Saturday, May 19, 2018 8:21 PM

Proof Systems

given language L : goal to prove $x \in L$

proof system for L requires verification algorithm V
Completeness : $x \in L \Rightarrow \exists$ proof x V accepts $(x, \text{proof}x)$
Soundness : $x \notin L \Rightarrow \forall$ proof' V rejects (x, proof')

The prover asserts $x \in L$
true assertions have proofs.
false assertions have no proofs.

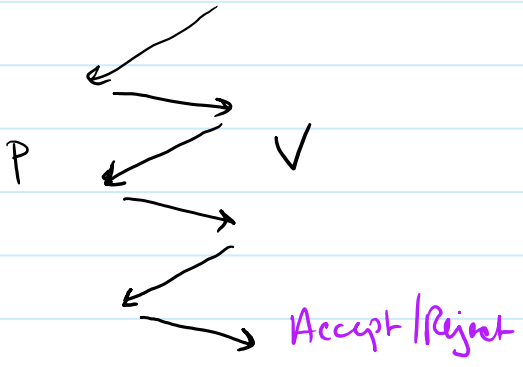
Efficiency: $\forall x$, proof $V(x, \text{proof})$ runs in time $\text{poly}(|x|)$

$L \in NP$ iff $L = \{x \mid \exists y \ |y| \leq |x|^k \ (x,y) \in R\}$ $R \in P$
Verifier = R Proof = y.

New ingredients: randomness (verifier can toss coins)
interaction - instead of reading the proof, verifier can ask questions.

Interactive proof systems:
both P (prover) and V (verifier) know x

rounds $\leq$ poly $(|x|)$
V can use a random string



Saturday, May 19, 2018 8:21 PM

An Interactive Proof System for L is an interactive protocol (P,V).

Completeness: $x \in L \Rightarrow \Pr[V \text{ accepts in } (P,V)(x)] \geq 2/3$
Soundness: $x \notin L \Rightarrow \forall P^* \Pr[V \text{ accepts in } (P^*,V)(x)] \leq 1/3$
By repeating, can reduce error to any ϵ.

$IP = \{L \mid L \text{ has an interactive proof system}\}$

$\Rightarrow$ philosophically captures more broadly what it means to be convinced that a statement is true

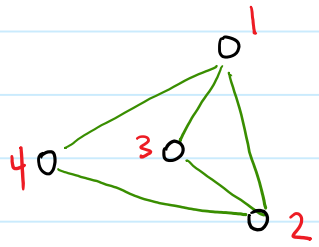
$NP \subseteq IP$ Verifier receives only a single message and uses no random bits.

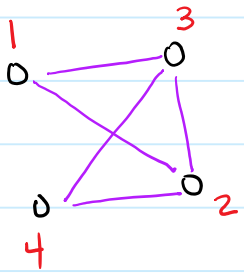
If $NP \subsetneq IP$ then randomness is essential.
If the Verifier is deterministic, the prover knows all of V's questions in advance and can send all the answers in one shot.

Graph Isomorphism: $G_0 = (V, E_0)$ $G_1 = (V, E_1)$

graphs are isomorphic $G_0 \cong G_1$ iff $\exists \pi : V \to V$
$(x,y) \in E_0 \iff (\pi(x), \pi(y)) \in E_1$

Saturday, May 19, 2018 8:21 PM





$1 \to 3$
$3 \to 1$
$4 \to 4$
$2 \to 2$

$GI = \{ (G_0, G_1) \mid G_0 \cong G_1 \}$

$GI \in NP$ but not known to be in P or NP-complete.

$GNI = \overline{GI}$ Not known if $GNI \in NP$

Theorem: $GNI \in IP$ (indication that IP may be more powerful than NP)

Verifier

Flips coin $c \in \{0,1\}$
pick random π
Apply π to G_c
to get H

$H = \pi(G_c)$

Prover

if $H \cong G_0$
$r = 0$
Else $r = 1$

Accept iff $r = c$

Completeness: if $G_0 \not\cong G_1$ then $H \cong G_0$ or $H \cong G_1$
but not both. Prover selects the correct one
prob of success = 1

Saturday, May 19, 2018 8:21 PM

Soundness : if $G_0 \cong G_1$, then prover sees the same distribution regardless of $c = 0, 1$.
Prover gets no information about c.
any prover succeeds w.p. $= 1/2$.

