Introduction to Computing: Task 2
Fall 2018
Dr. Ramesh Yerraballi
Designed by Matthew Barondeau
The University of Texas at Austin
Due: Friday 11/30, 11:59pm

Purpose

Finite state machines are used to model and simplify real-world problems, from traffic lights to vending machines to how computers respond to your actions. In this task, you will use JFLAP to create two Moore FSMs. You will also implement one state machine in Logisim using a PLA.

Task Requirements

1. Read the two problems described below. List the inputs and outputs and draw/sketch a Moore state transition graph for each system. Be sure to label each state with a name and its outputs and label each transition with its transition conditions.
2. Generate a State Transition Table for a state machine that models problem 1.
3. Using the State Transition Table, build the Moore FSM in JFLAP.
4. Run grading script to make sure your FSM works.
5. Use combination analysis to create a PLA implementation of the FSM
6. Generate a State Transition Table for a state machine that models problem 2.
7. Use the State Transition Table to build a State Transition Graph in JFLAP
8. Run sample grading script to ensure testability. Note this grading script is not comprehensive and it is your responsibility to test it in depth.
9. Answer the following questions:
    a. There are two types of FSMs. What are they?
    b. What is the difference between the two types of FSMs?
    c. What is the relationship between the number of states in the FSM and the size of the decoder in the PLA?
    d. Describe a problem that cannot be solved or modeled using an FSM
    e. Describe 3 other problems that a FSM could be used to model besides those discussed in lecture and in this assignment.

Software Installation

For this lab, you need to install JFLAP, an FSM simulation program. It can be found here: http://www.jflap.org/getjflap.html
You will also use Logisim, which should be installed on your computer.

When you run JFLAP, a small menu will pop up. On this menu, select Moore Machine and the menu will transform into a small window. It is in this window that you will implement your Moore FSM. I have included a tutorial on how to use JFLAP below if you are not satisfied with the tutorial here: http://www.jflap.org/tutorial/

Problem 1

In 2018, a boil order was issued in Austin. You and your friends had a brilliant idea to install a water bottle vending machine inside the EER. You have decided to price water bottles at $1. Your machine will process the change and determine if it should dispense a water bottle. Whenever an appropriate amount of money has been inserted into the machine, you will output water to the user and update your FSM accordingly. You have decided that you want to accept some coins, but anything below a quarter isn't of interest to you.

Specifically, you will need to handle the following change:

1. $1 Bills
2. Half Dollar Coins
3. Quarters

Any other bills or coins are invalid and not accepted by the machine; the machine remains unchanged if an invalid denomination is inserted. Whenever there is more than $1 in the machine, the machine should output a water bottle and subtract $1 from the balance. Assume there is no money in the machine initially.

Use the following encoding for inputs:
00: Invalid input: remain in the same state
01: Quarter inserted. Increase money amount by 25 cents
10: Half Dollar Coin inserted. Increase money amount by 50 cents
11: $1 Bill inserted. Increase money by $1

You must test your JFLAP implementation before implementing your FSM in Logisim. We will provide you a sample script that you can run as well as an expected output. Make sure this is correct before moving to the Logisim portion.

For your implementation in Logisim, you must implement the FSM in the form of a PLA. There should be a clock component in your Logisim implementation; otherwise, your FSM may detect one coin as many. You will be provided a starter file and an example of a PLA implementation of a different FSM. Be sure to test your system rigorously; we may use more test cases than we give you!

Please use combinational analysis to build the PLA. You can generate a circuit simply by inputting the truth table. PLEASE DON'T DRAW OUT BY HAND.

Problem 2

You will model a traffic light using JFLAP. Since this FSM will have many more inputs and outputs, we will not require you to implement it as a PLA. The traffic light is as follows:

There are two roads. One goes from East to West and the other goes from South to North. These roads are one way. Each road has a traffic light and a pedestrian walkway. It is your job to implement this traffic light using the following specifications:

There are three inputs:

1. A motion sensor on the South road
2. A motion sensor on the East road
3. A walk button – When on, pedestrians can cross both roads.

