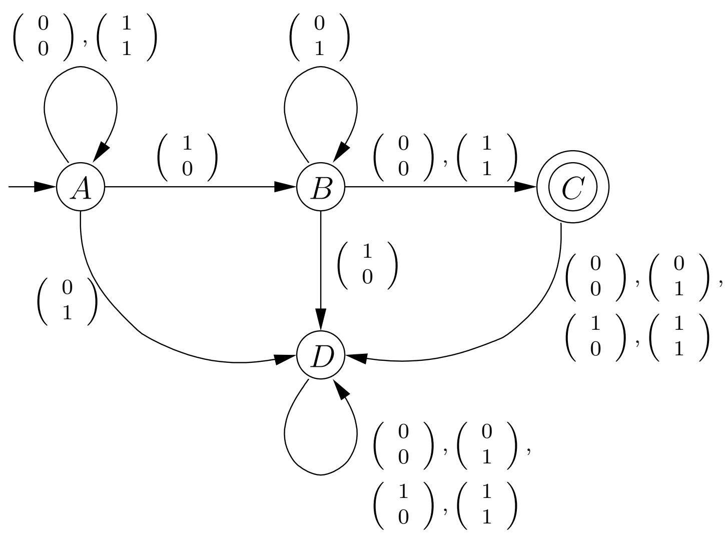
Homework Assignment #2 Solution

1.



Claim: After processing a string whose top row represents the number *top* and whose bottom row represents the number *bot*, the DFA is in

- state A iff $top = bot$,
- state B iff $top = bot + 1$,
- state C iff $top = bot + 2$, or
- state D if none of the above conditions apply.

It would be easy (but a bit tedious) to prove the above claim by induction on the length of the string that has been processed (although you were not required to do this for this assignment).

2.

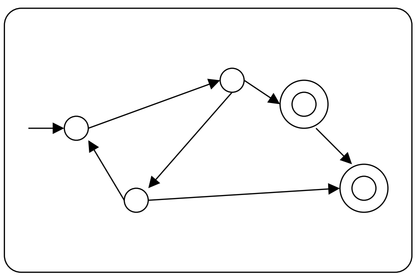
(a) $\hat{L}_1 = \{01, 11, 10\}$.

(b) Let L be any regular language. Let $M = (Q, \Sigma, q_0, \delta, F)$ be a DFA that recognizes L. We shall build a non-deterministic finite automaton M' that recognizes $\hat{L}$.

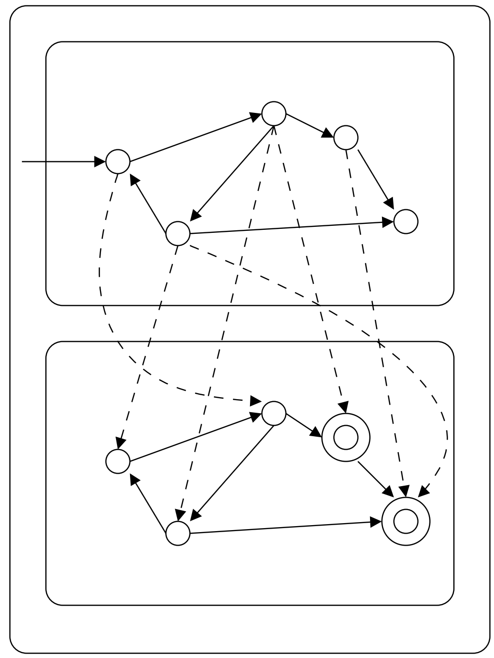
High-level idea: M' will behave like M, except that it must take exactly one transition of M without processing an input character. (This corresponds to the spot where a character a has been removed from a string that M accepts.) This will be accomplished by allowing M' to take exactly one ε-transition. In order to do this, M' must remember whether or not the ε-transition has already been used.

To do this, M' has two copies of each state of M. When processing a string, M' follows transitions of M within the first copy of M. Then, it non-deterministically chooses to take an ε-transition into the second copy of M, and then continues processing the rest of the string just as M would, staying in the second copy. (See the figure below for an example; the ε-transitions are shown as dashed lines.)

M



M'



Formally, $M' = (Q', \Sigma, q_0', \delta', F')$ whose components are defined as follows.

$$
\begin{array}{rcll}
Q' & = & Q \times \{1,2\} & \text{(use two copies of } Q\text{)} \\
q_0' & = & (q_0, 1) & \text{(start in the initial state of the first copy)} \\
\delta'((q,1),a) & = & \{(\delta(q,a),1)\} \text{ for } q \in Q, a \in \Sigma & \text{(transitions within first copy)} \\
\delta'((q,2),a) & = & \{(\delta(q,a),2)\} \text{ for } q \in Q, a \in \Sigma & \text{(transitions within second copy)} \\
\delta'((q,1),\varepsilon) & = & \{(q',2) : \exists a \in \Sigma \text{ such that } \delta(q,a) = q'\} \text{ for } q \in Q & \\
 & & & \text{(}\varepsilon\text{-transitions to switch from first to second copy)} \\
\delta'((q,2),\varepsilon) & = & \{\} & \text{(cannot do } \varepsilon\text{-transition in second copy)} \\
F' & = & \{(q,2) : q \in F\} & \text{(accept only in second copy)}
\end{array}
$$

You were not required to give a formal proof that M' accepts $\hat{L}$. But here are the claims that you would prove if you wanted to give a completely formal proof:

Claim 1: M' can be in state $(q,1)$ after processing a string z iff M is in state q after processing z.

Claim 2: M' can be in state $(q,2)$ after processing a string z iff $\exists x, y \in \Sigma^*$ and $a \in \Sigma$ such that $z = xy$ and M is in state q after processing xay.

Both claims could be proved by induction on the length of z. It follows from Claim 2 and the definition of F' that M' accepts a string z iff $z \in \hat{L}$.