**Definition** (Probabilistically Checkable Proofs (PCP)). A Turing machine with direct access is a Turing machine with:

- a special state, called the reading state,
- a reading oracle,
- two special working tapes, called the direct access tape, and the address tape.

The machine never reads directly the content of the direct access tape (in the sense that the normal transitions of the machine are independent of the content of the direct access tape). This tape is only accessed via the reading oracle in the following way: when the machine goes in the reading state, the content of the address tape is interpreted as the binary representation of a position $i$ of the direct access tape. The reading oracle then provides in one step, the symbol in position $i$ of the direct access tape. (You can assume this symbol is stored in the control state, or in a special output tape of the reading oracle.)

A $\text{PCP}(R(n), Q(n), T(n))$-verifier is a probabilistic Turing machine with direct access to a tape called the proof tape over alphabet $\{0, 1\}^*$. On input $x$ of size $n$ and proof tape content $\pi$, the machine uses $R(n)$ random bits and works in the following three phases:

1. It first computes $Q(n)$ positions $p_1, \ldots, p_{Q(n)}$ (in binary) in time polynomial in $n$, and with no calls to the reading oracle (i.e. these positions are only a function of $x$ and the random tape content).

2. Then it makes $Q(n)$ calls to the reading oracle, to retrieve the symbols of the proof tape $\pi$ in positions $p_1, \ldots, p_{Q(n)}$.

3. Finally, it computes a boolean value (either accept or reject) in time $T(n)$ and with no calls to the reading oracle (i.e. the answer computed in this phase is only a function of $x$, the random tape content, and the symbols $\pi[p_1], \ldots, \pi[p_{Q(n)}]$).

The class $\text{PCP}(R(n), Q(n), T(n))$ is the set of languages $L$ such that there exists a $\text{PCP}(R(n), Q(n), T(n))$-verifier $V$ such that:

- if $x \in L$, there exists a proof $\pi \in \{0, 1\}^*$ such that $Pr_r[V(x, \pi, r) \text{ rejects } ] = 0$;
- if $x \notin L$, then for all $\pi \in \{0, 1\}^*$ $Pr_r[V(x, \pi, r) \text{ accepts } ] \leq 1/2$.

Where the probability is computed over all random tape contents $r$ of size $R(n)$.

**Exercise 1** (PCP witnessing). Let $\text{PCP}^\prime(k_1 \cdot \log n, Q(n), T(n))$ be defined as $\text{PCP}(k_1 \cdot \log n, Q(n), T(n))$ except that only proofs $\pi$ of size $n^{k_1} \cdot Q(n)$ are considered, and addresses computed by the verifier have $\log(n^{k_1} \cdot Q(n))$ bits. Prove that $\text{PCP}(k_1 \cdot \log n, Q(n), T(n)) = \text{PCP}^\prime(k_1 \cdot \log n, Q(n), T(n))$. 

1
Exercise 2 (PCP and non-deterministic classes). Prove that, with $R(n) = \Omega(\log n)$, we have $\text{PCP}(R(n), Q(n), T(n)) \subseteq \text{NTIME}(2^{O(R(n))} \cdot Q(n) \cdot (T(n) + \text{poly}(n)))$.

Exercise 3 (Known classes). What are these versions of PCP equal to (in term of known complexity classes)?

$$\bigcup_{c \in \mathbb{N}, T(n) \text{ a polynomial}} \text{PCP}(0, c \cdot \log n, T(n))$$

$$\bigcup_{Q(n), T(n) \text{ polynomials}} \text{PCP}(0, Q(n), T(n))$$

$$\bigcup_{R(n), T(n) \text{ polynomials}} \text{PCP}(R(n), 0, T(n))$$

Exercise 4 (Graph non-isomorphism). Show that $\text{ISO} \in \text{PCP}(p(n), 1, c)$ for some polynomial $p$ and constant $c$.

Exercise 5 (Multi-prover protocol).