There are 6 outputs:

1. A Red light for the South Road
2. A Green Light for the South Road
3. A Red Light for the East Road
4. A Green Light for the South Road
5. A Walk Light
6. A Stop Walking Light

Your job is to make a functioning traffic light with a few guaranteed behaviors:

1. If all three inputs are 0, you should remain in this state
2. There is no yellow light state, so we must have an all red state between transitions
3. Walking should have priority over the other sensors. If the walk and another sensor are pressed, service the walk first
4. We don't want to starve the cars for the sake of the pedestrians. Therefore, at the end of a walk state, if walked is still pressed but another direction wants to go, let the other direction go. To prevent random behavior here as well, give the south priority over the East road

Implement your State Transition Table in JFLAP and test it. If there are any ambiguities, ask the Tas who can clarify.

Submission

When submitting your file, you need to name the JFLAP files firstname_lastname_Task2-1.jff and firstname_lastname_Task2-2.jff for the first and second problems. For Logisim, you need to submit your file as firstname_lastname_Task2. circ. Your pdf which contains the questions and the state diagrams/tables must be submitted as firstname_lastname_Task2.pdf. No late submissions shall be accepted.

Grading

Item	Points
State Transition Table Problem 1	10
Sample Script Runs Problem 1	10
Testing Script Runs Problem 1	15
PLA Implementation Works	20
State Transition Table Problem 2	10
Sample Script Runs Problem 2	10
Testing Script Runs Problem 2	15
Questions	10

JFLAP Tutorial.

1. Open up JFLAP and Select Moore FSM. The window is shown in Figure 1.

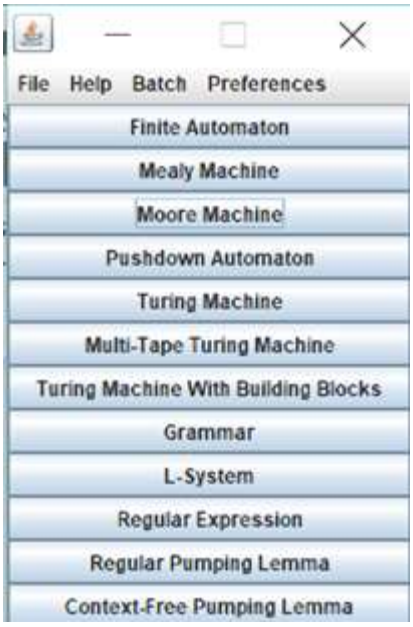


Figure 1. JFLAP Options window

2. If you have a higher resolution screen, when you open the Moore FSM, drag the automaton size bar at the bottom very far to the right. The bar is shown in Figure 2.

Figure 2. Zoom Bar for JFLAP

3. Once we have a Moore FSM window open, we need to create some states. To do this, we will go to the top left towards the file tab and select the circular button.

