Free University of Bozen-Bolzano – Faculty of Computer Science
Master of Science in Computer Science
Theory of Computing – A.Y. 2015/2016
Final exam – 20/6/2016 – Part 2

Time: 90 minutes

This is a closed book exam: the only resources allowed are blank paper, pens, and your head. Explain your reasoning. Write clearly, in the sense of logic, language and legibility. The clarity of your explanations affects your grade. Write your name and ID on every solution sheet. Good luck!

Problem 2.1 [2+2+2 points]

(a) Sketch the proof that, for languages X and Y, if $X <_{poly} Y$ and $Y \in P$, then also $X \in P$.
(b) Provide a formal justification of the following statement: "2SAT is *simpler* than 3SAT."
(c) Justify why a language in PSPACE is recognized by a TM whose running time is bounded by an exponential in the length of the input.

Problem 2.2 [6 points] For a string $w = a_1 \cdots a_n$, where each $a_i \in \Sigma$, and for $k \in \{1, \ldots, n\}$, let $w^{[-k]}$ denote the string obtained from w by removing every k-th character, starting from the first one, i.e., $w^{-[k]} = a_2 a_3 \cdots a_{k-1} a_{k+1} \cdots a_{2k-1} a_{2k+1} \cdots a_n$.
Consider a language L over an alphabet Σ, and let:

$$L_1 = \{ \ w \mid \text{ there exists a } k \in \{1, \ldots, |w|\} \text{ such that } w^{-[k]} \in L \ \}.$$

Show that, if L is in NP, then also L_1 is in NP.

[*Hint:* Use a non-deterministic TM M for L, and show how it can be used to construct a non-deterministic TM for L_1.]

Problem 2.3 [2+4 points] Consider the proof of Cook's theorem that CSAT is NP-hard.

(a) Describe how in that proof the computation of a non-deterministic TM with running time $p(n)$, where $p(n)$ is a polynomial in n, is represented *using propositional variables*.
(b) Consider then *only* the conditions holding between such propositional variables that *depend* on the actual transitions of the TM, and provide the CNF-formulas that encode such conditions. How many clauses of which length are necessary to encode such conditions?

Problem 2.4 [1+2+3 points]

(a) Define the problem stConn (source-to-target connectivity).
(b) Define the complexity classes L and NL, and the notion of NL-completeness. Which kinds of reductions are required?
(c) Provide a sketch of the proof that stConn is NL-hard.

Problem 2.5 [2+2+2 points]

(a) Provide the general definition of a tiling problem, both in terms of colored tiles, and in terms of adjacency relations.
(b) Given a generic tiling problem defined in terms of colored tiles, show the corresponding variant defined in terms of adjacency relations. Argue why, given a tiling problem T defined in terms of adjacency relations, in general it does not correspond *directly* to a variant defined in terms of colored tiles.
(c) Define formally the corridor tiling problem, and argue *briefly* (i.e., in a few sentences) how it can be solved in polynomial space, and why it is PSPACE-hard.