香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

SCHOOL OF DATA SCIENCE
數據科學學院

CSC 6011 - Theory of Computation

Lecture 20: Probabilistic Computation (cond.) & Interactive Proof

Time: Mon & Wed, 4:00 - 5:20 PM

Location: Teaching A, 107

Instructor: Tao LIN (林涛)

TA: Yefan GAO (高叶繁)

Fall 2026

Homework 4 has been posted.

Due: Dec 4 (Fri), 11:59 pm

Lecture 20

Last time:

- Probabilistic Computation, BPP
- Schwartz-Zippel Lemma
- Polynomial identity testing $\mathrm{EQ}_{\mathrm{poly}} \in$ BPP

Today:

- Checking matrix multiplication
- Interactive Proof System
- The class IP
- Graph isomorphism problem

Review: Probabilistic Computation

Defn: A PTM is equivalent to a deterministic TM with an extra read-only random tape containing independent, uniformly random bits.

Defn: BPP $= \{ A \mid$ some poly-time PTM decides A with error probability $\varepsilon \leq 1/3 \}$

Schwartz-Zippel algorithm for polynomial identity testing:

Let p_1, p_2 be polynomials with integer coefficients and degree $\leq d$ for each of m variables.
Let B be an upper bound on the absolute value of every coefficient (after expansion).

1. Choose a prime $q > \max\{3md, 2B\}$.
2. Pick $r_1, \ldots, r_m$ independently and uniformly from $\mathbb{F}_q = \{0, 1, \ldots, q-1\}$.
3. Evaluate $p_1(r)$ and $p_2(r)$ modulo q, without expanding the formulas.
4. If $p_1(r) = p_2(r)$, *accept*. Otherwise, *reject*.

The error probability is at most $\frac{md}{q} < \frac{1}{3}$.

Main idea: Searching for a witness.

Another Example: Checking Matrix Multiplication

Problem: Given $n \times n$ matrices A, B, and C, is $AB = C$?

$$\text{MatMul} = \{ \langle A, B, C \rangle \mid A, B, C \text{ are integer matrices satisfying } AB = C \}$$

Direct multiplication takes $O(n^3)$ time; latest progress is $O(n^{2.372})$.

Theorem: MatMul can be decided by a PTM in $O(kn^2)$ time with error probability $\frac{1}{2^k}$.

Freivalds' algorithm:

Repeat for k times:

1. Choose a vector $r \in \{0,1\}^n$ uniformly at random.
2. Compute ABr and Cr. [Computing $ABr = A(Br)$ only takes $O(n^2)$ time.]
3. If $ABr = Cr$, continue; otherwise, *reject*.

Accept.

For a "YES" instance $\langle A, B, C \rangle$: we always have $ABr = Cr$, so Pr[accept] = 1.

Error probability for "NO" instance

For a "NO" instance $\langle A, B, C \rangle$ (where $AB \neq C$):

Let $D = AB - C \neq 0$. There exists an entry $d_{ij} \neq 0$.

For the algorithm to accept, Dr should be 0. In particular, the i-th coordinate of Dr is 0:

$$(Dr)_i = \sum_{k=1}^{n} d_{ik} r_k = d_{ij} r_j + \sum_{k \neq j} d_{ik} r_k = 0 \quad \Longrightarrow \quad r_j = -\frac{1}{d_{ij}} \sum_{k \neq j} d_{ik} r_k$$

