Foundations of Probabilistic Proofs

A course by **Alessandro Chiesa**

Lecture 03

IP for PSPACE

Interactive Proofs for Polynomial Space

We proved an upper bound on the power of IPs: $IP \subseteq PSPACE$.

↑ languages decidable in polynomial space

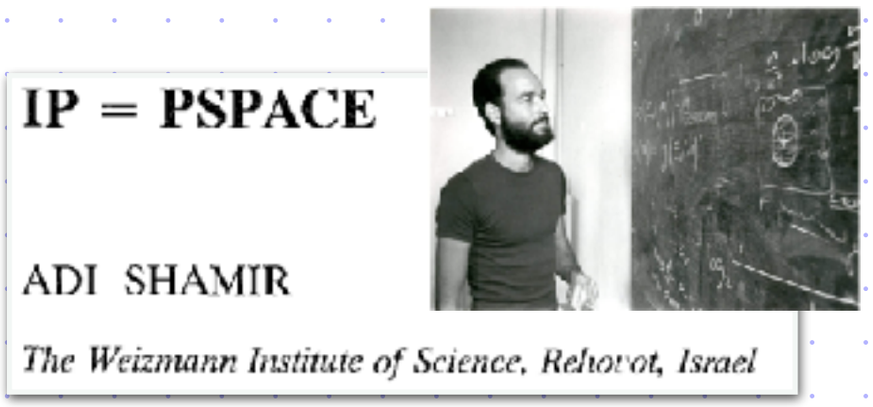
Today we prove that this upper bound is tight:

theorem: $PSPACE \subseteq IP$

We follow a similar approach as before:

	last time	today
① complete problem	UNSAT/#SAT	TQBF
② arithmetization	reduce to sumcheck problem	reduce to productcheck problem
③ protocol for algebraic problem	sumcheck protocol	Shamir protocol

The theorem was proved by Adi Shamir. We study a variant of the proof by Alexander Shen.



IP = PSPACE

ADI SHAMIR

The Weizmann Institute of Science, Rehovot, Israel

IP = PSPACE: Simplified Proof

A. SHEN

Academy of Sciences, Moscow, Russia, CIS

Quantified Boolean Formulas [1/2]

A quantified boolean formula (QBF) is a logical expression Φ that extends the notion of a boolean formula with the quantifiers $\forall$ and $\exists$.

The set of QBFs is the minimal set such that:
- {boolean formulas} $\subseteq$ QBF
- $\Phi \in$ QBF $\rightarrow \overline{\Phi} \in$ QBF
- $\Phi_1, \Phi_2 \in$ QBF $\rightarrow \Phi_1 \wedge \Phi_2, \Phi_1 \vee \Phi_2 \in$ QBF
- $\Phi \in$ QBF $\rightarrow$ for every (free) var x, $\forall x \Phi, \exists x \Phi \in$ QBF

Examples:

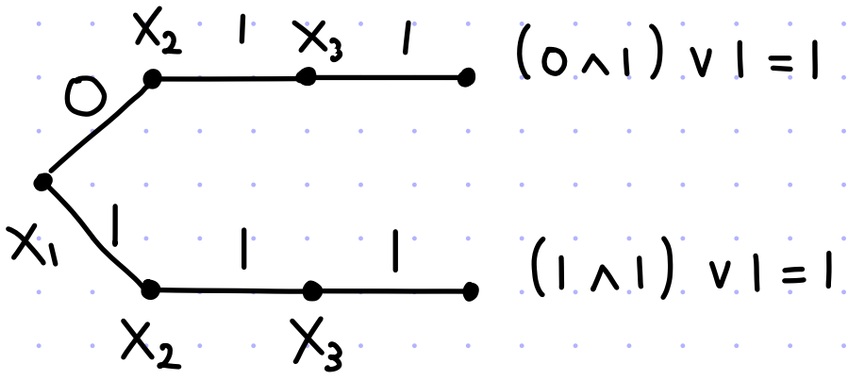
- $\Phi_1 = (x_1 \wedge (x_2 \vee \overline{x_3})) \wedge x_4$ $\quad \Phi_1 : \{0,1\}^4 \rightarrow \{0,1\}$ not quantified (open)
- $\Phi_2 = \forall x_1 \ (x_1 \vee \exists x_2 \ (x_2 \wedge x_3)) \vee x_4$ $\quad \Phi_2 : \{0,1\}^2 \rightarrow \{0,1\}$ partially quantified (open)
- $\Phi_3 = \forall x_1 \exists x_2 \ ((x_1 \wedge x_2) \wedge \forall x_3 \ (\overline{x_1} \wedge x_3))$ $\quad \Phi_3 \in \{0,1\}$ fully quantified (closed)

Each variable in Φ is free (unquantified) or bound (quantified).

