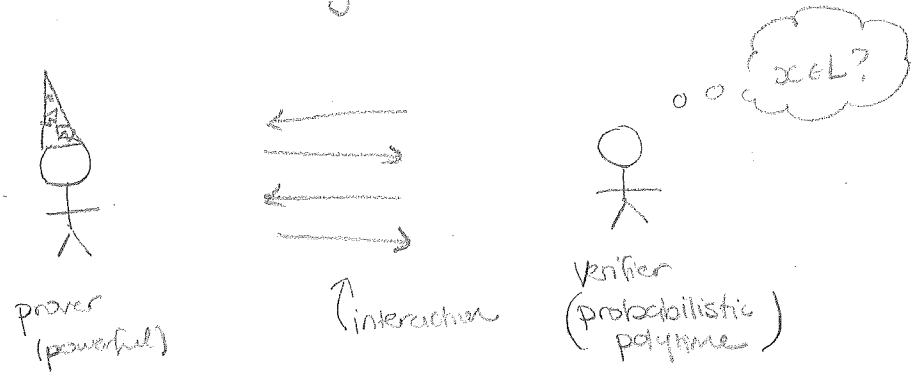
AGENDA
- coNP ⊆ IP
- PSPACE ⊆ IP
- AM / MA

Interactive proofs

Last time we defined IP, which was the class of languages that can be verified in the following model:

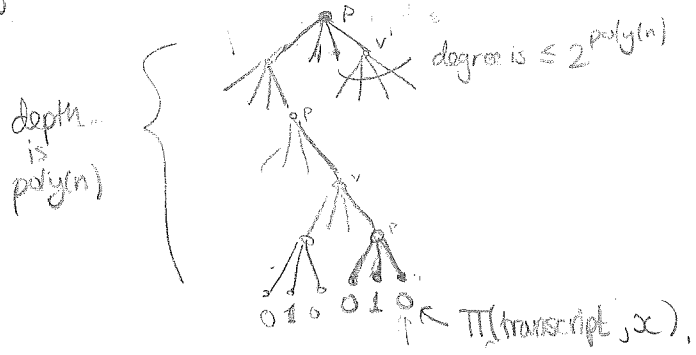


s.t. $\mathbb{P}\{(P \leftrightarrow V)(x) \neq \mathbb{1}_{x \in L}\} \leq 1/3$.

Q. How powerful is IP?

- $NP \subseteq IP$ (verifier just reads the certificate)
- Last time: $GNI \in IP$ (but we don't know if $GNI \in NP$)
- This time:
  $coNP \subseteq IP$, and in fact
  $PSPACE \subseteq IP$ (!)

UTOH, $IP \subseteq PSPACE$ → (why? SKETCH. Say that $L \in IP$, so $\exists$ protocol Π. Consider the tree of possible transcripts.



Here's a PSPACE alg for L.
Given x, search the tree and see if there's a way for the prover to win w/ prob. $\geq 2/3$.
(Need to think about how you'd do this in polyspace, but you can).

So, A. Thm IP = PSPACE.
[LFKN '90, Shamir '90]

(Finally, a question we can answer!)

Thm 1 (Warmup). coNP ⊆ IP.

Idea.

