Computational Complexity

Homework Sheet 2

Hand in before February 20, 17:00

Preferably by email to `J.Czajkowski@uva.nl`

Exercise 1 (2pt)**.** Show that coNP $\subseteq$ EXP.

Exercise 2 (4pt)**.** Consider the following problem Exactly-2-in-5-SAT:

> **Instance:** A propositional formula φ in 5CNF—that is, a formula of the form $\varphi = c_1 \wedge \cdots \wedge c_m$, where each c_i is of the form $c_i = l_{i,1} \vee l_{i,2} \vee l_{i,3} \vee l_{i,4} \vee l_{i,5}$, where $l_{i,1}, l_{i,2}, l_{i,3}, l_{i,4}, l_{i,5}$ are propositional literals.
>
> **Question:** Is there a truth assignment α to the variables occurring in φ that sets exactly 2 literals in each clause c_i to true?

Note: we allow clauses to contain complementary literals—e.g., $(x_1 \vee \neg x_1 \vee x_2 \vee x_3 \vee x_4)$ is a valid clause.

Prove that Exactly-2-in-5-SAT is NP-complete—that is, prove that is in NP and that it is NP-hard. To show NP-hardness, you may give a reduction from any known NP-complete problem.

- *Hint:* reduce from a suitable variant of 3SAT.[1]

Exercise 3 (2pt)**.** Let $A \subseteq \{0,1\}^*$ be an NP-complete language. Let p be a polynomial and let $\mathbb{M}_A$ be a polynomial-time Turing machine such that, for all $x \in \{0,1\}^*$:

$$x \in A \text{ if and only if there exists some } u \in \{0,1\}^{p(|x|)} \text{ such that } \mathbb{M}_A(x,u) = 1.$$

(a) Define the set $B = \{ \langle x, z \rangle \mid \text{there exists } z' \in \{0,1\}^* \text{ such that } |zz'| \leq p(|x|) \text{ and } \mathbb{M}_A(x, zz') = 1 \}$. Prove that B is in NP.

(b) Suppose that we have access to A as an oracle. Basically this means that we have a subroutine that, given a string y, tells in a single step whether $y \in A$. (See Definition 3.4 of Arora & Barak, 2009.) Construct a polynomial-time Turing machine $\mathbb{M}_{\text{search}}$ (with access to an A-oracle) that, given $x \in \{0,1\}^*$, if $x \in A$ outputs a string u such that $\mathbb{M}_A(x,u) = 1$ and if $x \notin A$ outputs 0. Use **(a)**.

- *Hint:* use the fact that A is NP-complete.

Exercise 4 (2pt)**.** Let $A \subseteq \{0,1\}^*$ be a language. When a Turing machine $\mathbb{M}$ has access to an A-oracle, we write $\mathbb{M}^A$. We say that A is *auto-reducible* if there is a polynomial-time Turing machine $\mathbb{M}^A$ with oracle access to A such that for all $x \in \{0,1\}^*$:

$$x \in A \text{ if and only if } \mathbb{M}^A(x) = 1,$$

with the special requirement that on input x the Turing machine $\mathbb{M}^A$ is not allowed to query the oracle A for x.

Suppose that A is NP-complete. Prove that A is auto-reducible. Use **Exercise 3**.

[1] See, e.g., `https://en.wikipedia.org/wiki/Boolean_satisfiability_problem`.