Today

- Public Coins = Private Coins
- Probabilistically Checkable Proofs (PCPs)
  - MIP = NEXP
  - Proof Checking
  - Approximation.

Lecture 24

Computational Complexity

(30 Apr, '20)

Instructor: Prahladh Harsha

Public Coins Interactive Proofs

$AM[\underbrace{k(n)}_{\#\text{rounds}}]$

- Arthur Merlin Proof Systems

$x \in L$

V Arthur		P Merlin
₹ r_1	$\xrightarrow{r_1}$	
	$\xleftarrow{a_1}$	
₹ r_2	$\xrightarrow{r_2}$	
	$\xleftarrow{a_2}$	
₹ r_3	$\xrightarrow{r_3}$	
	$\xleftarrow{a_3}$	

$V(x; r_1 \ldots r_3, a_1 \ldots a_3) = \text{acc/rej}?$

Arthur publicly reveals all the randomness in each round.

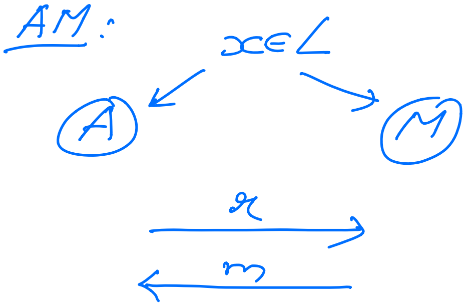
Properties:

1. $PSPACE \subseteq AM[\text{poly}]$
2. Goldwasser-Sipser
   $IP[k(n)] \subseteq AM[k(n)+2]$
3. $\forall$ constant k
   $AM[k] \subseteq AM[2] = AM$

Cor: GI is NP-complete $\Rightarrow$ PH $= \Sigma_2^P$.

AM, MA – Public Coins Interactive Proofs
2 rounds
(difference – who speaks first)

AM:



$A(x, r, m)$ = acc/rej

C: $x \in L$

$$\Pr_r[\exists m : A(x,r,m) = acc] \geq 2/3$$

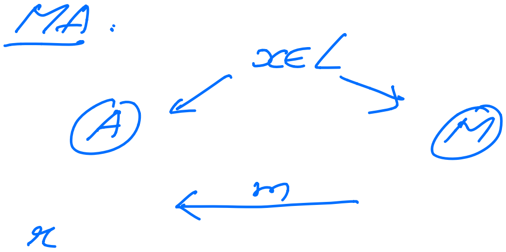
(can be increased to 1).

S: $x \notin L$

$$\Pr_r[\exists m \cdot A(x,r,m) = acc] \leq 1/3.$$

AM- randomized version of NP

MA:



$A(x, m, r)$ = acc/rej

C: $x \in L$

$\Downarrow$

$$\exists m, \Pr_r[A(x; m, r) = acc] \geq 1/3$$

$(= 1)$

S: $x \notin L$

$$\forall m, \Pr_r[A(x, m, r) = acc] \leq 1/3.$$

MA = NP except w/ a randomized verifier.

Round redn: MA $\subseteq$ AM. (AM[k] $\subseteq$ AM)

Public Coins = Private Coins
[Goldwasser-Sipser]

<u>Thm:</u> $IP[k(n)] \subseteq AM[k(n)+2]$.

<u>Special Case:</u> $GNI \subseteq AM$ - protocol.

Recall private coins protocol for $\overline{GNI}$

(G_0, G_1) $\qquad |V(G_0)| = |V(G_1)| = n$

$V \qquad P$

V		P
$b \in_R \{0,1\}$		
$\sigma \in_R S_n$		
$H \leftarrow \sigma(G_b)$	$\xrightarrow{H}$	Finds G_c s.t.
	$\xleftarrow{c}$	$G_c \cong H$
Ver acc if $b = c$.		

Private Coins.

(G_0, G_1)

$$S = \{H \mid H \cong G_0 \text{ or } H \cong G_1\}$$

Simplicity assume, both G_0 & G_1 have no automorphisms (no relabeling of G_0 & G_1 that is isomorphic to itself).

