1 IP = PSPACE

In the previous class we defined IP, the class of languages decidable by an interactive proof system. Given a language $L$, $L \in \text{IP}$ if there is a probabilistic polynomial-time Turing machine $V$ such that:

1. $x \in L \Rightarrow \exists P, V(\pi, x) = 1$ with probability $\geq \frac{2}{3}$ (Completeness)
2. $x \notin L \Rightarrow \forall P, V(\pi, x) = 1$ with probability $\leq \frac{1}{3}$ (Soundness)

Observations:

- $\pi$: transcript (message history), questions and answers exchanged between Prover and Verifier
- The probabilities only depend on the random bits of the verifier $V$.

**Theorem 1. (Shamir 1992)**

$\text{IP} = \text{PSPACE}$

We won’t give a full proof that $\text{PSPACE} \subseteq \text{IP}$; instead, we’ll show only that $\sharp\text{3SAT} \in \text{IP}$, which will show the main ideas of the proof of the more general theorem. Recall that $(\phi, k) \in \sharp\text{3SAT}$ if $\phi$ is a 3CNF formula with $k$ satisfying assignments. We will denote by $n$ the number of variables in $\phi$, and by $m$ the number of clauses.

Last time, we saw the following protocol for solving an instance of $\sharp\text{3SAT}$ with an IP protocol.

**IP protocol for $\sharp\text{3SAT}$:**

1. Both verifier and prover have $(\phi, k)$, $\phi$ is a 3CNF formula in $n$ variables having $m$ clauses. All calculations are done in $F_p$, with prime $p > 2^n$.
2. Verifier asks prover a polynomial

   $$P_1(x) = \sum_{x_2 \in \{0,1\}} \sum_{x_3 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} \tilde{\phi}(x, x_2, \ldots, x_n)$$

3. Prover replies with at most $3m$ coefficients of some polynomial $P_1^*(x)$.
4. If $P_1^*(0) + P_1^*(1) \neq k$, then the verifier will reject.
5. For $i = 1$ to $n - 1$
   - Verifier chooses a random element $\alpha_i$ in $F_p$ and asks prover for the polynomial:
     $$P_{i+1}(x) = \sum_{x_{i+2} \in \{0,1\}} \sum_{x_{i+3} \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} \tilde{\phi}(\alpha_1, \alpha_2, \ldots, \alpha_i, x, x_{i+2}, \ldots, x_n)$$
   - Prover replies with some polynomial $P_{i+1}^*(x)$ with degree at most $3m$.
   - Verifier checks if $P_{i+1}^*(0) + P_{i+1}^*(1) = P_i^*(\alpha_i)$, if not reject.
Confirming that this protocol can be computed by a polynomial time verifier is straightforward. The degree of each $P_i$ is at most $3m$ and it is presented as a list of coefficients, so evaluating the polynomial at a single point can be done in polynomial time.

It remains to confirm the completeness and soundness of the protocol.

The protocol has completeness 1; if $(\phi, k) \in \sharp 3\text{SAT}$, then the prover can make the verifier accept with probability 1 by sending the polynomial $P_i$ at each stage of the protocol.

If $(\phi, k) /\in \sharp 3\text{SAT}$, then either $P_i^*(0) + P_i^*(1) \neq k$, or $P_i^* \neq P_i$. If $P_i^* \neq P_i$, then $P_i^*$ and $P_i$ agree on at most $d = \max(\deg(P_i^*), \deg(P_i))$ points, so $Pr_a[P_i^*(a) = P_i(a)] \leq d/|F_p|$. More generally, in round $i$, if $P_{i-1}^* \neq P_i$, then either $P_i^*(0) + P_i^*(1) \neq P_{i-1}^*(\alpha_{i-1})$, or $Pr_a[P_i^*(a) = P_{i-1}(\alpha_{i-1})] \leq d/|F_p|$. By a union bound, assuming that $P_i^*(0) + P_i^*(1) = P_{i-1}(\alpha_{i-1})$ for all $i \in [1, n-1]$, then $Pr_{a_1, \ldots, a_n}[P_n(\alpha_n) = P_n(\alpha_n)] \leq n3m/|F_p|$. Since $p > 2^n$, the protocol has soundness at least $1 - n3m/|F_p|$. Since $m$ is polynomial in $n$, the soundness of the protocol is nearly 1 for large $n$.

1.1 Easy direction of IP = PSPACE: IP $\subseteq$ PSPACE

We will show that $\forall x$, there exists an optimal prover, one that maximizes the verifier’s acceptance probability, using poly($|x|$) space.

Fixing a proof system $(P, V)$ for a language $L$ and given an input $x \in \{0, 1\}^n$, we compute using polynomial space the maximum probability with which a prover can make $V$ accept.

