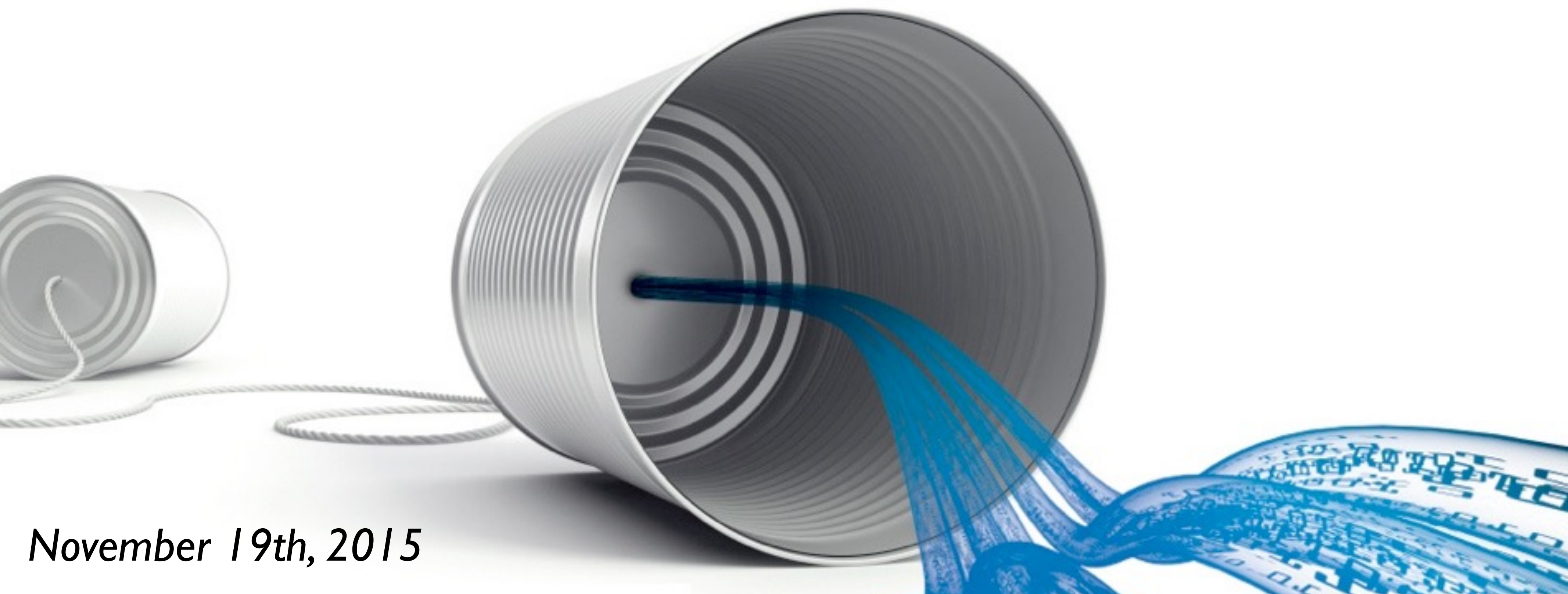
# 15-251
# Great Theoretical Ideas in Computer Science

## Lecture 24:
## Communication Complexity



*November 19th, 2015*

What are the limitations to what computers can learn?

Do certain mathematical theorems have short proofs?

Can quantum mechanics be exploited to speed up computation?

Is every problem whose solution is efficiently verifiable also efficiently solvable? *i.e.* P = NP?

What are the limitations to what computers can learn?

Do certain mathematical theorems have short proofs?

Can quantum mechanics be exploited to speed up computation?

Is every problem whose solution is efficiently verifiable also efficiently solvable? *i.e.* P = NP?

# Communication complexity

# Cool Things About Communication Complexity

Many useful applications:

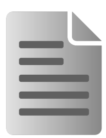
*machine learning, proof complexity, quantum computation, pseudorandom generators, data structures, game theory,...*

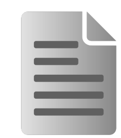
The setting is simple and neat.

Beautiful mathematics

*combinatorics, algebra, analysis, information theory, ...*

# Motivating Example 1: Checking Equality





01001010101110101 $\stackrel{?}{=}$ 01001010100110101

← $n$ bits → ← $n$ bits →

How many bits need to be communicated?

Naively: $n$ Actually: $n$

What if we allow 0.0000000001% probability of error?

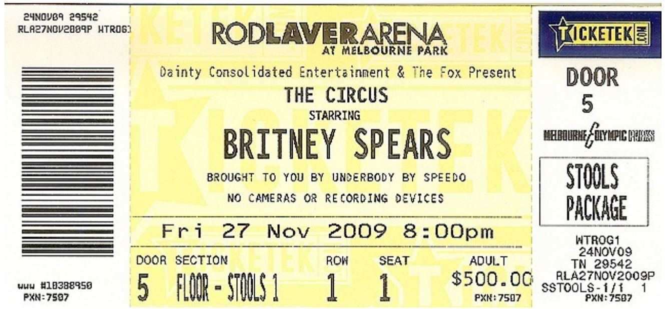
Naively: $\Omega(n)$ Actually: $O(\log n)$

# Motivating Example 2: Auctions

Alice

Bob

$100




$1000

# Defining the model a bit more formally

# 2 Player Model of Communication Complexity

$$F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$$

known to
both players

(Alice)

$x \in \{0,1\}^n$

(Bob)

$y \in \{0,1\}^n$

**Goal:** Compute $F(x, y)$. (both players should know the value)

**How:** Sending bits back and forth according to a protocol.

**Resource:** Number of communicated bits.

(We assume players have unlimited computational power individually.)

