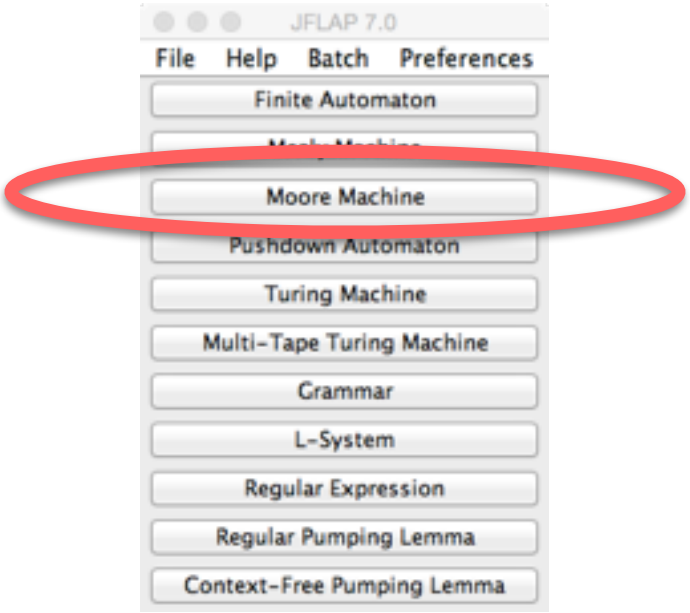
JFLAP: Getting Started
Creating Finite State Machine Models

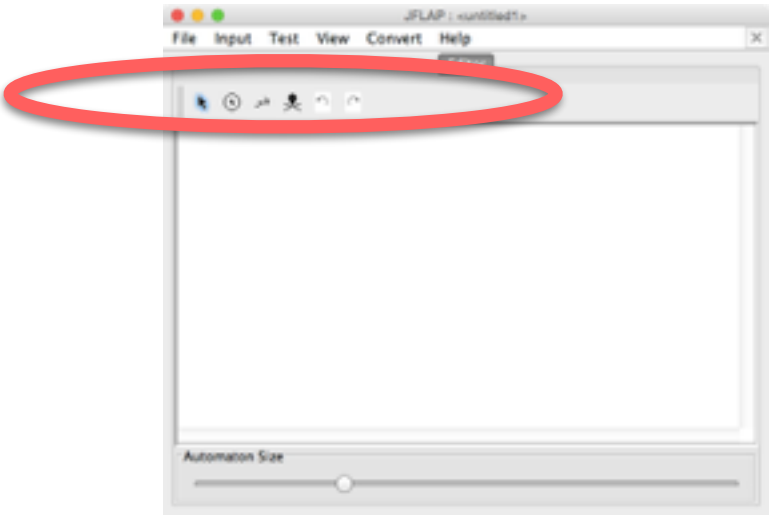
JFLAP is a simple tool for creating and evaluating finite state machine models. It can be downloaded as http://www.jflap.org. A full tutorial for JFLAP is at http://www.jflap.org/tutorial/

Here are some highlights to get you started. We will start by just creating a two node state machine with actions **a** and **b** to move between them, and where the states output RED and YELLOW.

Once you have started JFLAP, click on Moore Machine in the menu (the third entry). A Moore Machine is just a finite state machine where each state can output something.



It will open up an editor screen for you to draw new finite state machines.

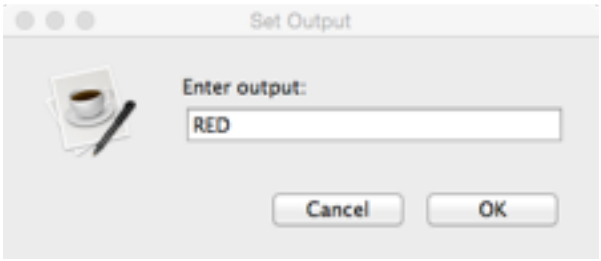


Click on the menu bar (circled in red) to change mode between
- editing states already drawn
- adding states
- adding transitions between states
- deleting things

