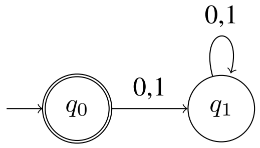
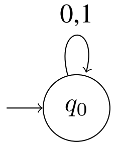
CMSC 303 Introduction to Theory of Computation, VCU
Assignment: 2
Name: Steven Hernandez

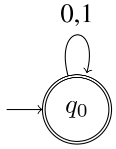
1. (a)



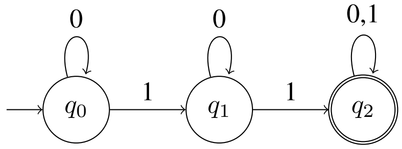
(b)



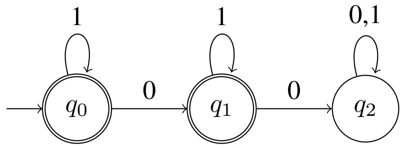
(c)



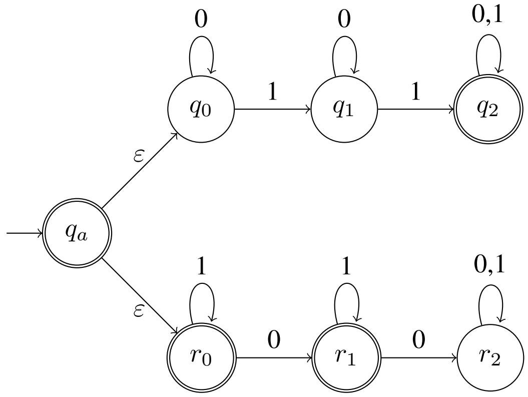
2. (a)



(b)



(c)



3. This implies regular languages are closed under negation. If a DFA M can be designed to recognize B, then applying negation you would have to switch accept and nonaccept states. Because the original DFA M was already determined to be have a finite number of steps, then $\bar{B}$ will have the exact same number of steps, thus $\bar{B}$ is also finite and regular.

4. Proof: Let M_1 recognize A, where $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, and M_2 recognize B, where $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

   Construct M to recognize $A \cdot B := \{x | x \in A$ and x does not contain any string in B as a substring.$\}$ where $M = (Q, \Sigma, \delta, q_0, F)$.

   (a) $Q = (r_1, r_2) | r_1 \in Q_1 andr_2 \in Q_2 \bigcup \emptyset$.

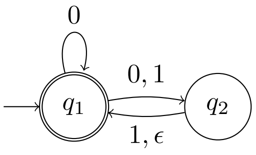
   (b) Define δ such that for $R \in Q$ and any $a \in \Sigma$.

$$\delta(R, a) = \begin{cases} \emptyset & R \in F_2 \\ \delta(q_1, a) \bigcup \delta(q_2, a) & R \notin F_2 \end{cases}$$

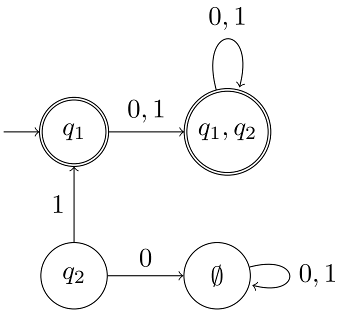
   (c) $q_0 = (q_1, q_2)$.

   (d) $F = \{(r_1, r_2) | r_1 \in F_1 andr_2 \notin F_2\}$.

5. M



M'



6. Claim: For any $k, C_k = \Sigma * 0\Sigma^{k-1}$ for $k \geq 1$ cannot be recognized by a DFA with less than 2^k states.

   Proof: We will begin by taking note of the construction of an NFA for the language C_k. Notice there must first be a state q_0 with transitions: $\delta(q_0, 0) = \{q_0, q_1\} and \delta(q_0, 1) = \{q_0\}$. This handles the beginning $\Sigma * 0$.

   From q_1 must have k additional states to handle counting exactly Σ^{k-1} characters after the 0. Each of these states has a transition $\forall a \in \Sigma$ to the next state except the final state. This final state is the accept state which has no transitions outbound. This is because if we have any additional input, we want the NFA to disregard this branch.

   The number of transitions required for this NFA would be $2 + (k - 1) = k + 1$.

   From our process of transforming NFAs to DFAs, we know that if an NFA has some integer x states, to create a full diagram of all possible transitions, our NFA must contain each element from the Powerset of x. We also know the $P(x) = 2^x$.

   Seemingly, this proves the claim because $2^{k+1} \geq 2^k$, however, we must take note about q_0 from our NFA.

   No matter the input into the NFA, q_0 will always be an option for the NFA. Thus, $\{q_0\}$ should not be factored into the powerset above. So instead, the number of states required for our DFA is $(2^k) + 1 \geq 2^k$ proving the claim.