# Poll 1

$x, y \in \{0,1\}^n$, $PAR(x,y) =$ parity of the sum of all the bits.

(i.e. it's 1 if the parity is odd, 0 otherwise.)

How many bits do the players need to communicate?
Choose the tightest bound.

$O(1)$

$O(\log n)$

$O(\log^2 n)$

$O(\sqrt{n})$

$O(n/\log n)$

$O(n)$

# Poll 1 Answer

$x, y \in \{0,1\}^n$, $PAR(x, y) =$ parity of the sum of all the bits.

(i.e. it's 1 if the parity is odd, 0 otherwise.)

How many bits do the players need to communicate?
Choose the tightest bound.

Once Bob knows the parity of $x$, he can compute $PAR(x, y)$.

- Alice sends $PAR(x)$ to Bob. 1 bit
- Bob computes $PAR(x, y)$ and sends it to Alice. 1 bit

2 bits in total

# 2 Player Model of Communication Complexity

$$F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$$

known to
both players

(Alice)

$x \in \{0,1\}^n$

(Bob)

$y \in \{0,1\}^n$

**Goal:** Compute $F(x,y)$. (both players should know the value)

**How:** Sending bits back and forth according to a protocol.

**Resource:** Number of communicated bits.

(we assume players have unlimited computational power individually.)

# 2 Player Model of Communication Complexity

**Goal:** Compute $F(x, y)$. (both players should know the value)

**How:** Sending bits back and forth according to a protocol.

**Resource:** Number of communicated bits.

A protocol $P$ is the "strategy" players use to communicate.

It determines what bits the players send in each round.

$P(x, y)$ denotes the output of $P$.

# 2 Player Model of Communication Complexity

**Goal:** Compute $F(x, y)$. (both players should know the value)

**How:** Sending bits back and forth according to a protocol.

**Resource:** Number of communicated bits.

A (deterministic) protocol $P$ computes $F$ if

$$\forall (x, y) \in \{0,1\}^n \times \{0,1\}^n, \qquad P(x, y) = F(x, y)$$

**Analogous to:** algorithm (TM) decision problem

$$\forall x \in \Sigma^* \quad A(x) = F(x)$$

# 2 Player Model of Communication Complexity

**Goal:** Compute $F(x, y)$. (both players should know the value)

**How:** Sending bits back and forth according to a protocol.

**Resource:** Number of communicated bits.

A randomized protocol $P$ computes $F$ with $\epsilon$ error if

$$\boxed{\forall (x, y) \in \{0,1\}^n \times \{0,1\}^n}, \quad \Pr[P(x, y) \neq F(x, y)] \leq \epsilon$$

**Analogous to:** Monte Carlo algorithms

# 2 Player Model of Communication Complexity

**Goal:** Compute $F(x, y)$. (both players should know the value)

**How:** Sending bits back and forth according to a protocol.

**Resource:** Number of communicated bits.

$$\text{cost}(P) = \max_{(x,y)} \# \text{ bits } P \text{ communicates for } (x, y)$$

if P is randomized, you take max over the random choices it makes.

*Deterministic communication complexity*

$\mathbf{D}(F)$ = min cost of a (deterministic) protocol computing $F$.

*Randomized communication complexity*

$\mathbf{R}^{\epsilon}(F)$ = min cost of a randomized protocol computing $F$ with $\epsilon$ error.

# 2 Player Model of Communication Complexity

**Goal:** Compute $F(x, y)$. (both players should know the value)

**How:** Sending bits back and forth according to a protocol.

**Resource:** Number of communicated bits.

$$\text{cost}(P) = \max_{(x} \ \# \text{ bits } P \text{ communicates for } (x, y)$$

max
makes.

*Deterministic co*

$\mathbf{D}(F) =$ mi … puting $F$.

*Randomized co*

$\mathbf{R}^{\epsilon}(F) =$ min … puting $F$
with $\epsilon$ error.

We usually fix $\epsilon$ to some constant.

e.g. $\epsilon = 1/3$

We can always boost the success probability if we want.

# What is considered hard or easy?

$$F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$$

$$0 \leq \quad \mathbf{R}_2^{\epsilon}(F) \leq \mathbf{D}_2(F) \quad \leq n+1$$

$$c \qquad \log^c(n) \qquad\qquad n^\delta \qquad \delta n$$

# Example

Equality: $$EQ(x,y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathbf{D}(EQ) = n + 1.$$

$$\mathbf{R}^{1/3}(EQ) = O(\log n).$$

# Poll 2

$MAJ(x, y) =$ 1 iff majority of all the bits in $x$ and $y$ are set to 1.