$G_0 \not\cong G_1 \implies |S| = 2n!$

$G_0 \cong G_1 \implies |S| = n!$

Remove the simplifying assumption.

$$S = \{(H, \pi) \mid \exists b \in \{0,1\},\ H \cong G_b \ \&\ \pi \in S_n,\ \text{s.t. } \pi(G_b) = G_b\}$$

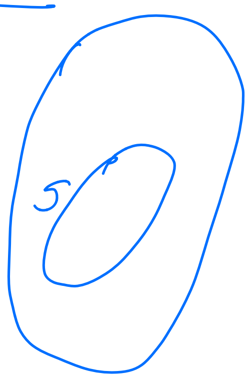
$$\left.\begin{array}{l} G_0 \not\cong G_1 \Rightarrow |S| = 2n! \\ G_0 \cong G_1 \Rightarrow |S| = n! \end{array}\right\}.$$

<u>Observations:</u>

① Size of S is different in YES & NO case

(In particular, large in YES case, small in NO case)

② Membership in S can be checked w/ a proof.



U = Graphs on vertices $\times S_n$

YES: $|S| \geq K$ $\quad (K = 2n!)$

NO: $|S| \leq K/2$.

Provers wants to convince that S is large via a public coins protocol

Idea: Use pairwise independent hash functions

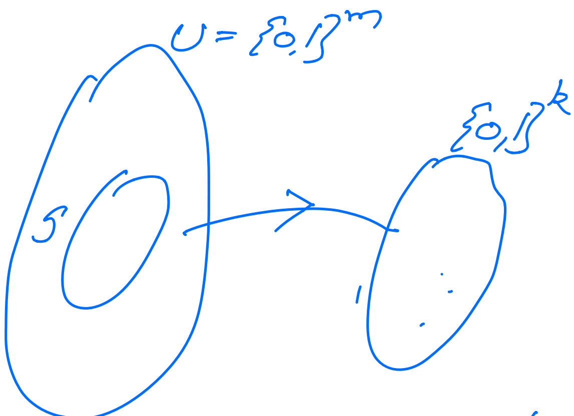
Set-Lower Bound Protocol

Input: m - $U = \{0,1\}^m$

$S \subseteq \{0,1\}^m$ specified implicitly using as NP oracle. (membership has short certificate)

K.

Goal: Distinguish $|S| \geq K$ or $|S| \leq K/2$.



Find k s.t. $2^{k-2} \leq K \leq 2^{k-1}$

$h: \{0,1\}^m \to \{0,1\}^k$

"S", K, m

(A) (M)

$h \in_R \mathcal{H}^k$ $\xrightarrow{\quad h \quad}$

$y \in_R \{0,1\}^k$ $\xrightarrow{\quad y \quad}$

$\xleftarrow{\quad s,\ \pi \quad}$ Find an $s \in S$ s.t. $h(s) = y$ & a proof π: "$s \in S$"

Arthur checks π is a valid proof & $h(s) = y$.

Soundness: $|S| \leq K/2$.

5

$$\Pr_{h,y}[\exists s \in S,\ h(s)=y] \le 1/4.$$

Pf. For every $h: \{0,1\}^m \to \{0,1\}^k$

$$\Pr_y[\exists s \in S,\ h(s)=y] \le \sum_{s\in S} \Pr_y[h(s)=y]$$

$$= |S|/2^k \le \frac{K}{2\cdot 2^k} \le \frac{1}{4}. \quad \boxtimes$$

Completeness:

$|S| \ge K$

$$\Pr_{h,y}[\exists s \in S,\ h(s)=y] \ge \ldots$$

Pf: Pick $S^* \subseteq S$ $\quad |S^*| = K$

For every y.