Can be repeated to get prob of success $= \epsilon$.

GNI $\in$ co-NP but not known if GNI is co-NP - complete.

Theorem : co-NP $\subseteq$ IP

Proof idea : Will actually show the following language is in IP

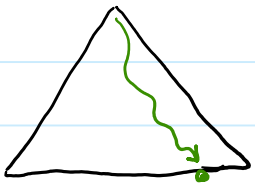
$$\{ (\phi, k) \mid \phi(x_1, \ldots, x_n) \text{ has exactly } k \text{ sat. assignments} \}$$

Prover claims that ϕ has exactly k satisfying assignments.
this is true iff :
$\phi(0, x_2, \ldots, x_n)$ has k_0 sat. assignments.
$\phi(1, x_2, \ldots, x_n)$ has k_1 sat. assignments
$k = k_0 + k_1$

Saturday, May 19, 2018 8:21 PM

Prover sends k_0 + k_1
Verifier selects a random $c \in \{0,1\}$ and asks prover to prove that $\phi(c, x_2, \ldots, x_n)$ has exactly k_c satisfying assignments
Continue recursively then check final step.

Problem: If k is only off by 1, verifier will only catch the problem is V's random choices lead exactly to the leaf that is the source of the discrepancy.



2^n leaves

Solution: Replace $\{0,1\}^n$ w/ $\{F_q\}^n$

↳ integers mod q

Verifier substitutes a random field element at each step. Vast majority of choices at each step will catch the prover (instead of just one.)

Theorem (LFKN) $L = \{(\phi, k) \mid$ CNF ϕ has exactly k satisfying assignments$\}$

$L \in IP$.

Saturday, May 19, 2018 8:21 PM

① Arithmetization: $\phi(x_1, x_2, \ldots, x_n) \Rightarrow P_\phi(x_1, \ldots, x_n)$

degree d polynomial over F_q.
q is prime and > 2.
degree $d \leq \text{poly}(n)$

$x_i \longrightarrow x_i$

$\phi \wedge \phi' \longrightarrow P_\phi \cdot P_{\phi'}$

$\neg\phi \longrightarrow 1 - P_\phi$

$\phi \vee \phi' \longrightarrow 1 - (1 - P_\phi)(1 - P_{\phi'})$

$\phi \vee \phi' = \neg(\neg\phi \wedge \neg\phi')$

Each clause has degree ≤ 3

$C_1 \wedge C_2 \cdots \wedge C_m$ has degree $\leq 3m = |\phi|$

* For all $\vec{x} \in \{0,1\}^n$ $P_\phi(x) = \phi(x)$ *

Can compute $P_\phi(x)$ in polytime, given ϕ & x.

Prover wants to show: $k = \sum_{x_1 \in \{0,1\}} \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} P_\phi(x_1, \ldots, x_n)$

Saturday, May 19, 2018 8:21 PM

fixed $z \in \mathbb{F}_q$.

Define $k_z = \sum_{x_2 \in \{0,1\}} \sum_{x_3 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}^n} P_\phi(z, x_2, \ldots, x_n)$

Prover sends k_z for all $z \in \mathbb{F}_q$

Verifier checks $k_0 + k_1 = k$
Picks a random z and asks prover to prove that
$k_z = \sum_{x_2} \cdots \sum_{x_n} P_\phi(z, x_2, \ldots, x_n)$

Continue recursively and at end Verifier checks that
$P_\phi(z_1, z_2, \ldots, z_n) = k_n$.

