Most basic, fundamental setting:

2 party communication protocols
(deterministic)

Two players Alice & Bob

Trying to compute a function \( F : X \times Y \rightarrow Z \)

\((X, Y, Z)\) - 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

Alice only knows \( x \)
Bob only knows \( y \)

Here: Focus only on #bits communicated
not number of steps for Alice & Bob to compute their responses
\[ F(x,y) = x \oplus y \oplus x \oplus y \oplus x \oplus y \]
\[ x = (x_1 \ldots x_n) \in \{0,1\}^n \]
\[ y = (y_1 \ldots y_n) \in \{0,1\}^n \]

Can compute with two bits exchanged.

\[
F = \text{EQ}
\]
\[
\text{EQ}(x,y) = \begin{cases} 
1 & \text{if } x = y \text{ (i.e. } x_i = y_i \text{ for every } i) \\
0 & \text{otherwise}
\end{cases}
\]

Alice

\[ x_1 \]
\[ x_2 \]
\[ x_n \]

Bob

\[ (x_i = y_i \text{ for every } i) \]

(\(N+1\) bits exchanged)

For every comm. prob. \( F : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z} \)

\( F \) 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 \( \text{EQ}(x,y) \) 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 specifics at each step:
1. Who's turn is it to send a bit
   (depends only on bits exchanged so far)
2. What bit to send
   (depends on bits exchanged 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:

\[ P \]

\[ \begin{array}{c}
  \text{Alice} \\
  f_1 : X \to \{0,1\}
\end{array} \]

\[ 0 \]

\[ \begin{array}{c}
  \text{Bob} \\
  f_1 : Y \to \{0,1\}
\end{array} \]

\[ 1 \]

\[ \begin{array}{c}
  \text{Alice} \\
  f_2 : X \to \{0,1\}
\end{array} \]

\[ 0 \]

\[ \begin{array}{c}
  \text{Bob} \\
  f_2 : Y \to \{0,1\}
\end{array} \]

Alice speaks first (Eddy)

Bob announces answer
For given pair \((x, y) \in X \times Y\)

\[ T_T(x, y) \] - labels on the root leaf path traversed on input pair \((x, y)\) for protocol \(P\)

\[ T_T: \text{transcript of } P \]

\[ \text{Cost}(P) := \max_{(x, y) \in X \times Y} |T_T(x, y)| \]

\[ \text{Det. comm. compl. of protocol } P \text{ for computing } F(x, y) \]

\[ \text{Cost}(P) = \text{height of its tree} \]

For \(F: X \times Y \rightarrow Z\), its det. comm. complexity:

\[ D(F) \overset{\text{def}}{=} \min \{ \text{cost}(P) \}
\]

(Protocol \(P\) that computes \(F\))

Earlier remark:

\[ F: \{0, 1\}^n \times \{0, 1\}^n \rightarrow Z \]

\[ D(F) \leq n + 1 \]

(Alice sends \(x\)

Bob announces answer)

\[ D(Eq) \leq n + 1 \]

\[ D(\oplus) \leq 2 \]

\[ \text{Maj}(x, y) \overset{1}{=} \begin{cases} 1 & \text{if there are at least } n + 1 \text{ in } x \& y \text{ combined} \\ 0 & \text{otherwise} \end{cases} \]
\[ D(\text{Maj}) \leq n + 1 \checkmark \]

Alice \quad Bob

\[ \text{left in binary} \quad \text{ask if } x_i = b_i \]

\[ \text{answer} \]

\[ D(\text{Maj}) \leq \lceil \log_2 n \rceil + 1 \]

**Thm:** \[ D(\text{EQ}) = n + 1 \]

Develop some structural understanding of protocol

Matrix view of \( F : X \times Y \rightarrow \{0,1\} \)

A submatrix \( S \times T \) where \( S \subseteq X \), \( T \subseteq Y \) is called a (combinatorial) rectangle

Note \( S \) & \( T \) need not be contiguous rows/columns
Def: A rectangle $S \times T$ is monochromatic if $M_f$ restricted to $S \times T$ has all 0's or all 1's
(for $F : x \times y \rightarrow \{0,1\}$)

Fundamental Proposition:
A protocol $P$ for computing $F : x \times y \rightarrow \{0,1\}$ with $\text{cost}(P) \leq c$ but induces a partition of $M_f$ into at most $2^c$ monochromatic rectangles.

Proof idea coming shortly.
Let's apply it to show $D(E_n) = n+1$

Look at $M_{EQ}$

$M_{EQ} = \frac{D}{2^{2^n}}$
How many monochromatic rectangles are needed to cover all the 1's (on the diagonal)?

Only monochromatic rectangle with all 1's are \( \times 1 \) rectangles.

So, need \( 2^n \) rectangles to cover all the 1's.

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

OTOH we know \( 2 \text{D}(\text{EQ}) \) rectangle suffices to cover all 0's & 1's (by Proposition)

\[
\Rightarrow 2 \text{D}(\text{EQ}) \geq 2^n + 1
\]

\[
\Rightarrow \text{D}(\text{EQ}) \geq n + 1
\]

We know \( \text{D}(\text{EQ}) \leq n + 1 \)

\[
\Rightarrow \text{D}(\text{EQ}) = n + 1.
\]
At leaves, value of $F$ is determined 
so rectangle is monochromatic (in $M_c$).

If $\text{cost}(t) = c \Rightarrow$ its protocol tree 
has $\leq 2c$ leaves.

**Exercise.**

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

Prove $D(D_{1}S_{1}) = n+1$. 
Application to Irregularity of Languages

(stream) lower bounds

\[ \text{PAL} = \{ww^r \mid w \in \{0,1\}^* \} \]

We know \( \text{PAL} \) is not regular

Key insight: DFA can be used to give a \( O(1) \) communication protocol; in this case for \( \text{Eq} \), (which is a contradiction)

Idea: Suppose \( \exists \) DFA for \( \text{PAL} \).

How can Alice & Bob use it in a protocol to compute \( \text{Eq}(x,y) \)?

**\( x \)**

- Alice

Run DFA for \( \text{PAL} \) on \( x \)

**\( y \)**

- Bob

(current) send state of DFA

Continue execution of DFA on \( y \)

1 if DFA ends in acc. state

0 otherwise

Note: DFA accepts

\( x=y \)

Alice & Bob together run DFA on \( x=y \)