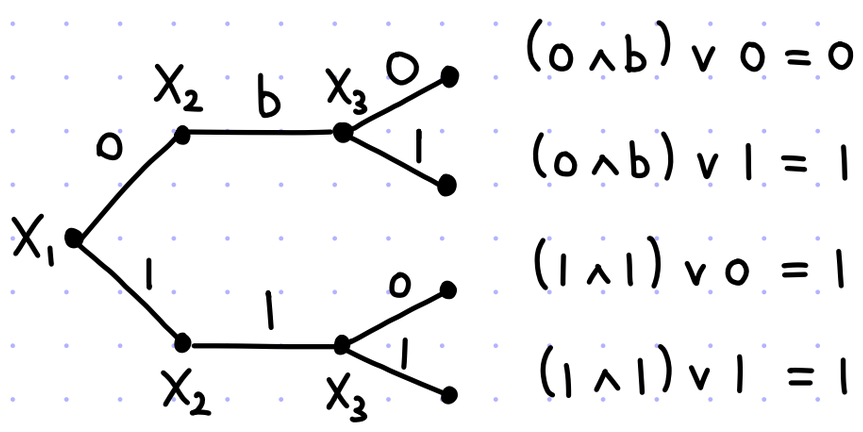
We say that Φ is closed if it has no free variables, and Φ is open otherwise.

Evaluating closed QBFs:

$\forall x_1 \ \exists x_2 \ \exists x_3 \quad (x_1 \wedge x_2) \vee x_3 \quad = 1$



$\forall x_1 \ \exists x_2 \ \forall x_3 \quad (x_1 \wedge x_2) \vee x_3 \quad = 0$



Quantified Boolean Formulas [2/2]

Quantified boolean formulas capture familiar classes:

- $\{ \varphi \mid \exists x_1 \exists x_2 \cdots \exists x_n \ \varphi(x_1,\ldots,x_n)=1 \}$ is NP-complete
- $\{ \varphi \mid \forall x_1 \forall x_2 \cdots \forall x_n \ \varphi(x_1,\ldots,x_n)=1 \}$ is coNP-complete

A QBF Φ is in prenex normal form (PNF) if $\Phi = Q_1 x_{i_1} Q_2 x_{i_2} \cdots \varphi(x_1,\ldots,x_n)$.

A list of quantifiers followed by a boolean formula.

Moreover, Φ is in P3CNF if φ in Φ is a 3CNF boolean formula.

We consider the language of closed QBFs, in P3CNF, that evaluate to true.

def: $\text{TQBF} = \{ \varphi(x_1,\ldots,x_n) \mid \forall x_1 \exists x_2 \forall x_3 \cdots \ \varphi(x_1,\ldots,x_n)=1 \}$

(true; 3CNF)

We use the following fact (more on this later):

TQBF is PSPACE-complete

Arithmetization for TQBF

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

We arithmetize the formula AND the quantifiers:

① formula: we use the arithmetization used for #SAT where

$\varphi(x_1,\dots,x_n) \mapsto p(x_1,\dots,x_n)$ s.t. $p|_{\{0,1\}^n} \equiv \varphi$ & $\deg_{tot}(p) \leq |\varphi|$ & $|p| \leq |\varphi|$.

② $\forall$ behaves like a conjunction: $\forall x_i \ \varphi(\dots,x_i,\dots) = \varphi(\dots,0,\dots) \wedge \varphi(\dots,1,\dots)$.

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: $\exists x_i \ \varphi(\dots,x_i,\dots) = \varphi(\dots,0,\dots) \vee \varphi(\dots,1,\dots)$.

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 obtain: $\prod_{x_1} \coprod_{x_2} \prod_{x_3} \cdots p(x_1,\dots,x_n)$.

Since $p|_{\{0,1\}^n} \equiv \varphi$ and $\prod, \coprod$ stay within $\{0,1\}^n$,

$\prod_{x_1} \coprod_{x_2} \prod_{x_3} \cdots p(x_1,\dots,x_n)$ equals $\forall x_1 \exists x_2 \forall x_3 \cdots \varphi(x_1,\dots,x_n)$ over every field.

Towards a Protocol [1/3]

We seek an IP for checking the value of $\prod_{x_1} \coprod_{x_2} \prod_{x_3} \cdots p(x_1, \ldots, x_n)$.