What is $\mathbf{D}(MAJ)$? Choose the tightest bound.

$O(1)$

$O(\log n)$

$O(\log^2 n)$

$O(\sqrt{n})$

$O(n/\log n)$

$O(n)$

# Poll 2 Answer

$MAJ(x,y) =$ 1 iff majority of all the bits in $x$ and $y$ are set to 1.

What is $\mathbf{D}(MAJ)$? Choose the tightest bound.

The result can be computed from

$$\sum_{i \in \{1,2,\ldots,n\}} x_i + \sum_{i \in \{1,2,\ldots,n\}} y_i$$

- Alice sends $\sum_i x_i$ to Bob. ~ log n bits
- Bob computes $MAJ(x,y)$ and sends it to Alice. 1 bit

$O(\log n)$ in total

# Another example

Disjointness: $DISJ(x,y) = \begin{cases} 0 & \text{if } \exists i : x_i = y_i = 1 \\ 1 & \text{otherwise} \end{cases}$

$$\mathbf{R}^{1/3}(DISJ) = \Omega(n).$$ hard

# The plan

1. Efficient randomized communication protocol for checking equality.

2. An application of communication complexity.

3. How to prove lower bounds.

# Efficient randomized communication protocol for checking equality

# The Power of Randomization

$$\mathbf{R}^{1/3}(EQ) = O(\log n).$$

Alice gets $x \in \{0,1\}^n$, Bob gets $y \in \{0,1\}^n$.

We treat $x$ and $y$ as numbers: $0 \leq x, y \leq 2^n - 1$.

**The Protocol:**

- Let $p_i$ be the $i$'th smallest prime number.

  $$p_1 = 2,\ p_2 = 3,\ p_3 = 5,\ p_4 = 7,\ \ldots$$

- Alice picks a random $i \in \{1, 2, \ldots, n^2\}$.

- Alice sends Bob: $i, \quad x \bmod p_i$

- Bob outputs 1 iff $x \bmod p_i = y \bmod p_i$. $(x \equiv_{p_i} y)$

# The Power of Randomization

$$\mathbf{R}^{1/3}(EQ) = O(\log n).$$

## Correctness:

*Want to show:* For all inputs $(x, y)$, probability of error is $\leq 1/3$.

For all $(x, y)$ with $x = y$ :

$$\Pr[\text{error}] = \Pr_i[x \not\equiv_{p_i} y] = 0.$$

For all $(x, y)$ with $x \neq y$ :

$$\Pr[\text{error}] = \Pr_i[x \equiv_{p_i} y] = \Pr_i[p_i \text{ divides } x - y]$$

**Claim:** $x - y$ has at most $n$ distinct prime factors.

$$\Pr[\text{error}] = \Pr[p_i \text{ is a prime factor of } x - y] \leq \frac{n}{n^2} = \frac{1}{n}.$$

# The Power of Randomization

$$\mathbf{R}^{1/3}(EQ) = O(\log n).$$

**Cost:**

The only communication is:

- Alice sends Bob: $\boxed{i,}$ $\boxed{x \bmod p_i}$

The first number $i$ is such that $i \leq n^2$.

Can represent it using $\sim \log_2 n^2 = 2 \log_2 n = O(\log n)$ bits.

The second number $x \bmod p_i$ is at most $p_{n^2}$.

By the Prime Number Theorem: $p_{n^2} \sim n^2 \log n^2 \leq 2n^3$

Can represent $p_{n^2}$ using at most $\log(2n^3) = O(\log n)$ bits. $\square$

# The plan

1. Efficient randomized communication protocol for checking equality.

2. An application of communication complexity.

3. How to prove lower bounds.

# An application of communication complexity

# Applications of Communication Complexity

- circuit complexity
- time/space tradeoffs for Turing Machines
- VLSI chips
- machine learning
- game theory
- data structures
- proof complexity
- pseudorandom generators
- pseudorandomness
- branching programs
- data streaming algorithms
- quantum computation
- lower bounds for polytopes representing NP-complete problems

# Applications of Communication Complexity

- circuit complexity

**- time/space tradeoffs for Turing Machines**

- VLSI chips

- machine learning

- game theory

- data structures

- proof complexity

- pseudorandom generators

- pseudorandomness

- branching programs

- data streaming algorithms

- quantum computation

- lower bounds for polytopes representing NP-complete problems

# How Communication Complexity Comes In

**Setting:** Solve some **task** while minimizing some **resource**.

e.g. *find a fast algorithm, design a small circuit, find a short proof of a theorem, …*

