Complexité avancée - TD 12

Simon Halfon

December 6, 2016

Exercise 1: Some old friends

1. $\mathsf{MA} \subseteq \mathsf{PP}$
2. Polynomial Identity Testing

Exercise 2: Another unlikely event around $\mathsf{P/poly}$
Show that if $\mathsf{PSPACE} \subseteq \mathsf{P/poly}$ then $\mathsf{PSPACE} = \mathsf{MA}$.

Definition 1 (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 $\mathsf{PCP}(R(n), Q(n), T(n))$-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 π, the machine uses $R(n)$ random bits and works in the following three phases:*

1. *It first computes $Q(n)$ positions $p_1, \dots p_{Q(n)}$ (in binary) in polynomial time 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 π in positions $p_1, \dots p_{Q(n)}$.*
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 $\pi[p_1], \dots \pi[p_{Q(n)}]$).*

The class $\mathsf{PCP}(R(n), Q(n), T(n))$ is the set of languages L such that there exists a $\mathsf{PCP}(R(n), Q(n), T(n))$-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 3: PCP and non-deterministic classes

Prove that, with $R(n) = \Omega(log\ n)$, we have $\mathsf{PCP}(R(n), Q(n), T(n)) \subseteq \mathbf{NTIME}(2^{O(R(n))} \cdot Q(n) \cdot T(n))$.

Exercise 4: PCP witnesses

Let $\mathsf{PCP}'(k_1 \cdot log\ n, Q(n), T(n))$ be defined as $\mathsf{PCP}(k_1 \cdot log\ n, Q(n), T(n))$ except that only proofs π of size $n^{k_1}Q(n)$ are considered, and addresses computed by the verifier have $log(n^{k_1}Q(n))$ bits. Prove that $\mathsf{PCP}(k_1 \cdot log\ n, Q(n), T(n)) = \mathsf{PCP}'(k_1 \cdot log\ n, Q(n), T(n))$.

Exercise 5: Known classes

Prove the following statements:

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

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

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

Exercise 6: Graph non-ismorphism

Show that $\overline{\mathbf{ISO}} \in \mathsf{PCP}(p(n), 1, c)$ for some polynomial p and constant c.

Exercise 7: A general note on self-reducibility

Define a language L to be *downward-self-reducible* if there is a polynomial-time Turing Machine R such that for any x of length n, $R^{L_{n-1}}(x) = L(x)$, where L_k denotes an oracle that decides L on input of size at most k. Prove that if L is such a language, then $L \in \mathsf{PSPACE}$.

Exercise 8: PCP, MIP and NEXPTIME

Prove that

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

(**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.