Lesson 11: IP = PSPACE

Theme: The equivalence between the class **IP** and **PSPACE**.

1 The verifier for the number of satisfying assignments of boolean formulas

Consider the following language $L_{\sharp\mathsf{SAT}}$:

$$L_{\sharp\mathsf{SAT}} \stackrel{\text{def}}{=} \left\{ (\varphi, k) \;\middle|\; \begin{array}{l} \varphi \text{ is a boolean formula} \\ \text{and } k \text{ is the number of its satisfying assignments (in binary)} \end{array} \right\}$$

We will describe its IP protocol.

The arithmetization of boolean formulas. Let $\varphi(x_1, \ldots, x_n)$ be a boolean formula with variables $x_1, \ldots, x_n$. We first convert it into a multi-variate polynomial $\widetilde{\varphi}(x_1, \ldots, x_n)$ by replacing the operators $\wedge$, $\vee$ and $\neg$ as follows.

$$\begin{array}{rcl} \neg\varphi_1 & \mapsto & 1 - \widetilde{\varphi_1} \\ \varphi_1 \wedge \varphi_2 & \mapsto & \widetilde{\varphi_1} \cdot \widetilde{\varphi_2} \\ \varphi_1 \vee \varphi_2 & \mapsto & 1 - (1 - \widetilde{\varphi_1}) \cdot (1 - \widetilde{\varphi_2}) \end{array}$$

By a straightforward induction on φ, it is not difficult to show that $\varphi(\bar{b}) = \widetilde{\varphi}(\bar{b})$, for every $\bar{b} = (b_1, \ldots, b_n) \in \{0, 1\}^n$. Thus,

$$\sharp\varphi \;=\; \sum_{x_1=0}^{1} \sum_{x_2=0}^{1} \cdots \sum_{x_n=0}^{1} \widetilde{\varphi}(x_1, \ldots, x_n).$$

The IP verifier for $L_{\sharp\mathsf{SAT}}$. Let (φ, k) be the input and $x_1, \ldots, x_n$ be the variables in φ. Let d be the maximal degree of each variable in $\widetilde{\varphi}$. Let $\mathbb{F}$ be some finite field with size $\geqslant 3d$.

Denote by $f_i(x_1, \ldots, x_i)$ the following polynomial:

$$f_i(x_1, \ldots, x_i) \stackrel{\text{def}}{=} \sum_{x_{i+1}=0}^{1} \cdots \sum_{x_n=0}^{1} \widetilde{\varphi}(x_1, x_2, \ldots, x_n)$$

In each round i, on some numbers $r_1, \ldots, r_i, t \in \mathbb{F}$, the prover tries to convince the verifier that the following holds.

$$f_i(r_1, \ldots, r_i) \;=\; t, \tag{1}$$

The protocol works by recursively on i.

In round 0, the prover "tells" the verifier that the value in (2) is k. Otherwise, the verifier rejects immediately.

For each $i \leqslant n-1$, round i works as follows Let $r_1, \ldots, r_i$ and t be the values that the prover tries to convince verifier that Eq.(1) holds.

- The verifier asks for the polynomial $f_{i+1}(r_1, \ldots, r_i, x_{i+1})$.
- Suppose the prover replies with $g(x_{i+1})$.

- The verifier checks if the following holds.

$$t \;=\; g(0) + g(1)$$

  Reject, if it does not. Otherwise, continue.

- The verifier chooses a random $r \in \mathbb{F}$ and proceeds to the next round to check:

$$g(r) \;=\; f_{i+1}(r_1, \ldots, r_i, r).$$

Note that $f_n(r_1, \ldots, r_n) = \widetilde{\varphi}(r_1, \ldots, r_n)$. Thus, in the last round $i = n-1$, the verifier can compute the value $f_n(r_1, \ldots, r_n)$ directly.

Proof of correctness. Note that if $(\varphi, k) \in L_{\sharp\mathsf{SAT}}$, the verifier always accepts when the prover always gives correct answers. That is, if in each round i the prover replies with $f_i(r_1, \ldots, r_{i-1}, x_i)$, the verifier always accepts.

