# Foundations of Computing

## Lecture 24

Arkady Yerukhimovich

April 18, 2024

# Outline

# Lecture 23 Review

- 3-Coloring is $\mathcal{NP}$-complete
- Ladner's Theorem
- The class co-$\mathcal{NP}$

# $\mathcal{NP}$ vs co-$\mathcal{NP}$

### $\mathcal{NP}$ – Yes instances are efficiently verifiable

$L \in \mathcal{NP}$ if there exists poly-time DTM $V$ s.t. for $x \in L$ there exists a witness $w$ s.t. $V(x, w) = 1$

# $\mathcal{NP}$ vs co-$\mathcal{NP}$

## $\mathcal{NP}$ – Yes instances are efficiently verifiable

$L \in \mathcal{NP}$ if there exists poly-time DTM $V$ s.t. for $x \in L$ there exists a witness $w$ s.t. $V(x, w) = 1$

## co-$\mathcal{NP}$ – No instances are efficiently verifiable

$L \in$ co-$\mathcal{NP}$ if there exists poly-time DTM $V$ s.t. for $x \notin L$ there exists a witness $w$ s.t. $V(x, w) = 1$

# $\mathcal{NP}$ vs co-$\mathcal{NP}$

**$\mathcal{NP}$ – Yes instances are efficiently verifiable**

$L \in \mathcal{NP}$ if there exists poly-time DTM $V$ s.t. for $x \in L$ there exists a witness $w$ s.t. $V(x, w) = 1$

**co-$\mathcal{NP}$ – No instances are efficiently verifiable**

$L \in$ co-$\mathcal{NP}$ if there exists poly-time DTM $V$ s.t. for $x \notin L$ there exists a witness $w$ s.t. $V(x, w) = 1$

Comments:

- $L \in$ co-$\mathcal{NP} \iff \bar{L} \in \mathcal{NP}$
- co-$\mathcal{NP}$ contains the languages whose complement languages are in $\mathcal{NP}$

# $\mathcal{NP}$ vs co-$\mathcal{NP}$

**$\mathcal{NP}$ – Yes instances are efficiently verifiable**

$L \in \mathcal{NP}$ if there exists poly-time DTM $V$ s.t. for $x \in L$ there exists a witness $w$ s.t. $V(x, w) = 1$

**co-$\mathcal{NP}$ – No instances are efficiently verifiable**

$L \in$ co-$\mathcal{NP}$ if there exists poly-time DTM $V$ s.t. for $x \notin L$ there exists a witness $w$ s.t. $V(x, w) = 1$

Comments:

- $L \in \text{co-}\mathcal{NP} \iff \overline{L} \in \mathcal{NP}$
- co-$\mathcal{NP}$ contains the languages whose complement languages are in $\mathcal{NP}$

Is $SAT \in$ co-$\mathcal{NP}$?

# Outline

# What is a Proof?

# What is a Proof?

## Traditional Definition

A proof is a string that convinces us of the truth of some mathematical statement $x$

# What is a Proof?

## Traditional Definition

A proof is a string that convinces us of the truth of some mathematical statement $x$

- $x$ is a satisfiable formula

# What is a Proof?

## Traditional Definition

A proof is a string that convinces us of the truth of some mathematical statement $x$

- $x$ is a satisfiable formula
- The Pythagorean Theorem is true

# What is a Proof?

## Traditional Definition

A proof is a string that convinces us of the truth of some mathematical statement $x$

- $x$ is a satisfiable formula
- The Pythagorean Theorem is true
- ...

# What is a Proof?

## Traditional Definition

A proof is a string that convinces us of the truth of some mathematical statement $x$

- $x$ is a satisfiable formula
- The Pythagorean Theorem is true
- ...

## New Definition

A proof is any process at the end of which one party (the prover) can convince the other party (the verifier) of the truth of some statement $x$

# What is a Proof?

## Traditional Definition

A proof is a string that convinces us of the truth of some mathematical statement $x$

- $x$ is a satisfiable formula
- The Pythagorean Theorem is true
- ...