**Goal:** Prove lower bounds on the **resource** needed.

**Sometimes:**

efficient solution to our problem →

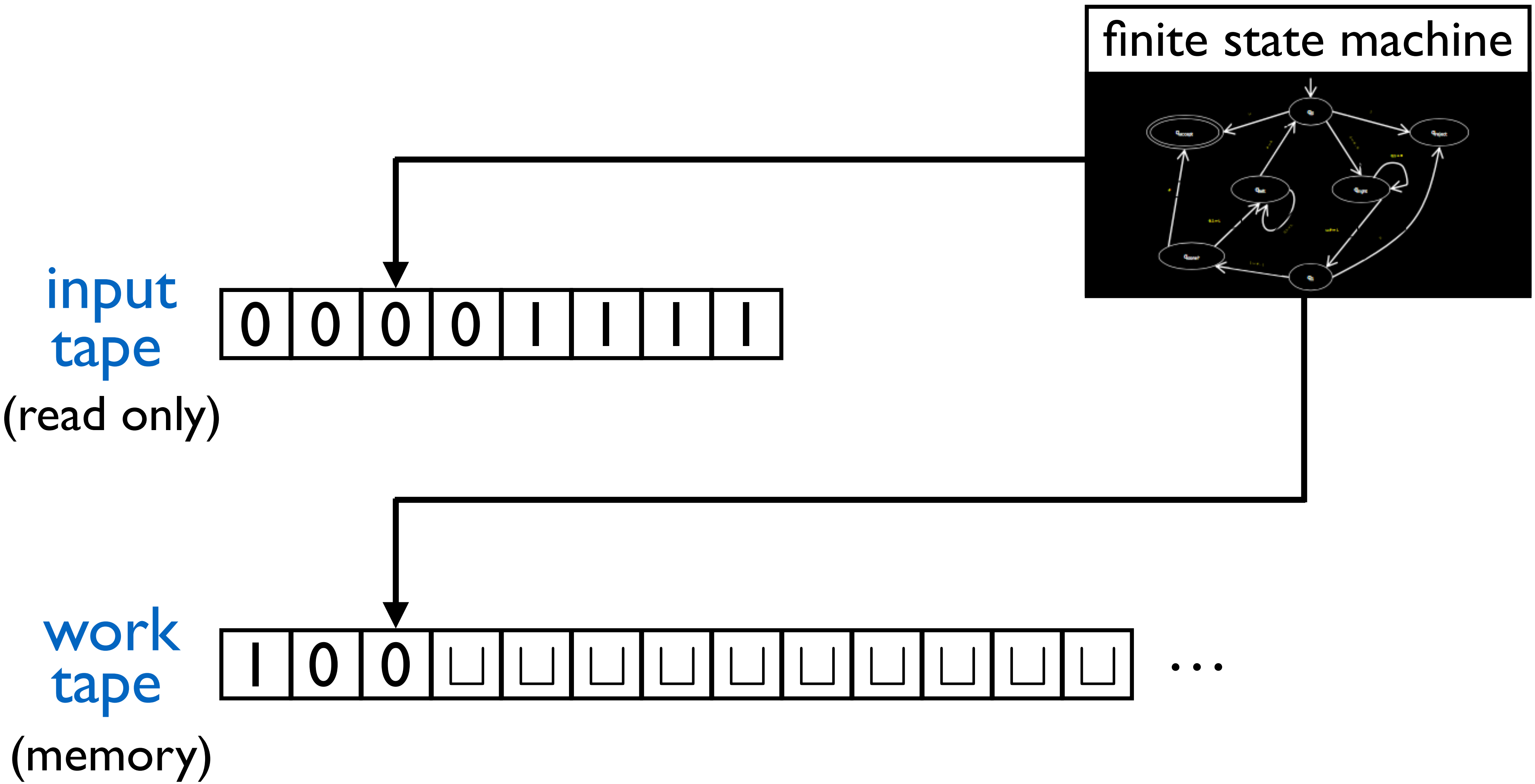
efficient communication protocol for a certain function.

i.e. no efficient protocol for the function →

no efficient solution to our problem.

