Lecture 01

Foundations of Probabilistic Proofs
2020.08.27
Alessandro Chiesa

Mathematical Proofs = NP

Recall the definition of NP:

verifier V

$L \in NP$ iff $\exists$ polynomial-time decider D s.t.

theorem ... proof π ... $V(x,\pi)=1$

(1) $\forall$ instance $x \in L$ $\exists$ witness w $D(x,w)=1$ } completeness

theorem ... proof π ... $V(x,\pi)=0$

(2) $\forall$ instance $x \notin L$ $\forall$ witness w $D(x,w)=0$ } soundness

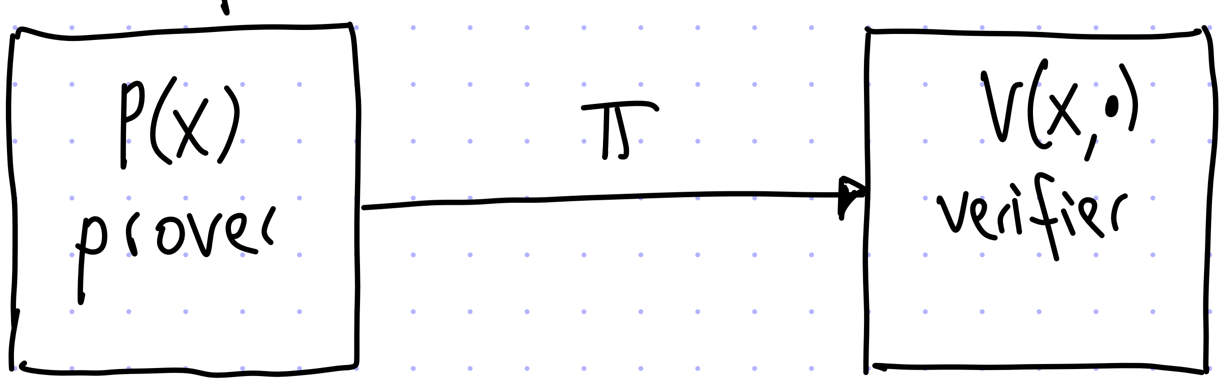
<u>Example: $L = SAT$</u>

x is a boolean formula $\phi(x_1,\dots,x_n)$

w is an assignment $(a_1,\dots,a_n) \in \{0,1\}^n$

D checks that $\phi(a_1,\dots,a_n)$ is true

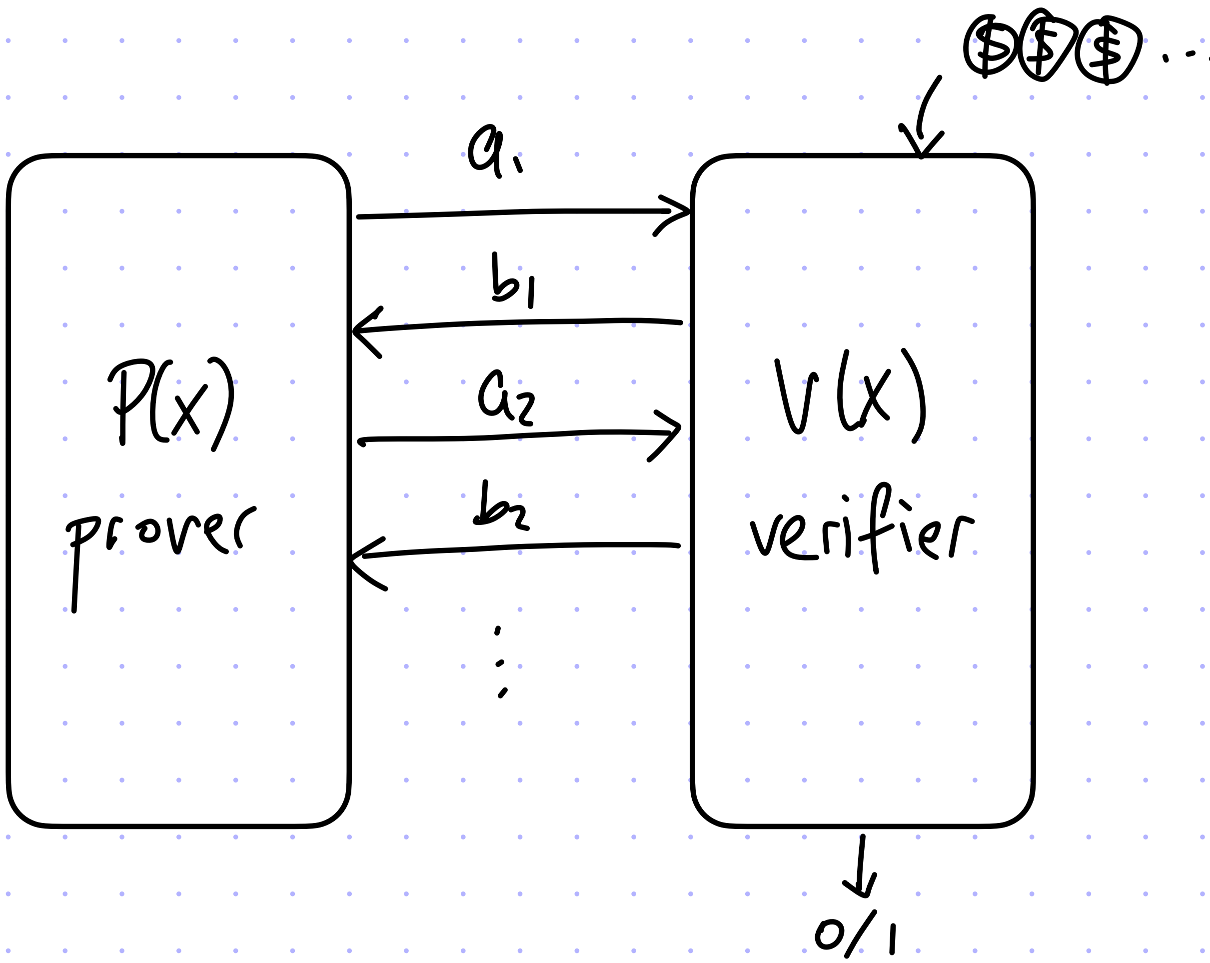
$\Rightarrow$ NP captures classical mathematical proofs



REVOLUTIONARY IDEA

the verifier V may
- use randomness
- interact with P