The probability of accepting $x$ given history:

$$P[x_1, q_1, a_1, x_2, q_2, a_2 \ldots q_{i+1}, a_{i+1}]$$

Figure 1: tree: maximum probability of acceptance

Imagine a tree as the one presented in Figure 1.1. Each node at level $i$ corresponds to a partial transcript, i.e., each node represents the state of the protocol so far with the sequence of messages.
exchanged between prover and verifier. Since $V$ can only run for poly $k$ rounds, we can affirm that the tree has polynomial depth, and knowing that the strings (messages) in the protocol have polynomial size implies that each node can have at most $2^{nc}$ children for some constant $c$. For every tree node the probability of acceptance is computed in the following way:

- A leaf node can assume only two values: 0, if the verifier rejects and 1, if the verifier accepts
- An internal node value is the weighted average over the values of its children
- Finally, the root value is the maximum probability of acceptance, i.e., the maximum probability which a prover can make the verifier accept given an input $x$

The root value can be computed in polynomial time – it is easy to see that the tree can be built in PSPACE. If this value is greater than $\frac{2}{3}$ then we know that $x \in L$, if it is less than $\frac{1}{3}$ then $x \notin L$.

2 Probabilistically Checkable Proofs - PCPs

NP:

$M$ : poly-time TM

$L \in NP$ if

\[
\forall x \in L, \exists \pi \text{ s.t. } M(x, \pi) = 1 \\
\text{if } x \notin L, \forall \pi^* M(x, \pi^*) = 0
\]

The verifier needs to read entire proof to be convinced whether $\varphi$ is satisfiable.

Definition 2. (Informal definition of PCP)

Rewrite the proof such that Verifier can probabilistically read the proof only in few constantly many locations and

- if $x \in L$, $V$ accepts with high probability
- if $x \notin L$, $V$ rejects with high probability

With this brief definition a couple of questions arise: How many coins are you allowed to toss? How many questions can you ask?

2.1 PCP verifier

Suppose,

- $L$: language
- $q, r : \mathbb{N} \to \mathbb{N}$
- $q$: number of queries
- $r$: number of tosses

$L$ has a $(r(n), q(n))$ PCP verifier if there exists a probabilistic poly-time verifier $V$ such that on input $x \in \{0,1\}^n$, given access to a proof $\pi \in \{0,1\}^*$:

- $V$ tosses at most $r(n)$ coins
- $V$ makes at most $q(n)$ non-adaptive “queries” to locations of $\pi$
• output 0, 1

Without loss of generality:

\[ |\pi| \leq q(n) \cdot 2^{r(n)} \]

1. Completeness:

   if \( x \in L \), \( \exists \pi \in \{0,1\}^* \) s.t. \( Pr[V^\pi(x) = 1] = 1 \)

2. Soundness:

   if \( x \notin L \), \( \forall \pi^* \in \{0,1\}^* \), \( Pr[V^{\pi^*}(x) = 1] \leq \frac{1}{2} \)

**Definition 3.** \( L \in \text{PCP}(r(n), q(n)) \)

If \( \exists \) PCP verifier for \( L \) and constants \( c \) and \( d \):

• number of random coins tosses \( \leq c \cdot r(n) \)
• number of queries \( \leq d \cdot q(n) \)

**Theorem 4.** PCP Theorem [ALMSS’92]

\[ \text{NP} = \text{PCP}(\log(n), 1) \]

The inclusion \( \text{PCP}(\log(n), 1) \subseteq \text{NP} \) is easy to prove. The other inclusion \( \text{NP} \subseteq \text{PCP}(\log(n), 1) \) is much harder to prove.

### 3 Hardness of Approximation

• 3-SAT
• Coloring
• \( k \)-clique
• Set cover
• Vertex cover
• TSP
• Max-cut

One of the motivations to study the PCP theorem lies in its implication to hardness results for approximating NP-hard problems. We will use the MAX-3SAT problem to explore this motivation.
\section*{3.1 MAX-3SAT}

\textbf{Definition 5.} Given $\varphi$, find an assignment satisfying $\text{val}(\varphi)m$ clauses.

- $\varphi$ is a 3CNF
- $r$ variables
- $m$ clauses
- $\text{val}(\varphi)$ = best possible fraction of clauses that can be satisfied by a single assignment
- $A$ is a $\rho$-approximation algorithm for MAX-3SAT if given $\varphi$, $A(\varphi)$ is an assignment satisfying $\rho \cdot \text{val}(\varphi)m$ clauses of $\varphi$

\textbf{Theorem 6.} \textit{(Informally - PCP Theorem)}

There exists some constant $\rho < 1$ such that if there is a poly-time algorithm which gives a $\rho$ approximation algorithm for MAX-3SAT, then $P = NP$. 