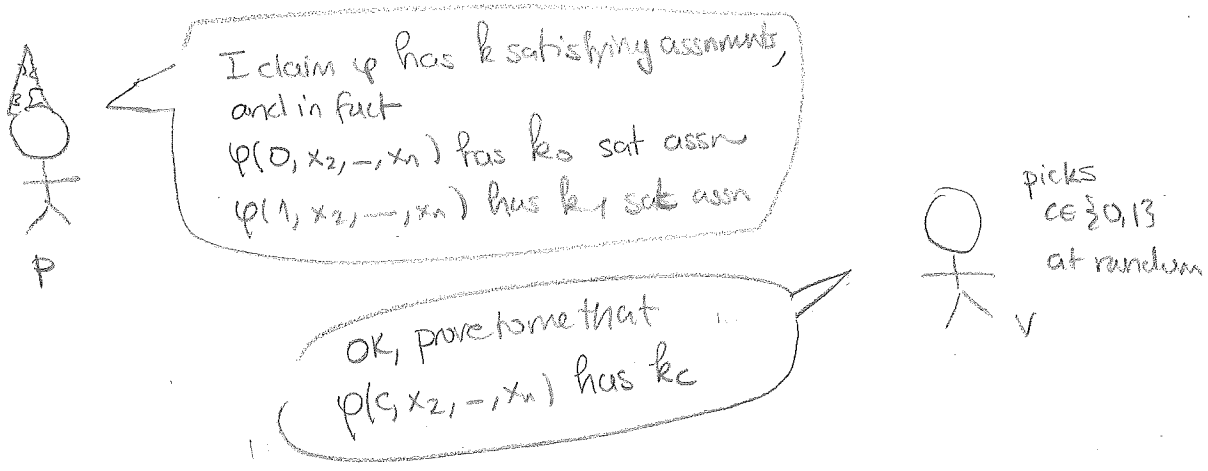
$\#SAT_D = \{ \langle \varphi, k \rangle : \varphi$ (3CNF) has exactly k satisfying assignments $\}$.

↑ decision

Note that $\overline{3SAT}$ is just $\#SAT_D$ w/ $k=0$, so if $\#SAT_D \in IP$ then $\overline{SAT} \in IP$.

We'll show $\#SAT_D \in IP$.

IDEAL.



— RECURSE —

If Prover is telling the truth, this should always work.

If Prover is lying, Verifier will catch him w/ prob. 2^{-n}.

(Why? If prover is lying by 1, then the verifier has to guess the c's exactly right to catch him.)

Ok, that's not good: want 2/3, not 2^{-n}.

Main ideas to fix this:

$\{0,1\}^n \leftarrow \mathbb{F}^n$

Instead of $x_1 = 0$ or $x_1 = 1$

$x_1 = \alpha$ or $x_1 = \beta$ or $x_1 = \gamma$ or ...

Now instead of guessing "right", Verifier just needs to guess "not wrong"

Details.

Arithmetization

Transform $\underbrace{\varphi(x_1, \dots, x_n)}_{\text{formula}} \longrightarrow \underbrace{P_\varphi(x_1, \dots, x_n)}_{P_\varphi \in \mathbb{F}[x_1, \dots, x_n]}$

$\deg(P_\varphi) = d$

$|\mathbb{F}|$ is a really big prime ($> 2^n$)

can do this recursively by:

$x_i \longrightarrow x_i$

$\varphi \wedge \varphi' \longrightarrow (P_\varphi) \cdot (P_{\varphi'})$

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

$\neg\varphi \longmapsto (1 - P_\varphi)$

eg $(x \vee y \vee \bar{z})$

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

and

$(x \vee y \vee \bar{z}) \wedge (\bar{x} \vee \bar{w} \vee \bar{z})$

$\mapsto (1 - (1-x)(1-y)(z-1))(1 - xwz)$

Then

(a) $\forall x \in \{0,1\}^n$, $P_\varphi(x) = \varphi(x)$

(b) $d = \deg(P_\varphi) \leq 3 \cdot (\#\text{clauses of } \varphi)$ (each clause $\Rightarrow$ deg 3 poly; multiply all the clause polys together)

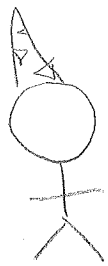
(c) $P_\varphi(x)$ can be computed in polytime from φ and x

IDEA 2.

Prover wants to show

$$k = \sum_{x_1=0,1} \sum_{x_2=0,1} \cdots \sum_{x_n=0,1} P_\varphi(x_1, \dots, x_n) \qquad (*)$$

for $\alpha \in \mathbb{F}$, $k_\alpha :=$ $\sum_{x_2=0,1} \cdots \sum_{x_n=0,1} P_\varphi(z, x_2, \dots, x_n)$



P

"I claim (*), and here all the values k_α"

"ok, prove to me that $k_\alpha = \sum_{x_2=0,1} \cdots \sum_{x_n=0,1} P_\varphi(z, x_2, \dots, x_n)$"

recurse.

