# Introduction to computability

## Final exam

8 January 2013

*Closed-book. Duration: 3h30*

*Please answer each question on a separate sheet with your name and section.* ***Motivate all your answers and give sufficient details.***

1. a) Is the set of well formed arithmetic expressions denumerable? For example $3*(2+4)$ is well formed and $3+*5$ as well as $(2+3)*3)$ are not.

   b) Consider the set of one argument predicates over the natural numbers and show that some of these predicates must be undecidable.

2. a) Give a DFA that accepts the language $L_1$ containing all the words on the alphabet $\{a, b\}$ that contain an odd number of letters $a$ and an even number of letters $b$. In other words,

   $$L_1 = \{w \mid w \in \{a, b\}^*,\ N_a(w) = 1 \mod 2 \text{ and } N_b(w) = 0 \mod 2\}$$

   where $N_\sigma(w)$ is the number of letters $\sigma$ contained in the word $w$.

   b) Give a DFA that accepts the language $L_2$ containing all the words on the alphabet $\{a, b\}$ that contain neither $aa$ nor $bb$. In other words,

   $$L_2 = \{w \mid w \in \{a, b\}^*,\ aa \notin \mathit{Fact}(w) \text{ and } bb \notin \mathit{Fact}(w)\}$$

   c) Give a DFA that accepts $L_1 \cup L_2$.

3. a) Show that the language $L = \{a^i b^j c^k \mid k \neq i \cdot j\}$ is not regular.

   b) Is it correct that every regular language is also a context-free language? And vice versa, is every context-free language also a regular language?

4. a) Show that the intersection of two context-free languages is not necessarily context-free. Use this to deduce that the complement of a context-free language is not necessarily context-free. Give a sufficient criterion for the intersection of two context-free languages to be context-free.

b) Given a context-free grammar $G$, give an algorithm for checking if $L(G) = \varnothing$.

5. a) For a Turing machine M, define the notions of *configuration*, *derivation*, *execution*, *accepted language* and *decided language*.

   b) Give a Turing machine that computes the function $n \mapsto 2n$. Consider that numbers are encoded using a unary alphabet, so that $n$ is represented by $n$ repetitions of the single letter of the alphabet. If $q_0$ is an initial state and $q_f$ a final state of the Turing machine, you must for instance have

$$q_0 \,.\, \underline{1}11\# \vdash^* q_f \,.\, 111111\underline{\#}$$

6. a) Show that the predicate $twins(x, y)$, that is true if and only if $x$ and $y$ are prime numbers and $y = x + 2$, is primitive recursive. For example, $twins(11, 13)$ is true.
   Remark: You can use any primitive recursive functions and predicates seen in the lectures.

   b) Define *$\mu$-recursive predicates*.

   c) Is the predicate $twins(x, y)$ a $\mu$-recursive predicate?

7. a) Show that there exists at least one undecidable language.

   b) Given two Turing machines $M_1$ and $M_2$, show that the problem of determining if there exists a word $w$ such that $M_1$ and $M_2$ both stop on $w$ is undecidable.

8. a) Define *polynomial transformations*.

   b) Define the *Travelling Salesman Problem* (TS) and the *Hamiltonian Circuit Problem* (HC) and give a polynomial transformation from HC to TS. What can you conclude from this transformation with respect to the membership in P of these two problems?

   c) Give an explicit statement of *Cook's theorem*. Explain in a few sentences how it can be proved.

   d) Give a deterministic algorithm to solve the SAT problem. What is its complexity?