CSCI 3130: Formal Languages and Automata Theory — Assignment 2

The Chinese University of Hong Kong, Fall 2021 — due 11:59pm Thursday October 7

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) (18 points) Briefly justify (in about 3-6 sentences) the following statements.

(a) If L is regular language over $\Sigma = \{0, 1\}$, then

$$L' = \{w \in L \mid w \text{ ends with } \texttt{1}\}$$

is also regular.

(b) There are irregular languages L_1 and L_2 such that $L_1 \cap L_2$ is regular.

(c) There is an irregular language L such that L^* is regular.
(Hint: use Lagrange's four-square theorem and a result in Week 5 tutorial)

(2) (32 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.

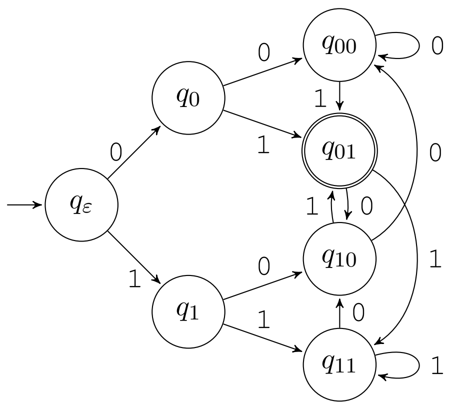
(a) $L_1 = \{\texttt{1}^n\texttt{1}^n \mid n \geqslant 0\}$

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

(c) $L_3 = \{w \in \{\texttt{0}, \texttt{1}\}^* \mid w \text{ has the same number of occurrences of } \texttt{00} \text{ and } \texttt{11}\}$
Note: 000 has two occurrences of 00

(d) $L_4 = \{u\#v \mid u, v \in \{\texttt{0}, \texttt{1}\}^* \text{ and } |u| \leqslant |v|\}$
Here $|u|$ denotes the length (number of symbols) of u

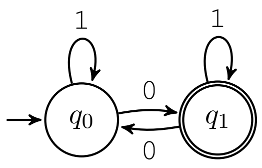
(3) (30 points) Consider the following DFA:



(a) Run the minimization algorithm on this DFA. Show the table of pairs of distinguishable states at the end of the algorithm (see page 22 of Lecture 7). Also draw the minimized DFA.

(b) Show that every pair of distinct states in the minimized DFA from part (a) is distinguishable (by giving a string to distinguish them, similar to page 10 of Lecture 7).

(c) Convert the following NFA 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) (20 points) This is the only problem in this course where we deal with POSIX-style regular expressions, as opposed to regular expressions in formal language theory.

The file `propernames` contains a partial list of first names. Every line contains a name. Each name consists of one or more English letters in upper or lower case. To search for each of the following piece of information in the file, write a `grep` command of the form

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

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

Recall that the option `-i` ignores distinction between upper and lower case. Test your commands with the file `http://www.cse.cuhk.edu.hk/~siuon/csci3130/other/propernames`. It's the best to run `grep` on Linux; `grep` in MacOS may behave a bit differently.

(a) Any name that has exactly four consonants, such as **Gr**a**nt**.
`y` is regarded as a vowel.

(b) Any name such that no consonant is followed by a vowel, so that **To**m is excluded.

(c) Any name whose every odd position is a consonant, such as **L**i**s**a.

(d) Any name with a substring *s* of length two such that *s* appears at least twice, such as **Ba**r**ba**ra.
For part (d) you may want to use the backreference feature of `grep`.

Your solution must not use `-v` flag of `grep` which swaps non-matches with matches.