Lesson 12: 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 $L_{\sharp\mathsf{SAT}}$:

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

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,

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

An intuitive description of the 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:

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

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.

For each $1 \leqslant i \leqslant n$, round i works as follows Let $r_1, \ldots, r_{i-1}$ and t be the values from the previous round and the prover tries to convince the verifier that the following holds.

$$f_{i-1}(r_1, \ldots, r_{i-1}) \;=\; t, \tag{2}$$

- The verifier asks for the polynomial $f_i(r_1, \ldots, r_{i-1}, x_i)$.

- Suppose the prover replies with $g(x_i)$.
- 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:

$$g(r) \ = \ f_i(r_1, \ldots, r_{i-1}, r).$$

Note that in round 1 the value t is k. In the last round $i = n$, the verifier can compute the value $f_n(r_1, \ldots, r_{n-1}, r)$ directly.

A more precise description of the 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 the polynomial $\widetilde{\varphi}(x_1, \ldots, x_n)$. Let $\mathbb{F}$ be some finite field with size $\geqslant 3d$. The verfiier works as follows, where all polynomial evaluation is computed in the field $\mathbb{F}$.

(Round 1)

- The verifier asks the prover for the polynomial $f_1(x_1)$.
- Suppose the prover replies with a polynomial $g_1(x_1)$.
- The verifier checks if $k = g_1(0) + g_1(1)$.

  If not, the verifier rejects immediately. Otherwise, continue.

(Round 2)

- The verifier randomly chooses a number $r_1 \in \mathbb{F}$ and asks the prover for the polynomial $f_2(r_1, x_2)$.
- Suppose the prover replies with a polynomial $g_2(x_2)$.
- The verifier checks if $g_1(r_1) = g_2(0) + g_2(1)$.

  If not, the verifier rejects immediately. Otherwise, continue.

... and so on, where each round $i \leqslant n$ is as follows.

(Round i)

- The verifier randomly chooses a number $r_{i-1} \in \mathbb{F}$ and asks the prover for the polynomial $f_i(r_1, \ldots, r_{i-1}, x_i)$.
- Suppose the prover replies with a polynomial $g_i(x_i)$.
- The verifier checks if $g_{i-1}(r_{i-1}) = g_i(0) + g_i(1)$.

  If not, the verifier rejects immediately. Otherwise, continue.

(Round $n+1$)

- The verifier randomly chooses a number $r_n \in \mathbb{F}$.
- The verifier checks if $g_n(r_n) = f_n(r_1, \ldots, r_n)$. It accepts if and only if the equality holds.

  Note that $f_n(r_1, \ldots, r_n) = \widetilde{\varphi}(r_1, \ldots, r_n)$.

Proof of correctness. Note that if $(\varphi, k) \in L_{\sharp\mathsf{SAT}}$, then the protocol works correctly. For each $r_1, \ldots, r_{i-1}$, the prover replies with $f_i(r_1, \ldots, r_{i-1}, x_i)$. So, $\mathbf{Pr}[\ V \text{ accepts }] = 1$.

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

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

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) $g_n(x_n) \neq f_n(r_1, \ldots, r_{n-1}, x_n)$.

(S2) $g_n(x_n) = f_n(r_1, \ldots, r_{n-1}, x_n)$.

That is, in (S1) the polynomial $g_n(x_n)$ sent by the prover is not "correct" whereas in (S2) $g_n(x_n)$ is correct.

In (S1) the probability that the verifier accepts in round $n+1$ is:

$$\mathbf{Pr}_{r_n}[\ V \text{ accepts }] \ = \ \mathbf{Pr}_{r_n}[\ g_n(r_n) = f_n(r_1, \ldots, r_n)\ ] \ \leqslant \ \frac{d}{|\mathbb{F}|} \ \leqslant \ \frac{1}{3}$$

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_n where $g_n(r_n) = f_i(r_1, \ldots, r_{n-1}, r_n)$.

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 12.1 (Lund, Fortnow, Karloff, Nisan 1990) $L_{\sharp\mathsf{SAT}} \in \mathbf{IP}$.

Corollary 12.2 $\mathbf{PH} \subseteq \mathbf{IP}$.

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 is 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:

$$\mathsf{Q}_1 x_1 \mathsf{L} x_1 \; \mathsf{Q}_2 x_2 \mathsf{L} x_1 \mathsf{L} x_2 \; \cdots \; \mathsf{Q}_n x_n \mathsf{L} x_1 \cdots \mathsf{L} x_n \; \widetilde{\varphi}(x_1, \ldots, x_n) \tag{3}$$

The IP protocol will verify that the value in Eq.(3) 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.

$$\mathsf{Q}_i z_i \cdots \mathsf{Q}_m z_m \; \widetilde{\varphi}(r_1, \ldots, r_k, x_{k+1}, \ldots, x_n) \; = \; t, \tag{4}$$

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 (3) 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 (4) is true as follows.

- If $\mathsf{Q}_i z_i$ 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:

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

- If $\mathsf{Q}_i z_i$ is $\mathsf{E} x_{k+1}$.

  Similar to above, but the verifier checks the following.

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

- If $\mathsf{Q}_i z_i$ 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.

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

  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:

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

Theorem 12.3 (Shamir 1990). $\mathsf{TQBF} \in$ **IP**. *Hence,* **IP** = **PSPACE**.

Theorem 12.4 *If* **PSPACE** $\subseteq$ $\mathbf{P}_{/\mathsf{poly}}$, *then* **PSPACE** = **MA**.