Suppose $(\varphi, k) \notin L_{\sharp\mathsf{SAT}}$. That is, the following holds.

$$k \;\neq\; \sum_{x_1=0}^{1} \sum_{x_2=0}^{1} \cdots \sum_{x_n=0}^{1} \widetilde{\varphi}(x_1, \ldots, x_n)$$

In the following let $g_i(x_i)$ denote the polynomial sent by the prover in round i.

We can assume that in round 1 the prover replies with a polynomial $g_1(x_1)$ where $k = g_1(0)+g_1(1)$. Otherwise, verifier rejects immediately. Note that this means that $g_1(x_1) \neq f_1(x_1)$.

We will calculate the probability that V rejects. Consider a fixed interaction between a prover and the verifier. Let $r_1, \ldots, r_n$ be the random strings generated by the verifier. There are two scenarios.

(S1) In round n, the prover's reply $g(x_n)$ is not correct, i.e., $g_n(x_n) \neq f_n(r_1, \ldots, r_{n-1}, x_n)$.

(S2) In round n, the prover's reply $g(x_n)$ is correct, i.e., $g_n(x_n) = f_n(r_1, \ldots, r_{n-1}, x_n)$.

In (S1) the probability that the verifier accepts in round n is:

$$\mathbf{Pr}_r[\ V \text{ accepts }\ ] \;=\; \mathbf{Pr}_r[\ g_n(r) = f_n(r_1, \ldots, r_{n-1}, r)\ ] \;\leqslant\; \frac{d}{|\mathbb{F}|} \;\leqslant\; \frac{1}{3}$$

The second last inequality comes from the fact that the degree of g_n and f_n are at most d, hence, there at most d such r where $g(r) = f_n(r_1, \ldots, r_{n-1}, r)$.

We now consider (S2). Since $g_1(x_1) \neq f_1(x_1)$ and $g_n(x_n) = f_n(r_1, \ldots, r_{n-1}, x_n)$, there is $1 \leqslant i \leqslant n$ such that:

$$g_{i-1}(x_{i-1}) \neq f_{i-1}(r_1, \ldots, r_{i-2}, x_{i-1}) \qquad \text{and} \qquad g_i(x_i) = f_i(r_1, \ldots, r_{i-1}, x_i)$$

The probability that the verifier continues in round i is:

$$\begin{aligned}
\mathbf{Pr}_{r_{i-1}}[\text{ the verifier continues in round } i\ ] &= \mathbf{Pr}_{r_{i-1}}[\ g_{i-1}(r_{i-1}) = g_i(0) + g_i(1)\ ] \\
&= \mathbf{Pr}_{r_{i-1}}[\ g_{i-1}(r_{i-1}) = f_{i-1}(r_1, \ldots, r_{i-1})\ ] \\
&\leqslant \frac{d}{|\mathbb{F}|} \;\leqslant\; \frac{1}{3}
\end{aligned}$$

Again, the second last inequality is due to the degree of g_n and f_n being at most d. In both scenarios (S1) and (S2), the probability that the verifier rejects is $\geqslant 2/3$. Thus, we have shown the IP protocol for the language $L_{\sharp\mathsf{SAT}}$. We state this result formally.

Theorem 11.1 (Lund, Fortnow, Karloff, Nisan 1990) $L_{\sharp\mathsf{SAT}} \in \mathbf{IP}$. *Hence,* $\mathbf{PH} \subseteq \mathbf{IP}$.

The inclusion $\mathbf{PH} \subseteq \mathbf{IP}$ follows from the algorithm for Toda's Theorem, i.e., Theorem 9.1.

2 The verifier for TQBF

We will now describe the IP protocol for TQBF. The idea is simple. To verify that $\forall x\ \varphi(x)$ is true, we check that $\widetilde{\varphi}(0) \cdot \widetilde{\varphi}(1) \neq 0$. Likewise, to verify that $\exists x\ \varphi(x)$ is true, we check that $1 - (1 - \widetilde{\varphi}(0)) \cdot (1 - \widetilde{\varphi}(1)) \neq 0$.

We formalize this intuition as follows. Let $q(\bar{x}, y_1, \ldots, y_n)$ be a polynomial where $\bar{x}$ is a vector of variables and $y_1, \ldots, y_n$ are variables. The expression $\mathsf{Q}_1 y_1 \cdots \mathsf{Q}_n y_n\ q(\bar{x}, y_1, \ldots, y_n)$, where each $\mathsf{Q}_i \in \{\mathsf{A}, \mathsf{E}\}$, defines a polynomial $p(\bar{x})$ as follows.