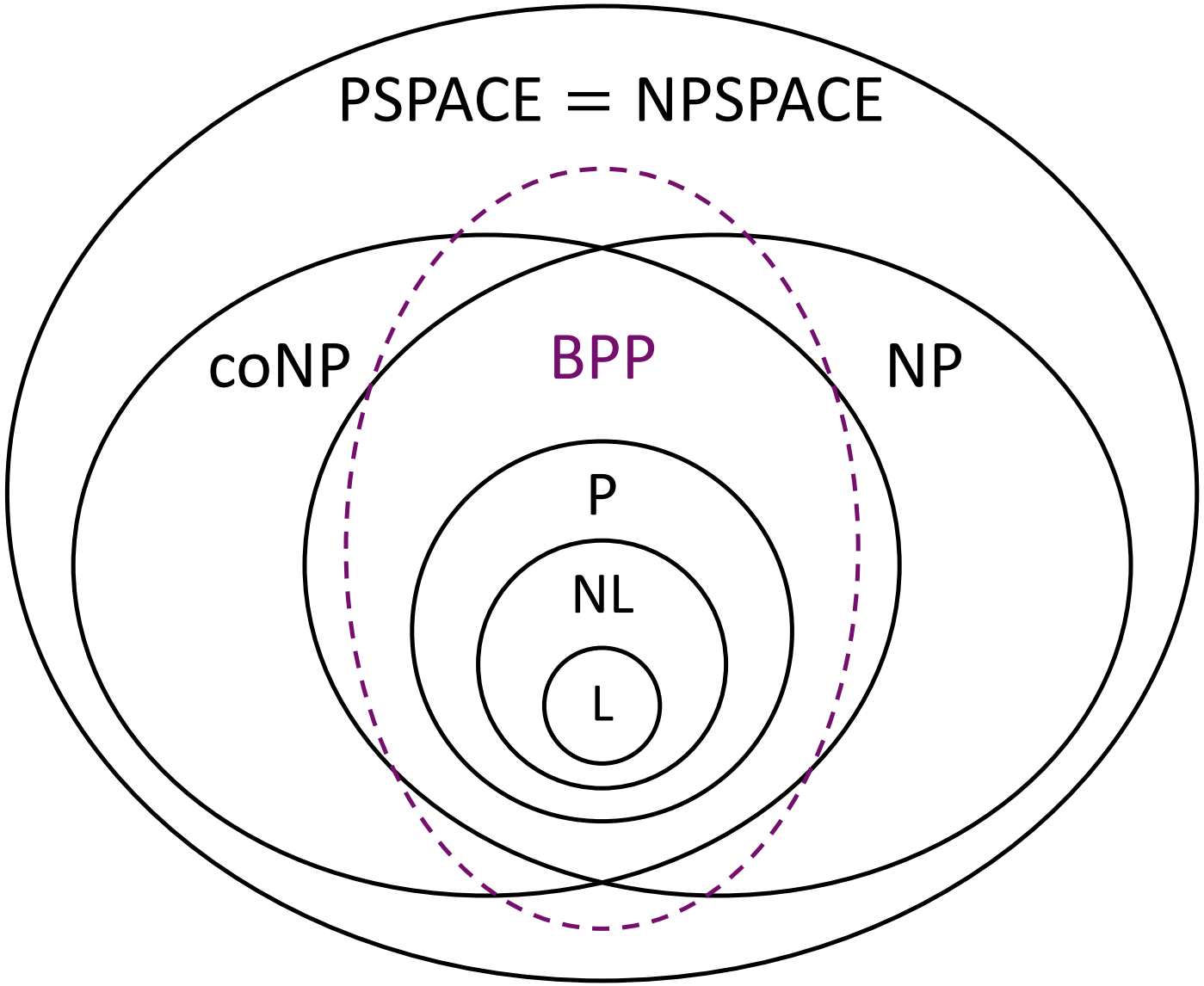
Fixing the values of r_k $(k \neq j)$, at most one of $r_j = 0$ and $r_j = 1$ can cause $(Dr)_i = 0$.

Thus, $\Pr[Dr = 0] \leq 1/2$.

Repeating k times independently, the accepting probability is at most $1/2^k$

Letting $k = 2$ gives error probability $1/4 < 1/3$, so $\mathrm{MatMul} \in$ BPP.

Complexity classes and BPP



Known inclusions

$L \subseteq NL \subseteq P$

$P \subseteq BPP \subseteq PSPACE$

Open questions

$P = BPP?$ (believed to be true)

