Complexity Theory

Homework Sheet 1

Hand in before the lecture of Tuesday 14 Feb.

Preferably by email to bannink@cwi.nl

7 February 2017

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

1. $f(n) = n^{\log n}$ $\quad g(n) = 2^{(\log n)^3}$ $\quad g \in \Omega(f)$? $\quad f \in \Theta(g)$?
2. $f(n) = \log n$ $\quad g(n) = n$ $\quad g \in \omega(f)$? $\quad f \in \mathcal{O}(g)$?
3. $f(n) = n^5$ $\quad g(n) = 100n^5$ $\quad f \in o(g)$? $\quad g \in \Theta(f)$?

Exercise 2. By the fundamental theorem of arithmetic, any natural number x can be uniquely written as a product

$$x = p_1 \cdot p_2 \cdots p_k$$

with $p_1, \ldots, p_k$ prime numbers such that $p_i \leq p_j$ if $i < j$. This yields a function

$$f : \mathbb{N} \to \{0,1\}^* : x \mapsto \langle p_1, p_2, \ldots, p_k \rangle,$$

which maps a natural number x to a binary encoding of the prime factorization of x. See section 0.1 of the book for more explanation of the notation used here.

(a) Using the fact that there is a polynomial-time algorithm for testing primality,[1] show that deciding whether $z = f(x)$, when given x and z as input, can be done in polynomial time.

(b) Show that the set

$$\text{FACTORIZATION} = \{\langle x, i \rangle \mid \text{the } i\text{-th bit of } f(x) \text{ is } 1\}$$

is in **NP**. Here $\langle x, i \rangle$ is a binary encoding of the integers x, i.

(c) Show: if FACTORIZATION is **NP**-complete, then **NP** = **coNP**.
Hint: First show that FACTORIZATION is in **coNP**.

(d) Define

$$\text{COMPOSITE} = \{\langle x \rangle \mid x \in \mathbb{N} \text{ has at least two prime factors}\}.$$

Show: COMPOSITE is **NP**-complete if and only if **P** = **NP**.

[1] Which has been an open problem for a very long time, but solved in 2002 by Agrawal, Kayal and Saxena, see `http://en.wikipedia.org/wiki/AKS_primality_test`.

Exercise 3. A given function $f : \{0,1\}^* \to \{0,1\}^*$ is called *honest* if there is some real constant $c \geq 0$ such that $|f(x)|^c > |x|$ for all x, where $|x|$ is the length of the bitstring x. We say the *inverse* of a function f is polynomial-time computable if there is a Turing machine that always halts in polynomial time and that given an input $y \in \{0,1\}^*$ computes and outputs an $x \in \{0,1\}^*$ such that $f(x) = y$ or outputs NONE if no such x exists.

(a) Show that if $\mathbf{P} = \mathbf{NP}$ then for every honest function that is polynomial-time computable, the inverse is also polynomial-time computable.

(b) Prove the converse of the previous statement, i.e., show that if every honest, polynomial-time computable function has a polynomial-time computable inverse, then $\mathbf{P} = \mathbf{NP}$. Hint: Which function would you have to invert to find the witness you are searching for?

Together, the result is sometimes known as the cryptographic theorem:

Theorem 1. $\mathbf{P} = \mathbf{NP}$ *if and only if every honest, polynomial-time computable function has a polynomial-time computable inverse.*

Exercise 4. Show that $\mathbf{NP} \subseteq \mathbf{EXP}$.

Hint. Answers will be graded with two criteria: they should be correct and intelligent, but also concise and to the point.