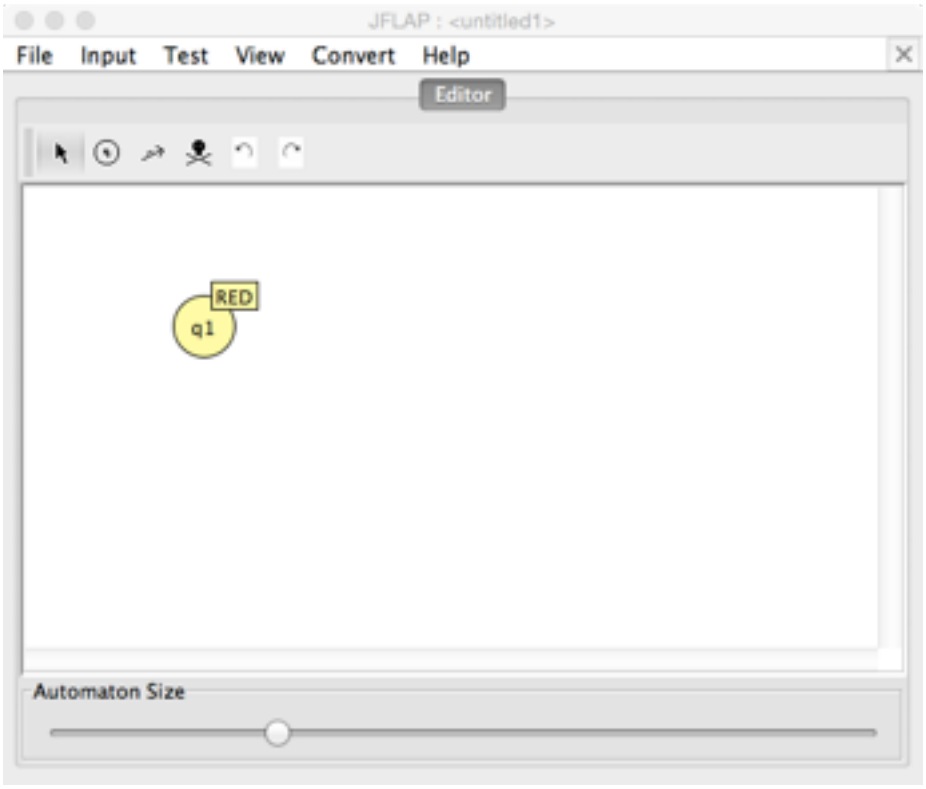
Remember to click back on the appropriate icon when you want to change action (eg going from deleting a node to adding a new node)

Adding States
You can add a new state to your finite state machine by clicking on the picture of a state (a circle) in the menu bar and then clicking in the editor at the point you want it to appear.

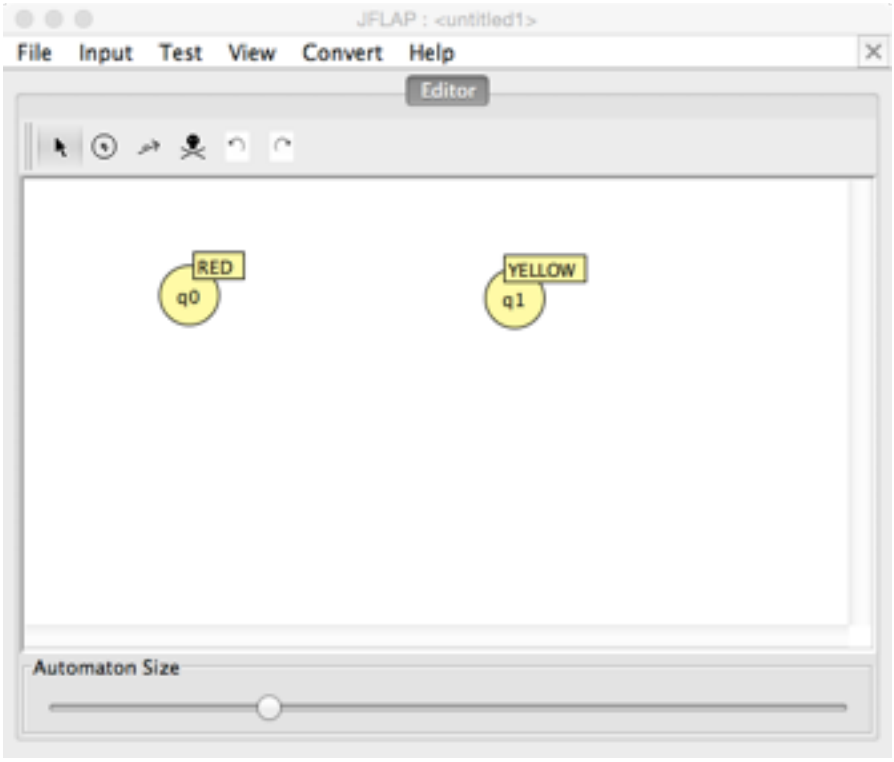
A popup box will appear asking you to “Enter Output”.