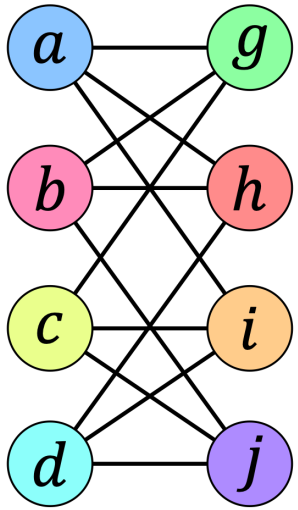
$NP \subseteq BPP?$

Interactive Proofs – Introduction

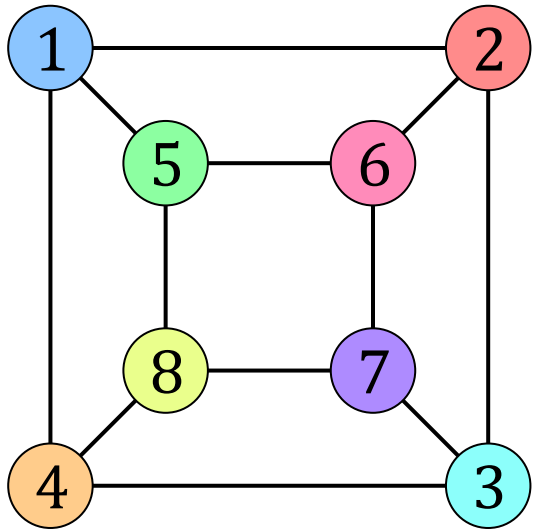
Problem: Graph isomorphism testing

Defn: Undirected graphs G and H are isomorphic if they are identical except for a re-labeling of the nodes: $\exists$ bijection $f: V_G \to V_H,\ (u,v) \in G \Leftrightarrow \big(f(u), f(v)\big) \in H$

G



H



An isomorphism f:

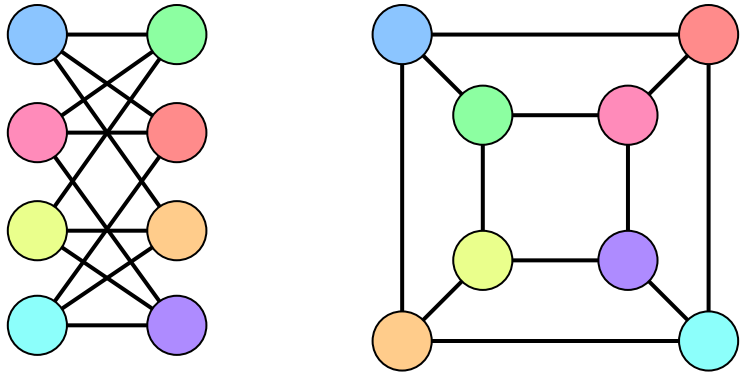
$f(a) = 1$ $\quad f(b) = 6$ $\quad f(c) = 8$ $\quad f(d) = 3$

$f(g) = 5$ $\quad f(h) = 2$ $\quad f(i) = 4$ $\quad f(j) = 7$

Interactive Proofs – Introduction

Problem: Graph isomorphism testing

Defn: Undirected graphs G and H are isomorphic if they are identical except for a re-labeling of the nodes: $\exists$ bijection $f: V_G \to V_H,\ (u,v) \in G \Leftrightarrow \big(f(u), f(v)\big) \in H$



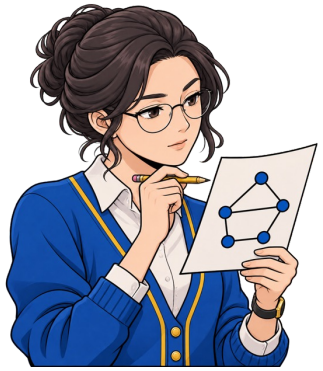
Defn: $\mathrm{ISO} = \{\, \langle G, H \rangle \mid G \text{ and } H \text{ are isomorphic graphs} \}$

$\mathrm{ISO} \in$ NP	the bijection is the certificate
$\mathrm{ISO} \in$ P ?	unknown
ISO is NP-complete ?	unknown
$\overline{\mathrm{ISO}} \in$ NP ?	unknown

$\mathrm{ISO} \in$ NP: a Prover can convince a poly-time Verifier that G and H are isomorphic (when true).

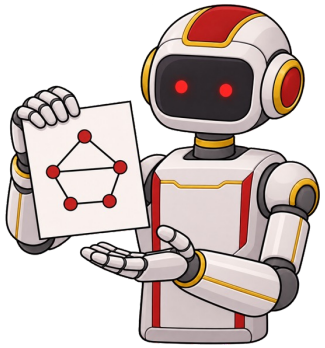
We will show: Even though whether $\overline{\mathrm{ISO}} \in$ NP is unknown,
a Prover can still convince a poly-time Verifier that G and H are not isomorphic (when true)
if we allow them to *interact* and *randomize*.

Interactive Proofs – informal model



Human Verifier

Probabilistic polynomial time TM



AI Prover

Unlimited computation

Human wants to know if graphs G and H are isomorphic.

