Today: Interactive Proofs

- AM
- IP
- IP ∈ PSPACE...

Classical Notion of Proof = NP

Theorem: \( T \leq E^* \)

Proof: \( \exists T \leq E^* \)

\( \exists \) proves \( T \) if \( \forall (T, \pi) = 1 \).

\( |\pi| = |T|^{O(1)} \leq V \) polytime

\( \Rightarrow \) "True-Theorem" in NP.
Interactive Proofs:

- Arise in context of cryptography;
- How do you prove X while keeping Y secret (modulo truth of X)?
- E.g. I am allowed to access account "madhu@mit.edu" without revealing my password is "blah-blah"
- Interactive proofs different from non-interactive ones.
Model

Prover \[ \leftarrow \frac{Q_i}{\pi_i} \rightarrow \text{Verifier} \]

\[ \leftarrow \frac{\exists_i}{\pi_i} \rightarrow \text{prob.} \]

\[ \leftarrow \frac{\pi_i}{\tau_i} \rightarrow \text{poly.} \]

\[ \vdash \leftarrow \frac{\pi_i}{\tau_i} \rightarrow \text{time} \]

\[ \pi_i^x \rightarrow \exists_i \left( Q_i, \pi_i, \pi_i \right) \]

\[ \text{L} \in \Pi \text{ if } \exists Q, V \]

\[ \uparrow \rightarrow \text{prob. poly time} \quad \text{s.t.} \]

for \( Q_i = Q(x, \pi_i, \pi_i, R, i) \)

\[ x \in L \Rightarrow \exists P \quad \text{s.t.} \quad V(x, \pi_i, \pi_i, R) = 1 \]

\[ x \notin L \Rightarrow \forall P \quad \cdots \quad \Rightarrow \text{w.p. } - C(n) \]

\[ x \notin L \Rightarrow \forall P \quad ... \quad \Rightarrow \leq C(n) \]
\[ T_i = P(x, z_1, \ldots, z_i). \]

Related notion

**Arthur-Merlin Proofs [Babai]**

Motivation: Some number-theoretic problems don't appear NP-hard.

\( L \subseteq \text{NP} \)

\( L \) not quite in \( \text{NP} \)

but close...

Formally: \( L \in AM \) ...

Merlin \[\begin{array}{c} \rightarrow \quad z_i \quad \text{Arthur} \\
\downarrow \quad T_i \end{array} \]

\( x \in L \) ---
Historical Issues

1. Coin of Verifier public or private
   \[\uparrow\] (Am) \[\uparrow\] (IP)

   \[\text{Goldwasser - Sipser}\]: Can convert private to public.

2. Error: one-sided or two-sided

   \[\text{Goldreich Mansour ?}\]: Can assume one-sided

3. \#Rounds: Constant? \(\Rightarrow\) (Am)

   \[\text{Poly} \ ?\ (IP)\].

   \(\text{Constant} \Rightarrow \text{2 rounds}\).
\([\text{GMR}]\) \( IP \subseteq \text{PSPACE} \)

\([\text{GmW}]\) \( \text{GNI} \in IP \Rightarrow \text{GNI} \in \text{AM} \)

Complexity Classes from \( IP \)

\( \text{NP} \subseteq \text{MA} \subseteq \text{AM} \subseteq \text{IP} = \text{IP}^\text{poly} \subseteq \text{PSPACE} \).

\[ \uparrow \]

prob. verifiere

Example:

\( \text{GNI} \subseteq \text{AM} \):

Merlin \( (G_0, G,) \)

\[ \xrightarrow{T\bar{I}(G_0)} \]

\( b' \)

Arthur

Pick \( T \), \( b \) at rand

\[ \text{accept if } b' = b. \]
1. \( IP \subseteq PSPACE \).

**Proof Idea:**

**Key Concept:** Optimal Prover

- \( P^* (x, q_1, \pi_1, q_2, \pi_2, \ldots, q_i) \):
  
  determines "optimal" answer to
  
  \( q_i \), given history \( x, q_1, \pi_1 \ldots q_{i-1}, \pi_{i-1} \).