# Time/space tradeoffs for TMs

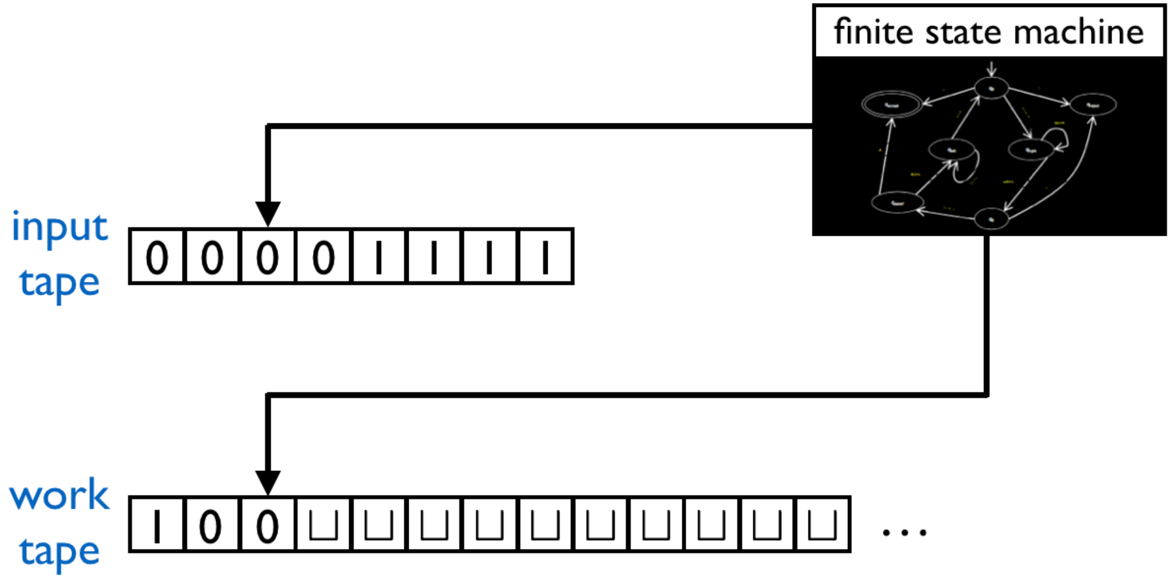
# Recall Turing Machines




$T(n)$ time: # steps the machine takes

$S(n)$ space: # work tape cells the machine uses

# An observation




Suppose we both know the TM $M$ and the input $w$.

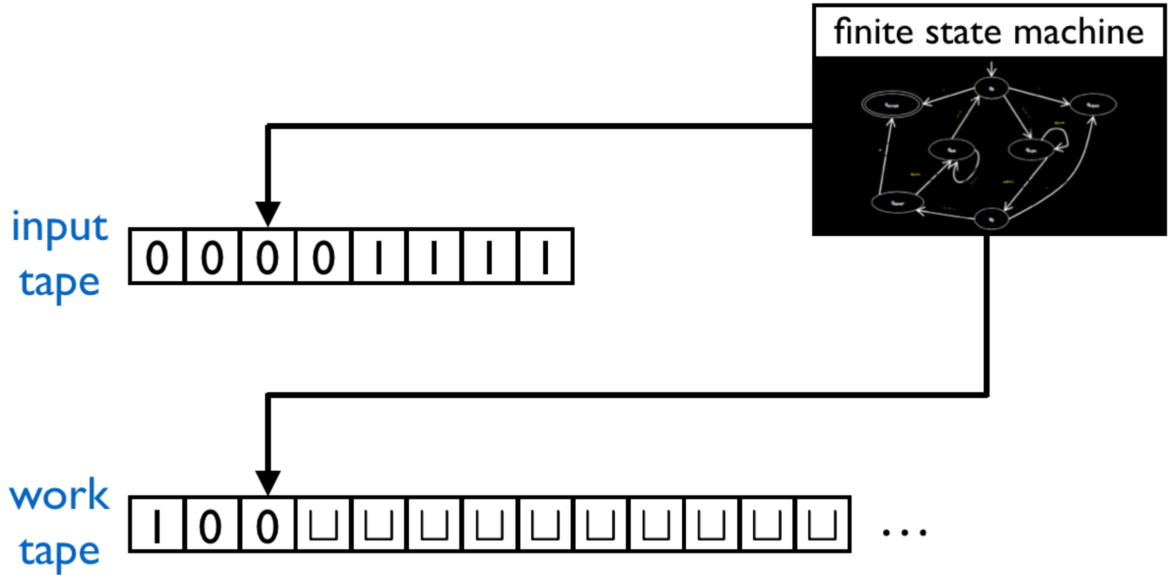
You start running $M(w)$.

You pause after a certain number of steps.

What information do I need to continue the computation from where you left it?

1. current state
2. positions of tape heads
3. contents of work tape

# An observation




Suppose we both know the TM $M$ and the input $w$.

You start running

You pause after a

What informatio
the computation

$O(1)$

$O(\log n) + O(\log S(n))$

$\boxed{O(S(n))}$

1. current state
2. positions of tape heads
3. contents of work tape

# Time/space tradeoff for a simple language

Let $L = \{x\#^{|x|}x : x \in \{0,1\}^*\}$

$$000\#\#\#000 \in L$$

$$1010\#\#\#\#1010 \in L$$

$$001\#\#\#000 \notin L$$

$$000\#\#000 \notin L$$

**Theorem:**

If a TM **M** decides $L$ in $T(n)$ time and $S(n)$ space on inputs of size $3n$, then $T(n) \cdot S(n) = \Omega(n^2)$.

# Time/space tradeoff for a simple language

Let $L = \{x\#^{|x|}x : x \in \{0,1\}^*\}$

**Theorem:**

If a TM **M** decides $L$ in $T(n)$ time and $S(n)$ space on inputs of size $3n$, then $T(n) \cdot S(n) = \Omega(n^2)$.

**Strategy:**

Using **M**, we design a communication protocol for $EQ$ of cost $\leq c\, T(n)S(n)/n$ for some constant $c$.

We know $EQ$ requires $\geq n$ bits of communication.

$$\implies c\, T(n)S(n)/n \geq n \implies c\, T(n)S(n) \geq n^2$$

# Time/space tradeoff for a simple language

Let $L = \{x\#^{|x|}x : x \in \{0,1\}^*\}$ . **M** decides $L$ .

