An interactive proof is an exchange between a prover and a verifier. The exchange has a polynomial number of rounds and the messages sent in each round have polynomial length. The prover has unbounded computational power and the verifier may use randomization. The prover wants to convince the verifier that a certain fact holds. If the fact holds, the prover should always succeed to convince the verifier. If the fact does not hold, the prover should succeed with probability at most one-half.

The presentation follows Sipser's book.

1 Warm-Up: Verifying the Number of Satisfying Assignments of a Boolean Formula

Let

$$\#SAT = \{\, (\Phi,k);\ \Phi \text{ is a boolean formula with exactly } k \text{ satisfying assignments.} \,\}.$$

We will show $\#SAT \in \mathrm{IP}$. The proof introduces the main technique needed to prove $QBF \in \mathrm{IP}$.

Let Φ be a boolean formula over the boolean variables x_1 to x_n. We construct a polynomial $p = A(\Phi)$ in real variables x_1 to x_n such that $p(x) = \Phi(x)$ for boolean arguments. We call p the *arithmetization* of Φ. p is constructed inductively.
(a) If $\Phi = x$ for a variable x, then $p = A(\Phi) = x$.
(b) If $\Phi = \neg\Phi_1$, then $p = A(\Phi) = 1 - A(\Phi')$.
(c) If $\Phi = \Phi_1 \wedge \Phi_2$, then $p = A(\Phi) = A(\Phi_1) \cdot A(\Phi_2)$.
The rules are readily extended to other connectives, e.g., if $\Phi = \Phi_1 \vee \Phi_2$ then $A(\Phi) = 1-(1-A(\Phi_1)) \cdot (1 - A(\Phi_2))$.

We next define a sequence of specializations of p. For $0 \le i \le n$, let

$$f_i(x_1,\ldots,x_i) = \sum_{a_{i+1},\ldots,a_n \in \{0,1\}} p(x_1,\ldots,x_i,a_{i+1},\ldots,a_n).$$

Then f_0 is the number of satisfying assignments of Φ, $f_n = p$, $\deg f_i \le n$ for all i, and we have the recursion

$$f_i(x_1,\ldots,x_i) = f_{i+1}(x_1,\ldots,x_i,0) + f_{i+1}(x_1,\ldots,x_i,1).$$

We are now ready for the interactive proof system for membership in $\#SAT$. Let (Φ,k) be a pair consisting of a boolean formula and an integer. The prover P wants to convince the verifier V that Φ has exactly k satisfying assignments. If Φ has exactly k satisfying assignments, the prover will succeed. If $(\Phi,k) \notin \#SAT$, the prover will succeed with probability at most $n^2/2^n$.

Let q be a prime larger than 2^n. Arithmetic is in F_q.

The protocol will consist of n rounds numbered 1 to n. At the beginning of the i-th round, the verifier will keep a number y_{i-1}. Also, it will have chosen $i-1$ random numbers r_1 to r_{i-1}. It will believe $(\Phi,k) \in \#SAT$ iff the prover convinces him that $y_{i-1} = f_{i-1}(r_1,\ldots,r_{i-1})$.

We start with $y_0 = k$.

After the n-th round (= before the $(n+1)$-th round), the verifier has a number y_n and it will have chosen n numbers r_1 to r_n. It will believe $(\Phi,k) \in \#SAT$ iff the prover convinces him that

$y_n = f_n(r_1,\ldots,r_n)$. However, there is no convincing to be done anymore, since $f_n = p$. The verifier simply checks whether $y_n = p(r_1,\ldots,r_n)$. It accepts if the equality holds and rejects otherwise.

We next describe the interaction in round i.

Round i.

(1) The prover sends to V the coefficients of a polynomial q_i in one variable z of degree at most n. Allegedly, these are the coefficients of $f_i(r_1,\ldots,r_{i-1},z) \in \mathbb{Z}_p[z]$.

(2) The verifier checks $y_{i-1} = q_i(0) + q_i(1)$. If this is not the case, it rejects. Otherwise, it chooses a random r_i and sets $y_i = q_i(r_i)$. It sends r_i to the prover.

Theorem 1 *If $(\Phi,k) \in \#SAT$, the prover can convince the verifier with certainty. If If $(\Phi,k) \notin \#SAT$, the prover can fool the verifier with probability at most $n \cdot n/2^n$.*

Proof: If $(\Phi,k) \in \#SAT$, the prover sends $q_i = f_i(r_1,\ldots,r_{i-1},z)$ in round i.