Interactive Proofs



randomness \ interaction	Y	N
Y	IP	MA
N	NP	NP

believed to equal NP (it does if strong PRGs exist)

(unbounded) honest prover, (efficient) honest verifier

An <u>interactive proof</u> for L is a pair (P, V) s.t.

(1) completeness: $\forall x \in L \quad \Pr_r\left[\langle P(x), V(x;r)\rangle = 1\right] = 1$

(2) soundness: $\forall x \notin L \ \forall \tilde{P} \quad \Pr_r\left[\langle \tilde{P}, V(x;r)\rangle = 1\right] \leq 1/2$

any noticeable gap suffices for definition

Q: which languages have interactive proofs? Any beyond NP?

IP for Graph Non-Isomorphism

Let $G_0=(V,E_0)$ and $G_1=(V,E_1)$ be two graphs on vertices V.

def: $G_0 \equiv G_1$ (G_0 & G_1 are isomorphic) if $\exists$ permutation $\pi: V\to V$ s.t.

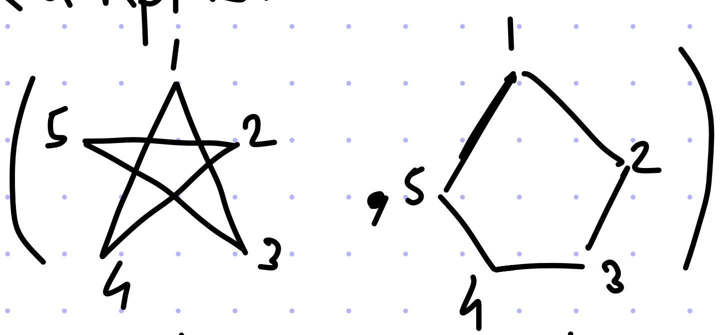
$$(u,v)\in E_0 \iff (\pi(u),\pi(v))\in E_1$$

