Free University of Bozen-Bolzano – Faculty of Computer Science
Master of Science in Computer Science
Theory of Computing – A.Y. 2006/2007
Final Exam – 12/6/2007 – Part 1

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 1.1 [4.5 points] Decide, if possible, which of the following statements is TRUE and which is FALSE. You must give a brief explanation of your answer to receive full credit.

(a) For all languages L, it holds that $L^* = (L \cup \varepsilon)^*$.

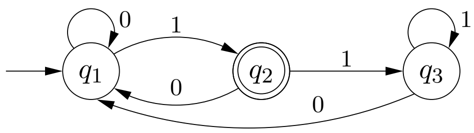
(b) If L_1 is regular and L_2 is context-free, then $L_1 \cap L_2$ is regular.

(c) For all languages L_1 and L_2, if $L_1 \subseteq L_2$ and $L_2^* \subseteq L_1^*$, then $L_1 = L_2$.

Problem 1.2 [1.5 points] Give a regular expression that represents the set of strings over the alphabet $\Sigma = \{a, b, c\}$ that contain the substring aa starting at an odd position and the substring bb starting at an even position.

Problem 1.3 [6 points] Describe an algorithm to construct from a DFA A a regular expression E such that $\mathcal{L}(E) = \mathcal{L}(A)$.

Illustrate the steps of the algorithm you have chosen to construct E, on the example of the following DFA A over $\{0, 1\}$:



Problem 1.4 [6 points] Describe an algorithm to solve the following problem: Given two regular expressions E_1 and E_2, respectively with associated languages $\mathcal{L}(E_1)$ and $\mathcal{L}(E_2)$ over the alphabet Σ, construct a regular expression E such that $\mathcal{L}(E) = \mathcal{L}(E_1) \cap \mathcal{L}(E_2)$. (Notice that set intersection is not an operator that can be used in regular expressions.) In describing the algorithm, you can make use of algorithms that have been discussed in class, without the need of detailing the various steps of these algorithms.

Illustrate the algorithm on the example of the regular expressions $E_1 = 0^*$ and $E_2 = 0$. Notice that E_1 and E_2 are sufficiently simple to allow you to calculate on them the results of the algorithms discussed in class, without the need of detailing the various steps of these algorithms.

Problem 1.5 [6 points] Show that the language $L = \{ a^n b^m \mid n \leq m \leq 2n \}$ is not regular.

[*Hint*: Use the pumping lemma for regular languages.]

Problem 1.6 [6 points] Consider a context-free grammar $G = (V_N, V_T, P, S)$ in which every production in P is of the form $A \longrightarrow xB$ or $A \longrightarrow x$, with $A, B \in V_N$ and $x \in V_T^*$. Show that the language generated by G is regular. Does this still hold if we allow in G also productions of the form $A \longrightarrow Bx$? Argue convincingly.