IDEA: take inspiration from the sumcheck protocol.

View the sum as n operators: $\sum_{\alpha_1, \ldots, \alpha_n \in \{0,1\}} p(\alpha_1, \ldots, \alpha_n) = \sum_{\alpha_1} \sum_{\alpha_2} \cdots \sum_{\alpha_n} p(\alpha_1, \ldots, \alpha_n)$.

The sumcheck protocol has n rounds, and each "peels off" one operator.

By analogy we could consider a protocol that works as follows:

$p_1(X_1) := \coprod_{x_2} \prod_{x_3} \cdots p(x_1, \ldots, x_n)$

$p_2(X_2) := \prod_{x_3} \cdots p(w_1, X_2, \ldots, x_n)$

$p_1 \in \mathbb{F}[X]$ →	$\prod_{x_1} p_1(x_1) \stackrel{?}{=} 1$
← $w_1 \in \mathbb{F}$	$w_1 \leftarrow \mathbb{F}$
$p_2 \in \mathbb{F}[X]$ →	$\coprod_{x_2} p_2(x_2) \stackrel{?}{=} p_1(w_1)$
← $w_2 \in \mathbb{F}$	$w_2 \leftarrow \mathbb{F}$
⋮	⋮
← $w_n \in \mathbb{F}$	$w_n \leftarrow \mathbb{F}$
	$p(w_1, \ldots, w_n) \stackrel{?}{=} p_n(w_n)$

PROBLEM: $\forall i \in [n]$, $p_i(X)$ may have degree $2^{n-i} \cdot 3m$, which is exponentially large.

Towards a Protocol [2/3]

OBSERVATION: boolean values are unaffected by degrees >0 ($0^k=0$ & $1^k=1$ $\forall k>0$).

For example, $X_1^3X_3 + X_2^5X_5^4 + X_4^2$ and $X_1X_3 + X_2X_5 + X_4$ agree on $\{0,1\}^n$.

IDEA: set all positive powers to 1.

This leads to the technique of **DEGREE REDUCTION**:

- $\forall i \in [n]$, define the new operator

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

- use ∇_{X_i} to reduce the degree of x_i to ≤ 1.

EXAMPLE: $\nabla_{X_1}(X_1^3X_3 + X_2^5X_5^4 + X_4^2) = X_1X_3 + X_2^5X_5^4 + X_4^2$, $\nabla_{X_1}\nabla_{X_2}\cdots\nabla_{X_5}(X_1^3X_3 + X_2^5X_5^4 + X_4^2) = X_1X_3 + X_2X_5 + X_4$.

Here is an expression without degree blowups that equals $\prod_{X_1}\coprod_{X_2}\prod_{X_3}\cdots p(x_1,\ldots,x_n)$:

$$\prod_{X_1}\nabla_{x_1}\coprod_{x_2}\nabla_{x_1}\nabla_{x_2}\prod_{x_3}\nabla_{x_1}\nabla_{x_2}\nabla_{x_3}\cdots\coprod_{x_n}/\prod_{x_n}\nabla_{x_1}\nabla_{x_2}\cdots\nabla_{x_n}\ p(x_1,\ldots,x_n)$$

reduce degree of each surviving variable to ≤ 1 right after degree doubling due to $\prod$ or $\coprod$

reduce degree of each variable to ≤ 1 right after arithmetizing φ to p

Towards a Protocol [3/3]

$$\prod_{x_1} \nabla_{x_1} \coprod_{x_2} \nabla_{x_1} \nabla_{x_2} \prod_{x_3} \nabla_{x_1} \nabla_{x_2} \nabla_{x_3} \cdots \coprod_{x_n} / \prod_{x_n} \nabla_{x_1} \nabla_{x_2} \cdots \nabla_{x_n} \; p(x_1,\dots,x_n)$$

Q: How to "peel off" the operator ∇_{x_i} ?

The operator ∇_{x_i} appears after (to the right of) $\prod_{x_i} / \coprod_{x_i}$.