→ Actually, since $q > 2^n$, prover can't send all of them. Instead send the polynomial

$$p(z) = \sum_{x_2} \cdots \sum_{x_n} P_\phi(z, x_2, \ldots, x_n)$$

$$\text{degree} \leq d \leq |\phi|$$

$P_\phi(x, \underline{0,1,1,0,0,1})$
$(x_1, \ldots, x_n)$

Saturday, May 19, 2018 8:21 PM

Input (ϕ, k)

Prover

$p_1(x) = \sum_{x_2 \cdots x_n} P_\phi(x, x_2, \ldots, x_n)$
$x_i \in \{0,1\}$

$p_1(x)$ →

Verifier

Check $p_1(0) + p_1(1) = k$ → if no → reject.
pick random $z_1 \in F_q$

z_1 →

$p_2(x) = \sum_{x_3 \cdots x_n} P_\phi(z_1, x, x_3, \ldots, x_n)$

$p_2(x)$ →

Check $p_2(0) + p_2(1) = p_1(z_1)$
Pick random $z_2 \in F_q$

z_2 →

$p_3(x) = \sum_{x_4 \cdots x_n} P_\phi(z_1, z_2, x, x_4, \ldots, x_n)$

$p_3(x)$ →

⋮

$p_n(x)$ →

Check $p_n(0) + p_n(1) = p_{n-1}(z_{n-1})$
pick random $z_n \in F_q$.
Check
$p_n(z_n) = P_\phi(z_1, \ldots, z_n)$

Completeness: $(\phi, k) \in L$
Honest prover will always cause the verifier to accept.

Thursday, May 24, 2018 8:39 AM

$$(x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) = \phi$$

x	y	z	
0	0	0	✓
0	0	1	✓
0	1	0	×
0	1	1	✓
1	0	0	×
1	0	1	✓
1	1	0	✓
1	1	1	✓

sat assignments = 6.

Poly for $(x \vee \neg y \vee z)$ is:

$1 - (1-x)y(1-z)$ degree 3

Poly for $(\neg x \vee y \vee z)$ is

$1 - x(1-y)(1-z)$ degree 3

Poly for whole formula: $P_\phi(x,y,z) =$

$$(1 - (1-x)y(1-z))(1 - x(1-y)(1-z))$$

$P(0,0,0) = 1$

$P(1,0,0) = 0$

$\vdots$

$$\sum_{x \in \{0,1\}} \sum_{y \in \{0,1\}} \sum_{z \in \{0,1\}} P_\phi(x,y,z) = 6.$$

$$P_1(x) = \sum_{y \in \{0,1\}} \sum_{z \in \{0,1\}} P_\phi(x,y,z) = P_\phi(x,0,0) = 1-x$$
$$+ P_\phi(x,0,1) = 1$$
$$+ P_\phi(x,1,0) = x$$
$$+ P_\phi(x,1,1) = 1$$

$$= 3$$

Check $P_1(0) + P_1(1) = 3 + 3 = 6.$ ✓

Thursday, May 24, 2018 8:50 AM

Poly for whole formula: $P_\phi(x,y,z) =$

$$(1-(1-x)y(1-z))(1-x(1-y)(1-z))$$

Verifier Select random $x \in \mathbb{Z}_q$ Say $x = 7$

$$P_2(y) = \sum_{z \in \{0,1\}} P_\phi(7,y,z) = P_\phi(7,y,0) + P_\phi(7,y,1)$$

$$= (1-(-6)y)(1-7(1-y)) + 1.$$

$$= (1+6y)(-6+7y)+1 = \underline{42y^2 - 29y - 5}$$

Check $P_1(7) = P_2(0) + P_2(1)$

$3 = -5 + 42 - 29 - 5 = 3$ ✓

Verifier Selects random $y \in \mathbb{Z}_q$ $z = 5$.

$$P_3(z) = \phi(7,5,z) = (1-(-6)\cdot 5\cdot(1-z))(1-7(-4)(1-z))$$

$$= (1+30-30z)(1+28-28z)$$

$$= \underline{(31-30z)(29-28z)}$$

Check $P_2(5) = P_3(0) + P_3(1)$

$900 = 31\cdot 29 + 1$

900 ✓.

Thursday, May 24, 2018 9:10 AM

Select random $z \in \mathbb{Z}_q$ $z = 2$.

