CS46 practice problems 4

These practice problems are an opportunity for discussion and trying many different solutions. They are **not counted towards your grade**, and **you do not have to submit your solutions.** The purpose of these problems is to get more comfortable with NFAs and regular expressions. I recommend trying to solve these problems on paper *first*, then trying with the online tool. Once you are ready to test your solutions, the Automata Tutor site will give you troubleshooting feedback.

Note that Automata Tutor uses slightly different notation for regular expressions.

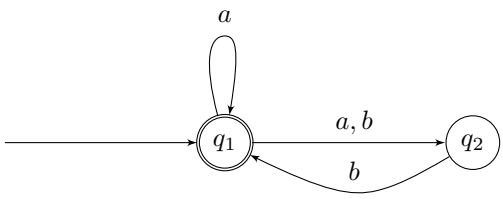
1. (a) Construct a DFA for the language $\{w \mid w \text{ begins with } a \text{ and ends with } b\}$.
   (b) Construct an NFA for the language $\{w \mid w \text{ begins with } a \text{ and ends with } b\}$. (Try to use nondeterminism so that it has fewer states than the previous DFA for the same language.)

2. Construct an NFA with *three states* that recognizes the language

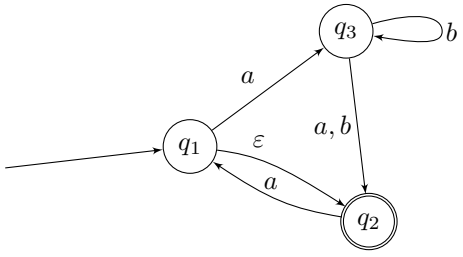
$$\{w \mid w \text{ ends with } bb\}$$

   You will need to use nondeterminism!

3. Construct a DFA equivalent to the following NFA:



4. Construct a DFA equivalent to the following NFA:



5. Construct an NFA for the language $L = (aa \cup ab)^*$ over the alphabet $\Sigma = \{a, b\}$.

6. Construct an NFA for each of the languages given by regular expressions:

   (i) `a(abb)`*$\cup$`b`
   (ii) `a`$^+$$\cup$`(ab)`*
   (iii) `(a`$\cup$`b`$^+$`)a`$^+$`b`$^+$

7. Construct a regular expression for the language Σ^* over $\Sigma = \{a, b\}$.

8. Construct a regular expression for the language $\{w \mid w \text{ contains the substring } ab\}$ over $\Sigma = \{a, b\}$.

9. Construct a regular expression for the language $\{aa, abba\}$ over $\Sigma = \{a, b\}$.

10. Construct a regular expression for the language $\{w \mid w \text{ contains exactly two } a\text{s}\}$ over $\Sigma = \{a, b\}$.

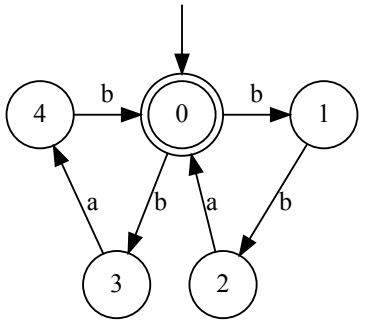
11. Let $L = \{w \mid \text{every appearance of } c \text{ in } w \text{ is in a contiguous substring with at least two other } c\text{s}\}$ over $\Sigma = \{a, b, c\}$. So for example, L contains *baaccca*, *aaab*, ε, and *cccca*. L does *not* contain *abcccbaca*, *bccb*, *c*, or *cabaaaabcc*.

    (a) Construct a DFA that recognizes L.

    (b) Construct an NFA that recognizes L.

    (c) Construct a regular expression that recognizes L.

12. Construct a DFA equivalent to the following NFA:



    Write a regular expression for the language accepted by this machine.