**<u>Protocol for $EQ$ :</u>**

Given input $x \in \{0,1\}^n$ to Alice, and $y \in \{0,1\}^n$ to Bob.

They want to decide if $x = y$ . They will make use of **M**.

Let $w = x\#^n y$.

They simulate **M**$(w)$.

If **M**$(w)$ accepts, they output $1$ .

If **M**$(w)$ rejects, they output $0$.

A correct protocol.

# Time/space tradeoff for a simple language

Let $L = \{x\#^{|x|}x : x \in \{0,1\}^*\}$.  **M** decides $L$.

**Protocol for $EQ$ :**

Given input $x \in \{0,1\}^n$ to Alice, and $y \in \{0,1\}^n$ to Bob.

They want to decide if $x = y$. They will make use of **M**.

Let $w = x\#^n y$.

They simulate **M**$(w)$.

If **M**$(w)$ accepts, they output $1$.

If **M**$(w)$ rejects, they output $0$.

How do they simulate **M**?

What is the cost?

A correct protocol.

# Time/space tradeoff for a simple language

Let $L = \{x\#^{|x|}x : x \in \{0,1\}^*\}$ . **M** decides $L$ .

**Protocol for $EQ$ :**

$\boxed{\text{They simulate } \mathbf{M}(x\#^n y).}$

Alice starts the simulation.

When input tape head reaches $y$ symbol,

she sends

1. current state
2. position of work tape head
3. contents of work tape

# Time/space tradeoff for a simple language

Let $L = \{x\#^{|x|}x : x \in \{0,1\}^*\}$. **M** decides $L$.

**Protocol for $EQ$ :**

They simulate **M**($x\#^n y$).

Bob continues the simulation.

When input tape head reaches $x$ symbol,

he sends
1. current state
2. position of work tape head
3. contents of work tape

This continues until **M** halts.

# Time/space tradeoff for a simple language

**Analysis:**

It is clear the protocol is correct. What is the cost?

In each transmission, players send

1. current state $\longrightarrow$ $O(1)$
2. position of work tape head $\longrightarrow$ $O(\log S(n))$
3. contents of work tape $\longrightarrow$ $+\ O(S(n))$

$$O(S(n))$$

What is the number of transmissions?

For each transmission, **M** takes $\geq n$ steps.

So $T(n) \geq (\#\ \text{transmissions}) \cdot n$.

$\Longrightarrow\ \#\ \text{transmissions} \leq T(n)/n.$

**Total cost:** $O(S(n)T(n)/n)$. $\square$

# The plan

1. Efficient randomized communication protocol for checking equality.

2. An application of communication complexity.

3. How to prove lower bounds.

# Lower bounds for
# deterministic communication complexity

# The function matrix

$$F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$$

$$M_F = \begin{array}{cc} & y \\ x & \boxed{\begin{array}{c}
011010111010111111001010010101000 \\
010101010110101001010100101111100 \\
010100001010111010101010111101011 \\
000101110101111010101100101010101 \\
001010101010101010110100010101011 \\
010111010111010010110101011110100 \\
010101110101010101000101000101101 \\
010101010111010101101101101110101 \\
111010101011010101010101011111100 \\
111011101010101010101010101001111 \\
110101101010101000101010101000101 \\
011110000111110000000001110101111 \\
011010111010111111001010010101000 \\
001010101010100101010101111110000 \\
101010101000001110101011101011000
\end{array}} \end{array}$$