Check $P_\phi(7,5,2) = P_3(2)$ $P_3(z) = (31-30z)(29-28z)$

$= -29 \cdot -27 = 783$

Poly for whole formula: $P_\phi(x,y,z) =$

$(1-(1-x)y(1-z))(1-x(1-y)(1-z))$

$(1-(-6)\cdot 5(-1))(1-7(-4)(-1))$ ✓

$-29 \cdot -27$

Sunday, May 20, 2018 2:21 PM

i^{th} var.

$$p_{i-1}(x) = \sum_{x_{i+1},\ldots,x_n \in \{0,1\}} P_\phi(z_1, z_2, \ldots, z_{i-1}, x, x_{i+1}, \ldots, x_n)$$

$i+1^{st}$ var.

$$p_i(x) = \sum_{x_{i+2},\ldots,x_n \in \{0,1\}} P_\phi(z_1, z_2, \ldots, z_{i-1}, z_i, x, x_{i+2}, \ldots, x_n)$$

$$p_{i-1}(z_i) = \sum_{x_{i+1},\ldots x_n \in \{0,1\}} P_\phi(z_1, z_2, \ldots, z_{i-1}, z_i, x_{i+1}, \ldots, x_n)$$

$$= \sum_{x_{i+2}\ldots x_n} P_\phi(z_1 z_2, \ldots, z_{i-1}, z_i, 0, x_{i+2}, \ldots, x_n)$$

$$+ \sum_{x_{i+2}\ldots x_n} P_\phi(z_1 z_2 \ldots z_{i-1} z_i, 1, x_{i+2}, \ldots, x_n)$$

$$= p_i(0) + p_i(1)$$

Sunday, May 20, 2018 2:06 PM

Soundness

Bad event i:
$p_i(z) \neq p_i^*(z)$
and $p_i(z_i) = p_i^*(z_i)$

$$\text{Prob}_z[\, p_i(z) = p_i^*(z)\,] \leq d/q \leq |\phi|/2^n$$

↳ if $p_i \neq p_i^*$ $\quad (p_i - p_i^*)(z) = 0$

↳ deg $\leq d$ polynomial has $\leq d$ roots.

By induction if $(\phi, k) \notin L$ and Verifier does not reject
and no bad event in first $i-1$ rounds
then $p_i(z) \neq p_i^*(z)$ ↳ which implies bad event in round i w/ prob $\leq \frac{|\phi|}{2^n}$.

$i = 1$: if $(\phi, k) \notin L$ then $p_1(0) + p_1(1) \neq k$.

if $p_1^*(0) + p_1^*(1) \neq k$ then verifier rejects
if $p_1^*(0) + p_1^*(1) = k$ then $p_1(z) \neq p_1^*(z)$

Assume no bad event in first $i-1$ steps
By inductive hypothesis $p_{i-1}(z) \neq p_{i-1}^*(z)$

Also $p_{i-1}(z_{i-1}) \neq p_{i-1}^*(z_{i-1})$ ⟸ no bad event in round $i-1$.

Sunday, May 20, 2018 2:16 PM

Prover sends $p_i^*(z)$

Verifier checks if $p_i^*(0) + p_i^*(1) = p_{i-1}^*(z_{i-1})$

If not reject.

$$p_i^*(0) + p_i^*(1) = p_{i-1}^*(z_{i-1}) \neq p_{i-1}(z_{i-1})$$

$$= p_i(0) + p_i(1)$$

$$\Rightarrow p_i^*(z) \neq p_i(z).$$

At the end we have $p_n^*(z)$ which prover claims is equal to $p_n^*(z) = P_\phi(z_1, z_2, \ldots, z_{n-1}, z)$

fixed ↳ variable.

If no bad event occurs then these polynomials are not the same.

Verifier selects random z_n and checks that

$$p_n^*(z_n) = P_\phi(z_1, \ldots, z_n)$$

↳ Fails to detect difference w/ prob $\leq |\phi|/2^n$.

If no bad event occurs then the verifier will detect the difference.

Prob no bad event $\leq n \cdot \frac{|\phi|}{2^n}$ < Small. //