## New Definition

A proof is any process at the end of which one party (the prover) can convince the other party (the verifier) of the truth of some statement $x$

- A proof doesn't have to be a string

# What is a Proof?

## Traditional Definition

A proof is a string that convinces us of the truth of some mathematical statement $x$

- $x$ is a satisfiable formula
- The Pythagorean Theorem is true
- ...

## New Definition

A proof is any process at the end of which one party (the prover) can convince the other party (the verifier) of the truth of some statement $x$

- A proof doesn't have to be a string
- Can be an interactive procedure

# What is a Proof?

## Traditional Definition

A proof is a string that convinces us of the truth of some mathematical statement $x$

- $x$ is a satisfiable formula
- The Pythagorean Theorem is true
- . . .

## New Definition

A proof is any process at the end of which one party (the prover) can convince the other party (the verifier) of the truth of some statement $x$

- A proof doesn't have to be a string
- Can be an interactive procedure
- The verifier (and prover) can use randomness to decide whether to accept

# An Example – Aladdin's Cave

# Why Interactive Proofs?

**Question**

We already know that all $L \in \mathcal{NP}$ have non-interactive proofs. Why study interactive ones?

# Why Interactive Proofs?

**Question**

We already know that all $L \in \mathcal{NP}$ have non-interactive proofs. Why study interactive ones?

- Can give proofs for languages not in $\mathcal{NP}$

# Why Interactive Proofs?

**Question**

We already know that all $L \in \mathcal{NP}$ have non-interactive proofs. Why study interactive ones?

- Can give proofs for languages not in $\mathcal{NP}$
- Interactive proofs can be much more efficient (e.g., shorter) than non-interactive ones

# Why Interactive Proofs?

**Question**

We already know that all $L \in \mathcal{NP}$ have non-interactive proofs. Why study interactive ones?

- Can give proofs for languages not in $\mathcal{NP}$
- Interactive proofs can be much more efficient (e.g., shorter) than non-interactive ones
- Can have additional properties that traditional proofs cannot satisfy.
  - Zero-knowledge

# Outline

# The Class $\mathcal{IP}$

**Definition**

$L \in \mathcal{IP}$ if there exist a pair of interactive algorithms $(P, V)$ with $V$ being poly-time (in $|x|$) s.t.

# The Class $\mathcal{IP}$

**Definition**

$L \in \mathcal{IP}$ if there exist a pair of interactive algorithms $(P, V)$ with $V$ being poly-time (in $|x|$) s.t.

1. (Completeness) If $x \in L$, then $\Pr[\langle P, V\rangle(x) = 1] = 1$

# The Class $\mathcal{IP}$

## Definition

$L \in \mathcal{IP}$ if there exist a pair of interactive algorithms $(P, V)$ with $V$ being poly-time (in $|x|$) s.t.

1. (Completeness) If $x \in L$, then $\Pr[\langle P, V\rangle(x) = 1] = 1$
2. (Soundness) If $x \notin L$, then for any (possibly unbounded) $P^*$, we have $\Pr[\langle P^*, V\rangle(x) = 1] \leq 1/2$

# The Class $\mathcal{IP}$

**Definition**

$L \in \mathcal{IP}$ if there exist a pair of interactive algorithms $(P, V)$ with $V$ being poly-time (in $|x|$) s.t.

1. (Completeness) If $x \in L$, then $\Pr[\langle P, V\rangle(x) = 1] = 1$
2. (Soundness) If $x \notin L$, then for any (possibly unbounded) $P^*$, we have $\Pr[\langle P^*, V\rangle(x) = 1] \leq 1/2$

Examples:

# The Class $\mathcal{IP}$

## Definition

$L \in \mathcal{IP}$ if there exist a pair of interactive algorithms $(P, V)$ with $V$ being poly-time (in $|x|$) s.t.

1. (Completeness) If $x \in L$, then $\Pr[\langle P, V\rangle(x) = 1] = 1$
2. (Soundness) If $x \notin L$, then for any (possibly unbounded) $P^*$, we have $\Pr[\langle P^*, V\rangle(x) = 1] \leq 1/2$

Examples:

- Aladdin's cave example from earlier

# The Class $\mathcal{IP}$

## Definition

$L \in \mathcal{IP}$ if there exist a pair of interactive algorithms $(P, V)$ with $V$ being poly-time (in $|x|$) s.t.