Definition. Let $P_1, \ldots, P_k$ be infinitely powerful machines whose output is polynomially bounded. Let $V$ be a probabilistic polynomial-time machine. $V$ is called the verifier, and $P_1, \ldots, P_k$ are called the provers.

A round of a multi-prover interactive protocol on input $x$ consists of an exchange of messages (i.e. words over a given alphabet) between the verifier and the provers, and works as follows:

- The verifier $V$ is executed on an input consisting of $x$, the history of all previous messages exchanged with all provers (both sent and received messages), and a random tape content of size polynomial in $|x|$. The output of the verifier is computed in time polynomial in $|x|$, and consists of messages to some or all of the provers.

- Each message $q_i$ sent from the verifier to prover $P_i$ is followed by an answer $a_i$, of size polynomial in $|x|$, sent from the prover $P_i$ to the verifier. The answer $a_i$ is computed by $P_i$ on input consisting of $x$ and the history of all messages previously exchanged between the verifier and the prover $P_i$ (and only $P_i$).

- Alternatively the verifier may decide not to produce messages, and terminates the protocol by either accepting or rejecting, based on the input $x$ and the history of all previous messages exchanged with all provers.

You can view the protocol as executed by the verifier sharing communication tapes with each $P_i$, where different provers $P_i$ and $P_j$ (for $i \neq j$) have no tapes they can both access, besides the input tape. In a round the verifier stores each message $q_i$ to prover $P_i$ on the $i$-th communication tape, shared between the prover and $P_i$. The answer of $P_i$ is put on tape $i$ as well. The verifier has access to the input and all communication tapes, while each prover $P_i$ has access only to the input and tape $i$.

$P_1, \ldots, P_k$ and $V$ form a multi-prover interactive protocol for a language $L$ if the execution of the protocol between $V$ and $P_1, \ldots, P_k$ terminates after a polynomial number of rounds (in the size of the input $x$) and:
• if $x \in L$, then $\Pr[(V,P_1,\ldots,P_k) \text{ accepts } x] > 1 - 2^{-n}$;

• if $x \notin L$, then for all provers $P'_1,\ldots,P'_k$, $\Pr[(V,P'_1,\ldots,P'_k) \text{ accepts } x] < 2^{-n}$;

where the probability is computed over all possible random choices of $V$.

In this case, we denote $L \in \text{MIP}_k$. The number of provers $k$ need not be fixed and may be a polynomial in the size of the input $x$. We say that $L \in \text{MIP}$ if $L \in \text{MIP}_{p(n)}$ for some polynomial $p$. Clearly $\text{MIP}_1 = \text{IP} = \text{PSPACE}$ (as you will see in the lecture), but allowing more provers makes the interactive protocol model potentially more powerful.

1. Let $M$ be a probabilistic polynomial-time Turing machine with access to a function oracle. A language $L$ is accepted by $M$ iff:

   • if $x \in L$, then there exists an oracle $O$ s.t. $M^O$ accepts $x$ with probability greater than $1 - 2^{-n}$;

   • if $x \notin L$, then for any oracle $O'$, $M^{O'}$ accepts $x$ with probability lower than $2^{-n}$.

Show that $L \in \text{MIP}$ if and only if $L$ is accepted by a probabilistic polynomial time oracle machine.

2. Show that $\text{MIP} = \text{MIP}_2$ (assuming we can use error-reduction).

Exercise 6 (PCP, MIP and NEXPTIME). Prove that

$$\bigcup_{R(n),Q(n),T(n) \text{ polynomials}} \text{PCP}(R(n),Q(n),T(n)) \subseteq \text{MIP} \subseteq \text{NEXPTIME}$$

(Hint. It is possible to prove (but you are not required to) that, as with IP, one can equivalently use perfect completeness in the definition of MIP. That is, in the case $x \in L$, we require that the protocol accepts with probability 1, rather than at least $1 - 2^{-q(n)}$. In this exercise use the definition of MIP with perfect completeness, and the corresponding notion of probabilistic oracle machine.)

Remark. Indeed MIP and this version of PCP coincide with NEXPTIME, but you are not required to prove the opposite inclusions.