COMP3630/6363 Assignment 1, 2008

Due date: end of lecture, Wednesday April 1

Hand-written answers are acceptable if written neatly. **Write on each sheet of paper that you use the page number, your student ID, and name, please.** Either hand your answers to me on paper, or send your answers to baueran@rsise.anu.edu.au as PDF, Postscript, or clear text. No Word, HTML, etc.

Correct answers might be given less than full credit if they are unnecessarily complicated or poorly written.

Question 1

Write a regular expression for each of the following languages.

(a) Non-empty strings over $\{a, b\}$ that start and finish with the same letter.

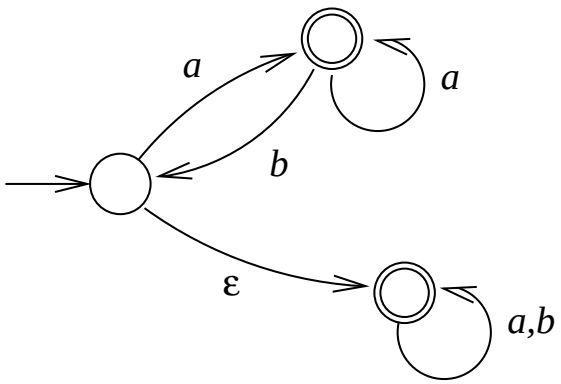
(b) Strings over $\{a, b, c\}$ such that no three symbols in a row are the same.

Question 2

Construct a DFA for the language of Question 1 (b).

Question 3

Using the vertex-elimination method, find a regular expression for the language accepted by the following ε-NFA. Show the situation after each elimination. (There is no need to simplify the answer, just give whatever expression the algorithm provides.)



Question 4

Consider the following language L: all strings over $\{a, b, c\}$ such that no more than half the symbols are the same. That is, for a string of length n, at most $n/2$ of the symbols are a, at most $n/2$ of the symbols are b, and at most $n/2$ of the symbols are c.

(a) Prove that L is not regular using the pumping lemma.

(b) Prove that L is not regular using the α-β lemma (see course web page for handout)

Question 5

Consider the following decision problem. Given are n regular languages, $L_1, \ldots, L_n$ (in the form of regular expressions or NFAs).

Does $L_1 \cap ... \cap L_n \neq \emptyset$ hold?

Answer the following:

- How would an algorithm look like that determines the answer to this question?
- What would be the "space requirements" of this algorithm in terms of automata sizes, expressions, etc. with respect to the size of the original input (i.e., the length of the expressions $l_1, \ldots, l_n$ or size of the NFAs, $\mathcal{A}_1, \ldots, \mathcal{A}_n$)?
- Do you get a more efficient algorithm if $L_1, \ldots, L_n$ are given as DFAs instead?

Justify your answers, but keep the algorithm somewhat abstract, i.e., do not actually write a program or compute an example. Just mention the crucial steps and how much space/time they might require with respect to the inputs of each such step! If you mention the "right steps", the answers should be rather short and concise.

Question 6 (COMP6363 and PhDs only)

Let L be a regular language over $\{a, b, c\}$. Define the language L' over $\{a, b\}$ to be language formed by deleting all strings from L that have no c, then deleting from the remaining strings the first c and everything following it. Formally:

$$L' = \left\{ x \in \{a, b\}^* \;\middle|\; xcy \in L \text{ for some } y \in \{a, b, c\}^* \right\}.$$

Either prove L' is regular, or give an example to show that L' might not be regular.