1. (Completeness) If $x \in L$, then $\Pr[\langle P, V\rangle(x) = 1] = 1$
2. (Soundness) If $x \notin L$, then for any (possibly unbounded) $P^*$, we have $\Pr[\langle P^*, V\rangle(x) = 1] \leq 1/2$

Examples:

- Aladdin's cave example from earlier
- $\mathcal{P} \subseteq \mathcal{IP}$

# The Class $\mathcal{IP}$

## Definition

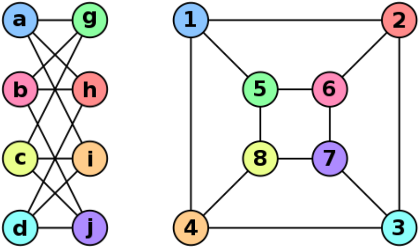
$L \in \mathcal{IP}$ if there exist a pair of interactive algorithms $(P, V)$ with $V$ being poly-time (in $|x|$) s.t.

1. (Completeness) If $x \in L$, then $\Pr[\langle P, V\rangle(x) = 1] = 1$
2. (Soundness) If $x \notin L$, then for any (possibly unbounded) $P^*$, we have $\Pr[\langle P^*, V\rangle(x) = 1] \leq 1/2$

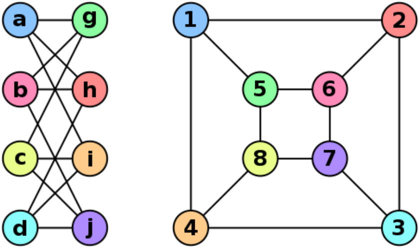
Examples:

- Aladdin's cave example from earlier
- $\mathcal{P} \subseteq \mathcal{IP}$
- $\mathcal{NP} \subseteq \mathcal{IP}$

# Another Example – Graph Isomorphism



# Another Example – Graph Isomorphism




**Claim**

Graph Isomorphism $\in \mathcal{IP}$

# Graph Non-Isomorphism

## Question

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

# Graph Non-Isomorphism

**Question**

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

# Graph Non-Isomorphism

**Question**

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0, 1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$

# Graph Non-Isomorphism

**Question**

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0,1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$
2. $P$ determines if $G^*$ is isomorphic to $G_0$ and sends $b' = 0$ if so, or $b' = 1$ otherwise back to $V$

# Graph Non-Isomorphism

**Question**

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0, 1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$
2. $P$ determines if $G^*$ is isomorphic to $G_0$ and sends $b' = 0$ if so, or $b' = 1$ otherwise back to $V$
3. $V$ accepts if $b' = b$

# Graph Non-Isomorphism

**Question**

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0, 1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$
2. $P$ determines if $G^*$ is isomorphic to $G_0$ and sends $b' = 0$ if so, or $b' = 1$ otherwise back to $V$
3. $V$ accepts if $b' = b$

Why This Works:

# Graph Non-Isomorphism

**Question**

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0, 1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$
2. $P$ determines if $G^*$ is isomorphic to $G_0$ and sends $b' = 0$ if so, or $b' = 1$ otherwise back to $V$
3. $V$ accepts if $b' = b$

Why This Works:

1. (Completeness) Suppose that $G_0$ and $G_1$ are not isomorphic.

# Graph Non-Isomorphism

**Question**

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0,1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$
2. $P$ determines if $G^*$ is isomorphic to $G_0$ and sends $b' = 0$ if so, or $b' = 1$ otherwise back to $V$
3. $V$ accepts if $b' = b$

Why This Works:

1. (Completeness) Suppose that $G_0$ and $G_1$ are not isomorphic.
    - Then $G^*$ can only be isomorphic to one of the two graphs

# Graph Non-Isomorphism

## Question

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0,1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$
2. $P$ determines if $G^*$ is isomorphic to $G_0$ and sends $b' = 0$ if so, or $b' = 1$ otherwise back to $V$
3. $V$ accepts if $b' = b$

Why This Works:

1. (Completeness) Suppose that $G_0$ and $G_1$ are not isomorphic.
    - Then $G^*$ can only be isomorphic to one of the two graphs
    - $P$ can perfectly determine which one this is

