Interactive Proof Systems

Interactive Proofs

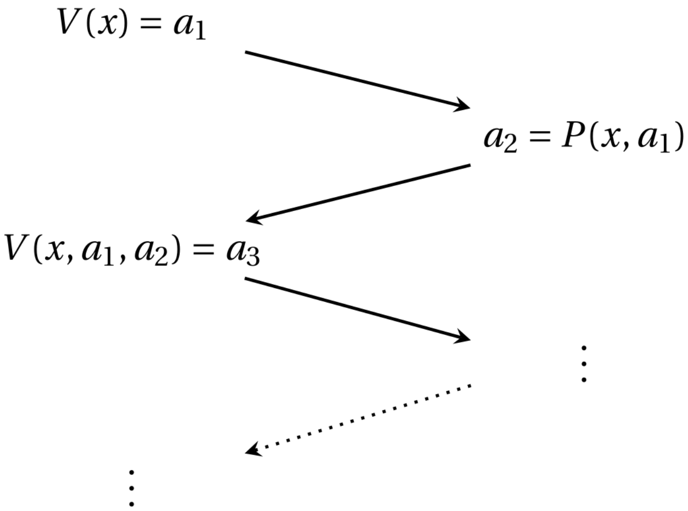
Mathematical Proof is related to Certificate definition of **NP**

Consists of :

- Prover with a unlimited computational power
- Verifier given limited computational resources

Formal Definition

- Let V,P : {0,1}* -> {0,1}* and k an integer
- A k-round interaction for input x
- V(x, a(1), a(2), …, a(k)) -> {0,1}



Warm-Up

- Alice as the prover with two socks
- Bob as a color-blinded verifier

- Do Alice's socks have different colors?

Warm-Up

Alice

Bob

Alice hands out the socks to bob

Bob holds one of the socks in each hands.

Alice has one last look and turns her back to bob

Bob tosses a coin; if it is head, he switches the socks between his left and right hand.

Alice turns around and tries to tell if the socks have switched

dIP

Language L has a k-round deterministic Proof system if there is polynomial deterministic TM V with input x and can have k-round interaction with a function P :

$$(Completeness)\ \ x \in L \Rightarrow \exists P : \{0,1\}^* \to \{0,1\}^* : out_V \langle V, P\rangle(x) = 1.$$

$$(Soundness)\ \ x \notin L \Rightarrow \forall P : \{0,1\}^* \to \{0,1\}^* : out_V \langle V, P\rangle(x) = 0.$$

- NP = dIP

- One can prove if we add randomness to the interactions, we can extend our power up to PSPACE.

Can we get stronger?

Interactive Proof Systems with Probabilistic Verifiers

Private Coin Scheme

Class IP

Language L is in **IP[k]** if there exists a k-round interaction between a Polynomial probabilistic TM V on input x and a Prover function P:

$$(Completeness)\;\; x \in L \Rightarrow \exists P : \{0,1\}^* \rightarrow \{0,1\}^* : \Pr[out_V\langle V, P\rangle(x) = 1] \geq 2/3.$$

$$(Soundness)\;\; x \notin L \Rightarrow \forall P : \{0,1\}^* \rightarrow \{0,1\}^* : \Pr[out_V\langle V, P\rangle(x) = 1] \leq 1/3.$$

$$We\ define\ \mathbf{IP} = \cup_{c\geq 1}\mathbf{IP}[n^c].$$

Probabilistic Interactive Proof for GNI

A simple Protocol for GNI G1 and G2:

P **V**

V chooses i in {1,2}. Then applies a random permutation U on Gi: H = Gi. Then Sends H to P

P figures out which of G1 or G2 was used to produce H. if Gj sends j.

Accepts if i = j

Public Coins and Class AM

Class AM

For every k, the complexity class **AM**[k] is defined as the subset of **IP**[k] obtained when we restrict the verifier's messages to be

- random bits and
- not allowing it to use any other random bits that are not contained in these messages.

- **AM = AM[2]** in which V sends a random string and P provides a response that V can verify it in polynomial.

Class AM Cont.

$$\mathbf{IP}[k] \subseteq \mathbf{AM}[k+2].$$

Instead of this general theorem known as Goldwasser-Sipser’87 , we try to prove:

$$\mathrm{GNI} \in \mathbf{AM}[2]$$

Set Lower Bound Protocol

For inputs G1 , G2

Consider set below :

- S = {H : H is isomorphic to G1 or G2}
- An n vertex graph has at most n! Equivalent graphs. For simplicity, assume G1 and G2 both have exactly n! Equivalent graphs.
- It is trivial that if G1 and G2 are isomorphic, |S| = n! And 2n! O.W.
- Assign D = |S * S * S * S|

Now Prover only needs to show |D| is at least 16(n!)^4 Or less than (n!)^4.

Set Lower Bound Protocol :

- D is a set s.t. Memberships in D can be certified by some u
- Both participants know a number K
- Prover goal is to convince V that |D| is at least K.
- Otherwise, V should reject with a high probability in |D| less than K/16.

Set Lower Bounds Cont.

Set Lower Bound Protocol Cont.