- If $\mathsf{Q}_1 = \mathsf{A}$:

$$p(\bar{x}) \stackrel{\mathsf{def}}{=} \Big(\mathsf{Q}_2 y_2 \cdots \mathsf{Q}_n y_n\ q(\bar{x}, 0, y_2, \ldots, y_n)\Big) \cdot \Big(\mathsf{Q}_2 y_2 \cdots \mathsf{Q}_n y_n\ q(\bar{x}, 1, y_2, \ldots, y_n)\Big)$$

- If $\mathsf{Q}_1 = \mathsf{E}$:

$$p(\bar{x}) \stackrel{\mathsf{def}}{=} 1 - \Big(1 - \mathsf{Q}_2 y_2 \cdots \mathsf{Q}_n y_n\ q(\bar{x}, 0, y_2, \ldots, y_n)\Big) \cdot \Big(1 - \mathsf{Q}_2 y_2 \cdots \mathsf{Q}_n y_n\ q(\bar{x}, 1, y_2, \ldots, y_n)\Big)$$

Intuitively, the IP protocol for TQBF works as follows. Let $\Psi \stackrel{\mathsf{def}}{=} Q_1 x_1 \cdots Q_n x_n\ \varphi(x_1, \ldots, x_n)$ be the input QBF. Its arithmetization is $\widetilde{\Psi} \stackrel{\mathsf{def}}{=} \mathsf{Q}_1 x_1 \cdots \mathsf{Q}_n x_n\ \widetilde{\varphi}(x_1, \ldots, x_n)$, where each $\forall x_i$ is replaced by $\mathsf{A} x_i$ and each $\exists x_i$ by $\mathsf{E} x_i$. It is not difficult to show that Ψ is true QBF if and only if $\widetilde{\Psi} = 1$.

Checking whether $\widetilde{\Psi} = 1$ can be done by similar method in the previous section. In each round i the verifier asks the prover for the polynomial:

$$f_i(r_1, \ldots, r_{i-1}, x_i) \stackrel{\mathsf{def}}{=} \mathsf{Q}_{i+1} x_{i+1} \cdots \mathsf{Q}_n x_n\ \widetilde{\varphi}(r_1, \ldots, r_{i-1}, x_i, x_{i+1}, \ldots, x_n)$$

for some randomly chosen numbers $r_1, \ldots, r_{i-1}$. However, note that the degree of x_i can be 2^{n-i}. For this, we introduce a new operator $\mathsf{L}x$, whose semantics are defined as follows. The expression $\mathsf{L}z\mathsf{Q}_1 y_1 \cdots \mathsf{Q}_n y_n\ q(\bar{x}, z, y_1, \ldots, y_n)$ defines the following polynomial $p(\bar{x}, z)$:

$$p(\bar{x}, z) \stackrel{\mathsf{def}}{=} (1 - z)\mathsf{Q}_1 y_1 \cdots \mathsf{Q}_n y_n\ q(\bar{x}, 0, y_1, \ldots, y_n) \ + \ z\mathsf{Q}_1 y_1 \cdots \mathsf{Q}_n y_n\ q(\bar{x}, 1, y_1, \ldots, y_n)$$

In the expression $\mathsf{L}z\mathsf{Q}_1 y_1 \cdots \mathsf{Q}_n y_n\ q(\bar{x}, z, y_1, \ldots, y_n)$, the variables $\bar{x}$ and z are free variables. The operator $\mathsf{L}z\ q(\bar{x}, z)$ means "linearize" the variable z in the polynomial $q(\bar{x}, z)$.

Since in the operators A and E we are only evaluating the polynomial on 0 and 1 and $x^k = x$ for $x \in \{0, 1\}$, the value $\mathsf{Q}_1 x_1 \cdots \mathsf{Q}_n x_n\ \widetilde{\varphi}(x_1, \ldots, x_n)$ is equal to:

$$\mathsf{Q}_1 x_1 \mathsf{L} x_1\ \mathsf{Q}_2 x_2 \mathsf{L} x_1 \mathsf{L} x_2\ \cdots\ \mathsf{Q}_n x_n \mathsf{L} x_1 \cdots \mathsf{L} x_n\ \widetilde{\varphi}(x_1, \ldots, x_n) \tag{2}$$

The IP protocol will verify that the value in Eq.(2) is 1.