Assume $(\Phi,k) \notin \#SAT$. Let q_1 to q_n be the sequence of polynomials sent by P. We claim

$$\mathrm{prob}(y_{i-1} = f_{i-1}(r_1,\ldots,r_{i-1})) \le (i-1)\cdot n/2^n$$

for $0 \le i \le n$. This is true for $i = 1$, since $y_0 = k$ and $f_0 \ne k$. Assume now that $y_{i-1} \ne f_{i-1}(r_1,\ldots,r_{i-1})$. The prover sends a polynomial q_i and the verifier checks $y_{i-1} = q_i(0)+q_i(1)$. If we not have equality, the verifier rejects and we are done. So assume otherwise. Then

$$q_i(0)+q_i(1) = y_{i-1} \ne f_{i-1}(r_1,\ldots,r_{i-1}) = f_i(r_1,\ldots,r_{i-1},0)+f_i(r_1,\ldots,r_{i-1},1)$$

and hence the polynomials $q_i(z)$ and $f_i(r_1,\ldots,r_{i-1},x)$ are not identical. Hence there are at most n arguments on which they agree. Thus $\mathrm{prob}(q(r_i) = f_i(r_1,\ldots,r_i) \,|\, y_{i-1} \ne f_{i-1}(r_1,\ldots,r_{i-1})) \le n/q \le n/2^n$ and hence

$$\begin{aligned}
&\mathrm{prob}(y_i = f_i(r_1,\ldots,r_i)) \\
&\qquad \le \mathrm{prob}(y_{i-1} = f_{i-1}(r_1,\ldots,r_{i-1})) + \mathrm{prob}(y_i = f_i(r_1,\ldots,r_i) \,|\, y_{i-1} \ne f_{i-1}(r_1,\ldots,r_{i-1})) \\
&\qquad \le (i-1)\cdot n/2^n + n/2^n = i \cdot n/2^n.
\end{aligned}$$

∎

2 PSPACE $\subseteq$ IP

We need to show that quantified boolean formula are in IP.

A First Approach. We proceed as in the preceding section. We first arithmetize the body of the quantified formula. This gives us a polynomial p in n variables. Then we eliminate the variables. We define

$$f_i(x_1,\ldots,x_i) = \begin{cases} 1 & \text{if } Q_{i+1}x_{i+1}\ldots Q_nx_n\Phi(x_1,\ldots,x_i,x_{i+1},\ldots,x_n) \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

Then f_0 is the truth value of the quantified boolean formula, and we have the identities

- if $Q_{i+1} = \forall$: $\quad f_i(x_1,\ldots,x_i) = f_{i+1}(x_1,\ldots,x_i,0)\cdot f_{i+1}(x_1,\ldots,x_i,1)$
- if $Q_{i+1} = \exists$: $\quad f_i(x_1,\ldots,x_i) = f_{i+1}(x_1,\ldots,x_i,0) * f_{i+1}(x_1,\ldots,x_i,1)$

where $a * b = 1-(1-a)\cdot(1-b)$. There is one problem here. Each quantifier might double the degree of the polynomial and hence we end up with polynomials of exponential degree. This is bad, as it would take exponential time for the prover to transfer the coefficients of such polynomials.

The Solution. We need one more technique: linearization. Let r be a polynomial in the variable z. Consider $\hat{r}$ defined as

$$\hat{r}(z) = (1-z)r(0) + zr(1).$$

Then $\hat{r}$ is linear in z and agrees with r on binary arguments. It may agree with r on more arguments.

Instead of $Q_1x_1\ldots Q_nx_np$, we consider

$$Q_1x_1Rx_1Q_2x_2Rx_1Rx_2\ldots Q_nx_nRx_1\ldots Rx_np;$$

here R stands for reduction or linearization. Let us write this as

$$S_1z_1\ldots S_mz_m\ p$$

where each S_i stands for $\forall$, $\exists$ or R and each z_i stands for one of the original variables.

Note that

$$Rx_1\ldots Rx_np = \sum_{(a_1,\ldots,a_n)\in\{0,1\}^n} \prod_i x_i^{(a_i)} p(a_1,\ldots,a_n),$$

where $x_i^{(1)} = x_i$ and $x_i^{(0)} = 1-x_i$, i.e., we really do disjunctive normal form.

We now define a sequence of polynomials f_0 to f_m. f_m has n variables and is the arithmetization of Φ. If S_i is equal to $\forall$ or $\exists$, then f_{i-1} has one less variable than f_i and

$$f_{i-1}(\ldots) = \begin{cases} f_i(\ldots,0)\cdot f_i(\ldots,1) & \text{if } S_i = \forall, \\ f_i(\ldots,0) * f_i(\ldots,1) & \text{if } S_i = \exists. \end{cases}$$

If S_iz_i is equal to Rx_j, f_{i-1} has the same number of variables as f_i and

$$f_{i-1}(\ldots,x_j,\ldots) = (1-x_j)f_i(\ldots,0,\ldots) + x_jf_i(\ldots,1,\ldots).$$

Then $f_0 = 1$ if and only if the quantified boolean formula is true, $f_0 = 0$ otherwise. Note that p, the arithmetization of Φ has degree at most n. We then do n linearization steps, bringing all the degrees down to 1. Forall and exists-quantors increase the degree again to 2. Note that the rules for $\forall$ and $\exists$ square the degrees of the remaining variables, i.e, if Q is equal to $\forall$ or $\exists$, we obtain a quadratic polynomial which we linearize then by reductions on all variables.

