Free University of Bozen-Bolzano – Faculty of Computer Science
Master of Science in Computer Science
Theory of Computing – A.Y. 2013/2014
Final exam – 28/1/2014 – 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] Decide which of the following statements is TRUE and which is FALSE, or believed to be so, under certain complexity-theoretic assumptions, which you should state. You must give an explanation of your answer to receive full credit.

(a) $\mathrm{P}^{\mathrm{NP}} = \mathrm{NP}^{\mathrm{P}}$.

(b) Let L_1 and L_2 be languages. If $L_1 <_{poly} L_2$ and $L_2 \in \mathrm{NPS{\small PACE}}$, then $L_1 \in \mathrm{PS{\small PACE}}$.

(c) Connectivity in undirected graphs is *simpler* than connectivity in directed graphs.

Problem 2.2 [6 points] Let L be a language over an alphabet Σ, and let:

$$L_1 = \{\ w \mid w = x_0 \cdot y_1 \cdot x_1 \cdot y_2 \cdot x_2,\ \text{with } y_1 \cdot y_2 \in L \text{ and } x_0, x_1, x_2 \in \Sigma^*\ \}$$

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

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 *do not 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 [3+2+1 points] Let $\Sigma = \{0, 1\}$,

- let $f_1 : \Sigma^* \to \Sigma^*$ be a function computable by a TM A_1 in space $s_1(n)$,
- let $f_2 : \Sigma^* \times \Sigma^* \to \Sigma^*$ be a function computable by a TM A_2 in space $s_2(n)$,

and consider the computation of the function $f(x) = f_2(x, f_1(x))$.

(a) Describe the technique of *emulative composition* for "composing" the computations of A_1 and A_2 in order to obtain a computation of $f(x)$.

(b) Elaborate on the space bound resulting from this technique.

(c) What is the impact on computations that require logarithmic space?

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 problems 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 $\mathrm{PS{\small PACE}}$-hard.