1. (10 points) Consider the DFAs $M_1$ (on the left) and $M_2$ (on the right):

(a) If we use the general constructions discussed in class and in the book for building a DFA whose language is $L(M_1) \cup L(M_2)$, how many states would be in this DFA? Briefly justify your calculation.

(b) Draw the state diagram that results from this construction and remove any unreachable states. How many states are left?

(c) Describe the language $L(M_1) \cup L(M_2)$ in set builder notation. You may find it useful to first describe $L(M_1)$ and $L(M_2)$ in set builder notation first.

(d) Use the description from (c) to draw a DFA with fewer states even than what you saw in part (b). Draw the state diagram in JFLAP and include the image in your submission.

Note: For parts (b) and (d), you can use the “test equivalence” feature of JFLAP to check your work.
2. Show that the class of regular languages over the alphabet \( \{0, 1\} \) is closed under the operation \( \text{FlipBits}(L) \), defined as

\[
\text{FlipBits}(L) = \{ w | w \text{ can be obtained from some } w' \text{ in } L \text{ by flipping each 0 in } w' \text{ to 1 and each 1 to 0} \}
\]

A full proof would have three stages: setup, construction, and proof of correctness. In this exercise you will focus on the setup and construction, and then apply your construction to an example.

**Setup** Consider an arbitrary DFA \( M = (Q, \{0, 1\}, \delta, q_0, F) \), and call the language of this DFA \( L \).

**Construction** Build a new DFA whose language is \( \text{FlipBits}(L) \). To do so, fill in the blanks

\[
M' = (Q', \{0, 1\}, \delta', q', F')
\]

where

\[
Q' = \underline{\text{This will be the set of states for your new machine.}} \\
\delta'(r, x) = \underline{\text{For each possible input to the transition function, specify the output. Notice that } r \text{ is a state in } Q' \text{ and } x \in \{0, 1\}.} \\
q' = \underline{\text{What is the initial state of } M'\text{? Make sure you choose an element of } Q'.} \\
F' = \underline{\text{What is the set of accepting states of } M' \text{? Choose a subset of } Q'.}
\]

**Application** Consider the language, \( L \), recognized by this DFA (from HW1):

![DFA Diagram](image)

Apply your construction to this DFA and confirm that the language recognized by the resulting DFA is \( \text{FlipBits}(L) \).

[[**Bonus (not for credit):** To prove that the construction of correct, we would need to prove that \( L(M') = \text{FlipBits}(L) \) for any \( L \). Fix an arbitrary but unknown language \( L \). Let \( M \) be a DFA recognizing \( L \), and construct \( M' \) from \( M \) as shown above. Two claims are required

(1) Assume that some string, call it \( w \), is accepted by \( M' \). Prove that \( w \) is in \( \text{FlipBits}(L) \).

(2) Assume that some string, call it \( y \), is in \( \text{FlipBits}(L) \). Prove that \( y \) is accepted by \( M' \).

Practice your proof techniques by carrying out this justification.]]
3. Consider the NFAs $N_1$ (on the left) and $N_2$ (on the right):

(N$_1$ is described in example 1.33 on page 52.)

(a) Write out the formal definition of $N_1$ and $N_2$. Notice that $N_2$ appears to be a DFA but we are asking for its formal description as a NFA. Be careful of types, especially when describing the transition function of each NFA.

(b) Write out $L(N_1)$ and $L(N_2)$ in set builder notation, justifying each briefly by making specific references to the state diagrams.

(c) In fact, $L(N_1) = L(N_2)$. You can check this by using the “test equivalence” feature of JFLAP to compare the two state diagrams. Explain why these two NFAs are equivalent.