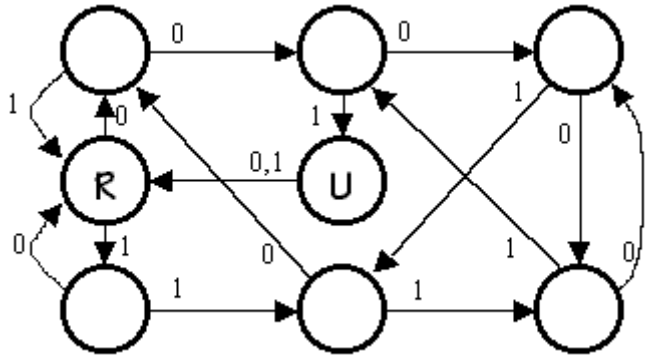
CS270 State machine Name_____

Pepsi Machine

You are hired by Pepsi to design their next generation Pepsi (Generation Next) machine, the reason being that Pepsi is going to drop the price of their product to a quarter. The new machine will accept the industry standard nickel, dime and quarter (for simplicity sake assume it does not accept bills…and pennies who uses those anyway). Design a totally synchronous FSM and be sure to anticipate any coin combination that can occur including strange combinations such as the user has already but in 20 cents and puts in another quarter. Assume the only inputs to the system are the type of change put in (nickel, dime, or quarter). Your outputs will be a change amount, and a Pepsi release signal that goes high when enough change has been put in.

Hint: Try to draw your Mealy state diagram neatly…you could end up with a lot of arcs.

Megan Bitminder has designed an electronic lock with three buttons: "reset", "0" and "1". She has provided the following state transition diagram showing how the lock responds to a sequence of inputs.



The lock makes a transition from its current state to a new state whenever one of the three buttons is pressed and released. It ignores its inputs if more than one button is pressed. Pressing "reset" returns the lock to the state marked "R" in the diagram (arcs showing the transitions to the reset state have been omitted from the diagram to make it easier to read). Pressing "0" or "1" will cause the lock to follow the appropriately labeled transition from its current state. The lock opens if it reaches the state marked "U".

A. After pressing the "reset" button what is the length of the shortest sequence of button presses that will open the lock?

B. After pressing the "reset" button what is the length of the longest sequence of button presses that will cause the lock to open after the last button in the sequence is pressed but not open any earlier in the sequence?

C. After much use, the "reset" button breaks. Is it still possible to open the lock using only the "0" and "1" buttons assuming you know nothing about the lock's state (except that its locked!) when you start?

D. Suppose Megan wanted to design a lock that required exactly 10 button presses to open after pressing "reset". Not counting the "reset" and "unlock" states, what is the minimum number of states her FSM would need?