# Computational Complexity

Homework Sheet 1

Hand in via Canvas before February 17 at 17:00

**Exercise 1** (3pt)**.** For the following pairs of functions and relations (i.e., $O$, $o$, $\omega$, $\Omega$, $\Theta$), prove for the two relations at each pair whether they hold or do not hold.

**1.** $f(n) = 4^{(\log n)^2}$ $\quad g(n) = n^{\log n}$ $\quad$ **(a)** $g \in \Omega(f)$? $\quad$ **(b)** $f \in \Theta(g)$?

**2.** $f(n) = 2n$ $\quad g(n) = (\log n)^2$ $\quad$ **(a)** $g \in O(f)$? $\quad$ **(b)** $f \in \omega(g)$?

**3.** $f(n) = n^3$ $\quad g(n) = n^3 - 100n^2$ $\quad$ **(a)** $g \in o(f)$? $\quad$ **(b)** $f \in \Theta(g)$?

**Exercise 2** (2pt)**.** Let $\mathbb{M} = (\Gamma, Q, \delta)$ be a 2-tape Turing machine that computes some function $f : \{0,1\}^* \to \{0,1\}$ in time $t(n)$, for some function $t$. Give a 3-tape Turing machine $\mathbb{M}' = (\Gamma', Q', \delta')$ that computes the same function $f$ in time $O(n) + t(n)/2$.

- Use the conventions and notation from the book (see Section 1.2 of Arora & Barak, 2009)—for example, the first tape is the input tape and is read-only.
- No need to specify $\mathbb{M}'$ in full detail; explain how $\mathbb{M}'$ is constructed.
- (*Hint:* create the alphabet $\Gamma'$ in such a way that sequences $(\sigma_1, \dots, \sigma_k)$ of symbols from $\Gamma$ (of up to a certain length $k$) are encoded by a single symbol $\sigma' \in \Gamma'$.)

**Exercise 3** (2pt)**.** Show that coNP $\subseteq$ EXP.

**Exercise 4** (3pt)**.** Consider the following problem Exactly-2-in-5-SAT:

> **Instance:** A propositional formula $\varphi$ 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 $\alpha$ to the variables occurring in $\varphi$ 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]

**Remark 1.** Answers will be graded on two criteria: they should (1) be correct and intelligent, and also (2) concise and to the point.

**Remark 2.** If you find a solution to one of the exercises in a paper or book, you can use this to inform your solution. Make sure that you write down the solution in your own words, conveying that you understand what is going on.

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