable fn

<u>Here</u>: Focus only on # bits communicated, not number of steps for Alice & Bob to compute their responses

$F(x,y) = x_1 \oplus \cdots \oplus x_n \oplus y_1 \oplus \cdots \oplus y_n$

$x = (x_1 \ldots x_n) \in \{0,1\}^n$

$y = (y_1 \ldots y_n) \in \{0,1\}^n$

Can compute with two bits exchanged.

Alice ($x$) — Bob ($y$)

$$\text{Alice} \xrightarrow{b = x_1 \oplus x_2 \oplus \cdots \oplus x_n} \text{Bob}$$

$$\text{Alice} \xleftarrow{b \oplus y_1 \oplus \cdots \oplus y_n} \text{Bob}$$

---

$F := EQ$

$$EQ(x,y) = \begin{cases} 1 & \text{if } x = y \text{ (i.e. } x_1 = y_1, x_2 = y_2, \ldots, x_n = y_n) \\ 0 & \text{otherwise} \end{cases}$$

Alice ($x$) — Bob ($y$) ($x_i = y_i$ for every $i$)

$$\text{Alice} \xrightarrow{x_1} \text{Bob}$$
$$\xrightarrow{x_2}$$
$$\vdots$$
$$\xrightarrow{x_n}$$
$$\xleftarrow{EQ(x,y)}$$

$(n+1)$ bits exchanged

For every comm. prob. $F : \{0,1\}^n \times \{0,1\}^n \to Z$

$F$ can be computed with $(n+1)$ bits exchanged

(Alice sends $x$ to Bob,
Bob computes $F(x,y)$ & announces the answer)

Intuitively exchanging $n$ bits seems necessary for computing $EQ(x,y)$ since for each $i \in \{1 \ldots n\}$, we need to know if $x_i = y_i$

So Alice & Bob must "talk about" $x_i$ or $y_i$

HOW TO TURN THIS INTO A RIGOROUS PROOF?

Communication protocol formally:
(for $F: X \times Y \to Z$)

Protocol specifies at each step

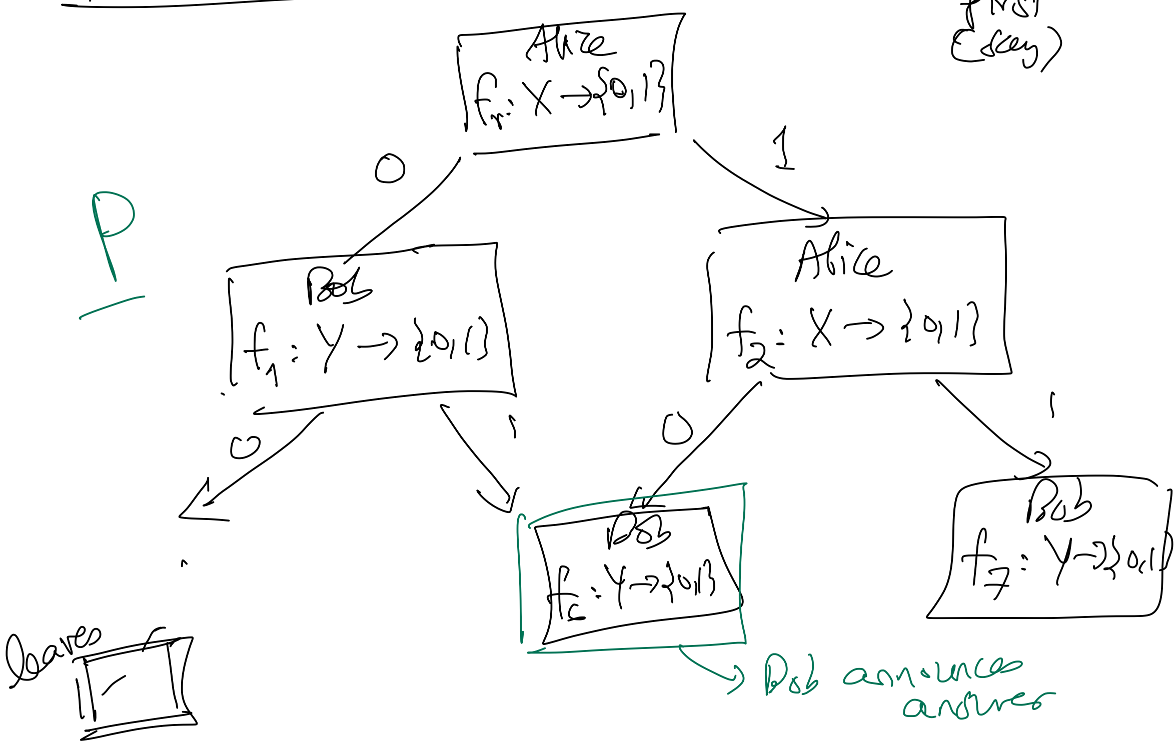
(1) Whose turn is it to send a bit
(depends only on bits exchanged so far)