The protocol is quite similar to the protocol above. Initially, the verifier sets $y_0 = 1$. The prover wants to convince the verifier that $y_0 = f_0$. We proceed in rounds.

Consider the round corresponding to $S_i z_i$. f_{i-1} has a certain degree, say d_{i-1}. The verifier has already chosen the corresponding number of random values $r_1, \ldots, r_{d_{i-1}}$. It also has a value y_{i-1}. The prover still has to convince the verifier that $y_{i-1} = f_{i-1}(r_1, \ldots)$.

The prover sends the coefficients of some univariate polynomial $q(z)$ of degree at most n. In an exchange corresponding to a true input, this is $f_i(r_1, \ldots, z)$ if $S_i = \forall$ or $S_i = \exists$, and it is $f_i(r_1, \ldots, r_{j-1}, z, r_{j+1}, \ldots, r_{d_i})$ if $S_i z_i = Rx_j$. The verifier checks that q satisfies the recurrence formulas for the f_i's where the left hand side is replaced by y_{i-1}. $y_{i-1} = q(0) \cdot q(1)$ if $S_i = \forall$, and $y_{i-1} = q(0) * q(1)$ if $S_i = \exists$.
$y_{i-1} = (1 - r_j) q(0) + r_j q(1)$ if $S_i z_i = Rx_j$. If the check fails, V rejects.

Otherwise, V chooses a random value r, sets y_i to $q(r)$, and sends r to P. The r either extends the sequence of random values (if S_i is a quantifier) or replaces the value r_j (if $S_i z_i = Rx_j$).

Once we have worked through the prefix, V checks whether $f_m(r_1, \ldots, r_n) = y_m$. It accepts if the equality holds.

Theorem 2 *TQBF $\in$ IP.*

Proof: If the formula is true, the prover plays according to the rules and the verifier accepts.

If the formula is false, we have $y_0 \neq f_0$. If the verifier accepts, we have $f_m(r_1, \ldots, r_n) = y_m$. Hence there must be an i such that $y_{i-1} \neq f_{i-1}(r_1, \ldots)$ and $y_i = f_i(r_1, \ldots, r)$.

If $S_i = \forall$ or $\exists$, we have ($\oplus$ stands for $\cdot$ or $*$)

$$q_i(0) \oplus q_i(1) = y_{i-1} \neq f_{i-1}(r_1, \ldots) = f_i(r_1, \ldots, 0) \oplus f_i(r_1, \ldots, 1)$$

and hence $q_i(z)$ and $f_i(r_1, \ldots, z)$ are distinct as polynomials in z. Thus the probability that $y_i = f_i(r_1, \ldots, r)$ is at most $n/$size of the field.

If $S_i z_i = Rx_j$, we have

$$(1 - r_j) q_i(0) + r_j q_i(1) = y_{i-1} \neq f_{i-1}(r_1, \ldots, r_j, \ldots, r_{d_{i-1}}) = (1 - r_j) f_i(r_1, \ldots, 0, \ldots, r_{d_{i-1}}) + r_j f_i(r_1, \ldots, 1, \ldots, r_{d_i}$$

and hence $q_i(z)$ and $f_i(r_1, \ldots, z, \ldots, r_{d_{i-1}}) \ldots, z)$ are distinct as polynomials in z. Thus the probability that $y_i = f_i(r_1, \ldots, r_{d_i})$ is at most $n/$size of the field. In the last equation, r_j has the new random value chosen in this round.

∎

.

3 Graph Isomorphism

Given two graphs G_1 and G_2, the prover wants to convince the verifier that G_1 and G_2 are isomorphic.

1. The prover generates a graph H (isomorphic to G_1 and G_2) and shows it to the verifier.
2. The verifier chooses $i \in \{1,2\}$ at random and asks the verifier to show him an isomorphism between H and G_i. He accepts if the prover can do so.

If G_1 and G_2 are isomorphic, the prover always wins. If G_1 and G_2 are not isomorphic, he is caught with probability $1/2$.

4 Graph Non-Isomorphism

Given two graphs G_1 and G_2, the prover wants to convince the verifier that G_1 and G_2 are not isomorphic.

1. The verifier chooses $i \in \{1,2\}$ at random and produces an isomorphic copy H of G_i. He asks the verifier to tell him whether H is isomorphic to G_1 or to G_2.

If G_1 and G_2 are non-isomorphic, the prover always wins. If the two graphs are isomorphic, he is caught with probability $1/2$.