$$\Pr_h[\exists s \in S,\ h(s)=y]$$

$$\ge \sum_{s\in S} \Pr_h[h(s)=y]$$

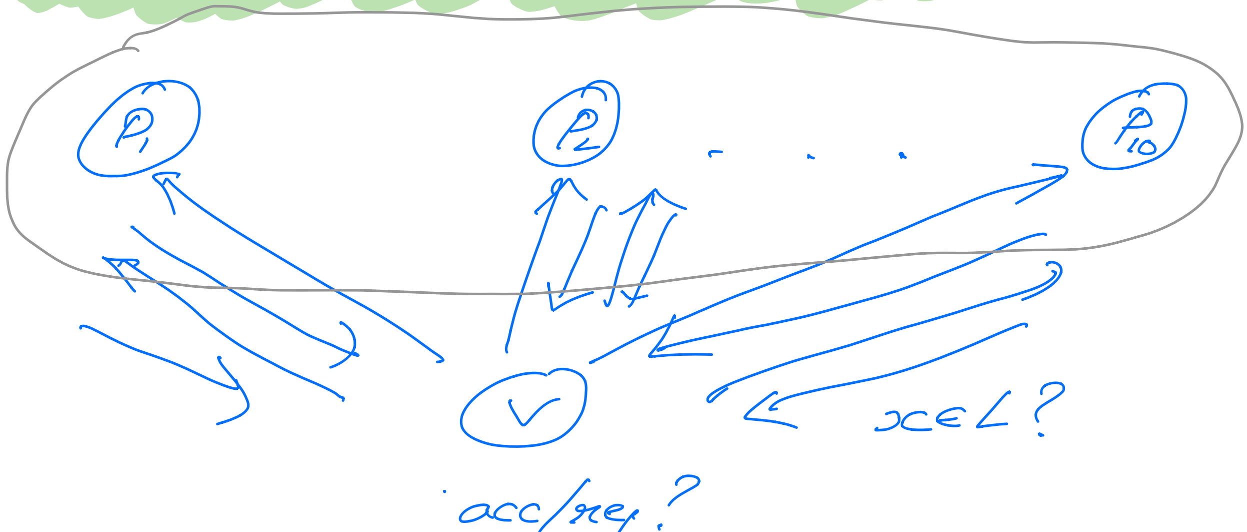
$$- \sum_{s, s'} \Pr_h[h(s)=y=h(s')]$$

$$= \frac{|S|}{2^k} - \binom{|S|}{2}\frac{1}{2^{2k}} \ge \frac{3p}{4} - \frac{p}{2^k}$$

$p = |S^*|/2^k.$

Hence, $GNI \in AM[2]$.

Probabilistically Checkable Proofs: (PCPs)



Does having multiple provers help?

(increase verifier's ability to check computation)

Possibly not, one can simulate multiple prover using a single prover.

— Not correct

Can simulate only if the provers are aware of each other's qns & answers.

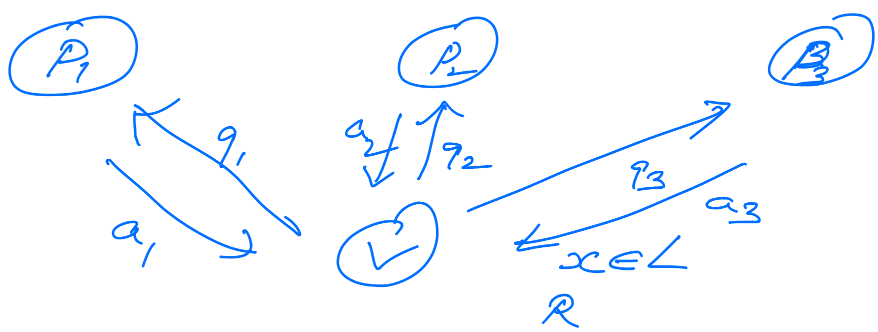
Multi prover - Can use one prover against another.

Multi provers: MIP - multi-prover Interactive Proofs

Surprising Result

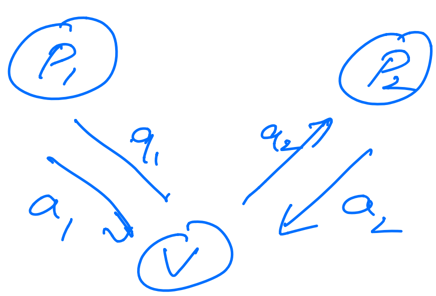
Thm [Babai Fortnow Lund] MIP = NEXP

- In fact true even in the following settings:
- 2 round protocols



$V(x, R, a_1, a_2, a_3) =$ acc/rej?

