CSCI 3130: Formal Languages and Automata Theory — Homework 2
The Chinese University of Hong Kong, Fall 2015 — due 11:59pm Thursday October 8

Collaborating on homework is encouraged, but you must write your own solutions in your own words and list your collaborators. Copying someone else's solution will be considered plagiarism and may result in failing the whole course.

Please answer clearly and concisely. Explain your answers. Unexplained answers will get lower scores or even no credits.

(1) (6 points) Which of the following statements are correct? If you think it is correct, give a short justification (about 2-5 sentences). If you think it is incorrect, give a counterexample and a short justification of your counterexample.

(a) If L_1 is not regular and L_2 is not regular, then $L_1 \cup L_2$ is not regular.

(b) A string u is an anagram of a string w if u is obtained from w by rearranging the symbols. For example, `listen` is an anagram of `silent`. Formally, if $w = w_1 w_2 \dots w_n$, then u is its anagram if $u = w_{\sigma(1)} w_{\sigma(2)} \dots w_{\sigma(n)}$ for some permutation σ.
If L is regular, then $L^A = \{u \mid u \text{ is an anagram of } w \in L\}$ is also regular.

(c) If L is regular, then

$$L' = \{ww' : w, w' \text{ are strings in } L \text{ such that } w \neq w'\}$$

is also regular.

(2) (8 points) Which of the following languages are regular, and which are not? To show a language is regular, give a DFA, NFA, or regular expression for it. To show a language is not regular, prove it using the pumping lemma or using pairwise distinguishable strings.

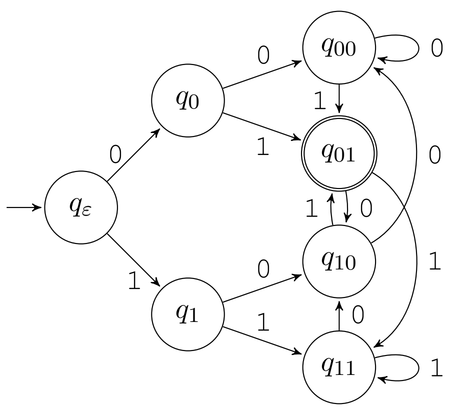
(a) $L_1 = \{\mathtt{0}^n \mathtt{0}^n \mid n \geqslant 0\}$

(b) $L_2 = \{\mathtt{0}^n \mathtt{1} \mathtt{0}^n \mid n \geqslant 0\}$

(c) $L_3 = \{\mathtt{1}^{n^3} \mid n \geqslant 0\}$

(d) $L_4 = \{w \mid w \text{ has the same number of occurrences of } \mathtt{010} \text{ and } \mathtt{101}\}$
Note: `01010` has two occurrences of `010` and one occurrence of `101`

(3) (12 points) Consider the following DFA:



(a) Run the minimization algorithm on this DFA. Show the table of pairs of distinguishable states at the end of the algorithm (see page 25 of Lecture 7). Also draw the minimized DFA.

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

(c) Convert the minimized DFA into a regular expression using the conversion algorithm from class. Show the preprocessing step and how the NFA changes after each state is eliminated.

(4) (8 points) The file `propernames` contains a list of English first names. To search for each of the following information in the file, write a `grep` command of the form

`grep -iE '`*regex*`' propernames`

Also give a short explanation (1-3 sentences) how it works in each case.

Recall that the option `-i` ignores distinction between upper and lower case. You can test your commands with the file `http://www.cse.cuhk.edu.hk/~siuon/csci3130/other/propernames`.

(a) Any name that has exactly three consonants, such as **Tr**a**c**y. `y` is regarded as a vowel.

(b) Any name that has no consecutive vowels, so `Mich`**ae**`l` is excluded.

(c) Any name that doesn't end in `ll` (double lower case L).

(d) Any name with two groups of consecutive repeated letters (and the two groups don't overlap), such as `Ru`**ss**`e`**ll**.

For part (d) you may want to use the backreference feature of `grep`.

(5) (6 points) For an integer $k \geqslant 1$, define L_k to be the set of strings (over $\Sigma = \{0, 1\}$) that have a `1` at the kth-to-last position. For example, **1**`00` and `0`**1**`01` are in L_3, but `0` and `011` are not.

(a) Prove that every DFA for L_k has at least 2^k states.

(b) Describe (e.g. with a diagram) an NFA for L_k that has at most $k + 1$ states.