

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, then

$$L' = \{w \in L \mid w \text{ is of odd length}\}$$

is also regular.

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

(c) There is an irregular language L such that $LLLL$ is regular.

(Hint: use Lagrange's four-square theorem)

(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.

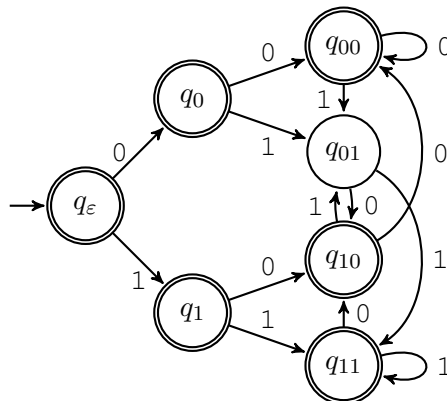
(a) $L_1 = \{0^m 0^n \mid m \geq 0, n \geq 0, m \geq n\}$

(b) $L_2 = \{w \in \{0, 1\}^* \mid w \text{ has at most as many 0's as 1's}\}$
 e.g. $01 \in L_2$ but $100 \notin L_2$

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

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

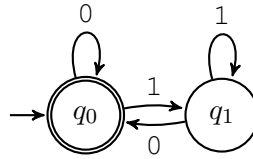
(3) (30 points) Consider the following DFA:



(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 9 of Lecture 7).

- (c) Convert the following 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) (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 `proprnames` contains a partial list of first names. Each name consists of one or more English letters in upper or lower case. To search for each of the following information in the file, write a `grep` command of the form

```
grep -iE 'regex' proprnames
```

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. You can test your commands with the file <http://www.cse.cuhk.edu.hk/~siuon/csci3130/other/proprnames>.

- (a) Any name that begins and ends with consonant(s), such as **Ben**. `y` is regarded as a vowel.
- (b) Any name without consecutive `l`, so **Molly** is excluded.
- (c) Any name containing an even number of vowels, such as **Patricia**
- (d) Any name where the same letter appears (at least) three times, such as **Vassos**.

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

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