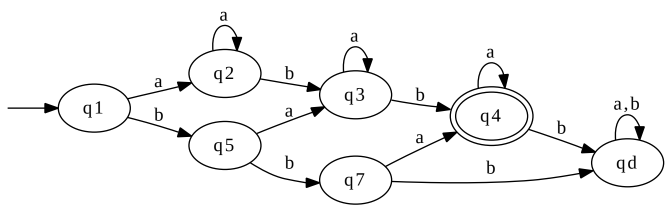
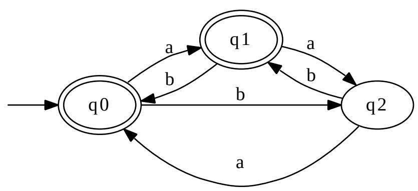
Homework 2
CS 321

1. Give a DFA for the following problems. Assume the alphabet is $\{a, b\}$.

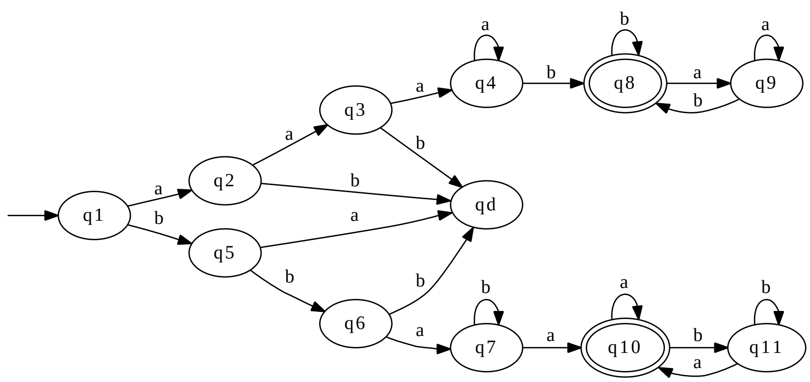
   (a) All strings over with at least one a and exactly two b's.



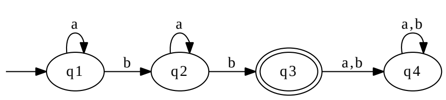
   (b) $L = \{w : n_a(w) + 2n_b(w) \bmod 3 < 2\}$.
   (q_x represents the state where $(n_a(w) + 2n_b(w)) mod 3 = x$)



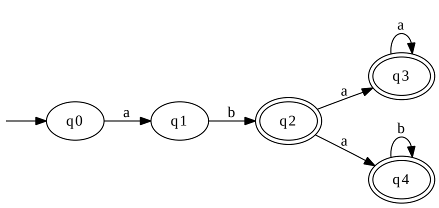
   (c) All strings of length 4 or greater in which the leftmost 3 symbols are the same, but different from the rightmost symbol.



2. Let L be the language accepted by the automaton of Figure 2.2. Find a DFA that accepts L^2.



3. Design an NFA with no more than 5 states for the set $\{abab^n : n \geq 0\} \bigcup \{aba^n : n \geq 0\}$.



4. Use Definition 2.5 to show that for any NFA,

$$\delta^*(q, wv) = \bigcup_{p \in \delta*(q,w)} \delta^*(p, v), \tag{1}$$

for all $q \in Q$ and all $w, v \in \Sigma^*$.

We have to show that $\delta^*(q, wv) = \bigcup_{p \in \delta^*(q,w)} \delta^*(p, v)$

From hint: You have to argue that any state r you can reach by reading wv from q will have a state p such that (a) you can reach p from q by reading w and (b) you can reach r from p by reading v.

We will use induction on the length of v. We will show both sides of the argument at the same time.

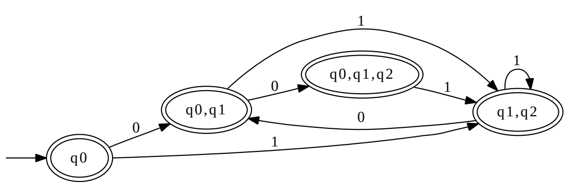
Base case: Suppose $v = \lambda$. Let $\delta^*(q, wv)$ contain a state r. Since $v = \lambda$, this means that $\delta^*(q, w)$ contains state r. Set $p = r$. Since $\delta^*(r, \lambda)$ contains r, r is in the right hand side set. Conversely, let r be a state in the right hand side when $v = \lambda$. Then there is a walk labeled w from q to p and then from p to r labeled λ. So r is in the left hand side set.