# Graph Non-Isomorphism

## Question

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0, 1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$
2. $P$ determines if $G^*$ is isomorphic to $G_0$ and sends $b' = 0$ if so, or $b' = 1$ otherwise back to $V$
3. $V$ accepts if $b' = b$

Why This Works:

1. (Completeness) Suppose that $G_0$ and $G_1$ are not isomorphic.
    - Then $G^*$ can only be isomorphic to one of the two graphs
    - $P$ can perfectly determine which one this is
    - So $\Pr[b' = b] = 1$

# Graph Non-Isomorphism

**Question**

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0,1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$
2. $P$ determines if $G^*$ is isomorphic to $G_0$ and sends $b' = 0$ if so, or $b' = 1$ otherwise back to $V$
3. $V$ accepts if $b' = b$

Why This Works:

1. (Completeness) Suppose that $G_0$ and $G_1$ are not isomorphic.
   - Then $G^*$ can only be isomorphic to one of the two graphs
   - $P$ can perfectly determine which one this is
   - So $\Pr[b' = b] = 1$
2. (Soundness) Suppose that $G_0$ and $G_1$ are isomorphic

# Graph Non-Isomorphism

## Question

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0,1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$
2. $P$ determines if $G^*$ is isomorphic to $G_0$ and sends $b' = 0$ if so, or $b' = 1$ otherwise back to $V$
3. $V$ accepts if $b' = b$

Why This Works:

1. (Completeness) Suppose that $G_0$ and $G_1$ are not isomorphic.
    - Then $G^*$ can only be isomorphic to one of the two graphs
    - $P$ can perfectly determine which one this is
    - So $\Pr[b' = b] = 1$
2. (Soundness) Suppose that $G_0$ and $G_1$ are isomorphic
    - Then $G^*$ is isomorphic to both $G_0$ and $G_1$

# Graph Non-Isomorphism

**Question**

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0, 1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$
2. $P$ determines if $G^*$ is isomorphic to $G_0$ and sends $b' = 0$ if so, or $b' = 1$ otherwise back to $V$
3. $V$ accepts if $b' = b$

Why This Works:

1. (Completeness) Suppose that $G_0$ and $G_1$ are not isomorphic.
   - Then $G^*$ can only be isomorphic to one of the two graphs
   - $P$ can perfectly determine which one this is
   - So $\Pr[b' = b] = 1$
2. (Soundness) Suppose that $G_0$ and $G_1$ are isomorphic
   - Then $G^*$ is isomorphic to both $G_0$ and $G_1$
   - $P$ has no way to tell which one $V$ started from

# Graph Non-Isomorphism

**Question**

How can we prove that two graphs $G_0$ and $G_1$ are NOT isomorphic?

The Protocol:

1. $V$ chooses $b \leftarrow \{0, 1\}$, and applies a random permutation $\pi$ to the vertices of $G_b$ and sends this graph $G^*$ to $P$
2. $P$ determines if $G^*$ is isomorphic to $G_0$ and sends $b' = 0$ if so, or $b' = 1$ otherwise back to $V$
3. $V$ accepts if $b' = b$

Why This Works:

1. (Completeness) Suppose that $G_0$ and $G_1$ are not isomorphic.
   - Then $G^*$ can only be isomorphic to one of the two graphs
   - $P$ can perfectly determine which one this is
   - So $\Pr[b' = b] = 1$
2. (Soundness) Suppose that $G_0$ and $G_1$ are isomorphic
   - Then $G^*$ is isomorphic to both $G_0$ and $G_1$
   - $P$ has no way to tell which one $V$ started from
   - Thus, $\Pr[b' = b] = 1/2$

# Important Takeaways

# Important Takeaways

- GNI $\in$ co-$\mathcal{NP}$

# Important Takeaways

- GNI $\in$ co-$\mathcal{NP}$
- It is not believed that there is a short witness $w$ s.t. $V((G_0, G_1), w) = 1$ if $G_0$ and $G_1$ are not isomorphic. I.e., GNI $\notin \mathcal{NP}$

# Important Takeaways

- GNI $\in$ co-$\mathcal{NP}$
- It is not believed that there is a short witness $w$ s.t. $V((G_0, G_1), w) = 1$ if $G_0$ and $G_1$ are not isomorphic. I.e., GNI $\notin \mathcal{NP}$
- The power of interaction and randomness has allowed us to do what we couldn't do before

# Boosting Soundness

So far, we defined soundness as:

$$\Pr[\langle P^*, v\rangle(x) = 1] \leq 1/2$$

# Boosting Soundness

So far, we defined soundness as:

$$\Pr[\langle P^*, v\rangle(x) = 1] \leq 1/2$$

What if we don't want malicious prover to win so often?