V

pick $\alpha \in \mathbb{F}$ at random

What's wrong here?

(9) OK, the prover can't send all $> 2^n$ values k_α.

But he CAN send

$$q_1(X) = \sum_{x_2=0,1} \cdots \sum_{x_n=0,1} P_\varphi(X, x_1, \ldots, x_n) \in \mathbb{F}[X].$$

(notice $\deg(q_0) \leq 3 \cdot (\#\text{clauses in } \varphi)$)

So the actual protocol is: (IDEA 3)

I claim #sat. assn for φ is k ← aka $q_0 \equiv \sum_{\vec{x} \in \{0,1\}^n} P_\varphi(\vec{x}) = k$

P

"q_1" →

V

check: "q_1"(0) + "q_1"$(1) = k$

pick random $\alpha_1 \in \mathbb{F}$

α_1 ←

I claim $q_2(\alpha_1)$ is what I said.

→ "q_2"

$q_2(X) = \sum_{x_3=0,1} \cdots \sum_{x_n=0,1} P_\varphi(\alpha_1, X, x_3, \ldots, x_n)$

check:

"q_2"(0) + "q_2"(1) = "q_1"(α_1)

pick random $\alpha_2 \in \mathbb{F}$

α_2 ←

etc.

I claim $q_{n-1}(\alpha_{n-1})$ is what I said

→ "q_n"

$q_n(X) = P_\varphi(\alpha_1, \alpha_2, \ldots, \alpha_{n-1}, X)$

check

"q_n"(0) + "q_n"(1) = "q_{n-1}"(α_{n-1})

pick random α_n

Check

"q_n"$(\alpha_n) = P_\varphi(\alpha_1, \alpha_2, \ldots, \alpha_n)$

Analysis of protocol:

Completeness:

If $\langle \varphi, k \rangle \in \#SAT_D$, the honest prover will convince the verifier ✓

Soundness:

If $\langle \varphi, k \rangle \notin \#SAT_D$:

- $q_1(0) + q_1(1) \neq k$, so the prover must send "q_1" $\neq q_1$.
- If "q_1" $\neq q_1$,

$$\mathbb{P}\{ q_1(\alpha_1) = \text{"}q_1\text{"}(\alpha_1) \} \leq \frac{\deg(\text{"}q_1\text{"})}{|\mathbb{F}|} \leq \frac{|\varphi|}{2^n}$$

- If that happens, then Prover can win.
- But if not, then he has to lie in the next round too,

so "q_2" $\neq q_2$

$$\mathbb{P}\{ q_2(\alpha_2) = \text{"}q_2\text{"}(\alpha_2) \} \leq |\varphi|/2^n$$

- again, if that happens, P is off the hook.
- But if not ....

at the end of the day, by the union bound,

$$\mathbb{P}(\text{P got } q_i(\alpha_i) = \text{"}q_i\text{"}(\alpha_i) \text{ but } q_i \neq \text{"}q_i\text{"} \text{ for some } i \leq n) \leq \frac{n|\varphi|}{2^n} \leq \frac{\text{poly}(n)}{2^n} < 1/3$$

so

$$\mathbb{P}\{\text{Prover got away w/ cheating}\} < 1/3. \quad \checkmark$$

Extension for PSPACE ⊆ IP

Show TQBF ∈ IP.

Recall, TQBF is $\{\langle\varphi\rangle : \varphi \text{ is a true QBF}\}$

Try the same type of thing

QBFs look like

$\forall x_1 \exists x_2 \forall x_3 \cdots \exists x_n \varphi(\vec{x})$

Arithmetization:

$\exists x\ \varphi(x) \longrightarrow \sum_{x=0,1} P_\varphi(x)$

$\forall x\ \varphi(x) \longrightarrow \prod_{x=0,1} P_\varphi(x)$

So turn QBF Ψ into P_Ψ, and then Ψ true $\iff P_\Psi \neq 0$

Claim: $P_\Psi = k$

$q_1(X) = \sum_{x_2=0,1} \prod_{x_3=0,1} \cdots \sum_{x_n=0,1} P_\varphi(X, x_2, \ldots, x_n)$ ⟶

Check $q_1(0) \cdot q_1(1) = k$

Choose α_1 random

α_1 ⟵

etc.