$$M_F[x, y] = F(x, y)$$

$2^n$ by $2^n$ matrix

# The function matrix

**Equality:** $EQ(x,y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$

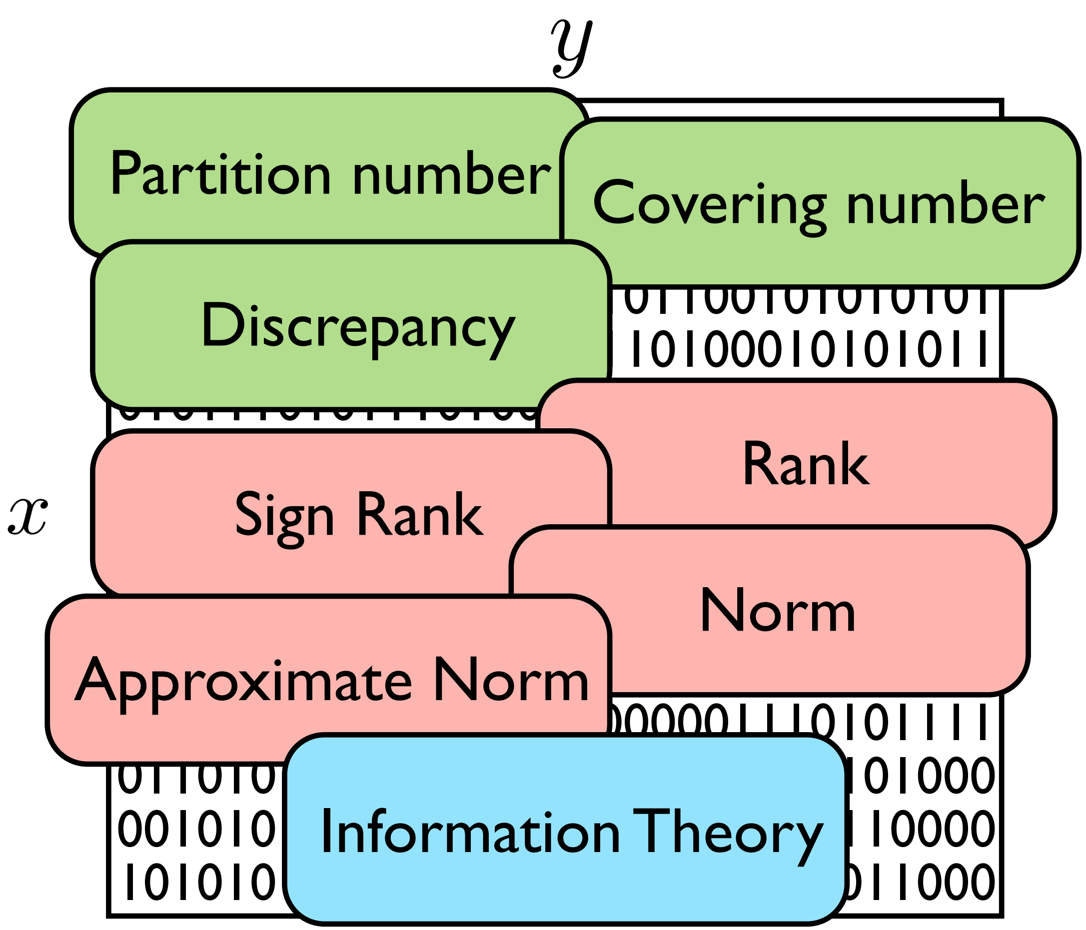
n = 3

$$M_{EQ} = \begin{array}{cc} & y \\ x & \begin{array}{c|cccccccc|} & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ \hline 000 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 001 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 010 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 011 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 100 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 101 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 110 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 111 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline \end{array} \end{array}$$

$2^n$ by $2^n$ matrix

# The function matrix

How do you prove lower bounds on communication complexity?




$$M_F[x, y] = F(x, y)$$

You study this matrix!

# IMPORTANT(!) property of communication protocols:

A protocol partitions *function matrix* into monochromatic rectangles.

What is a rectangle?

| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| 000 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 001 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 010 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 011 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 100 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 101 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 110 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 111 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

A rectangle is of the form $S \times T$ for $S, T \subseteq \{0,1\}^n$

# IMPORTANT(!) property of communication protocols:

A protocol partitions *function matrix* into monochromatic rectangles.

What is a rectangle?

$T$

$S$

| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| 000 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 001 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 010 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 011 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 100 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 101 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 110 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 111 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

A rectangle is of the form $S \times T$ for $S, T \subseteq \{0,1\}^n$

# IMPORTANT(!) property of communication protocols:

Suppose we have a deterministic protocol of cost $c$ that computes a function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

This protocol partitions $M_F$ into at most $2^c$ monochromatic rectangles.

You will prove this in the homework.

# Example

$$PAR(x,y) = \sum_{i=1}^{n} x_i + y_i \pmod 2$$

**Protocol:**

Alice sends the parity of her input bits.

Bob sends the output of the function.

The cost of the protocol is **2** bits.

This protocol partitions the function matrix into at most **4** monochromatic rectangles.

# Example

$$PAR(x,y) = \sum_{i=1}^{n} x_i + y_i \pmod 2$$

| | 000 | 011 | 110 | 101 | 111 | 100 | 010 | 001 |
|---|---|---|---|---|---|---|---|---|
| 000 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 011 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 110 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 101 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 111 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 100 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 010 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 001 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

# Example

$$PAR(x,y) = \sum_{i=1}^{n} x_i + y_i \pmod 2$$

| | 000 | 011 | 110 | 101 | 111 | 100 | 010 | 001 |
|---|---|---|---|---|---|---|---|---|
| 000 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 011 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 110 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 101 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 111 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 100 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 010 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 001 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

0

1

# Example

$$PAR(x,y) = \sum_{i=1}^{n} x_i + y_i \pmod 2$$

| | 000 | 011 | 110 | 101 | 111 | 100 | 010 | 001 |
|---|---|---|---|---|---|---|---|---|
| 000 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 011 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 110 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 101 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 111 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 100 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 010 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 001 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

# Example

$$PAR(x,y) = \sum_{i=1}^{n} x_i + y_i \pmod 2$$

| | 000 | 011 | 110 | 101 | 111 | 100 | 010 | 001 |
|---|---|---|---|---|---|---|---|---|
| 000 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 011 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 110 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 101 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 111 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 100 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 010 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 001 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