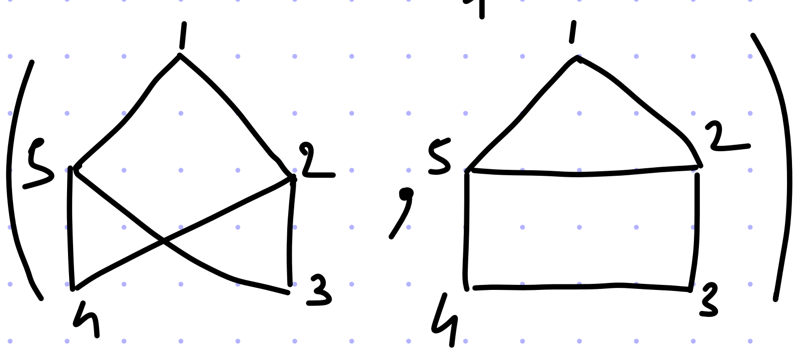
[if so, we write $G_1=\pi(G_0)$]

def: $GI := \{(G_0,G_1) \mid G_0\equiv G_1\}$ $\quad$ $GNI := \{(G_0,G_1) \mid G_0\not\equiv G_1\}$

- $GI \in NP$
- $GNI \in coNP$
- not known if in P

Examples:

- () $\in GI$. Why? $\pi = \begin{cases} 1\to 1 \\ 3\to 2 \\ 5\to 3 \\ 2\to 4 \\ 4\to 5 \end{cases}$

- () $\in GNI$. Why? Harder to see (in general).

How to prove that $G_0 \not\equiv G_1$?

theorem: GNI $\in$ IP

$P(G_0, G_1)$		$V(G_0, G_1)$
		$b \leftarrow \{0,1\}$
		$\pi \leftarrow$ (permutations on V)
		$H := \pi(G_b)$
choose $\tilde{b}$ s.t. H is in equiv class of $G_{\tilde{b}}$	$\xleftarrow{\quad H \quad}$	
	$\xrightarrow{\quad \tilde{b} \quad}$	$\tilde{b} \stackrel{?}{=} b$

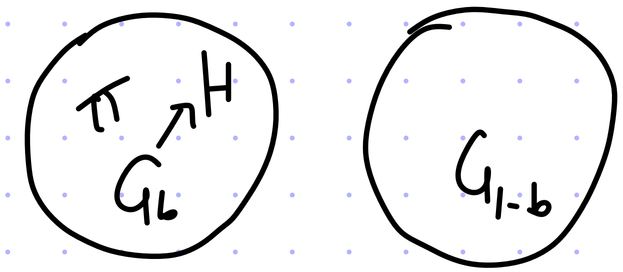
we can use interactive proofs to show that $G_0 \not\equiv G_1$!

Note: for now ignore prover time

Note: it is crucial that b is secret

... but later we'll see how to not rely on "private randomness"

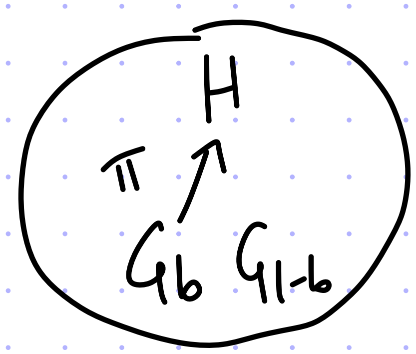
Completeness $(G_0, G_1) \in$ GNI $[G_0 \not\equiv G_1]$



G_b and G_{1-b} are in different equiv classes

$\Rightarrow$ honest prover can figure out which graph is H isomorphic to (& hence b) ∎

Soundness $(G_0, G_1) \notin$ GNI $[G_0 \equiv G_1]$



the random variable $\pi(G_b)$ is identical to $\pi(G_{1-b})$ so H and b are independent. No matter what $\tilde{b}$ is sent, $\Pr[\tilde{b} = b] = 1/2$ ∎

An Upper Bound on IP

theorem: $IP \subseteq PSPACE$

Let $L \in IP$, and let (P,V) be an IP for L.
We need to show that $L \in PSPACE$.

Fix an instance x and define $q_x := \max_{\tilde{P}} \Pr_r\left[\langle \tilde{P}, V(x;r)\rangle = 1\right]$.

If $x \in L$ then $q_x = 1$.
If $x \notin L$ then $q_x \leq 1/2$.
It suffices to compute q_x in polynomial space.