What's wrong? (1) $\deg(q_1(X))$ might be 2^n ← this one we should deal with

(2) P_Ψ might be $2^{2^{|\varphi|}}$ ← sweep this under the rug — work mod a big prime and it will be ok

To fix (1), observe that for $x \in \{0,1\}$, $x^j = x\ \forall j$.

So these polynomials may as well be multilinear (deg of each x_i is ≤ 1).

More precisely, let $L_i : \mathbb{F}[X_1, \ldots, X_n] \to \mathbb{F}[X_1, \ldots, X_n]$ be

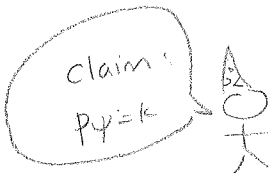
$$L_i \cdot p(X_1, \ldots, X_n) = X_i\, p(X_1, \ldots, X_{i-1}, 0, X_{i+1}, \ldots, X_n) + (1 - X_i)\, p(X_1, \ldots, X_{i-1}, 1, X_{i+1}, \ldots, X_n)$$

So $L_i(p)$ is linear in X_i, and $L_i(p)(\vec{x}) = p(\vec{x})$ whenever $x_i = 1$.

So instead of $\prod_{x_1} \sum_{x_2} \prod_{x_3} \cdots \sum_{x_n} P_\varphi(x_1, \ldots, x_n)$, use

$$\prod_{x_1} L_1 \sum_{x_2} L_1 L_2 \cdots \sum_{x_n} L_1 L_2 L_3 \cdots L_n\ P_\varphi(x_1, \ldots, x_n)$$

Now we can make this work.



$q_1(X) = L_1 \Sigma_{x_2} L_1 L_2 \cdots P_\varphi(X, -, x_n)$

$\longrightarrow$ "q_1"

check
"$q_1(0) \cdot q_1(1) = k$"
choose α_1 randomly

$\longleftarrow \alpha_1$

$q_2(X) = \Sigma_{x_2} L_1 L_2 \cdots P_\varphi(X, x_2, -, x_n)$

$\longrightarrow$ "q_2"

check
α_1 "$q_2(0) + (1-\alpha_1)$ "$q_2(1)$
$=$ "$q_1(\alpha_1)$"
Choose α_2 at random.

$\longleftarrow \alpha_2$

$q_3(X) = L_1 L_2 \Pi_{x_3} \cdots P_\varphi(\alpha_2, X, x_3, \ldots, x_n)$

$\longrightarrow$ "q_3"

check that
"$q_3(0)$" + "$q_3(1)$" = "$q_2(\alpha_2)$"
Choose α_3 at random

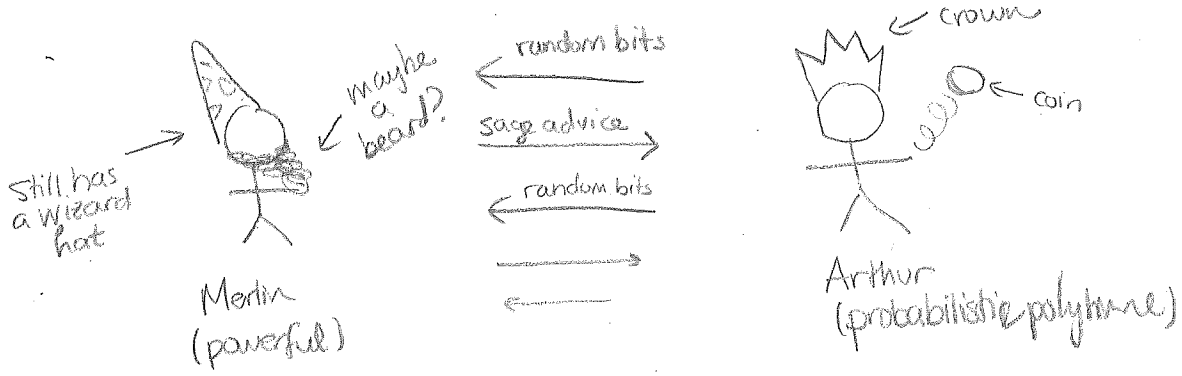
$\longleftarrow \alpha_3$

etc.

- At each step, the polynomial q_i that Prover sends is degree $\leq |\varphi|$, thanks to linearization.
- The only change in the analysis is this step, with the L_i. again, if the prover is lying, and also "$q_i(\alpha_i)$" $\neq q_{i-1}(\alpha_{i-1})$, then "$q_i$"$(\alpha_i) \neq \alpha_1 q_2(0) + (1-\alpha_1) q_2(1)$, and so he has to lie about q_2 also. But then
$$P\{ q_2(\alpha_2) = \text{"}q_2\text{"}(\alpha_2) \} \leq |\varphi| / |F|,$$
and he has to keep lying.
- now the union bound is over n^2 rounds instead of n, but $\frac{n^2 \cdot \text{poly}(n)}{2^n}$ is still $< 1/3$.

Arthur-Merlin Games

- IP crucially allows the Verifier to keep his randomness private.
- In an AM-game, the verifier (Arthur) has PUBLIC randomness.



AM[k] is the class of languages L w/ a k-round AM protocol s.t.

$x \in L \Rightarrow \mathbb{P}\{\text{Arthur accepts}\} \geq 2/3$ ← HW: Can make this 1 w/out changing anything.

$x \notin L \Rightarrow \mathbb{P}\{\text{Arthur accepts}\} \leq 1/3$

AM := AM[2] (Arthur sends random bits; Merlin responds)

MA := the class where Merlin goes first. (still one message each).

$AM[k] \subseteq IP[k]$ (k rounds), clearly. "private coins are better than public"

Thm (Goldwasser-Sipser)

$IP[k] \subseteq AM[k+2]$ "not THAT much better"

When $k = \text{poly}(n)$, this actually follows from the proof we just did:

the Verifier's randomness was all public, so

$TQBF \in AM[\text{poly}] \Rightarrow PSPACE \subseteq AM[\text{poly}] \subseteq IP \subseteq PSPACE$

$\Rightarrow AM[\text{poly}] = IP = PSPACE.$

HW: $AM[k] = AM[2]$ for constant k.

(9)

Some facts about AM:

- $NP \subseteq MA$ (Arthur doesn't randomize)
- $BPP \subseteq MA$ (HW)
- $MA \subseteq AM$ (HW)
- $AM \subseteq \Pi_2$ (HW)

$L \in AM$ if $\exists$ poly-time A s.t.

$x \in L \Rightarrow \mathbb{P}_r\{\exists m\, A(x,m,r)\} = 1$

$x \notin L \Rightarrow \mathbb{P}_r\{\exists m\, A(x,m,r)\} \leq 1/2$

$L \in MA$ if $\exists$ poly-time A s.t.

$x \in L \Rightarrow \exists m\ \mathbb{P}_r\{A(x,m,r)\} = 1$

$x \notin L \Rightarrow \forall m\ \mathbb{P}_r\{A(x,m,r)\} \leq 1/2$

- On your hw, you'll show:

  $GNI \in AM$

- Thm (HW.) [Boppana-Hastad-Zachos]

  $coNP \subseteq AM \Rightarrow PH = AM = \Sigma_2$

- Punchline: GI is unlikely to be NP-hard:

  GI NP-hard $\Rightarrow$ GNI coNP-hard

  $\Rightarrow$ $coNP \subseteq AM$

  $\Rightarrow$ $PH = \Sigma_2$