Hence when we reach ∇_{x_i} the claim has this form:

$$\begin{bmatrix} x_1 := \omega_1 \\ \vdots \\ x_i := \omega_i \\ \vdots \\ x_s := \omega_s \end{bmatrix} \Big(\nabla_{x_i} \underbrace{O_{j+1} \cdots O_k \; p(x_1,\dots,x_n)}_{p_j(x_1,\dots,x_s)} \Big) = \gamma_{j-1} \qquad \text{for some } s \in \{i,\dots,n\} .$$

EXAMPLE: $\begin{bmatrix} x_1 := \omega_1 \\ x_2 := \omega_2 \\ x_3 := \omega_3 \end{bmatrix} \Big(\nabla_{x_2} x_1^5 x_2^2 + x_2 x_3 \Big) = \begin{bmatrix} x_1 := \omega_1 \\ x_2 := \omega_2 \\ x_3 := \omega_3 \end{bmatrix} \big(x_1^5 x_2 + x_2 x_3 \big) = \omega_1^5 \omega_2 + \omega_2 \omega_3 =: \gamma .$

The prover sends $\tilde{p}_j(X_2)$. The honest prover sends $p_j(X_2) := \begin{bmatrix} x_1 := \omega_1 \\ x_2 \text{ (free)} \\ x_3 := \omega_3 \end{bmatrix} \big(\quad x_1^5 x_2^2 + x_2 x_3 \big) = \omega_1^5 x_2^2 + x_2 \omega_3 .$

(x_2 is free; operator removed)

The verifier checks that $\nabla_{x_2} \tilde{p}_j(X_2)$ evaluated at $x_2 := \omega_2$ equals γ.

The verifier sends $\omega_2' \leftarrow \mathbb{F}$.

The new expression is $\begin{bmatrix} x_1 := \omega_1 \\ x_2 := \omega_2' \\ x_3 := \omega_3 \end{bmatrix} \big(x_1^5 x_2^2 + x_2 x_3 \big)$ and claimed value is $\gamma' := \tilde{p}_j(\omega_2')$.

Intuition: the new claim is tantamount to $\tilde{p}_j(\omega_2') = p_j(\omega_2')$.

Shamir's Protocol

Statement: $\prod_{x_1} \nabla_{x_1} \coprod_{x_2} \nabla_{x_1} \nabla_{x_2} \prod_{x_3} \nabla_{x_1} \nabla_{x_2} \nabla_{x_3} \cdots \coprod/\prod_{x_n} \nabla_{x_1} \nabla_{x_2} \cdots \nabla_{x_n} \; p(x_1,\ldots,x_n) \stackrel{?}{=} \gamma_0$

There are $K := n + \sum_{i=1}^{n} i = \frac{n^2+3n}{2}$ operators. We peel off one at a time.

For $j \in [K]$, let $i_j \in [n]$ be the variable of the j-th operator O_j.

For $j = 1,\ldots,K$ in round j of the protocol:

$O_j \cdots O_K \; p \stackrel{?}{=} \gamma_{j-1}$ when $\{x_i := \omega_i\}_{i \in S_{j-1}}$

$\tilde{p}_j \in \mathbb{F}[x_{i_j}]$ →

← $\omega_{i_j} \in \mathbb{F}$

check $\tilde{p}_j$ vs γ_{j-1}

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

$S_j := S_{j-1} \cup \{x_{i_j} := \omega_{i_j}\}$

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

- if $O_j = \prod_{x_{i_j}}$ then
  $\tilde{p}_j(0) \cdot \tilde{p}_j(1) \stackrel{?}{=} \gamma_{j-1}$
- if $O_j = \coprod_{x_{i_j}}$ 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
  $(\nabla_{x_{i_j}} \tilde{p}_j)(\omega_{i_j}^{\text{old}}) \stackrel{?}{=} \gamma_{j-1}$

$O_{j+1} \cdots O_K \; p \stackrel{?}{=} \gamma_j$ when $\{x_i := \omega_i\}_{i \in S_j}$

replaces old value of x_{i_j} if one exists

After K rounds the verifier checks that $p(\omega_1,\ldots,\omega_n) \stackrel{?}{=} \gamma_K$.

Analysis of Shamir's Protocol [1/3]

Consider a round $j \in [k]$ where $O_j = \prod_{x_{i_j}}$ for $i_j \in [n]$. (The case $O_j = \coprod_{x_{i_j}}$ is similar.)

COMPLETENESS: Suppose that $\left[\begin{smallmatrix} x_1 := \omega_1 \\ \vdots \\ x_{i_j-1} := \omega_{i_j-1} \end{smallmatrix}\right]\left(\prod_{x_{i_j}} O_{j+1} \cdots P\right) = \gamma_{j-1}$.

The honest prover sends $p_j(x_{i_j}) := \left[\begin{smallmatrix} x_1 := \omega_1 \\ \vdots \\ x_{i_j-1} := \omega_{i_j-1} \end{smallmatrix}\right]\left(O_{j+1} \cdots P\right)$ of degree ≤ 1.

