CS5.203.1
Computational Complexity
— Lecture #22
Instructor: (5 May 21)
Prahladh Harsha

<u>Today</u>

Interactive Proofs
- Graph Non-isomorphism
- Formal Defn
- Permanent

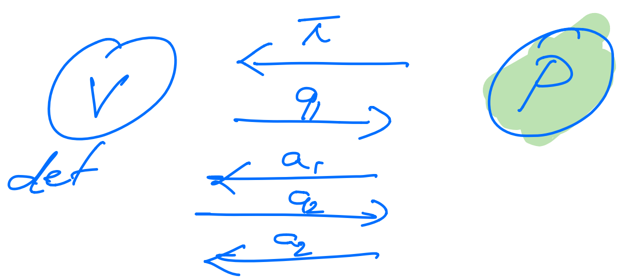
Interactive Proofs

"$x \in L$"

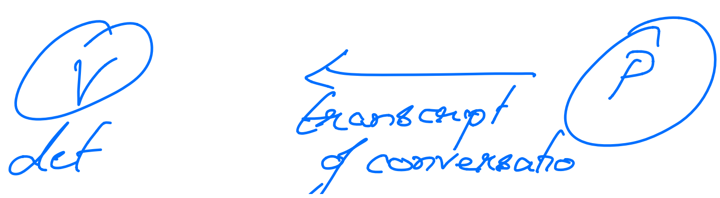
Verifier
(deterministic)
V

$\boxed{\pi}$

<u>Qn</u>: What if verifier had access to the prover and not just the proof?



Interaction w/ the prover



Does interaction increase power of verifier?
No; not really.

However, not true if verifier is randomized

Model: Verifier – randomized

– interaction w/ a powerful prover.

Toy Example:

Graph Non-Isomorphism

$GNI = \{(G_0, G_1) \mid G_0 \not\cong G_1\}$.

$\overline{GNI} = GI \in NP$; $GNI \in coNP$

(G_0, G_1)

V $\qquad$ P

1. $b \in_R \{0,1\}$

2. $\sigma \in_R S_n$ $(n = |V(G_0)| = |V(G_1)|)$

3. $H = \sigma(G_b)$

$V \xrightarrow{H} P$

$V \xleftarrow{c \in \{0,1\}} P$

4. Accept if $b = c$

$G_0 \not\cong G_1$. then the correct prover P

$\Pr[(V \leftrightarrow P)(G_0, G_1) = acc] = 1$

$G_0 \cong G_1$, $\forall$ prover P^*

$\Pr[(V \leftrightarrow P^*) = acc] = \frac{1}{2}$

Interactive Proofs.

$NP \subseteq IP$ | $GNI \in IP$

$BPP \subseteq$ | We don't know if $GNI \in NP$?

Formal Definition:

Recall defn of NP

$x \in L$

$V \xleftarrow{\pi} P$

$V(x, \pi) = acc/rej$

$L \in NP$ if $\exists$ a verifier V w/ the following properties.

(1) Efficiency : V is polytime computable.

(2) Completeness:
$x \in L \Rightarrow \exists \pi, \quad V(x, \pi) = acc$

(3) Soundness
$x \notin L \Rightarrow \forall \pi, \quad V(x, \pi) = rej$

Extend this defn to Interactive Proofs.
Model the verifier.

Inputs: x (original input)

R (randomness input)

Next message fn V

$(x, R, \underset{\text{transcript}}{\tau}) \longmapsto$ next message or acc/rej

x

R $\boxed{\text{V}}$ $\boxed{\text{P}}$

$\xrightarrow{m_1}$

$m_1 = V(x, R, \{\})$ $\xleftarrow{a_1}$

$m_2 = V(x, R, m_1 a_1)$ $\xrightarrow{m_2}$

$m_3 = V(x, R, m_1 m_2, a_1 a_2)$ $\xleftarrow{a_2}$

$\vdots$

Acc/Rej $\longleftarrow$

Prover is also a next message fn however w/ no efficiency restrictions.

$L \in IP$ (interactive proof) if there exists a randomized verifier (next message fn) V s.t.

(1) Efficiency

V runs in time poly($|x|$).

(2) Completeness:

$x \in L \Rightarrow \exists$ prover P

$$\Pr_R[(V \leftrightarrow P)(x; R) = acc] \geq 2/3$$

(3) Soundness:

$x \notin L \Rightarrow \forall$ provers P^*

$$\Pr_R[(V \leftrightarrow P)(x; R) = acc] \leq 1/3$$

Remarks: Definition of IP.

(0) $NP \subseteq IP$; $BPP \subseteq IP$

(1) The error (in defn) is 1/3, but can be reduced to exp(-n) just by repeating the above protocol sequentially O(n) times.

[An alternate repetition can be performed by asking qns in parallel. Also reduces error, but this requires a proof].

(2) The prover can be randomized but this does not give the prover any additional power.

(3) Private vs Public Coins:

Private: IP protocol in which the verifier does not reveal their randomness

Public: Verifier reveals the random coins.

Surprisingly, for every language that has a private-coins IP, there is an equivalent public-coins IP.

