Each of the problems is worth 10 points. Write your name, student ID, and your TA's name on the solution sheet.

Please write your solutions clearly and concisely. If you do not explain your answer you will be given no credit. You are encouraged to collaborate on the homework, but you *must* write your own solutions and list your collaborators on your solution sheet. Copying someone else's solution will be considered plagiarism and may result in failing the whole course.

Please turn in the solutions by 11.59pm on Monday 10 October. The homework should be dropped off in the box labeled CSC 3130 on the 9th floor of SHB. Late homeworks will not be accepted.

## Problem 1

Which of the following statements are correct? If you think a statement is correct, give a proof. If you think it is incorrect, give a counterexample. You may assume the alphabet is $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

(a) If $L$ is regular, then $L' = \{x \colon \mathtt{a}x \in L \text{ or } x\mathtt{b} \in L\}$ is regular.

(b) If $L$ is regular, then $L' = \{xx^- \colon x \in L\}$ is regular.
Here $x^-$ is $x$ without its last symbol, e.g. $(\mathtt{bab})^- = \mathtt{ba}$, $\mathtt{b}^- = \varepsilon$. (We let $\varepsilon^- = \varepsilon$.)

(c) If $L$ is regular, then $L' = \{x \colon xy \in L \text{ for some string } y\}$ is regular.

(d) If $L_1L_2$ is regular, then $L_2L_1$ is regular.
($L_1$ and $L_2$ can be any pair of languages, not necessarily regular.)

(e) **(Extra credit)** If $L$ is regular, then $L' = \{wz \colon zw \in L \text{ for some strings } w, z\}$ is regular.

## Problem 2

Which of the following languages are regular, and which are not? To show a language is regular, describe a DFA, NFA, or regular expression for it (with explanation). To show a language is not regular, prove it using the pumping lemma. The alphabet is $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

(a) $L_1 = \{wz \colon |w| = |z|, w \in (\mathtt{a} + \mathtt{b})^* \text{ and } z \in (\mathtt{b} + \mathtt{c})^*\}$.

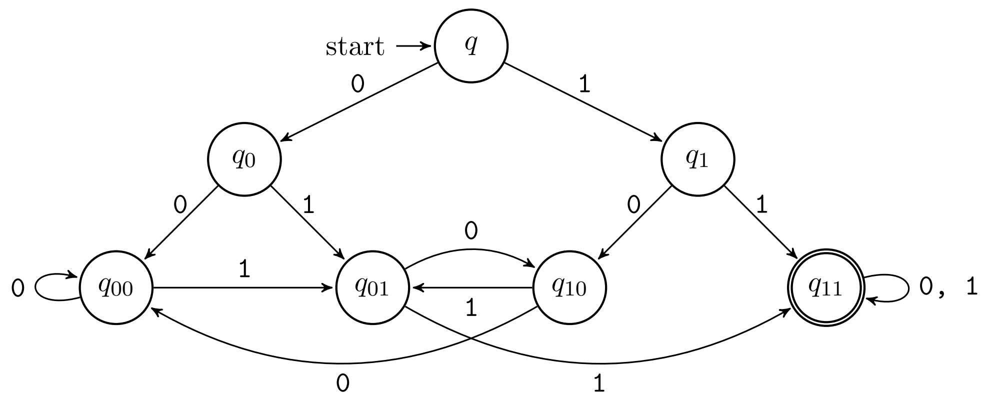
(b) $L_2 = \{w \colon \text{every } \mathtt{a} \text{ in } w \text{ is followed by at least one } \mathtt{b} \text{ and at least one } \mathtt{c}\}$.
For example, $\varepsilon, \mathtt{abaacb} \in L_2$, but $\mathtt{abacc} \notin L_2$.

(c) $L_3 = \{w \colon w \text{ does } \textit{not} \text{ have the same number of } \mathtt{a}\text{s}, \mathtt{b}\text{s, and } \mathtt{c}\text{s}\}$.

(d) $L_4 = \{w \colon w \text{ contains the same number of patterns } \mathtt{ac} \text{ and } \mathtt{abc}\}$.

## Problem 3

This problem concerns the following DFA.




(a) Run the minimization algorithm on this DFA. Clearly show the different stages that the minimization algorithm goes through.

(b) Show that every pair of states in the minimized DFA is distinguishable.

(c) Convert the minimized DFA into a regular expression. You may use the conversion algorithm from class, or write a regular expression by hand and argue that it is equivalent to the DFA.

## Problem 4

You have a file `propernames` which contains a list of first names in English. Write `grep` commands (including a short explanation about how they work) that will search for the following information in the file. You can use `grep -E -i` to ignore distinctions between uppercase and lowercase.

(a) Any name that contains exactly two vowels, as in `Amy`.

(b) Any name that any `a` is followed by at least two consonants, as in `Ahmed`.

(c) Any name that does not contain the pattern `ale`, as in `Emma` but not `Alex`.

(d) Any name that contains both the patterns `in` and `ri`, as in `Kristin` (but not `Nina`).

You can test your search patterns with the file
`http://www.cse.cuhk.edu.hk/~andrejb/csc3130/other/propernames`.