# Boosting Soundness

So far, we defined soundness as:

$$\Pr[\langle P^*, v\rangle(x) = 1] \leq 1/2$$

What if we don't want malicious prover to win so often?

## Soundness Amplification

# Boosting Soundness

So far, we defined soundness as:

$$\Pr[\langle P^*, v\rangle(x) = 1] \leq 1/2$$

What if we don't want malicious prover to win so often?

## Soundness Amplification

1. Run the proof $n$ times sequentially on same input $x$, but different randomness

# Boosting Soundness

So far, we defined soundness as:

$$\Pr[\langle P^*, v \rangle(x) = 1] \leq 1/2$$

What if we don't want malicious prover to win so often?

## Soundness Amplification

1. Run the proof $n$ times sequentially on same input $x$, but different randomness
2. Accept if ALL proofs accept

# Boosting Soundness

So far, we defined soundness as:

$$\Pr[\langle P^*, v\rangle(x) = 1] \leq 1/2$$

What if we don't want malicious prover to win so often?

## Soundness Amplification

1. Run the proof $n$ times sequentially on same input $x$, but different randomness
2. Accept if ALL proofs accept
3. $P^*$ wins with probability $\leq 1/2$ in each run, so

$$\Pr[\langle P^*, V\rangle(x) = 1] \leq 1/2^n$$

# Outline

# Another Example – Polynomial Identity Testing

## Polynomial

A polynomial is an equation in one-variable

$$f(x) = x^3 - 6x^2 + 11x - 7 = (x-1)(x-2)(x-3)$$

# Another Example – Polynomial Identity Testing

## Polynomial

A polynomial is an equation in one-variable

$$f(x) = x^3 - 6x^2 + 11x - 7 = (x-1)(x-2)(x-3)$$

Properties:

- A root of a polynomial $f$ is a value $x$ s.t. $f(x) = 0$

# Another Example – Polynomial Identity Testing

## Polynomial

A polynomial is an equation in one-variable

$$\begin{aligned} f(x) &= x^3 - 6x^2 + 11x - 7 = \\ &\quad (x-1)(x-2)(x-3) \end{aligned}$$

Properties:

- A root of a polynomial $f$ is a value $x$ s.t. $f(x) = 0$
- The degree of a polynomial $f(x)$ is the maximum exponent in $f$

# Another Example – Polynomial Identity Testing

## Polynomial

A polynomial is an equation in one-variable

$$f(x) = x^3 - 6x^2 + 11x - 7 = (x-1)(x-2)(x-3)$$

Properties:

- A root of a polynomial $f$ is a value $x$ s.t. $f(x) = 0$
- The degree of a polynomial $f(x)$ is the maximum exponent in $f$
- A polynomial of degree $d$ has at most $d$ roots

# Another Example – Polynomial Identity Testing

## Polynomial

A polynomial is an equation in one-variable

$$f(x) = x^3 - 6x^2 + 11x - 7 = (x-1)(x-2)(x-3)$$

Properties:

- A root of a polynomial $f$ is a value $x$ s.t. $f(x) = 0$
- The degree of a polynomial $f(x)$ is the maximum exponent in $f$
- A polynomial of degree $d$ has at most $d$ roots
  - unless $f(x) = 0$

# Another Example – Polynomial Identity Testing

## PIT Problem

# Another Example – Polynomial Identity Testing

## PIT Problem

- Prover $P$ chooses a degree $d$ polynomial $f$ and wants to prove that

$$\forall x, f(x) = 0$$

# Another Example – Polynomial Identity Testing

## PIT Problem

- Prover $P$ chooses a degree $d$ polynomial $f$ and wants to prove that

$$\forall x, f(x) = 0$$

- Completeness: If $f(x) = 0$, $V$ should accept after interacting with $P$
- Soundness: If $f(x) \neq 0$, $V$ should reject*