Inductive hypothesis. Suppose that the claim is true for q and wv.

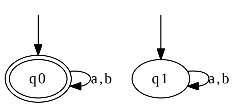
Inductive step. We need to show the claim for q and wva. Let $\delta^*(q, wva)$ contain state r, which means that there is a walk labeled wv from q to some state x and a walk labeled a from x to r. By inductive hypothesis, there is a walk labeled w from q to p and a walk labeled v from p to x. Concatenating this latter walk with a, there is a walk labeled va from p to r. Hence r is in the right hand side set.

Conversely, let r be a state in the right hand side set for w and va. This means that there is a state p which is reached by a walk labeled w from q, and from p there is a walk labeled va that reaches r. Concatenating these two walks, there is a walk labeled wva from q to r, which means that $r \in \delta^*(q, wva)$.

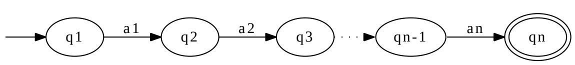
5. If we construct the DFA according to the assumption of q1 being the only final state:



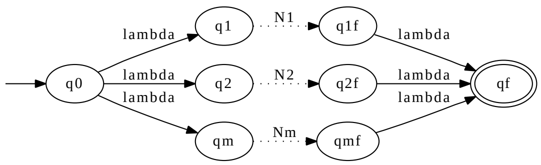
but as the figure doesn't contain any information on final states, these solutions, and their modifications are also true.



6. We can do this by showing how to construct an NFA for any finite language. First we show how to construct an NFA for a single string. Let the string be a1a2..an. Then we can build an NFA for it as follows.



qn is the final state and q0 is the initial state. Suppose the finite set contains m strings, w1, ... wm. For each string wi, we construct an NFA Ni. To accept all the strings w1...wm, we build a finite union of all the NFA's. We introduce a new initial state q and a new final state qf. From the initial state q, we have lambda transitions to the initial states of all m NFAs. From the final states of all the NFAs we have lambda transitions to the final state qf. This is represented schematically by the following diagram.



It is clear that this union machine accepts a string iff one of the NFA's accepts it, which happens if and only if it is in the finite set of strings from which it is built.

7. Suppose that N is an NFA that has some λ transitions and has possibly many final states.

   Build another NFA N_1 by adding a new final state f to N with λ transitions from each of the final states in N to f. Now make all the final states in N non-final. It is clear that N_1 accepts if and only if N accepts. Besides N_1 has a single final state.

   Now construct another NFA N_2 over the same states in N_1 as follows. They have the same initial and final states.

   For every state p, if there is a walk labeled $a \neq \lambda$ and some λ transitions to state q in N_1, then put an arc from p to q labeled a in N_2.

   We will show by induction on w that there is a walk labeled $w \neq \lambda$ from the initial state to any state in N_1 if and only if it is also true in N_2.

   Basis: $w = a$. If there is a walk labeled w from the initail state to any state in N_1, there is an edge labeled a from the initial state to that state in N_2 by construction.

   Inductive step: Suppose the claim is true for w. So there is a walk from the initial state to any state q in N_1 iff it is also true in N_2.

   Let there be a walk labeled wa from the initial state q_0 to some state r in N_1. Hence there must be a walk from q_0 to some state q in N_1, and hence in N_2 by inductive hypothesis.

   There must be an edge labeled a from q to r by construction. Hence there must be a walk labeled wa in N_2.

   Similarly if there is a walk labeled wa in N_2 from q_0 to r where q occurs just before r in that walk, there must be a walk labeled w from q_0 to q (by inductive hypothesis) and a walk labeled a from q to r (by construction). Hence there is a walk labeled wa from q_0 to r, proving the induction.

   Since there is no λ in the original language, $L(N) = L(N_1) = L(N_2)$.