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

The next statement is always true: $\forall \omega_{i_j} \in \mathbb{F}$, $\left[\begin{smallmatrix} x_1 := \omega_1 \\ \vdots \\ x_{i_j-1} := \omega_{i_j-1} \\ x_{i_j} := \omega_{i_j} \end{smallmatrix}\right]\left(O_{j+1} \cdots P\right) \overset{\text{def of } p_j}{=} p_j(\omega_{i_j}) \overset{\text{def of } \gamma_j}{=} \gamma_j$.

SOUNDNESS: Suppose that $\left[\begin{smallmatrix} x_1 := \omega_1 \\ \vdots \\ x_{i_j-1} := \omega_{i_j-1} \end{smallmatrix}\right]\left(\prod_{x_{i_j}} O_{j+1} \cdots P\right) \neq \gamma_{j-1}$.

The malicious prover sends $\tilde{p}_j(X_{i_j})$ of degree ≤ 1.

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

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

By definition of p_j, $\left[\begin{smallmatrix} x_1 := \omega_1 \\ \vdots \\ x_{i_j-1} := \omega_{i_j-1} \\ x_{i_j} := \omega_{i_j} \end{smallmatrix}\right]\left(O_{j+1} \cdots P\right) = p_j(\omega_{i_j})$.

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

Hence the output claim $\left[\begin{smallmatrix} x_1 := \omega_1 \\ \vdots \\ x_{i_j-1} := \omega_{i_j-1} \\ x_{i_j} := \omega_{i_j} \end{smallmatrix}\right]\left(O_{j+1} \cdots P\right) \overset{?}{=} \gamma_j$ is $p_j(\omega_{i_j}) \overset{?}{=} \tilde{p}_j(\omega_{i_j})$.

This last equality holds w.p. $\leq \frac{1}{|\mathbb{F}|}$ over the choice of $\omega_{i_j} \in \mathbb{F}$.

Analysis of Shamir's Protocol [2/3]

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

COMPLETENESS: Suppose that $\left[\begin{smallmatrix} X_1 := \omega_1 \\ X_{i_j} := \omega_{i_j}^{old} \\ X_s := \omega_s \end{smallmatrix}\right]\left(\nabla_{X_{i_j}} O_{j+1} \cdots P\right) = \gamma_{j-1}$ for some $s \geq i_j$.

The honest prover sends $p_j(X_{i_j}) := \left[\begin{smallmatrix} X_1 := \omega_1 \\ X_{i_j} \text{ (no value)} \\ X_s := \omega_s \end{smallmatrix}\right]\left(O_{j+1} \cdots P\right)$ of degree $\leq 3m$ (first reductions) or ≤ 2 (other reductions).

The verifier's check passes: $(\nabla_{X_{i_j}} p_j)(\omega_{i_j}^{old}) = \gamma_{j-1}$.

The next statement is always true: $\forall \omega_{i_j} \in \mathbb{F}$, $\left[\begin{smallmatrix} X_1 := \omega_1 \\ X_{i_j} := \omega_{i_j} \\ X_s := \omega_s \end{smallmatrix}\right]\left(O_{j+1} \cdots P\right) \overset{\text{def of } p_j}{=} p_j(\omega_{i_j}) \overset{\text{def of } \gamma_j}{=} \gamma_j$.

SOUNDNESS: Suppose that $\left[\begin{smallmatrix} X_1 := \omega_1 \\ X_{i_j} := \omega_{i_j}^{old} \\ X_s := \omega_s \end{smallmatrix}\right]\left(\nabla_{X_{i_j}} O_{j+1} \cdots P\right) \neq \gamma_{j-1}$ for some $s \geq i_j$.

The malicious prover sends $\tilde{p}_j(X_{i_j})$ of degree $\leq 3m$ (first reductions) or ≤ 2 (other reductions).

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

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

By definition of p_j, $\left[\begin{smallmatrix} X_1 := \omega_1 \\ X_{i_j} := \omega_{i_j} \\ X_s := \omega_s \end{smallmatrix}\right]\left(O_{j+1} \cdots P\right) = p_j(\omega_{i_j})$.

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

Hence the output claim $\left[\begin{smallmatrix} X_1 := \omega_1 \\ X_{i_j} := \omega_{i_j} \\ X_s := \omega_s \end{smallmatrix}\right]\left(O_{j+1} \cdots P\right) \overset{?}{=} \gamma_j$ is $p_j(\omega_{i_j}) \overset{?}{=} \tilde{p}_j(\omega_{i_j})$.