- The Human asks the AI to figure out the answer.

- But the Human doesn't trust its answer. The Human must be convinced.

If the AI claims that G and H are isomorphic,
it can give the isomorphism and convince the Human.

But what if it claims that G and H are <u>not</u> isomorphic?

- The Human randomly and secretly picks G or H and relabels it, then sends the result to the AI.

- If AI can identify which graph the Human picked reliably (repeat this 100 times), then the Human is convinced.

Interactive Proof (System) – formal model

Two interacting parties

Verifier (V): Probabilistic polynomial time TM

Prover (P): Unlimited computational power

Both P and V see input w.

They exchange a polynomial number of polynomial-size messages.

Then V *accepts* or *rejects*.

Defn: Pr[(V $\leftrightarrow$ P) accepts w] = probability that V accepts when V interacts with P, given input w.

Defn: IP $= \{A \mid$ for some V and P (This P is an "honest" prover)

$w \in A \rightarrow$ Pr [(V $\leftrightarrow$ P) accepts w] $\geq {}^{2}/_{3}$

$w \notin A \rightarrow$ for any prover $\tilde{P}$, Pr [(V $\leftrightarrow \tilde{P}$) accepts w] $\leq {}^{1}/_{3}\}$

Think of $\tilde{P}$ as a "cheating" prover trying to make V accept when V shouldn't accept.

An amplification lemma can improve the error probability from ${}^{1}/_{3}$ to ${}^{1}/_{2^{\text{poly}(n)}}$

$\overline{\text{ISO}} \in$ IP

Theorem: $\overline{\text{ISO}} \in$ IP

Proof: Protocol for V and (the honest) P on input $\langle G, H \rangle$

1) Repeat twice:

2) V→P Randomly choose G or H and relabel to get K, then send K

3) P→V Compare K with G and H. Send "G" or "H" (V's choice in step 2)

4) V *accepts* if P was correct both times. Otherwise V *rejects*.

If G and H are not isomorphic, then P can determine which graph was chosen by V.

Thus Pr [($V \leftrightarrow$ P) accepts $\langle G, H \rangle$] $= 1 \geq {}^{2}/_{3}$

If G and H are isomorphic, then any $\tilde{\text{P}}$ has no way to tell which graph was chosen by V.

So $\tilde{\text{P}}$ has a 50% chance of answering correctly each time and a 25% chance of being correct twice.

Thus Pr [(V $\leftrightarrow \tilde{\text{P}}$) accepts $\langle G, H \rangle$] $= {}^{1}/_{4} < {}^{1}/_{3}$.

$\overline{\text{ISO}} \in$ IP

Theorem: $\overline{\text{ISO}} \in$ IP

Proof: Protocol for V and (the honest) P on input $\langle G, H \rangle$

1) Repeat twice:

2) V→P Randomly choose G or H and relabel to get K, then send K

3) P→V Compare K with G and H. Send "G" or "H" (V's choice in step 2)

4) V *accepts* if P was correct both times. Otherwise V *rejects*.

Mini-Quiz 20.1

Suppose we change the model to allow the Prover access to the Verifier's random choices. Now consider the same protocol as described above. What language does it describe?

a) $\{\langle G, H \rangle | \ G \neq H\}$

b) $\{\langle G, H \rangle |$ G and H are not isomorphic $\}$

c) $\{\langle G, H \rangle |$ G and H are any two graphs $\}$

d) $\emptyset$

Facts about IP

Mini-Quiz 20.2

Which of the following is true? Check all that apply:

a) NP ⊆ IP [V is deterministic]

b) BPP ⊆ IP [V ignores P]

c) IP ⊆ PSPACE

21–22	Interactive proofs, IP = PSPACE.

Theorem (Shamir, 1990): IP = PSPACE

We will prove IP ⊆ SPACE, and give the main idea for PSPACE ⊆ IP

Quick review of today

1. BPP algorithm for checking matrix multiplication

2. Introduced the interactive proof system

3. Defined the class IP

4. Showed $\overline{\text{ISO}} \in \text{IP}$

These slides are adapted from Michael Sipser, *18.404J Theory of Computation*, Fall 2020. Massachusetts Institute of Technology:

- Source: MIT OpenCourseWare https://ocw.mit.edu

Except for separately identified third-party material, this adapted presentation is licensed under CC BY-NC-SA 4.0.