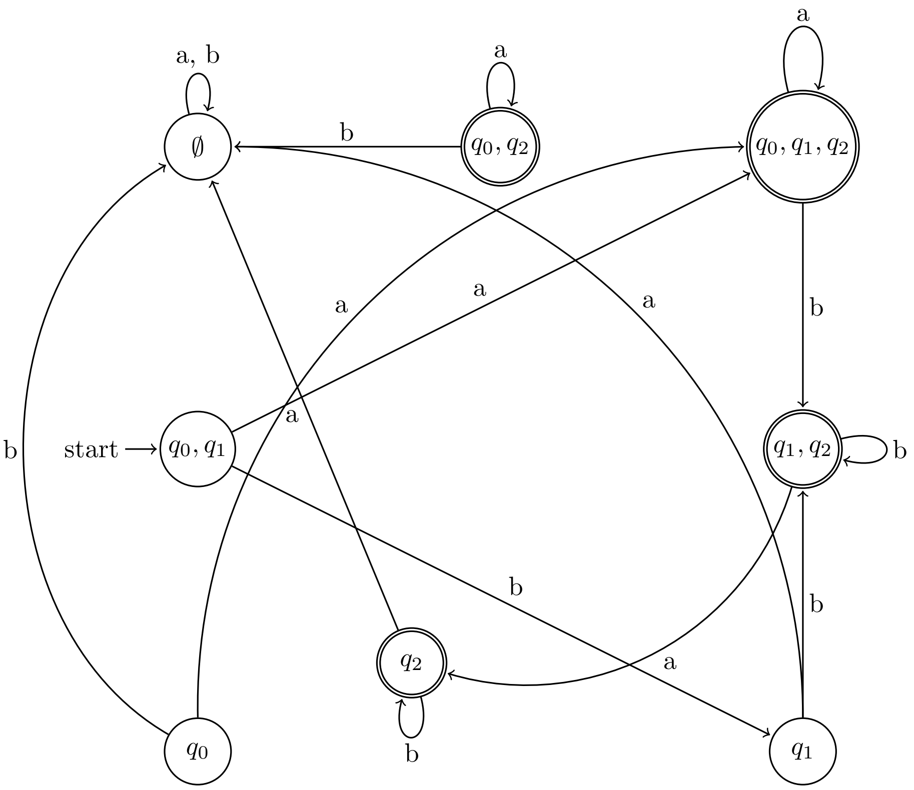
Exercise 1.

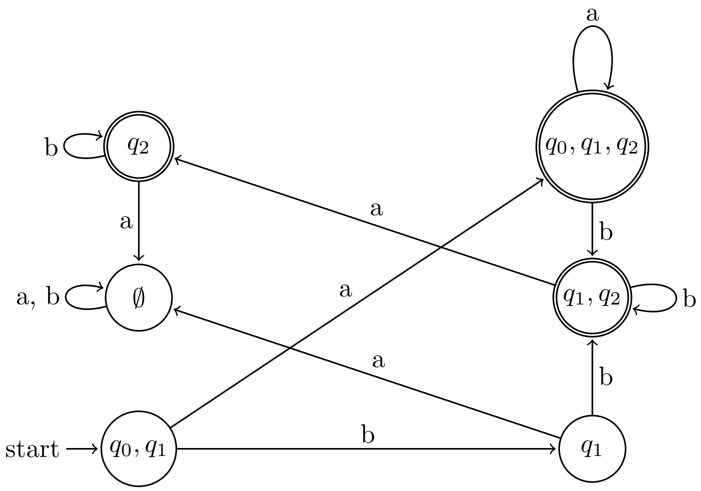
The determinization of the given NFA: (I literally tried my best.)



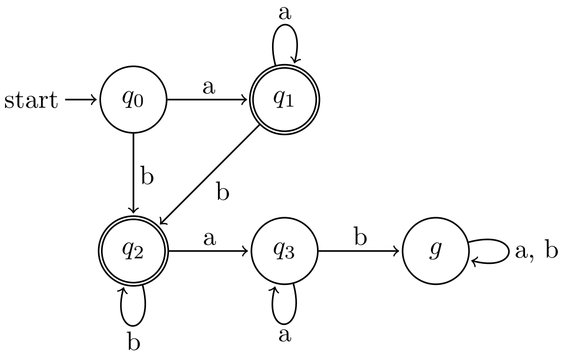
Exercise 2.