# Another Example – Polynomial Identity Testing

## PIT Problem

- Prover $P$ chooses a degree $d$ polynomial $f$ and wants to prove that

$$\forall x, f(x) = 0$$

- Completeness: If $f(x) = 0$, $V$ should accept after interacting with $P$
- Soundness: If $f(x) \neq 0$, $V$ should reject*

The rules:

- $V$ is allowed to query $f(x)$ at points $x$ of its choice

# Another Example – Polynomial Identity Testing

## PIT Problem

- Prover $P$ chooses a degree $d$ polynomial $f$ and wants to prove that

$$\forall x, f(x) = 0$$

- Completeness: If $f(x) = 0$, $V$ should accept after interacting with $P$
- Soundness: If $f(x) \neq 0$, $V$ should reject*

The rules:

- $V$ is allowed to query $f(x)$ at points $x$ of its choice
- $P$ is required to answer honestly, but

# Another Example – Polynomial Identity Testing

## PIT Problem

- Prover $P$ chooses a degree $d$ polynomial $f$ and wants to prove that

$$\forall x, f(x) = 0$$

- Completeness: If $f(x) = 0$, $V$ should accept after interacting with $P$
- Soundness: If $f(x) \neq 0$, $V$ should reject*

The rules:

- $V$ is allowed to query $f(x)$ at points $x$ of its choice
- $P$ is required to answer honestly, but
- $P$ knows $V$'s strategy (i.e., how he chooses the points $x$)

# Another Example – Polynomial Identity Testing

## PIT Problem

- Prover $P$ chooses a degree $d$ polynomial $f$ and wants to prove that

$$\forall x, f(x) = 0$$

- Completeness: If $f(x) = 0$, $V$ should accept after interacting with $P$
- Soundness: If $f(x) \neq 0$, $V$ should reject*

The rules:

- $V$ is allowed to query $f(x)$ at points $x$ of its choice
- $P$ is required to answer honestly, but
- $P$ knows $V$'s strategy (i.e., how he chooses the points $x$)

Question: What should $V$ do? How many queries does he need?

# Another Example – Polynomial Identity Testing

## PIT Problem

- Prover $P$ chooses a degree $d$ polynomial $f$ and wants to prove that

$$\forall x, f(x) = 0$$

- Completeness: If $f(x) = 0$, $V$ should accept after interacting with $P$
- Soundness: If $f(x) \neq 0$, $V$ should reject*

The rules:

- $V$ is allowed to query $f(x)$ at points $x$ of its choice
- $P$ is required to answer honestly, but
- $P$ knows $V$'s strategy (i.e., how he chooses the points $x$)

Question: What should $V$ do? How many queries does he need?

- Suppose that $V$ is deterministic.
- What if you allow $V$ to be randomized?

# The Power of Randomness in Interactive Proofs

- By allowing $V$ to be randomized we went from $d+1$ queries to 1 query

# The Power of Randomness in Interactive Proofs

- By allowing $V$ to be randomized we went from $d + 1$ queries to 1 query
- We have strong evidence that derandomizing PIT will be very hard – it implies strong complexity results that we have no idea how to prove

# The Power of Randomness in Interactive Proofs

- By allowing $V$ to be randomized we went from $d + 1$ queries to 1 query
- We have strong evidence that derandomizing PIT will be very hard – it implies strong complexity results that we have no idea how to prove

**Take away**

Randomness and interaction are key to the power of $\mathcal{IP}$

# The Power of Randomness in Interactive Proofs

- By allowing $V$ to be randomized we went from $d + 1$ queries to 1 query
- We have strong evidence that derandomizing PIT will be very hard – it implies strong complexity results that we have no idea how to prove

**Take away**

Randomness and interaction are key to the power of $\mathcal{IP}$

# Next Week

We have seen the power of interactive proofs in convincing a verifier of the truth of some statement.

**Question:**

What does the verifier learn from seeing the proof?