(2) What bit to send
(depends on bits exchange so far as well as input of player sending the bit)

Protocol also tells when comm. stops & the value of the output (based on transcript of comm. bits)

Protocol as a tree:

Alice speaks first (say)



For given pair $(x,y) \in X \times Y$

$\Pi_P(x,y)$ — labels on the root leaf path traversed on input pair $(x,y)$ for protocol $P$

$\Pi_P$: transcript of $P$

$$\text{Cost}(P) := \max_{(x,y)\in X\times Y} |\Pi_P(x,y)|$$

Det. comm. compl. of protocol $P$ for computing $F(x,y)$

$\hookrightarrow$ worst case measure

$\boxed{\text{cost}(P) = \text{height of its tree}}$

For $F: X \times Y \to Z$, its det. comm. complexity

$$D(F) \stackrel{\text{def}}{=} \min_{\text{protocol } P \text{ that computes } F} \text{cost}(P)$$

Earlier remarks: $\forall\ F: \{0,1\}^n \times \{0,1\}^n \to Z$

$$D(F) \leq n+1$$ (Alice sends $X$, Bob announces answer)

---

$D(EQ) \leq n+1$

$D(\oplus) \leq 2$

$$\text{Maj}(x,y) \begin{cases} = 1 & \text{if there are at least } n \text{ 1's in } x \,\&\, y \text{ combined} \\ = 0 & \text{otherwise} \end{cases}$$

$D(\mathrm{Maj}) \le n+1$ ✓

Alice $\xrightarrow{|x| \text{ in binary}}$ Bob

$\xleftarrow{\text{answer}}$

$|x| = |\{i \mid x_i = 1\}|$

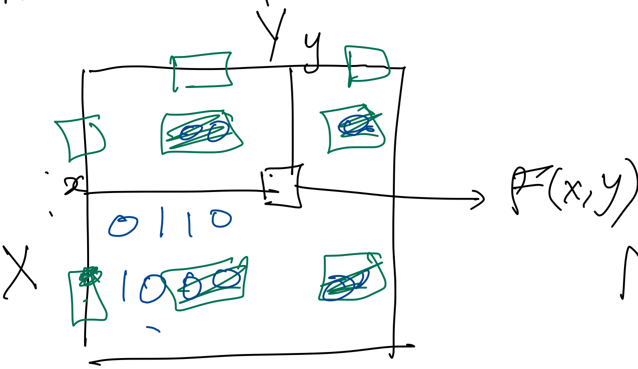
$$D(\mathrm{Maj}) \le \lceil \log_2 n \rceil + 1$$

---

<u>Thm</u>: $D(EQ) = n+1$

Develop some structural understanding of protocols

Matrix view of $F : X \times Y \to \{0,1\}$




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

A submatrix $S \times T$ where $S \subseteq X$, $T \subseteq Y$ is called a (combinatorial) $\boxed{\text{rectangle}}$

<u>Note</u> S & T need not be contiguous rows/columns

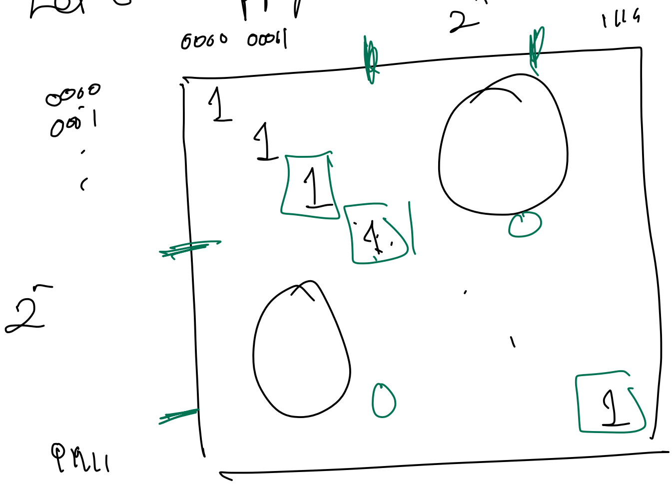
**Def$^n$:** A rectangle $S \times T$ is monochromatic if $M_F$ restricted to $S \times T$ has all 0's or all 1's
(for $F: X \times Y \to \{0,1\}$)

**Fundamental Proposition:**

A protocol $P$ for computing $F: X \times Y \to \{0,1\}$ with $\text{cost}(P) \le c$ bits induces a partition of $M_F$ into at most $2^c$ monochromatic rectangles.

Pf idea coming shortly.

Let's apply it to show $D(EQ) = n+1$




Look at $M_{EQ}$.

$M_{EQ} = I_{2^n \times 2^n}$

How many monochromatic rectangles are needed to cover all the 1's (on the diagonal).

Only monochromatic rectangle with all 1's are $1 \times 1$ rectangles

$\therefore$ Need $2^n$ rectangles to cover all the 1's.