Here I'm going to trim it so you can actually make some sense of it :) (and also because I have to)

I moved q_2 to make the diagram a little simpler.



Exercise 3



To show that there is no DFA with strictly fewer states that recoginzes $L(\mathcal{A})$, we attempt to minimize it. If the minimized DFA is the same, then we know there is no DFA with fewer states that recognizes the same language.

We know the given DFA is trimmed because no states are inaccessible. So we must determine whether any of the states are indistinguishable. The only two states we have that have the same residual language are q_0 and q_1. They would be indistinguishible, however q_0 is not final, whereas q_1 is. The rest of the states do not share residual languages, which means all of the states are distinguishible. Since all of the states are distinguishable, the minimization of $\mathcal{A}$ would have the same number of states as $\mathcal{A}$. Since a minimization is inherently unique, and the amount of states is by definition minimal, there is no DFA with strictly fewer states than $\mathcal{A}$ that recognizes $L(\mathcal{A})$.

Exercise 4.
Define $\mathcal{A}$ as follows:

$$\mathcal{A} := (\Sigma, Q, \delta, q_0, F)$$

For a DFA to recognize $\Sigma^* \setminus L(\mathcal{A})$, it must not recognize any successful run in $\mathcal{A}$, and it must recognize every unsuccessful run in $\mathcal{A}$. What this entails is that every run that ended in a non-final state must now end in a final state, and every run that ended in a final state must no longer be final. Call this new DFA $\mathcal{B}$. We can recognize $\Sigma^* \setminus L(\mathcal{A})$ by defining $\mathcal{B}$ as such:

$$\mathcal{B} := (\Sigma, Q, \delta, q_0, F')$$

where F' is defined as $F' := \{q \in Q \mid q \notin F\}$. This causes every run that ended in a final state to no longer be successful, which now means it is no longer recognized in $L(\mathcal{B})$, and everything in Σ^* that was not recognized now has a successful run in $\mathcal{B}$.

Exercise 5.

Since L is finite, then L is by extension regular. Since L is regular, then we can ascertain that there exists an NFA that recogninzes L. Call this NFA $\mathcal{A}$. We can then determinize $\mathcal{A}$ into $\mathcal{A}^d$. Determinization, by definition, creates a DFA that recognizes the same language as the original NFA. Therefore, there exists a DFA (namely $\mathcal{A}^d$) that recognizes L.

Exercise 6.

We want to find a process to convert a given DFA to recognize the language $\{w^R \mid w \in L(\mathcal{A})\}$, where w^R recognizes the reversal of a word. We can achive this by essentially creating a DFA that is the "opposite" of the given DFA. Let the DFA $\mathcal{A}$ be defined $\mathcal{A} := (\Sigma, Q, \delta, q_0, F)$. We can the construct an NFA through the following process, and determinize it to get a DFA that recognizes the reversed language. Define the new NFA $\mathcal{N} := (\Sigma, Q, \delta_N, F, q_0)$, where δ_N is defined as follows:

$$\delta_N := \{\forall a \in \Sigma, \forall q \in Q, \delta(a, q) = \{q_0, q_1, \ldots, q_n\} \implies \forall i \in \{0, 1, \ldots, n\}, \delta_N(a, q_i) = q\}$$

Which is essentially stating that, if by saying a letter $a \in \Sigma$ from a given state $q \in Q$ we get a set of states, from that set of states by saying the same letter, we should get back to the original state. In laymans terms, this is reversing all of the arrows in the DFA. In order to turn $\mathcal{N}$ into a DFA, we can simply determinize it into $\mathcal{N}^d$, which gives us a DFA that recognizes the language $\{w^R \mid w \in L(\mathcal{A})\}$.