This last equality holds w.p. $\leq \frac{3m}{|\mathbb{F}|}$ or $\leq \frac{2}{|\mathbb{F}|}$ over the choice of $\omega_{i_j} \in \mathbb{F}$.

Analysis of Shamir's Protocol [3/3]

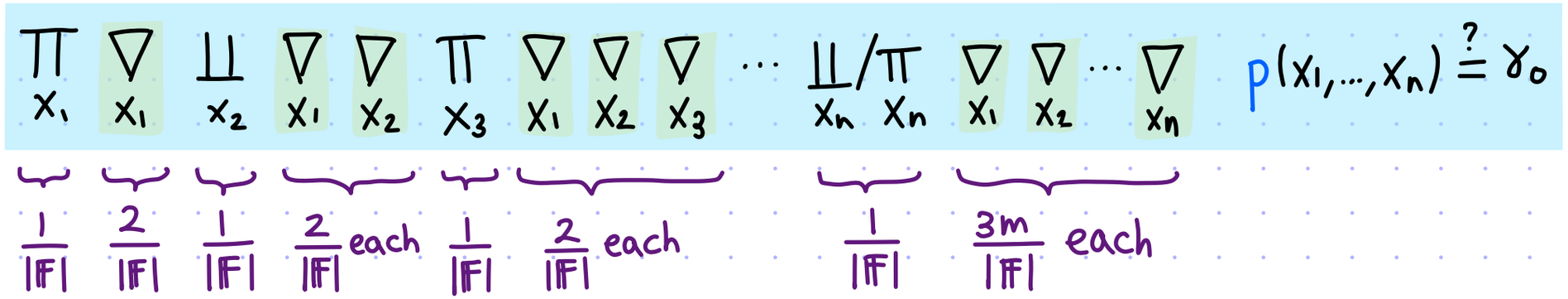
Overall Completeness:

In each round, if the current claim is true then the next claim is true w.p. 1.

After the last round, the final check ($p(\omega_1,\dots,\omega_n) \stackrel{?}{=} \gamma_k$) passes.

Overall Soundness:

The soundness error is at most the sum of the round errors:



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

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

Shamir's Protocol for a Simple Example

Evaluate: $\prod_{x_1} \nabla_{x_1} \prod_{x_2} x_1^2 + x_2 = \prod_{x_1} \nabla_{x_1} x_1^4 + x_1^2 = \prod_{x_1} 2x_1 = 0$. Hence $\gamma_0 := 0$.

Protocol execution:

$$\prod_{x_1} \nabla_{x_1} \prod_{x_2} x_1^2 + x_2 \stackrel{?}{=} \gamma_0$$

$p_1 := \nabla_{x_1} \prod_{x_2} x_1^2 + x_2 = 2x_1$ — $p_1(x_1)$ → — $p_1(0) \cdot p_1(1) \stackrel{?}{=} \gamma_0 \leftrightarrow (2 \cdot 0) \cdot (2 \cdot 1) \stackrel{?}{=} 0$ ✓

← a — $a \leftarrow \mathbb{F}$

$\gamma_1 := p_1(a) = 2 \cdot a$

$$[x_1 \to a]\left(\nabla_{x_1} \prod_{x_2} x_1^2 + x_2\right) \stackrel{?}{=} \gamma_1$$

$p_2 := \prod_{x_2} x_1^2 + x_2 = x_1^4 + x_1^2$ — $p_2(x_1)$ → — $(\nabla_{x_1} p_2)(a) \stackrel{?}{=} \gamma_1 \leftrightarrow a + a \stackrel{?}{=} 2 \cdot a$ ✓

← b — $b \leftarrow \mathbb{F}$

$\gamma_2 := p_2(b) = b^4 + b^2$

$$[x_1 \to b]\left(\prod_{x_2} x_1^2 + x_2\right) \stackrel{?}{=} \gamma_2$$

$p_3 := [x_1 \to b](x_1^2 + x_2) = b + x_2$ — $p_3(x_2)$ → — $p_3(0) \cdot p_3(1) \stackrel{?}{=} \gamma_2 \leftrightarrow b^2 \cdot (b^2 + 1) \stackrel{?}{=} b^4 + b^2$ ✓

← c — $c \leftarrow \mathbb{F}$

$\gamma_3 := p_3(c) = b^2 + c$

final check: $[x_1 \to b, x_2 \to c](x_1^2 + x_2) \stackrel{?}{=} \gamma_3 \leftrightarrow b^2 + c \stackrel{?}{=} b^2 + c$ ✓

On Shamir's Original Proof

The IP for TQBF that we saw is due to Alexander Shen.