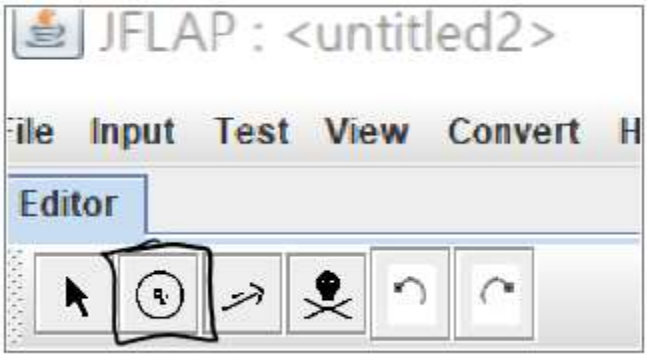


Figure 3. State Addition menu

4. Then you will need to drop this state onto the canvas, and you can specify the outputs for that individual state.

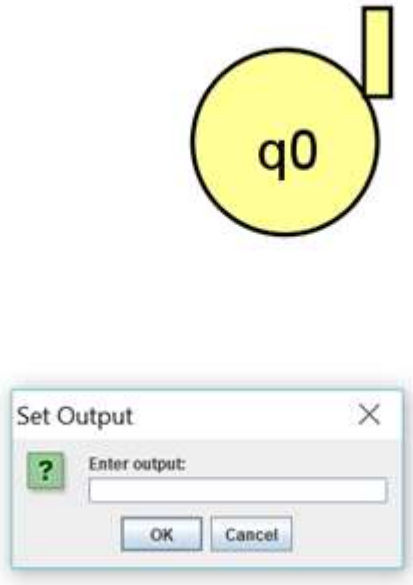


Figure 4. Adding a state

5. To modify the state, select the point option back up on the top left and then right click on a state to edit it.

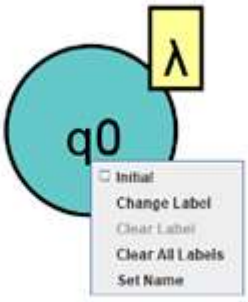


Figure 6. Setting the initial state

6. You will need to specify an initial state that the FSM will begin to execute at or else you cannot run the testing script.

7. You will now need to add connections between states. This can be done with the arrow button up on the top panel. You will need to drag a beginning point to the end point and then specify the input on which the graph should take that option.

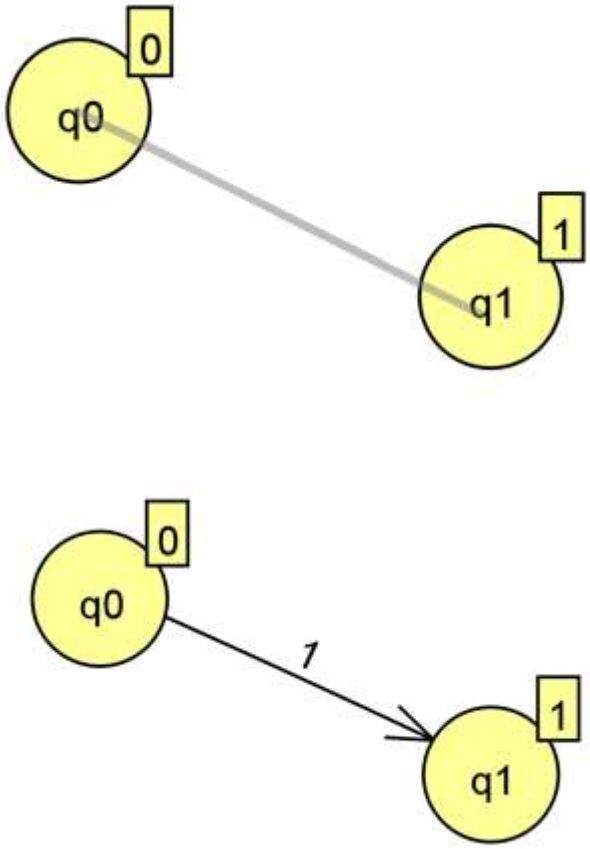
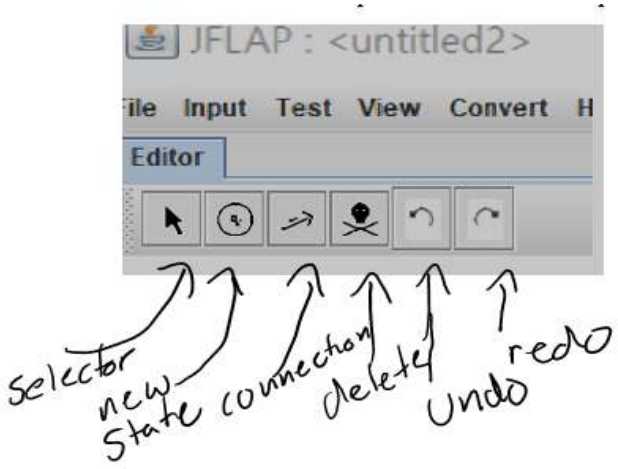


Figure 7. Adding a transition arrow

8. Below are the options available on the quick launch button bar:



9. To test your input, we will be using the input tab. In the input tab, we can select an option to load a file, or to input a testbench one at a time.