Type in whatever you want to be output whenever the machine enters that state. Then click OK. For example, our first state will output RED, so we type in “RED “ in to the dialogue box that appears as above. We get:



If you keep clicking in the window, while the circle in the menu is selected, more states will appear. Create a second state that outputs YELLOW in the same way.



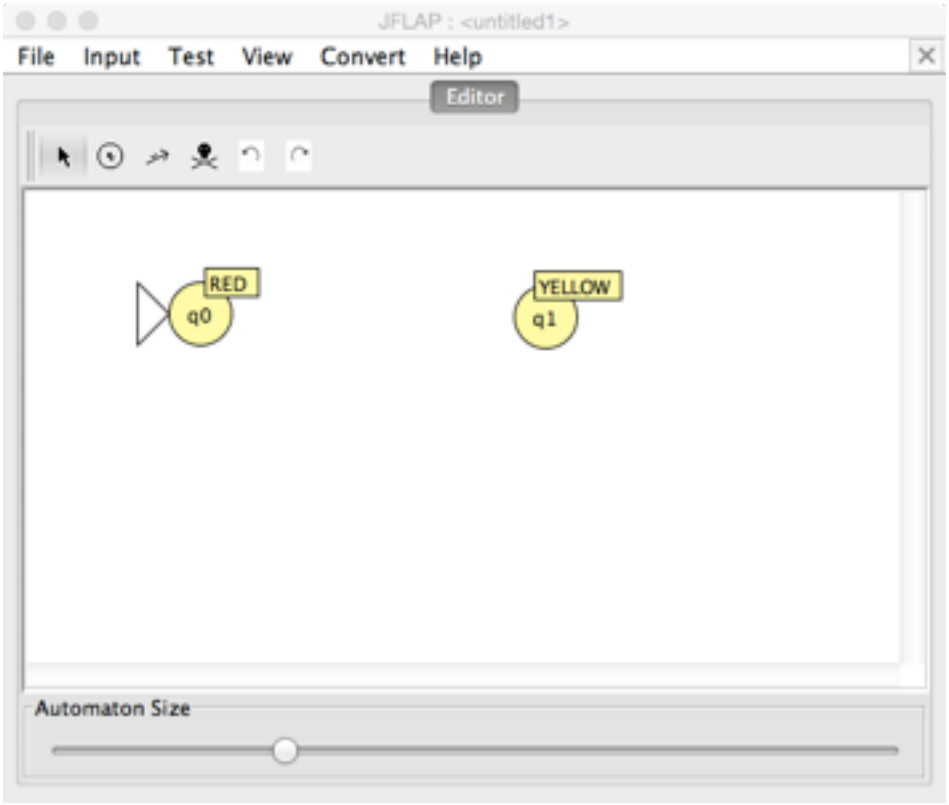
Editing the output of a state
To change the output of a state, select the arrow icon in the menu bar. Then click on the state you want to change. The “Enter Output” popup box will reappear allowing you to type in something new as the output of that state.

In this mode you can also move nodes around on the screen, and change other information.

Deleting a state (or transition)
To delete a state or transition, select the skull and crossbones icon in the menu bar. Then click on the thing you want to delete.

