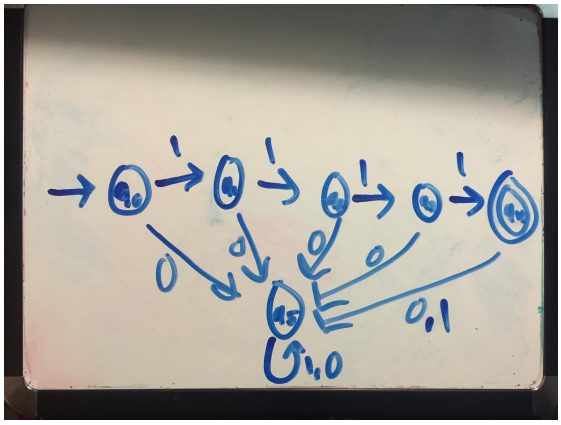
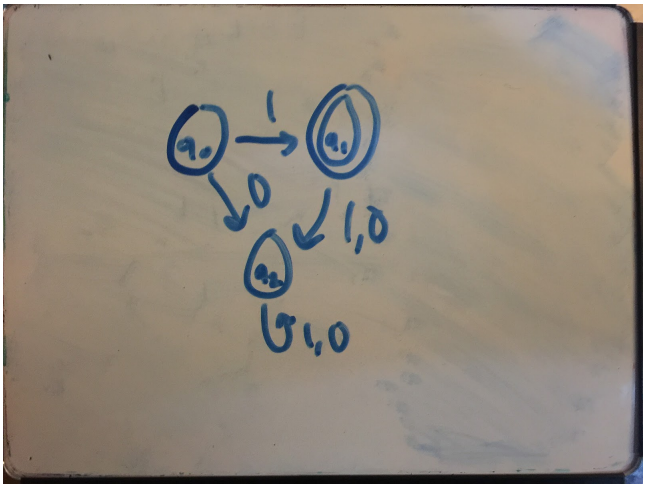
Simrun Heir and Michael Sanchez
I pledge my honor that I have abided by the Stevens Honor System.
Problem 1:

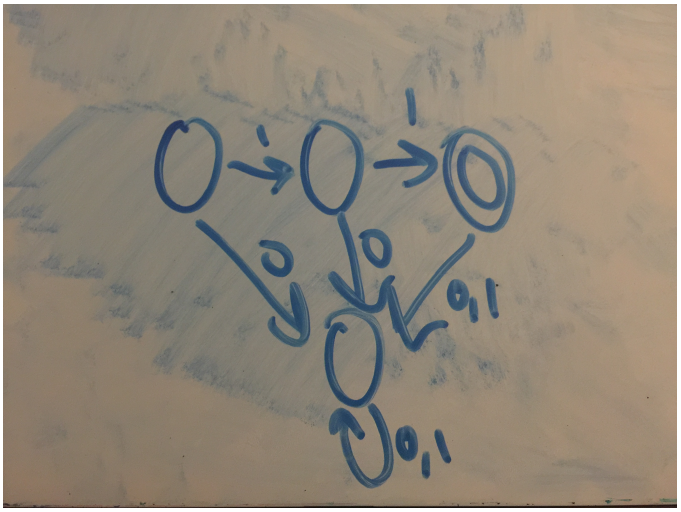
a. 

Explanation: In order to ensure that the only input accepted is 1111, any input of 0 results in the machine getting caught in the state q_5, which loops back into itself for any input, 1 or 0, and if too many 1's are input, the machine will still get caught in q_5. n

b. For each number *k* greater than or equal to 3, there exists a language that can be accepted by an FA with *k* states that cannot be recognized by an FSA with fewer than *k* states.
   i. Let M_1 be a k state FSA $M_1=(Q_1, \Sigma_1, \delta_1, q0, F_1)$,where k = 3, and $\Sigma_1 = \{0,1\}$, constructed as follows



   ii. Let L_1 be a regular language, where $L_1 \in \Sigma$ and $L_1 = \{1\}$
   iii. Let M_2 be a k+1 state FSA $M_2 = (Q_2, \Sigma_2, \delta_2, p0, F_2)$, where $\Sigma_2 = \{0,1\}$, constructed as follows



iv. Let L_2 be a regular language, where $L_2 \in \Sigma_2$ and $L_2 = \{11\}$

v. M_1 does not accept L_2

vi. By induction, for all k>=3 there is a language that is accepted by a k-state FSA but not an FSA with less than k states

Problem 2:

Let A be regular. This means there exists a DFA S that recognizes this language. w represents a string that is part of the language A. w1 is the first input of the string, it causes the transition out of the start state q0. wn is the last input of the string, which brings the S into its accept state. Since w^R is the reverse of w, wn is now the first input and w1 is now the last input. Using this, you are able to create a DFA R by going through S backwards according to the reverse order given by w^R. This means that the accept state of S becomes the start state of R, q'0, and the start state of S is the accept state of R. Since A^R = {w^R | w is an element of A} and w^R is accepted by R, A^R is regular.

Problem 3: