Lecture 03

Foundations of Probabilistic Proofs
Fall 2020
Alessandro Chiesa

Interactive Proofs for Polynomial Space

We have proved an upper bound on the power of interactive proofs: IP ⊆ PSPACE.
Today we prove that this upper bound is tight:

theorem: PSPACE ⊆ IP

We follow a similar approach as before:

		last lecture	today
①	choose complete problem	UNSAT / #SAT	TQBF
②	arithmetization	reduce to sumcheck problem	reduce to sum-product problem
③	protocol for the algebraic problem	sumcheck protocol	Shamir's protocol

Quantified Boolean Formulas

A fully quantified boolean formula is a logical expression such as

$\forall x_1 \exists x_2 \exists x_3 \; (x_1 \wedge x_2) \vee x_3$, or $\forall x_1 \exists x_2 \forall x_3 \; (x_1 \wedge x_2) \vee x_3$.

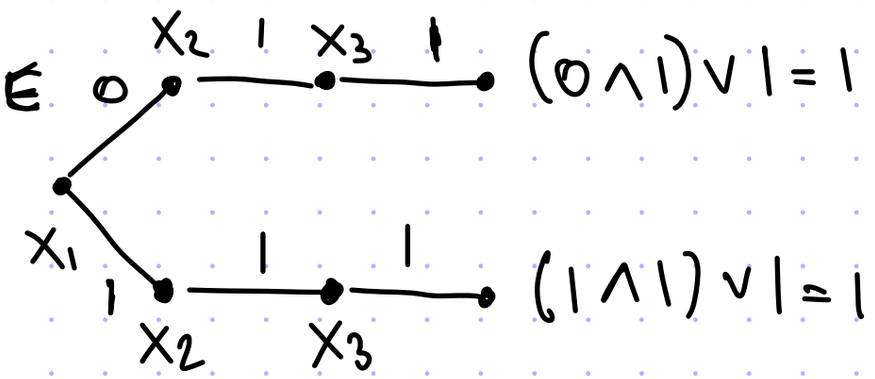
every variable is quantified via $\forall$ or $\exists$

boolean formula

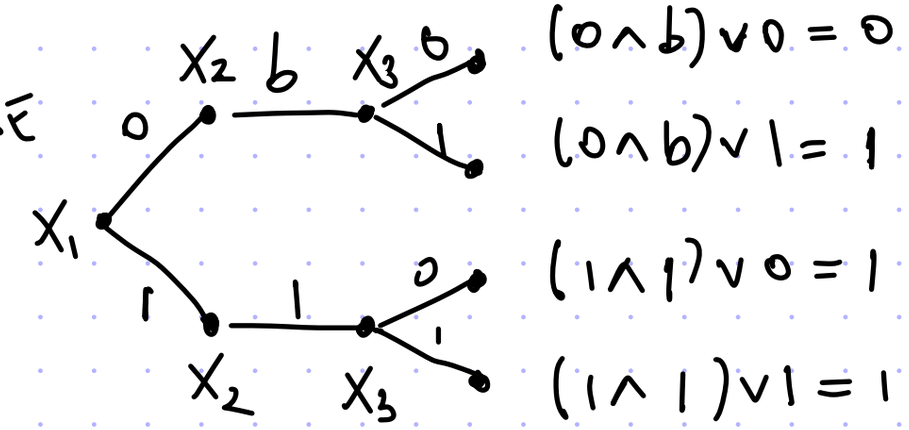
The expression evaluates to true or false.

Let us evaluate the examples:

1) TRUE



2) FALSE



Note: $NP \sim \{ \phi \mid \exists x_1 \exists x_2 \cdots \exists x_n \; \phi(x_1 \ldots x_n) = 1 \}$ & $coNP \sim \{ \phi \mid \forall x_1 \forall x_2 \cdots \forall x_n \; \phi(x_1, \ldots, x_n) = 1 \}$.

Def: $TQBF = \{ \phi(x_1, \ldots, x_n)$ s.t. $\forall x_1 \exists x_2 \forall x_3 \ldots \; \phi(x_1, \ldots, x_n) = 1 \}$ (3CNF)

Fact: TQBF is PSPACE-complete [more on this later]

Arithmetization for TQBF

We wish to arithmetize an expression such as: $\forall x_1 \exists x_2 \forall x_3 \dots \phi(x_1, \dots, x_n)$.

We arithmetize the formula and the quantifiers:

① formula: we use the arithmetization used for #SAT
where $\phi(x_1, \dots, x_n) \mapsto p(x_1, \dots, x_n)$ s.t. $p|_{\{0,1\}^n} \equiv \phi$ & $\deg_{tot}(p) \leq |\phi|$.

② $\forall$ behaves like a conjunction $\left(\forall x_i \ \phi(\dots, x_i, \dots) = \phi(\dots, 0, \dots) \wedge \phi(\dots, 1, \dots)\right)$
so we define an operator for this:
$$\prod_{x_i} p(\dots, x_i, \dots) := p(\dots, 0, \dots) \cdot p(\dots, 1, \dots)$$

③ $\exists$ behaves like a disjunction $\left(\exists x_i \ \phi(\dots, x_i, \dots) = \phi(\dots, 0, \dots) \vee \phi(\dots, 1, \dots)\right)$
so we define an operator for this:
$$\coprod_{x_i} p(\dots, x_i, \dots) := 1 - (1 - p(\dots, 0, \dots)) \cdot (1 - p(\dots, 1, \dots))$$

In sum we get: $\prod_{x_1} \coprod_{x_2} \prod_{x_3} \dots p(x_1, \dots, x_n)$.

Since $p|_{\{0,1\}^n} \equiv \phi$ and $\prod, \coprod$ stay within $\{0,1\}^n$, the expressions equal over any field.

Towards a Protocol: Degree Reduction

We want a protocol to evaluate $\prod_{x_1} \coprod_{x_2} \prod_{x_3} \cdots p(x_1, \ldots, x_n)$.

Idea: take inspiration from sumcheck protocol!

View sum as n operators: $$\sum_{\alpha_1, \ldots, \alpha_n \in \{0,1\}} p(\alpha_1, \ldots, \alpha_n) = \sum_{\alpha_1 \in \{0,1\}} \cdots \sum_{\alpha_n \in \{0,1\}} p(\alpha_1, \ldots, \alpha_n).$$

In the sumcheck protocol, each round peels off 1 operator.

E.g., in round 1 the prover sends: $$p_1(X_1) := \coprod_{x_2} \prod_{x_3} \cdots p(x_1, \ldots, x_n).$$

Problem: p_i may have degree $2^{n-i} \cdot 3m$ — exponentially large!

Degree reduction: on $\{0,1\}^n$, $X_1^3 X_3 + X_2^5 X_5^4 + X_4^2 \equiv X_1 X_3 + X_2 X_5 + X_4$ so we can set all positive powers to 1!

New operator ∇_{x_i} := "replace each occurrence of x_i^k, $k>0$, with x_i".

This leads to a new expression:

$$\prod_{x_1} \nabla_{x_1} \coprod_{x_2} \nabla_{x_1} \nabla_{x_2} \prod_{x_3} \nabla_{x_1} \nabla_{x_2} \nabla_{x_3} \cdots \coprod/\prod_{x_n} \nabla_{x_1} \nabla_{x_2} \cdots \nabla_{x_n} p(x_1, \ldots, x_n).$$

Shamir's Protocol

We need to check: $\prod_{x_1}\nabla_{x_1}\amalg_{x_2}\nabla_{x_1}\nabla_{x_2}\prod_{x_3}\nabla_{x_1}\nabla_{x_2}\nabla_{x_3}\cdots\amalg/\prod_{x_n}\nabla_{x_1}\nabla_{x_2}\cdots\nabla_{x_n} p(x_1,\dots,x_n) \stackrel{?}{=} \gamma_0$.

There are $k := n + \left(\sum_{i=1}^{n} i\right) = \frac{n^2+3n}{2}$ operators, and we will peel off one at a time.

So the protocol will have k rounds.

For $j = 1,\dots,k$ in round j:

P $O_j \cdots O_k\, p \stackrel{?}{=} \gamma_{j-1}$ **V**

$\xrightarrow{\tilde{p}_j \in \mathbb{F}[X]}$

$i_j \in [n]$ is var of O_j

CHECK $\tilde{p}_j$ vs γ_{j-1}

$\xleftarrow{w_{i_j} \in \mathbb{F}}$

$w_{i_j} \leftarrow \mathbb{F}$

$\gamma_j := \tilde{p}_j(w_{i_j})$

$[\&\ w_i := w_{i_j}]$

$O_{j+1} \cdots O_k\, p \stackrel{?}{=} \gamma_j$

- If $O_j = \prod_{x_{i_j}}$ then
  $\tilde{p}_j(0) \cdot \tilde{p}_j(1) \stackrel{?}{=} \gamma_{j-1}$
- If $O_j = \amalg_{x_{i_j}}$ then
  $1 - (1 - \tilde{p}_j(0)) \cdot (1 - \tilde{p}_j(1)) \stackrel{?}{=} \gamma_{j-1}$
- If $O_j = \nabla_{x_{i_j}}$ then
  $(\nabla_{x_i} \tilde{p}_j)(w_i^{\text{old}}) \stackrel{?}{=} \gamma_{j-1}$

After k rounds, the verifier checks that $p(w_1,\dots,w_n) \stackrel{?}{=} \gamma_k$.

Analysis of Shamir's Protocol

Consider a round $j \in [k]$ where $O_j = \Pi_{X_i}$ for $i \in [n]$. [Similar to $O_j = \mathbb{E}_{X_i}$.]

Completeness: Suppose that $\Pi_{X_i} O_{j+1} \cdots p(w_1, \ldots, w_{i-1}, X_i, \ldots, X_n) = \gamma_{j-1}$.

The honest prover sends $p_j(X_i) := O_{j+1} \cdots p(w_1, \ldots, w_{i-1}, X_i, \ldots, X_n)$.

The verifier's check passes: $p_j(0) p_j(1) = \gamma_{j-1}$.

Next, for every choice of $w_{ij} \in \mathbb{F}$, $O_{j+1} \cdots p(w_1, \ldots, w_{i-1}, w_{ij}, X_{i+1}, \ldots, X_n) = p_j(w_{ij}) = \gamma_j$.

Soundness: Suppose that $\Pi_{X_i} O_{j+1} \cdots p(w_1, \ldots, w_{i-1}, X_i, \ldots, X_n) \neq \gamma_{j-1}$.

The malicious prover sends $\tilde{p}_j(X_i)$ of degree at most 1.

If $\tilde{p}_j \equiv p_j$ (the honest polynomial) then the verifier's check fails: $\tilde{p}_j(0) \tilde{p}_j(1) = p_j(0) p_j(1) \neq \gamma_{j-1}$.

So suppose that $\tilde{p}_j \not\equiv p_j$.

By definition of p_j, $O_{j+1} \cdots p(w_1, \ldots, w_{i-1}, w_{ij}, X_{i+1}, \ldots, X_n) = p_j(w_{ij})$.

By definition of γ_j, $\gamma_j = \tilde{p}_j(w_{ij})$.

So the output claim is $p_j(w_{ij}) \stackrel{?}{=} \tilde{p}_j(w_{ij})$. This holds w.p. at most $1/|\mathbb{F}|$ over $w_{ij} \in \mathbb{F}$.

Analysis of Shamir's Protocol

syntactic sugar for:
$f(x_1,\dots,x_s) := \nabla_{x_i} O_{j+1} \cdots p(x_1,\dots,x_i,\dots,x_s,x_{s+1},\dots,x_n)$ evaluated at $(w_1,\dots,w_i,\dots,w_s)$

Consider a round $j \in [k]$ where $O_j = \nabla_{x_i}$ for $i \in [n]$.

Completeness: Suppose that $\nabla_{x_i} O_{j+1} \cdots p(w_1,\dots,w_i,\dots,w_s,x_{s+1},\dots,x_n) = \gamma_{j-1}$ for $s \geq i$.

The honest prover sends $p_j(X_i) := O_{j+1} \cdots p(w_1,\dots,w_{i-1},X_i,w_{i+1},\dots,w_s,x_{s+1},\dots,x_n)$.

The verifiers check passes: $(\nabla_{x_i} p_j)(w_i) = \gamma_{j-1}$.

Next, for every choice of $w_{ij} \in \mathbb{F}$, $O_{j+1} \cdots p(w_1,\dots,w_i,w_{ij},w_{i+1},\dots,w_s,x_{s+1},\dots,x_n) = p_j(w_{ij}) = \gamma_j$.

Soundness: Suppose that $\nabla_{x_i} O_{j+1} \cdots p(w_1,\dots,w_i,\dots,w_s,x_{s+1},\dots,x_n) \neq \gamma_{j-1}$ for $s \geq i$.

The malicious prover sends $\tilde{p}_j(X_i)$ of degree at most $3m$ or 2.

If $\tilde{p}_j \equiv p_j$ (the honest polynomial) then the verifier's check fails: $(\nabla_{x_i} \tilde{p}_j)(w_i) = (\nabla_{x_i} p_j)(w_i) \neq \gamma_{j-1}$.

So suppose that $\tilde{p}_j \not\equiv p_j$.

By definition of p_j, $O_{j+1} \cdots p(w_1,\dots,w_{i-1},w_{ij},w_{i+1},\dots,w_s,x_{s+1},\dots,x_n) = p_j(w_{ij})$.

By definition of γ_j, $\gamma_j = \tilde{p}_j(w_{ij})$.

So the output claim is $p_j(w_{ij}) \stackrel{?}{=} \tilde{p}_j(w_{ij})$. This holds w.p. at most $\frac{3m}{|\mathbb{F}|}$ or $\frac{2}{|\mathbb{F}|}$ over $w_{ij} \in \mathbb{F}$.

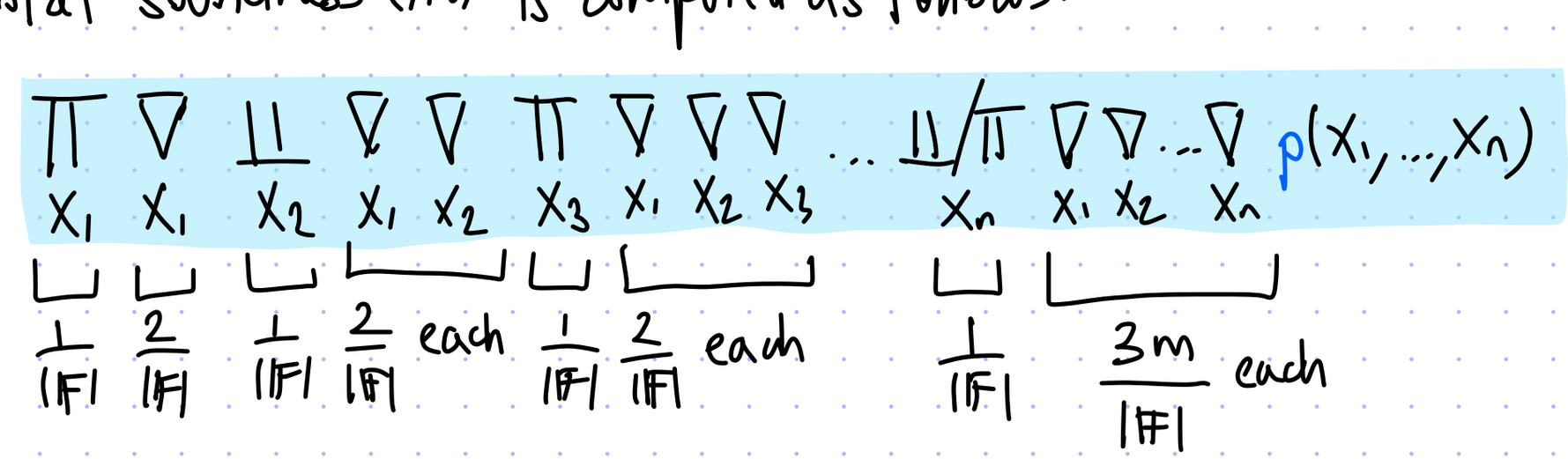
Analysis of Shamir's Protocol

Overall completeness:

In each round, if current claim is true then new claim is true w.p. 1.
After the last round, the final check ($p(w_1,\dots,w_n) \stackrel{?}{=} r_k$) passes.

Overall soundness:

The total soundness error is computed as follows:



$$\Rightarrow \quad n \cdot \frac{1}{|\mathbb{F}|} + n \cdot \frac{3m}{|\mathbb{F}|} + \frac{(n-1)\cdot n}{2} \cdot \frac{2}{|\mathbb{F}|} = \frac{3mn + n^2}{|\mathbb{F}|}.$$

So for $\mathbb{F}$ sufficiently large, Shamir's protocol is sound.

TQBF is in PSPACE

Let $\Phi = Q_1 x_1 Q_2 x_2 \cdots Q_n x_n \; \phi(x_1, \ldots, x_n)$ be a (fully) quantified boolean formula, where each $Q_i \in \{\forall, \exists\}$. We wish to evaluate Φ in poly(m,n) space.

Define: $\Phi_n := \Phi$ and for each $i \in \{n-1, n-2, \ldots, 0\}$

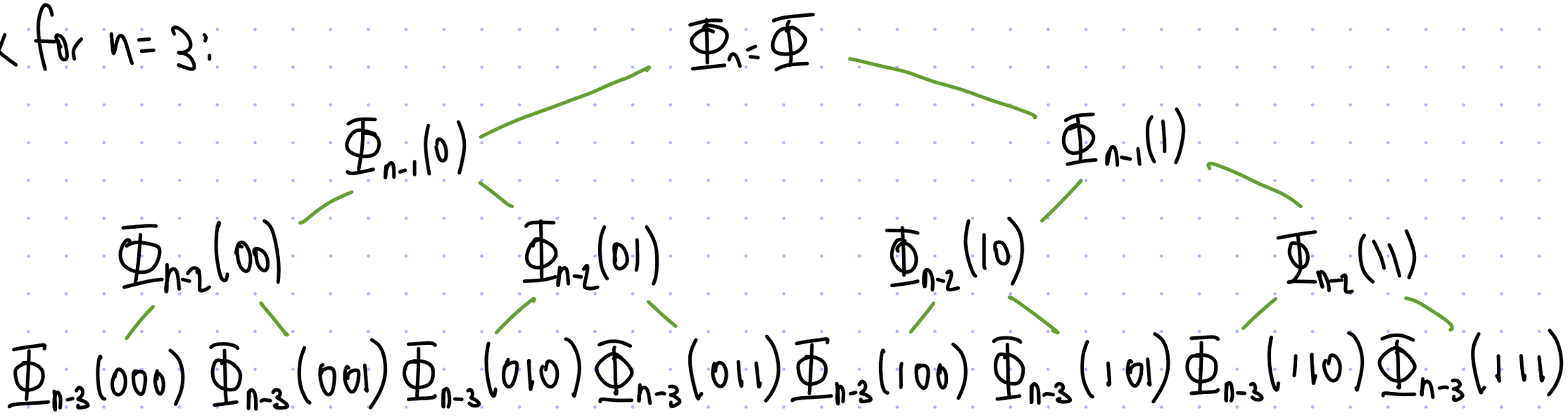
$$\Phi_i(x_1, \ldots, x_{n-i}) := Q_{n-i+1} x_{n-i+1} \cdots Q_n x_n \; \phi(x_1, \ldots, x_{n-i}, x_{n-i+1}, \ldots, x_n).$$

Observe that $\Phi_0 = \phi$ and a recurrence holds:

$\Phi_n = Q_1 x_1 \Phi_{n-1}(x_1)$, $\Phi_{n-1}(x_1) = Q_2 x_2 \Phi_{n-2}(x_1, x_2)$, and so on.

This yields a full binary tree on 2^n leaves that we can evaluate in poly(m,n) space.

Ex for $n=3$:



TQBF is PSPACE-Hard

Suppose that a language L is decidable by a machine M running in space $S(n) = \text{poly}(n)$.

GOAL: given x of size n, we wish to construct a QBF Φ, of size $\text{poly}(n)$, that is true iff $x \in L$.

Given instance x, define:

$G :=$ "configuration graph of the computation of M on x".

The graph has $2^{O(S(n))}$ vertices (the possible states) and directed edges represent transitions.

There is a unique starting state C_s and unique accepting state C_a.

Observation: $x \in L \Leftrightarrow \exists$ path in G from C_s to C_a.

We recursively define, for increasing i, a QBF Φ_i s.t.

$\forall$ configs C, C', $\Phi_i(C, C') = 1 \Leftrightarrow \exists$ path in G from C to C' of length $\leq 2^i$.

The totally quantified boolean formula that we seek is $\Phi := \Phi_{O(S(n))}(C_s, C_a)$.

We are left to show that we can construct Φ_i with size $\text{poly}(n, i)$.

Want: $\Phi_i(C,C')=1 \Leftrightarrow \exists$ path in G from C to C' of length $\leq 2^i$.

Base case ($i=0$):

$\Phi_0(C,C') :=$ "the boolean formula obtained by applying the Cook-Levin Theorem to the transition function of M on input x" (no quantifiers).

Recursive case ($i>0$): consider a halfway configuration

$$\Phi_i(C,C') := \exists C'' \quad \Phi_{i-1}(C,C'') \wedge \Phi_{i-1}(C'',C')$$

Problem: the formula is correct but doubles in size at each recursion

Solution: use more quantifiers to use Φ_{i-1} only once:

$$\Phi_i(C,C') := \exists C'' \forall D_1, D_2 \quad \big((D_1=C \wedge D_2=C'') \vee (D_1=C'' \wedge D_2=C')\big) \Rightarrow \Phi_{i-1}(D_1,D_2)$$

Above we use the syntactic sugar $\phi_1 \Rightarrow \phi_2$ which stands for $\overline{\phi_1} \vee \phi_2$.

Now we have that $|\Phi_i| = |\Phi_{i-1}| + \text{poly}(s(n))$, so $\Phi_{O(s(n))}$ has poly(n)-size. ∎