P | **V**

- Randomly picks a function h : D -> {1,2,...k/4}
- Pick y randomly from h.
- Send h,y to P

- Tries to find a x in D s.t h(x) = y
- Sends x and u for x membership in D for V

- Accepts iff h(x) = y and u is a valid certificate.

Completeness

|D| = k

The probability that h matches an element x from D to y :

$$1 - \Pr[h(x) \neq y \text{ for all } x \in S] = 1 - \left(1 - \frac{4}{K}\right)^K \geq 2/3.$$

Soundness

$|D| = k/16$

The probability that h matches an element x from D to y :

Is equal to ¼ since size of D covers at most ¼ of y domain which is {1,2,..., k/4 }

Zero-Knowledge Proof Systems

And why do we need them.

- Completeness
- Soundness
- Perfect Zero Knowledge
  - For all strategies V* exists a polytime algorithm S*, s.t.

$$out_{V^*}\langle P(x,u), V^*(x)\rangle \equiv S^*(x)$$

Definition

Relaxations

Statistical Zero Knowledge

- Small L1 Norm
- Statistical Distance
- **SZK**
  - Is believed to lie strictly between P and NP

Computational Zero Knowledge

- If one-way functions exists then all languages in NP have Computational Zero Knowledge Proofs.

Graph Isomorphism

Prover already knows a permutation $\pi : [n] \to [n]$ s.t. $G_0 = \pi(G_1)$ but wants to prove isomorphism without revealing it.

P **V**

$$\xrightarrow{\pi_1(G_1)}$$

random permutation $\pi_1 : [n] \to [n]$

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

$$\longleftarrow$$

If $b = 1$ π_1

$$\longrightarrow$$

If $b = 0$ $\pi_1 \circ \pi$

The Verifier accepts iff $H = \pi(G_b)$

- Correctness
- Soundness
  - If NO instance, H can’t be isomorphic to both!

S **V**

$b' \in_R \{0, 1\}$ random permutation π

$\longrightarrow$

$H = \pi(G_{b'})$

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

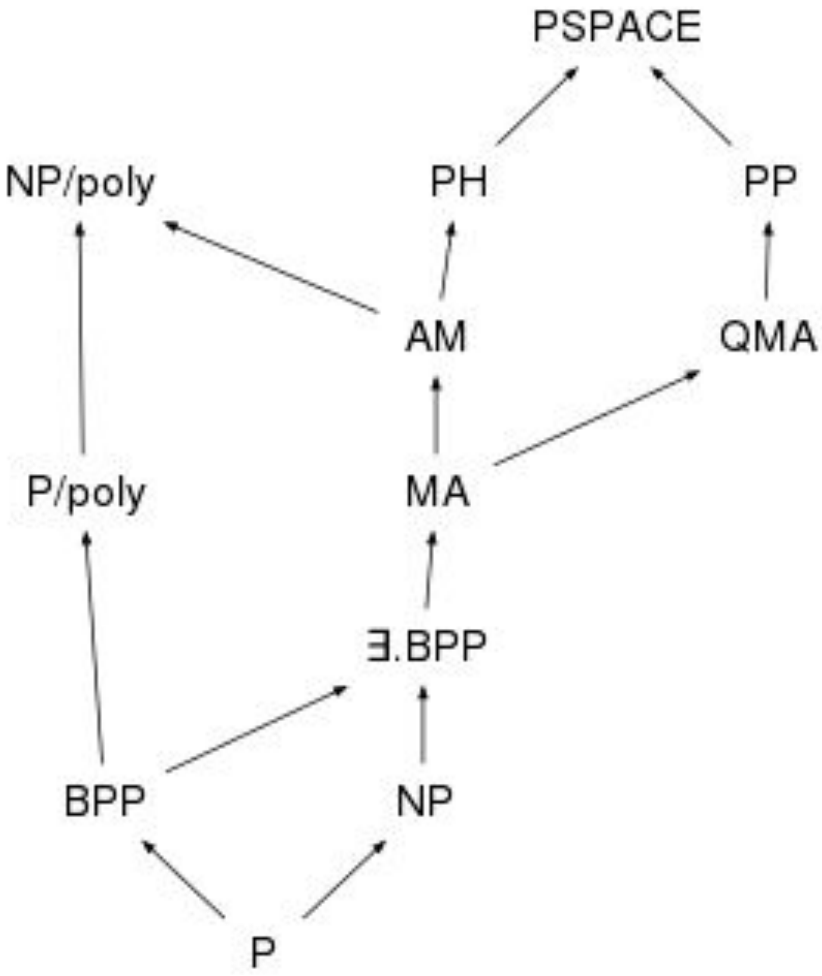
$\longleftarrow$

If $b = b'$ π

$\longrightarrow$

If $b \neq b'$ Start Over

Simulator


PSPACE
NP/poly
PH
PP
AM
QMA
P/poly
MA
∃.BPP
BPP
NP
P

Thank You

References

- Sanjeev Arora, Boaz Barak , Computational Complexity : A Modern Approach. 2009
- Ola Svensson, Computational Complexity Lecture Notes, Lectures No. 3, 7, 8. 2018
- Dieter van Melkebeek, Quantum Information Processing, Lecture Notes, Lecture No. 24. 2010