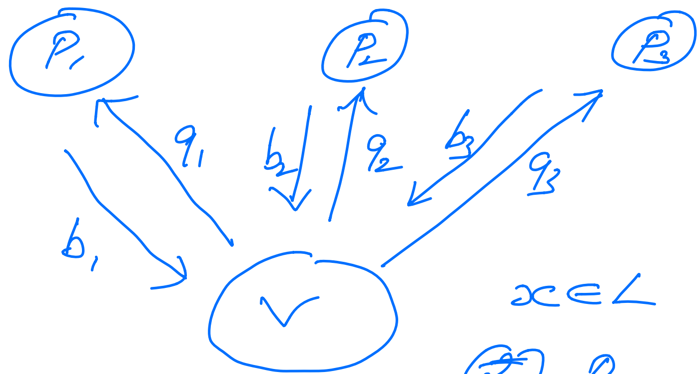
- 2 provers are suff.



- The answers can be just a bit each (in this case we need 3 provers).

[LFKN, Shamir, BF, RS, FS, BFL]

$MIP = NEXP$ $\qquad$ $L \in NEXP$



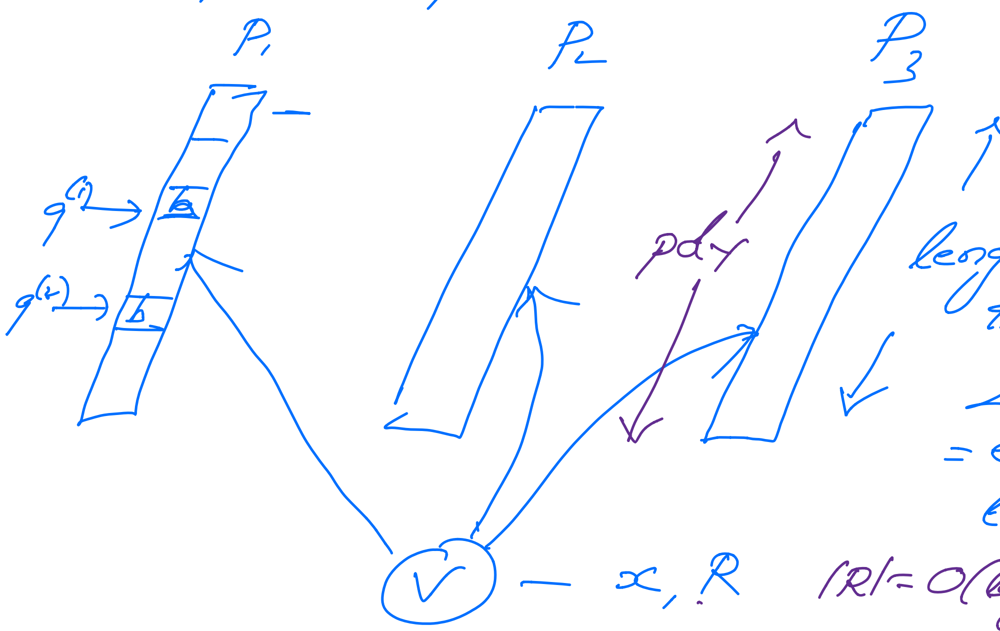
$P_i : \{q\} \to \{a\}$

$x \in L$

R – poly $\quad |R| = poly(|x|)$

$V(x; R, b_1, b_2, b_3) = acc/rej$?

View the 3 provers as tables.



length

#questions

$2^{|randomness|}$

= exponentially long table.

— x, R $\quad |R| = O(\log n)$

$V(x, R, b_1, b_2, b_3) = acc/rej$?

Qn: Can one scale the above to NP.

PCP Theorem [FGLSS, BFLS, AS, ALMSS]

MIP = NEXP can be scaled down to get PCP = NP