(4) Perfect Completeness Qn: Can $2/3 \to 1$

Any IP-protocol can be converted to one w/ perfect completeness
($\text{proof: } BPP \subseteq \Sigma_2^P$)

(5) Perfect Soundness Qn: Can $1/3 \to 0$.

Possibly No.
(then can make the verifier deterministic by the prover just sending the random coins that cause the verifier to accept in YES case)

perf-soundness-IP = det-IP = NP

Parameters IP protocol.

$L \in IP$

- # rounds. k-round protocol.
  $L \in IP[k]$; $IP = IP[poly]$

(eg: GNI $\in$ IP[2])

- Public vs Private Coins.

Private Coins: $L \in IP[\text{poly}]$

Public Coins: $L \in AM[\text{poly}]$
(Arthur-Merlin)

$AM \neq AM[\text{poly}]$

AM- always specify the #rounds

Public-coins IP/AM protocol for computing the permanent

$A = (a_{ij})_{\substack{i=1 \\ j=1}}^{n}$ $\qquad a_{ij} \in \mathbb{F}$ - finite field

$|\mathbb{F}| > 2n^3.$
(field is large enough)

$$\text{Perm} = \left\{ (\mathbb{F}, n, A, \alpha) \;\middle|\; \begin{array}{l} \mathbb{F} - \text{finite field.} \\ A - n \times n \text{ matrix} \\ A \in \mathbb{F}^{n \times n} \\ \text{perm}(A) = \alpha \end{array} \right\}.$$

$$\text{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

$$= \sum_{i=1}^{n} a_{1,i} \, \mathrm{Perm}(A_{1,i})$$

where $A_{1,i}$ – refer to the $(n-1) \times (n-1)$ matrix obtained by removing the 1st row & ith column.

Candidate IP-protocol:

(A, α)

V   P

$\alpha_1, \ldots \alpha_n$

$\alpha_1, \ldots \alpha_n$

$\alpha_i = \mathrm{Perm}(A_{1,i})$

1. Checks if

$$\alpha = \sum_{i=1}^{n} a_{1,i} \cdot \alpha_i$$

$i \in_R [n]$

i

$(A_{1,i}, \alpha_i)$

Reduced the problem to a $(n-1) \times (n-1)$ setting to check if $\mathrm{perm}(A_{1,i}) = \alpha_i$

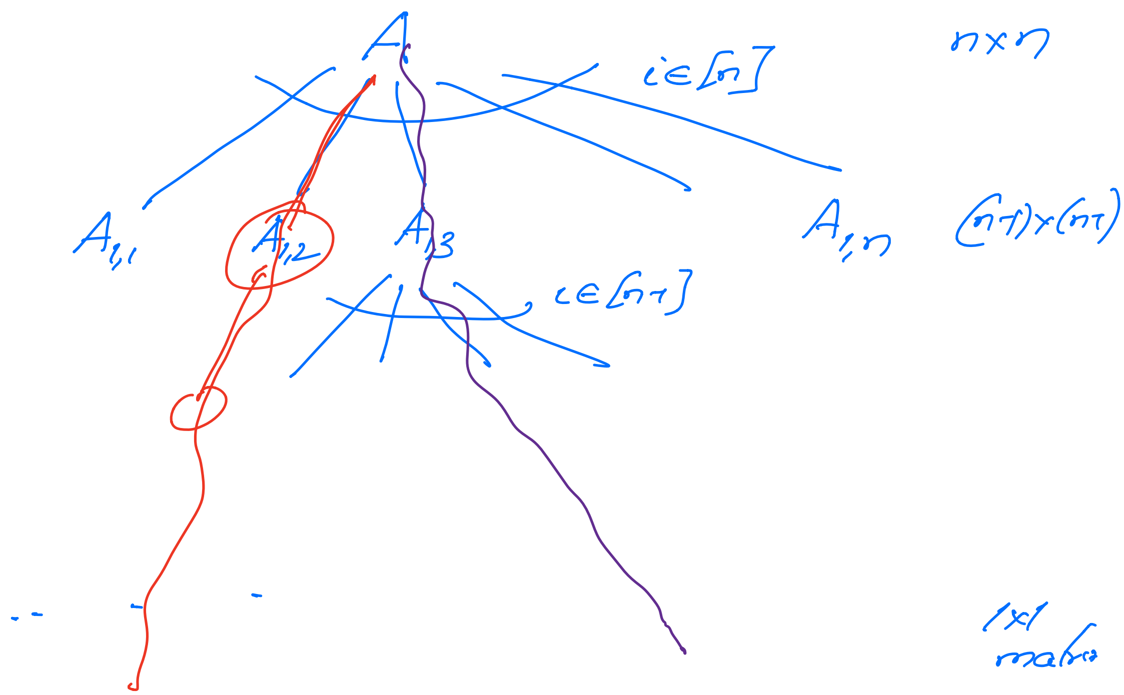
Qn: Is this a valid IP-protocol?

Efficiency ✓

Completeness ✓

Soundness: ???

Suppose $\text{perm}(A) \neq \alpha$.



Prover could cheat on just one of the paths.

Prob that the verifier catches the cheating prover $= \frac{1}{n!}$

Protocol is not sound.

Next time: modify protocol to improve rejecting prob from $\frac{1}{n!}$ to $\frac{1}{2}$ (or any constant).