Computational Complexity 2018–2019

Homework Sheet 1

Hand in before February 13, 17:00
Preferably by email to `J.Czajkowski@uva.nl`

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

1.	$f(n) = 4^{(\log n)^2}$	$g(n) = n^{\log n}$	**(a)** $g \in \Omega(f)$?	**(b)** $f \in \Theta(g)$?
2.	$f(n) = 2n$	$g(n) = (\log n)^2$	**(a)** $g \in O(f)$?	**(b)** $f \in \omega(g)$?
3.	$f(n) = n^3$	$g(n) = n^3 - 100n^2$	**(a)** $g \in o(f)$?	**(b)** $f \in \Theta(g)$?

Exercise 2 (1pt)**.** Give a function $g(n)$ such that $g \in o(2n)$ and $g \in \omega(1)$—and prove that this is the case.

Exercise 3 (3pt)**.** 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 Γ' in such a way that sequences $(\sigma_1, \ldots, \sigma_k)$ of symbols from Γ (of a certain length k) are encoded by a single symbol $\sigma' \in \Gamma'$.)

Exercise 4 (3pt)**.** In this exercise, we are going to compare algorithms with different running times. Suppose we have two algorithms A, B that compute the same function $f : \{0,1\}^* \to \{0,1\}$. Moreover, suppose that A runs in time $a(n) = 1.01^n$ and that B runs in time $b(n) = 100,000 \cdot n^2$, where n denotes the size of the input x for which $f(x)$ is to be computed.

(a) Which of these running times (if any) is better in terms of asymptotic growth? That is, is $a \in o(b)$, $b \in o(a)$, or $a \in \Theta(b)$?

(b) Is there a concrete input size n_0 so that one of A and B outperforms the other for all inputs of size $\geq n_0$ (in terms of running time)? If so, what is this n_0 and which algorithm performs better on inputs of size $\geq n_0$? Can an input of size n_0 be stored on a floppy disk?

(c) Answer the above two questions—that is, (a) and (b)—for two algorithms C, D computing the same function $g : \{0,1\}^* \to \{0,1\}$, such that C runs in time $c(n) = (1 + \frac{1}{10^9})^n$ and D runs in time $d(n) = n^{25}$.

(d) If you were to use one of the algorithms C and D on your computer, which one would you choose, and why?

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.