00

01

1

# Example

$$PAR(x,y) = \sum_{i=1}^{n} x_i + y_i \pmod 2$$

| | 000 | 011 | 110 | 101 | 111 | 100 | 010 | 001 |
|---|---|---|---|---|---|---|---|---|
| 000 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 011 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 110 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 101 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 111 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 100 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 010 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 001 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

00 01

# Example

$$PAR(x,y) = \sum_{i=1}^{n} x_i + y_i \pmod 2$$

| | 000 | 011 | 110 | 101 | 111 | 100 | 010 | 001 |
|---|---|---|---|---|---|---|---|---|
| 000 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 011 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 110 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 101 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 111 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 100 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 010 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 001 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

00 01

11 10

# Lower bound for Equality function

A protocol for $EQ$ of cost $c$ partitions $M_{EQ}$ into at most $2^c$ monochromatic rectangles.

$$\begin{array}{c|cccc|} & 00 & 01 & 10 & 11 \\ \hline 00 & 1 & 0 & 0 & 0 \\ 01 & 0 & 1 & 0 & 0 \\ 10 & 0 & 0 & 1 & 0 \\ 11 & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

**Observe**:

Any protocol computing $EQ$ must cover the 1's with monochromatic rectangles.

# Lower bound for Equality function

A protocol for $EQ$ of cost $c$ partitions $M_{EQ}$ into at most $2^c$ monochromatic rectangles.

$$\begin{array}{c|cccc|} & 00 & 01 & 10 & 11 \\ \hline 00 & 1 & 0 & 0 & 0 \\ 01 & 0 & 1 & 0 & 0 \\ 10 & 0 & 0 & 1 & 0 \\ 11 & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

**Observe**:

Any protocol computing $EQ$ must cover the 1's with monochromatic rectangles.

**Claim**: No two 1's can be in the same monochromatic rectangle

Proof:

Suppose two 1's are in the same rectangle.

# Lower bound for Equality function

A protocol for $EQ$ of cost $c$ partitions $M_{EQ}$ into at most $2^c$ monochromatic rectangles.

| | 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 0 |
| 01 | 0 | 1 | 0 | 0 |
| 10 | 0 | 0 | 1 | 0 |
| 11 | 0 | 0 | 0 | 1 |

**Observe**:

Any protocol computing $EQ$ must cover the 1's with monochromatic rectangles.

**Claim**: No two 1's can be in the same monochromatic rectangle

Proof:

Suppose two 1's are in the same rectangle.

# Lower bound for Equality function

A protocol for $EQ$ of cost $c$ partitions $M_{EQ}$ into at most $2^c$ monochromatic rectangles.

| | 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 0 |
| 01 | 0 | $\boxed{1}$ | 0 | $\boxed{0}$ |
| 10 | 0 | 0 | 1 | 0 |
| 11 | 0 | $\boxed{0}$ | 0 | $\boxed{1}$ |

**Observe**:

Any protocol computing $EQ$ must cover the 1's with monochromatic rectangles.

**Claim**: No two 1's can be in the same monochromatic rectangle

Proof:

Suppose two 1's are in the same rectangle.

Then there must also be 0's in the rectangle. Contradiction.

# Lower bound for Equality function

A protocol for $EQ$ of cost $c$ partitions $M_{EQ}$ into at most $2^c$ monochromatic rectangles.

| | 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 0 |
| 01 | 0 | $\boxed{1}$ | 0 | $\boxed{0}$ |
| 10 | 0 | 0 | 1 | 0 |
| 11 | 0 | $\boxed{0}$ | 0 | $\boxed{1}$ |

**Observe**:

Any protocol computing $EQ$ must cover the 1's with monochromatic rectangles.

**Claim**: No two 1's can be in the same monochromatic rectangle

**Conclusion**: We need a separate rectangle for each 1.

$\Longrightarrow$ We need at least $2^n$ rectangles to cover the 1s.

# Lower bound for Equality function

A protocol for $EQ$ of cost $c$ partitions $M_{EQ}$ into at most $2^c$ monochromatic rectangles.

<u>Conclusion</u>: We need a separate rectangle for each 1.

$\implies$ We need at least $2^n$ rectangles to cover the 1s.

We also need at least one rectangle to cover the **0**s.

$$2^c \geq 2^n + 1$$

$$\implies \quad c \geq n + 1$$

This is true for any protocol computing $EQ$.

In particular, it is true for the most efficient protocol.

$$\mathbf{D}(EQ) \geq n + 1.$$

# Summary of the lower bound technique

Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

A lower bound on # monochromatic rectangles needed to partition $M_F$

$$\Downarrow$$

A lower bound on $\mathbf{D}(F)$.

**Interesting corollary** (not hard to prove):

$$\mathbf{D}(F) \geq \log_2 \operatorname{rank}(M_F)$$

# The plan

1. Efficient randomized communication protocol for checking equality.

2. An application of communication complexity.

3. How to prove lower bounds.

# Take-Home Message

Communication complexity studies natural distributed tasks.

Communication complexity (lower bounds) has **<u>many</u>** interesting applications.

Lower bounds can be proved using a variety of tools:
*combinatorial, algebraic, analytic, information theoretic,…*