Making a state the initial state
To make a state the initial state (the place the machine starts), you also click on the arrow icon at the left of the menu bar. Then while holding down the **crtl** key on the keyboard (this is the equivalent to right clicking on a PC), click and hold on the state you want to change. A drop down menu appears. Click on **initial.**

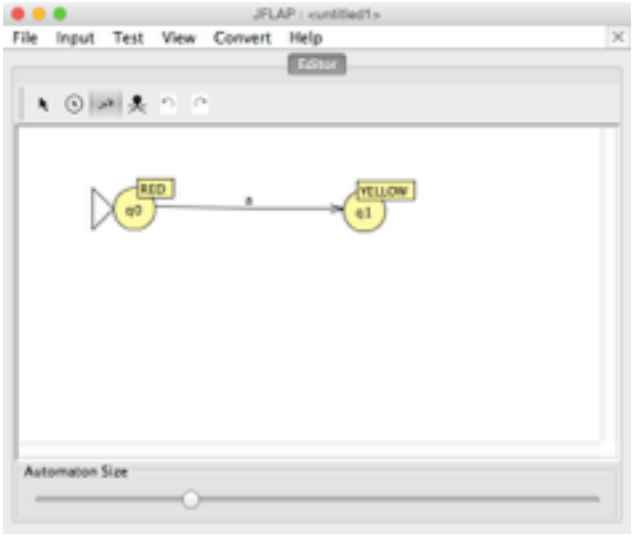
A triangle appears on the edge of the state showing that this is the initial state**.** We will make state q0 the initial state.



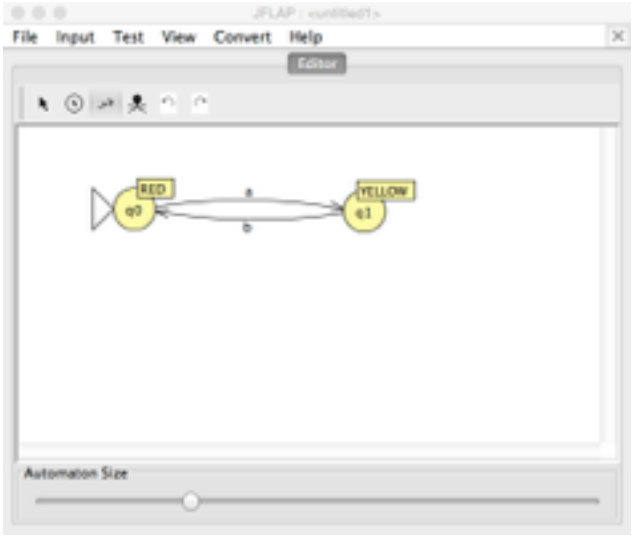
Adding a transition (an arrow) between states
To add a transition, select the lighter arrow with the dash above it from the menu bar.

Click and hold on the start state for the transition, then drag the arrow to the end state of the transition before releasing the button.

Type in the name of the action needed to take that transition into the box that appears and hit the return key. We will make a transition with action a that takes you from state q0 (marked as the initial state) to state q1. We click on q0 and then drag the mouse to q1.



To make a transition go the other way on action b, click on q1 and drag the mouse to q0 before releasing. Then type in b as the action into the box and hit return.

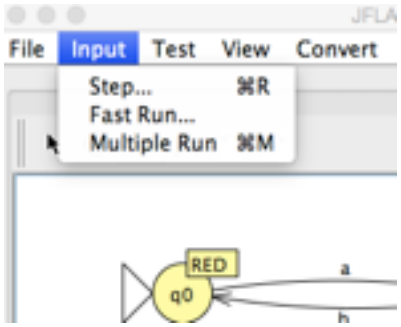


You can also make a transition go from a state to itself just by clicking on the node when in this mode. When that action is done the state does not change.

