CS46 practice problems 2

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 DFAs and with using the Automata Tutor site. You may want to try to solve these problems on paper *first*, then trying with the online tool. Once you are ready to test your solutions, the site will give you feedback on how to improve your solution. For each practice problem, you are allowed unlimited attempts.

For all of these problems, $\Sigma = \{a, b\}$.

0. Go to Automata Tutor and create a login (use your preferred first and last name, as we will use this tool for actual graded problems as well). Enroll in this course with:
   Course ID: CS46-2020s
   Course Password: G4FSWXM3
1. Construct a DFA for the language $\{w \mid w \text{ contains the substring } ab\}$.
2. Construct a DFA for the language $\{w \mid w \text{ does } \textit{not} \text{ contain the substring } ab\}$.
3. Construct a DFA for the language $\{w \mid w \text{ contains the substring } baba\}$.
4. Construct a DFA for the language $\{w \mid w \text{ does } \textit{not} \text{ contain the substring } baba\}$.
5. Construct a DFA for the language $\{aa, abba\}$.

   You might consider breaking this problem into pieces:

   (a) Construct a DFA for the language $\{aa\}$.
   (b) Construct a DFA for the language $\{abba\}$.
   (c) Use the proof idea from theorem 1.25 (regular languages are closed under union) to construct a new DFA for the union language from your two simpler DFAs.
6. Construct a DFA for the language $\{w \mid w \text{ contains exactly two } a\text{s and at least two } b\text{s}\}$.

   You might consider breaking this problem into pieces:

   (a) Construct a DFA for the language $L_1 = \{w \mid w \text{ contains exactly two } a\text{s}\}$
   (b) Construct a DFA for the language $L_2 = \{w \mid w \text{ contains at least two } b\text{s}\}$.
   (c) We want to construct a DFA for $L_1 \cap L_2$, so we can use an idea like the footnote (page 46) on the proof of theorem 1.25 to construct the states and transitions for this new DFA.
7. Construct a DFA for the language $\{w \mid w \text{ begins with } a \text{ and ends with } b\}$.
8. 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.)
9. Construct an NFA with *three states* that recognizes the language

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

   . You will need to use nondeterminism!

10. Construct a DFA for the language $L = \{\varepsilon\}$ over the alphabet $\Sigma = \{a, b\}$.

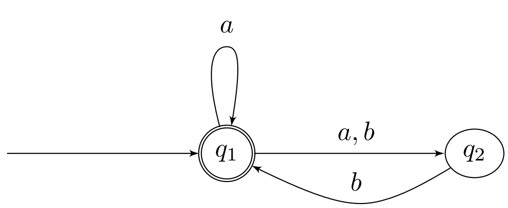
11. Construct a DFA for the language $L = \{w \mid w \text{ does not contain exactly two } a\text{s}\}$ over the alphabet $\Sigma = \{a, b\}$.

12. Construct a DFA for the language $L = \{w \mid 3 \leq |w| \leq 5\}$ over the alphabet $\Sigma = \{a, b\}$.

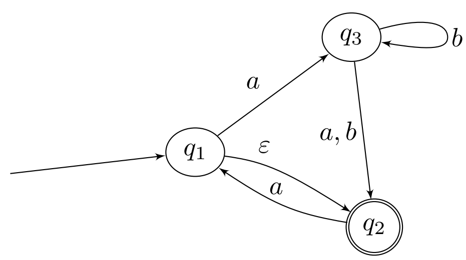
13. Construct a DFA for the language $L = \{w \mid a \text{ appears } k \text{ times in } w \text{ where } k + 1 \text{ is divisible by } 3\}$ over the alphabet $\Sigma = \{a, b\}$.

14. Construct a DFA for the language $L = \{w \mid \text{ every } b \text{ in } w \text{ is immediately followed by two } a\text{s}\}$ over the alphabet $\Sigma = \{a, b\}$.

15. Construct a DFA equivalent to the following NFA:



16. Construct a DFA equivalent to the following NFA:



17. **Serious challenge.** With only 3 attempts: Construct a DFA for the language

$$L = \{w \mid w \text{ is a binary number equal to 1 mod 3}\}$$

over alphabet $\Sigma = \{0, 1\}$. (So $0 \notin L$, $1 \in L$, $100 \in L$, etc.)

(**Advice:** You absolutely need to figure this one out on paper first, or you will run out of attempts.)