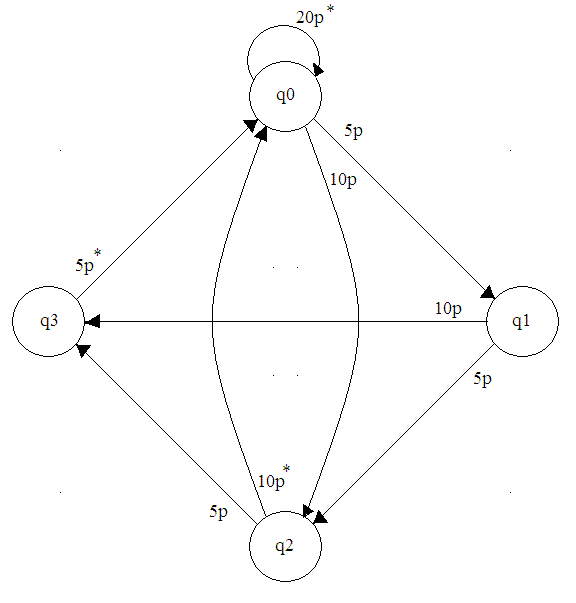
**Finite State Machines**

1. Get in groups of 3 or 4 and as for the washing machine in the lecture, go through the process demonstrated in the lecture for a lift system with just ground and 1st floor.
2. Consider a rather crude coffee-vending machine - Mark 1 - a cup of coffee costs 20p, and the machine accepts 5p, 10p and 20p coins. No change is given, when the machine has at least 20p, it disgorges a cup of coffee. Mark 1 is also provided with an inexhaustible supply of water, cups, instant coffee and milk etc, factors that a real implementer would have to deal with, and thus introduce unnecessary complication. Here is the diagram for Mark 1:



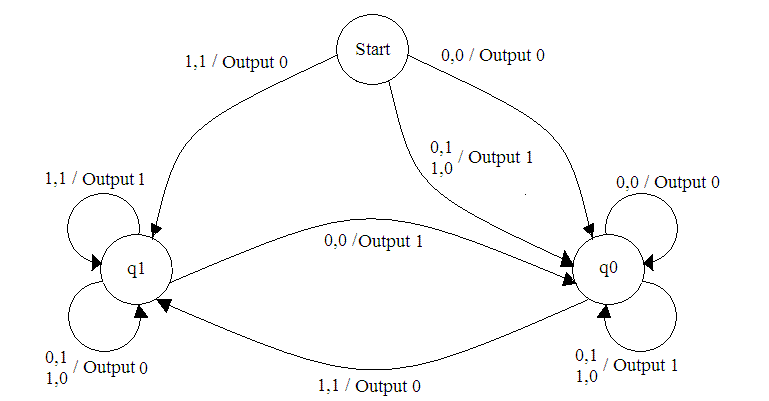
q0 is the start, or ‘reset’ state, in which no coins have been inserted; in q1, 5p has been inserted; in q2, 10p, and q3, 15p has been inserted. The arrows indicate the transitions, and are labelled with the coin value that enacts this transition. A \* denotes the delivery of a cup of coffee.

Depict the diagram above as a state table.

1. Modify the Mark 1 - to Mark 2 - such that it will accept 5p, 10p and 20p coins as before, but in all states, i.e. Mark 2’s actions are defined for all the three coins in all states. Mark 2 doesn’t give change; it delivers a cup of coffee if at least 20p is inserted. Excess coins are lost to the system: the next customer doesn’t benefit from the previous customer’s over-payment.
2. Modify Mark 2 - to Mark 3 - such that a customer pays only 20p for a cup and the machine gives change.
3. Now modify Mark 1 to Mark 4 - such that excess coins entered contribute to the next customer’s payment.
4. Minsky describes a binary adder in FSM form. To add two binary numbers conventionally, say 10 (=01010) and 7 (=00111), the two numbers are written down in corresponding positions, and the addition performed starting with the Least Significant Bits

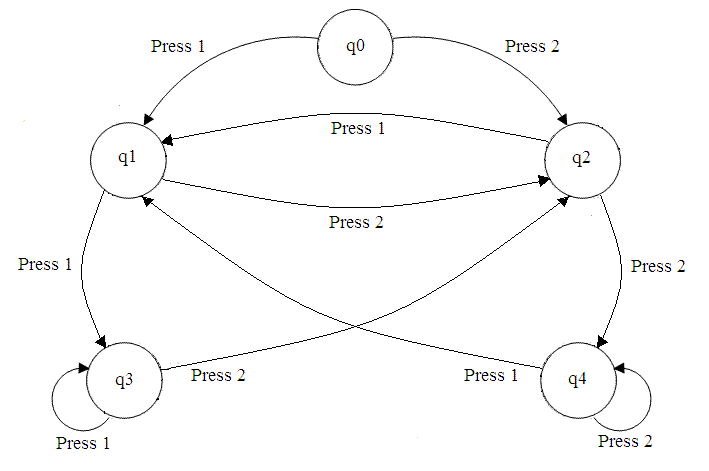
In Minsky's adder, each corresponding pair of binary digits (one from each of the two numbers to be added, is presented to the FSM, which moves from state to state, outputting 0 or 1, which is written down in the answer position, under the corresponding digits of the numbers to be added.

Determine the state table from this diagram. Use either the diagram or the table to verify the binary addition of 10 and 7



1. A system uses two buttons (1 and 2) and two lights, (Red and Green). The rule-set is :-
   * If button 1 is pressed then SET the Red light ON. Additionally,
     + If the previous press was button 1 then FLIP the Green light
   * If button 2 is pressed then RESET the Green light OFF. Additionally
     + If the previous press was button 2 then FLIP the Red light
   * When starting for the first time regard both previous button inputs as zero

Given the Mealy State Transition Diagram below, identify the states represented on the State Transition Diagram which correspond to those given by the above rule set.



Depict the  state transition diagram as a state  transition table.

Why is there no way back to the state q0? Is this a fault in the diagram, or is there a reason or it?

What is the sequence of actions of this system if initially both lights are off and the buttons are pressed in the following sequence (1,1,1,2,2,2,1,2,1,2,2,1,1). Produce an Action Table showing the sequence and the subsequent states.

1. Implement the above Mealy Machine as a Moore Machine
2. A system uses three buttons (1, 2 and 3) and three lights, (Red, Blue and Green). The rule-set is :
   * If button 1 is pressed then SET the Red light ON.
   * If button 2 is pressed then
     + If the previous press was button 1 then set Blue light ON and FLIP the Green light
     + If the previous press was button 2 then RESET Red OFF
     + If the previous press was button 3 then RESET Blue OFF
   * If button 3 is pressed then FLIP Blue. Additionally
     + If the previous press was button 1 then FLIP Red
     + If the previous press was button 2 then SET Green ON

Draw a state transition diagram for a Mealy Machine that satisfies the above rule set.

Show the state transition diagram as a statetransition table.

Produce an Action Table showing the sequence and the subsequent states of the lights after the following sequences of buttons are pressed:- 1,1,2,3,1,3,2,1,3,3

Devise a sequence to visit every state once only and produce an action table.