Adi Shamir's original proof that IP = PSPACE relies on simple QBFs.

Let Φ be a quantified boolean formula where $\forall/\exists$ quantifiers may appear anywhere (not just prefix).

We say that Φ is simple if $\forall i \in [n]$ every occurrence of x_i is separated from its quantification point by ≤ 1 universal quantifier (& any number of other symbols).

Example: • $\forall x_1 \forall x_2 \exists x_3 \; ((x_1 \vee x_2) \wedge \forall x_4 \; (x_2 \wedge x_3 \wedge x_4))$ ← simple

• $\forall x_1 \forall x_2 \; ((x_1 \wedge x_2) \wedge \forall x_3 \; (\bar{x}_1 \wedge x_3))$ ← NOT simple

Define $TSQBF := \{ \Phi \mid \Phi$ is a closed quantified boolean formula that is simple and evaluates to true $\}$.

Lemma: $\exists$ efficient f s.t. $\forall \Phi \quad \Phi \in TQBF \leftrightarrow f(\Phi) \in TSQBF$.

(The rough idea is to introduce a new variable for each occurrence of each variable. This squares the number of variables.)

Can arithmetize a simple QBF Φ to obtain an arithmetic expression that:

(i) involves the operators $\prod_{x_i}$ and $\sum_{x_i}$, and (ii) has no degree problems.

One can then design an IP for the arithmetic expression.

Additional Slides:

TQBF is PSPACE-complete

TQBF is in PSPACE

Let $\Phi = Q_1 x_1 \, Q_2 x_2 \cdots Q_n x_n \; \varphi(x_1, \ldots, x_n)$ be a closed quantified boolean formula.

Here each $Q_i \in \{\forall, \exists\}$. Also: m = size of boolean formula φ, n = # variables.

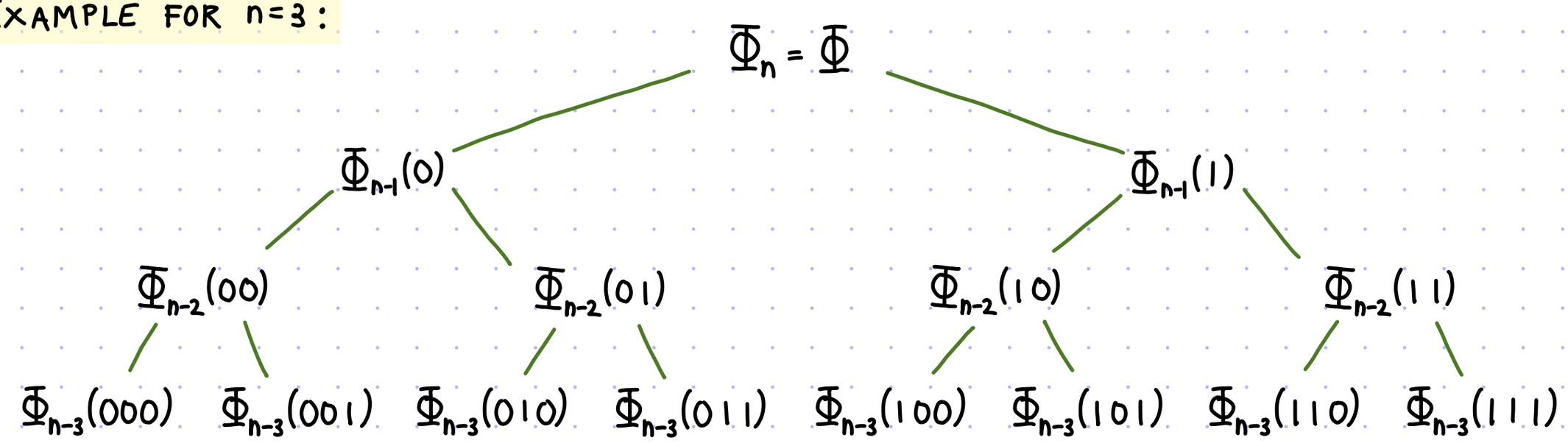
GOAL: evaluate Φ in poly(m,n) space

Define: $\begin{cases} \Phi_n := \Phi \\ \forall i \in \{n-1, n-2, \ldots, 0\}, \;\; \Phi_i(x_1, \ldots, x_{n-i}) := Q_{n-i+1} x_{n-i+1} \cdots Q_n x_n \;\; \varphi(x_1, \ldots, x_{n-i}, x_{n-i+1}, \ldots, x_n) \end{cases}$

Observe :
- $\Phi_0(x_1, \ldots, x_n) = \varphi(x_1, \ldots, x_n)$
- recurrence: $\Phi_n = Q_1 x_1 \; \Phi_{n-1}(x_1)$, $\Phi_{n-1}(x_1) = Q_2 x_2 \; \Phi_{n-2}(x_1, x_2)$, ...

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

EXAMPLE FOR n=3:



TQBF is PSPACE-Hard [1/2]

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, construct QBF Φ in time $\text{poly}(n)$ s.t. $\Phi = 1$ iff $x \in L$
(and hence of size $\text{poly}(n)$)

Define $G = G(M,x)$ to be the configuration graph of the computation of M on x:

$G = (V,E)$ where $V = \{ C : C \text{ is a possible state of } M(x) \}$ $\quad |V| = 2^{O(S)}$

$E = \{ (C_1, C_2) : M(x) \text{ in state } C_1 \text{ transitions to } C_2 \text{ in 1 step} \}$

There is a unique initial state C_{init} and a unique accepting state C_{acc}.

OBSERVATION: $x \in L \leftrightarrow \exists$ path in G from C_{init} to C_{acc}

We recursively define, for $i = 0, 1, 2, \ldots$, a QBF Φ_i s.t.

$\forall\, C_1, C_2 \in V \quad \Phi_i(C_1, C_2) = 1 \leftrightarrow \exists$ path in G from C_1 to C_2 of length $\leq 2^i$

The QBF that we seek is $\Phi := \Phi_{O(S)}(C_{init}, C_{acc})$.

We are left to show that we can construct Φ_i in time $\text{poly}(n,i)$.
(and hence of size $\text{poly}(n,i)$)

TQBF is PSPACE-Hard [2/2]

We want to construct Φ_i s.t. $\Phi_i(C_1, C_2) = 1 \leftrightarrow \exists$ path in G from C_1 to C_2 of length $\leq 2^i$

BASE CASE: $i = 0$

$\Phi_0(C_1, C_2)$:= "the boolean formula (with no quantifiers) obtained by applying the Cook-Levin theorem to the transition function of $M(x)$"

RECURSIVE CASE: $i > 0$ consider a state half-way

$\Phi_i(C_1, C_2) := \exists C_3 \quad \Phi_{i-1}(C_1, C_3) \wedge \Phi_{i-1}(C_3, C_2)$

The QBF Φ_i computes the correct boolean function.

PROBLEM: $|\Phi_i| \geq 2 \cdot |\Phi_{i-1}| \geq 2^i$, so this construction is inefficient

(Also, Φ_i has only existential quantifiers so we do not expect to capture PSPACE.)

SOLUTION: use extra quantifiers to include Φ_{i-1} only ONCE

$$\Phi_i(C_1, C_2) := \exists C_3 \ \forall D_1, D_2 \quad \big((D_1 = C_1 \wedge D_2 = C_3) \vee (D_1 = C_3 \wedge D_2 = C_2)\big) \rightarrow \Phi_{i-1}(D_1, D_2)$$

syntactic sugar: $\varphi_1 \rightarrow \varphi_2$ stands for $\overline{\varphi_1} \vee \varphi_2$

Now $|\Phi_i| = |\Phi_{i-1}| + \text{poly}(S) = \text{poly}(S, i) = \text{poly}(n, i)$. (Since $S(n) = \text{poly}(n)$.)

Bibliography

IP=PSPACE

- [Shamir 1992]: IP = PSPACE, by Adi Shamir. — original proof
- [Shen 1992]: IP = PSPACE: simplified proof, by Alexander Shen. — alternate proof
- [Meir 2010]: IP = PSPACE using error-correcting codes, by Or Meir. (▸Video) — combinatorial proof

Related

QIP is a quantum analogue of IP

- [JJUW 2009]: QIP = PSPACE, by Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, John Watrous.
- [CFS 2017]: A zero knowledge sumcheck and its applications, by Alessandro Chiesa, Michael Forbes, Nicholas Spooner. — ZK variants of sumcheck and Shamir's protocols
- [GOS 2024]: Perfect zero-knowledge PCPs for #P, by Tom Gur, Jack O'Connor, Nicholas Spooner.
- [Babai 1990]: E-mail and the unexpected power of interaction, by László Babai. — historical retrospective