It works recursively where in each round i, on some numbers $r_1, \ldots, r_k$ and t, the prover tries to convince the verifier that the following holds.

$$\mathsf{Q}_i z_i \cdots \mathsf{Q}_m z_m\ \widetilde{\varphi}(r_1, \ldots, r_k, x_{k+1}, \ldots, x_n) \ = \ t \tag{3}$$

where $x_{k+1}, \ldots, x_n$ are the variables quantified by A or E in $\mathsf{Q}_i z_i \cdots \mathsf{Q}_m z_m$.

In round 0, the prover "tells" the verifier that the value in (2) is 1. Otherwise, the verifier rejects immediately.

In round i, suppose the values $r_1, \ldots, r_k$ and t are already given. The verifier tries to verify that (3) is true as follows. There are three cases.

Case 1: $\mathsf{Q}_i z_i$ is $\mathsf{A} x_{k+1}$.

- The verifier asks for the polynomial:

$$\mathsf{Q}_{i+1}z_{i+1}\cdots\mathsf{Q}_m z_m\ \widetilde{\varphi}(r_1,\ldots,r_k,x_{k+1},\ldots,x_n)$$

- Suppose the prover replies with $g(x_{k+1})$.
- The verifier checks the following.

$$t\ =\ g(0)\cdot g(1)$$

  Reject, if it does not hold. Otherwise, continue.

- The verifier chooses a random number $r\in\mathbb{F}$ and proceeds to the next round to verify:

$$g(r)\ =\ \mathsf{Q}_{i+1}z_{i+1}\cdots\mathsf{Q}_m z_m\ \widetilde{\varphi}(r_1,\ldots,r_k,r,x_{k+2},\ldots,x_n)$$

Case 2: $\mathsf{Q}_i z_i$ is $\mathsf{E}x_{k+1}$.
Similar to above, but the verifier checks the following.

$$t\ =\ 1\ -\ (1-g(0))\cdot(1-g(1))$$

Case 3: $\mathsf{Q}_i z_i$ is $\mathsf{L}x_j$, for some $1\leqslant j\leqslant k$.

- The verifier asks for the polynomial:

$$\mathsf{Q}_{i+1}z_{i+1}\cdots\mathsf{Q}_m z_m\ \widetilde{\varphi}(r_1,\ldots,r_{j-1},x_j,r_{j+1},\ldots,r_k,x_{k+1},\ldots,x_n)$$

- Suppose the prover replies with $g(x_j)$.
- The verifier checks the following.

$$t\ =\ (1-r_j)\cdot g(0)\ +\ r_j\cdot g(1)$$

  Reject, if it does not hold. Otherwise, continue.

- The verifier chooses a random number $r\in\mathbb{F}$ and proceeds to the next round to verify:

$$g(r)\ =\ \mathsf{Q}_{i+1}z_{i+1}\cdots\mathsf{Q}_m z_m\ \widetilde{\varphi}(r_1,\ldots,r_{j-1},r,r_{j+1},\ldots,r_k,x_{k+1},\ldots,x_n)$$

The probabilistic analysis is similar to the one in the previous section. If Ψ is a true QBF, then the verifier always accepts provided that the prover always answers correctly. If Ψ is not correct, then in some round i the polynomial $g(x_j)$ sent by the prover is not correct. We can show that in such round the probability that the verifier chooses the value r that invalidates the prover's claim is at least 2/3.

Theorem 11.2 (Shamir 1990). $\mathsf{TQBF}\in\mathbf{IP}$. *Hence,* **IP = PSPACE**.*

Theorem 11.3 *If* $\mathbf{PSPACE}\subseteq\mathbf{P}_{/\mathsf{poly}}$, *then* **PSPACE = MA**.

*The IP protocol described in this note is from "A. Shen. **IP = PSPACE**: Simplified proof. JACM, vol. 39, no. 4, Oct. 1992, pp. 878–880."