- \( P^* (x, q_1, \ldots, q_i, \pi_i) \):
  
  computes prob. acceptance given
  
  history \( x, q_1, \ldots, q_i, \pi_i \).
  
  using \( P^* \) for answers to future questions.
Given big computation tree of max 2 arg nodes...

Can be computed in PSPACE.

One-Sided Error? Public coins?

Easy in poly rounds [Kilian]

Idea: Assume questions binary; a random coins revealed at end;

So fixing optimal prover \( P^* \);

Verifier’s view is a tree & he wants to know how many paths accepts
\( V \) arbitrary

\( V' \) public-win

\( V' \) one-sided ...

\( N \)

\( q = 0 \)

\( N_0 \)

\( q = 1 \)

\( N_1 \)

\( V' \) wants to know how many accepting paths at root.

\( P' \): Send \( N \) to \( V' \).

also for two children.

\( V' \): picks \( q \) with prob. \( \frac{N_q}{N} \) (if \( N = N_0 + N_1 \))
Analysis: Suppose prover claims $N > N'$

Then (inductively) caught w.p. $1 - \frac{N}{N'}$

Proof: Prob. Catching

\[
= \left(1 - \frac{N_0}{N_0'}\right) \cdot \frac{N_0'}{N'}
\]

\[
+ \left(1 - \frac{N_1}{N_1'}\right) \cdot \frac{N_1'}{N'}
\]

\[
= \frac{1}{N'} \left[ N_0' + N_1' - N_0 + N_1 \right]
\]

\[
= 1 - \frac{N}{N'} \quad \Box
\]

(Should verify base case...)
Harder version: \( O(1) \) rounds.

Will show it for 2 round protocol:

**Step 1**: Verifier picks \( R \in \{0, 1\}^r \)

: Computes \( q = q(R, x) \)

**Step 2**: Sends \( q \rightarrow \) prover

Prover answers with \( \Pi = \Pi(q) \)

**Step 3**: Accept if \( V(x, R, \Pi) = 1 \).

Want to convert this to public coin version.
1. Key Ingredient: Protocol for "approx counting"
   - Given: \( S \subseteq \{0,1,\}^* \)
     \[ \text{"membership in } S \text{" } \in \text{ AM} \]
   - \( |S| \geq f(n) \) \Rightarrow \text{ accept w.p. } 1
   - \( |S| \leq \frac{f(n)}{?} \) \Rightarrow \text{ accept w.p. } \leq \frac{1}{2}

2. Use above to prove that for fixed \((q,a)\)

\[ S_{q,a} = \{ R \mid q(R) = q \text{ and } \forall (x,r,a) \text{ accepts } ? \} \]

is large.
3. Can we use above to prove
\[ \exists S_q, a \text{ is large.} \]

Protocol for approx. Counting

Verifier \( V' \):
\[ \begin{align*}
\text{Picks } h: \{0,1\}^r \rightarrow 2 \cdot [f(n)] \\
\text{Asks for } R \text{ s.t. } & \text{ "RE S"} \\
\text{& "} h(R) = ? \text{"} \leftarrow \\
V' & \xrightarrow{h} P' \\
\end{align*} \]

\( P' \): Picks shifts \( S_{i \ldots S_t} \)
\[ t = O(\log f(n)) \]
\[ \leftarrow S_{i \ldots S_t} \]
\[ V' \xrightarrow{y} \]

Now \( P' \) proves to \( V' \) that

\[ \exists \ RES \ ( \exists \ proves \ this ) \]

\[ h(R \oplus S_i) = y \quad \text{for some} \quad i. \]

Combine [Valiant - Vazirani] \[ \cup \] [Haastenaire, Sipser]

Similar proofs yield

\[ IP[k] \leq AM[O(k)] \leq AM[O(\frac{k}{c})] \]

\[ AM[O(\frac{1}{c})] = AM[2] = BPE \quad \forall c. \]