Free University of Bozen-Bolzano – Faculty of Computer Science
Bachelor in Computer Science and Engineering
Formal Languages and Compilers– A.Y. 2012/2013 Mid-Term Exam – 9/11/2012
Dr. Alessandro Artale – *Time: 120 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 [6 points] Decide 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_1 and L_2, if L_1 is regular and L_2 is non-regular then $L_1 \cap \overline{L_2}$ is non-regular.

(b) For all languages L, we have that $L^* \cdot \varepsilon = (L \cdot \varepsilon)^*$.

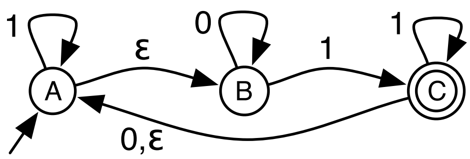
(c) For all languages L_1, L_2, and L_3, if $L_1 \cdot L_3 \subseteq L_2 \cdot L_3$, then $L_1 \subseteq L_2$.

Problem 2 [8 points]

(a) Construct a regular expression $R.E.$ that generates the language over the alphabet $\{1, 2, 3\}$ constituted by all strings in which each 1 is (not necessarily immediately) preceded by some 2, and this 2 comes after any other 1.
E.g., $\varepsilon \in \mathcal{L}(E)$, $2332 \in \mathcal{L}(E)$, $2133223132 \in \mathcal{L}(E)$, $213312 \notin \mathcal{L}(E)$, $1332133 \notin \mathcal{L}(E)$.

(b) Construct an ε-NFA A that accepts the language specified above.

Problem 3 [6 points] Consider the following ε-NFA A_ε over $\{0, 1\}$:



(a) Construct an NFA A_n such that $\mathcal{L}(A_n) = \mathcal{L}(A_\varepsilon)$. The algorithm you have followed to construct A_n should become evident in your construction. In particular, construct the ε-closure of each state.

(b) Show all sequences of transitions of A_n that lead to acceptance of the string 0101.

Problem 4 [8 points]

(a) Apply the steps that are necessary to Clean-up a context free grammar (Eliminate: ε-productions, unit productions, non-generating and non-reachable symbols) to the context free grammar $G = (\{S, A, B, C, D\}, \{a, b\}, P, S)$, where P consists of the following productions:

$$\begin{array}{lll} S & \longrightarrow & B \mid ABa \mid BaDb \\ A & \longrightarrow & Ab \mid AC \\ B & \longrightarrow & BaD \mid \varepsilon \end{array} \qquad \begin{array}{lll} C & \longrightarrow & CB \mid CA \mid bA \\ D & \longrightarrow & DA \mid B \mid aDD \mid ABC \end{array}$$

Problem 5 [8 points]

(a) Write a grammar for Boolean Expressions between **identifiers** using the operators **or**, **and**, **not** and parentheses. Be sure to give **or** the lower precedence, then **and**, and finally **not** with the highest precedence. Allow for repeated **not**, as in the Boolean Expression **not not id**. Be sure that the grammar is not ambiguous.

(b) Show the parse tree for the strings:

(i) id_1 **and not** id_2 **or** id_3;

(ii) **not**(id_1 **or** (id_2 **and** id_3)).

(c) Show the right-most derivation for the string: **not** id_1 **or** id_2 **and** id_3.

(d) Rewrite the grammar in such a way that the operators **or**, **and** have the same precedence.