Also need 1 rectangle (at least) to cover the 0's.

OTOH we know $2^{D(EQ)}$ rectangles suffice to cover all 0's & 1's (by Proposition)

$$\Rightarrow \quad 2^{D(EQ)} \geq 2^n + 1$$

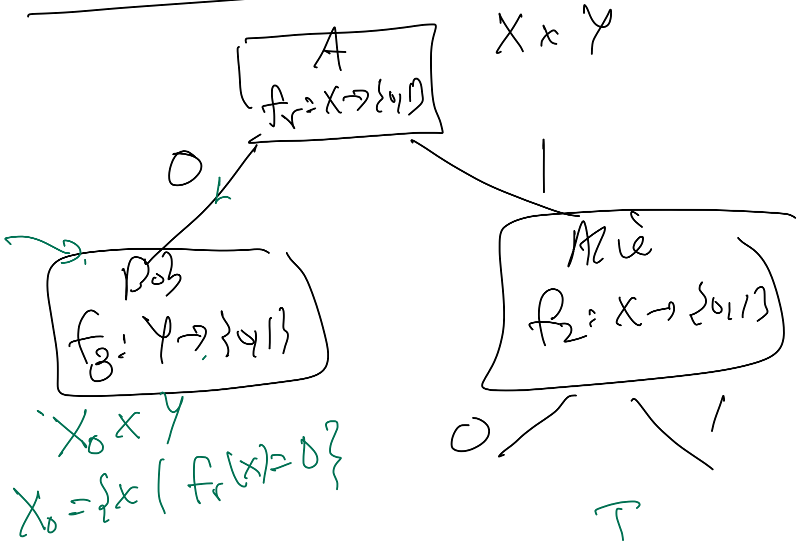
$$\Rightarrow \quad D(EQ) \geq n+1$$

We know $D(EQ) \leq n+1$

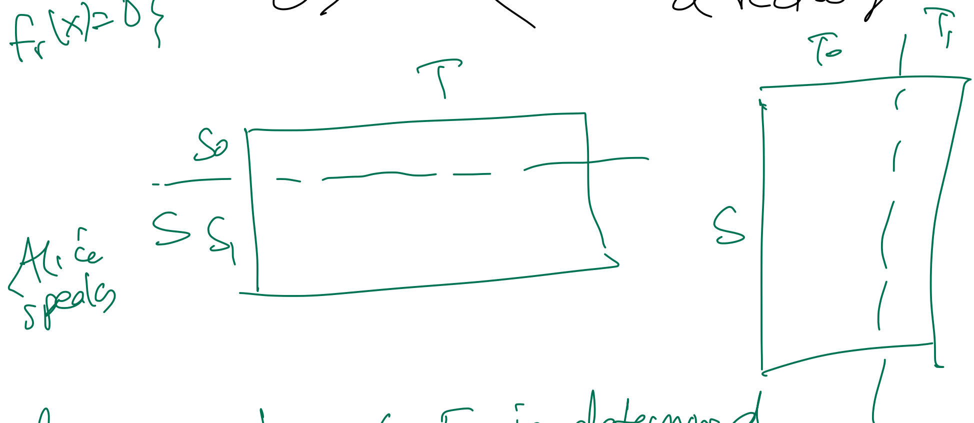
$$\Rightarrow \quad D(EQ) = n+1.$$

$\boxtimes$

# Pf Sketch for Proposition




Prove the invariant that the set of inputs which reach a particular node ~~of~~ of the tree a rectangle.



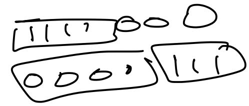

At leaves, value of F is determined

So rectangle is monochromatic (in $M_F$).

If cost(P) $= c \implies$ its protocol tree has $\leq 2^c$ leaves

Exercise. $DISJ(x,y) = \begin{cases} 1 & \text{if there is no } i \text{ s.t. } x_i = y_i = 1 \\ 0 & \text{otherwise} \end{cases}$

Viewed As sets, $x \cap y = \emptyset$



Prove $D(DISJ) = n+1$.

## Application to irregularity of languages

(streaming lower bounds)

$$PAL = \{ w w^R \mid w \in \{0,1\}^* \}$$

We know PAL is not regular

**Key insight**: DFA can be used to give a $O(1)$ communication protocol, in this case for EQ.

(Which is a contradiction)

**Idea**: Suppose $\exists$ DFA for PAL.

How can Alice & Bob use it in a protocol to compute $EQ(x, y)$

$x$

Alice ($x$) — Bob ($y$)

Run DFA for PAL on $x$ $\xrightarrow[\text{of DFA}]{\text{send (current) state}}$ Continues execution of DFA on $y^R$

$\xleftarrow{}$ 1 if DFA ends in acc. state, 0 otherwise

Note: DFA accepts $\iff x = y$

Alice & Bob together run DFA on $xy^R$