Problem: $\max_{\tilde{P}}$ we cannot expect to iterate over all provers because this includes provers that require large space to simulate

Idea: any transcript has polynomial size (the verifier reads it)
so we can afford to iterate over all transcripts in polyspace

$\Rightarrow$ the OPTIMAL prover strategy is computable in polyspace, and so is the probability q_x.

A partial transcript is a tuple $(\underbrace{a_1, b_1}_{\text{round } 1}, \underbrace{a_2, b_2}_{\text{round } 2}, \dots, \underbrace{a_i, b_i}_{\text{round } i})$.

def: $P^*(x, (a_1, b_1, \dots, a_i, b_i))$ output a_{i+1}^* that maximizes convincing probability conditioned on interaction so far being $tr = (a_1, b_1, \dots, a_i, b_i)$.

claim: $P^* \in PSPACE \rightarrow q_x \in PSPACE$

proof: Since P^* is optimal, we have $q_x = \frac{\sum_{r \in R} d(x,r)}{|R|}$ where $d(x,r)$ is the decision of $V(x;r)$ when interacting with P^*.

For any fixed r, $d(x,r)$ is computable in polynomial space:

$$a_1^* = P^*(x, \perp), \quad a_2^* = P^*(x, (a_1^*, b_1)), \quad \dots, \quad a_k^* = P^*(x, (a_1^*, b_1, \dots, a_{k-1}^*, b_{k-1}))$$

$$b_1 = V(x, r, a_1^*), \quad b_2 = V(x, r, a_1^*, a_2^*), \quad \dots, \quad b_k = V(x, r, a_1^*, \dots, a_k^*)$$

$\left.\begin{array}{l} \rightarrow \text{each in polyspace} \\ \rightarrow \text{each in polytime} \end{array}\right\}$ $d(x,r)$ in polynomial space.

∎

claim: $P^* \in \mathrm{PSPACE}$

proof: Let $tr = (a_1, b_1, \ldots, a_i, b_i)$ be a transcript of i rounds, and let $R[x, tr]$ be the set of random strings r consistent with (x, tr):

$$b_1 = V(x, r, a_1), \quad b_2 = V(x, r, a_1, a_2), \quad \ldots, \quad b_i = V(x, r, a_1, \ldots, a_i).$$

Proof is by induction on i:

- Base case is $i = k-1$:

$$P^*(x, tr) = \underset{a_k}{\operatorname{argmax}} \mathop{\mathbb{E}}_{r \in R[x,tr]} \left[V(x, r, a_1, \ldots, a_{k-1}, a_k) \right]$$

We can iterate over all messages a_k and randomness r in polyspace.

- Inductive case is $i < k-1$ (& assuming $P^* \in \mathrm{PSPACE}$ for $|tr| > i$):

$$P^*(x, tr) = \underset{a_{i+1}}{\operatorname{argmax}} \mathop{\mathbb{E}}_{r \in R[x,tr]} \left[V(x, r, a_1, \ldots, a_i, a_{i+1}, a_{i+1}^*, \ldots, a_k^*) \right].$$

- Inductive case is $i < k-1$ (& assuming $P^* \in PSPACE$ for $|tr| > i$):

$$P^*(x,tr) = \underset{a_{i+1}}{\arg\max} \mathop{\mathbb{E}}_{r \in R[x,tr]} \left[V(x, r, a_1, \ldots, a_i, a_{i+1}, a_{i+2}^*, \ldots, a_k^*) \right].$$

where $a_{i+2}^*, \ldots, a_k^*$ are optimal prover messages for (x, tr, r, a_{i+1}):

$$b_{i+1} = V(x, r, a_1, \ldots, a_i, a_{i+1})$$

$$a_{i+2}^* = P^*(x, (a_1, b_1, \ldots, a_i, b_i, a_{i+1}, b_{i+1}))$$

$$b_{i+2} = V(x, r, a_1, \ldots, a_i, a_{i+1}, a_{i+2}^*)$$

$$\vdots$$

$$a_k^* = P^*(x, (a_1, b_1, \ldots, a_i, b_i, a_{i+1}, b_{i+1}, a_{i+2}^*, b_{i+2}, \ldots, a_{k-1}^*, b_{k-1})).$$

P^* on transcripts longer than i rounds

Each of the above is computable in polyspace, given (x, tr, r, a_{i+1}).

We can iterate over all messages a_{i+1} and randomness r.

We deduce that P^* is computable in polynomial space. ∎