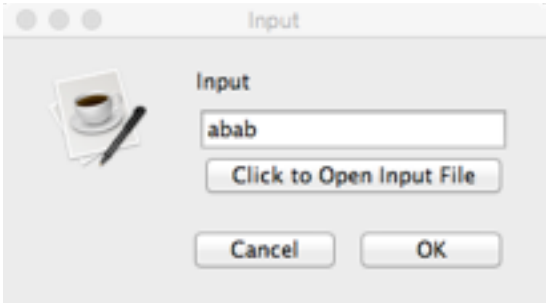
JFLAP: Running Finite State Machine Models

To Simulate your model

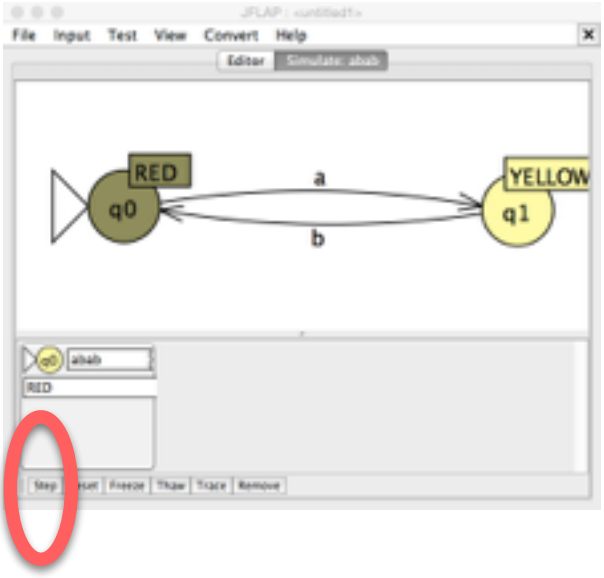
Click on Input and then Step in the drop down menu.



Type in a series of input actions you want to test then click OK. For example we could type in abab to indicate we want to see what happens when the sequence of inputs when we simulate our model are **a** then **b** then **a** then **b**.

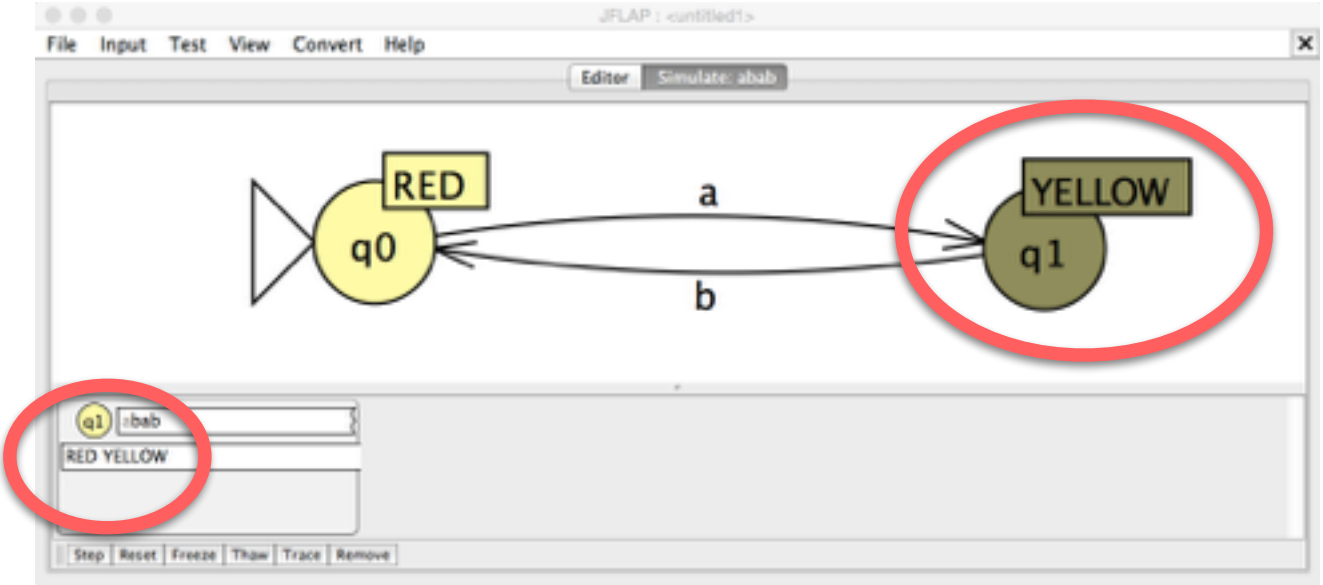


This will open a simulation window.



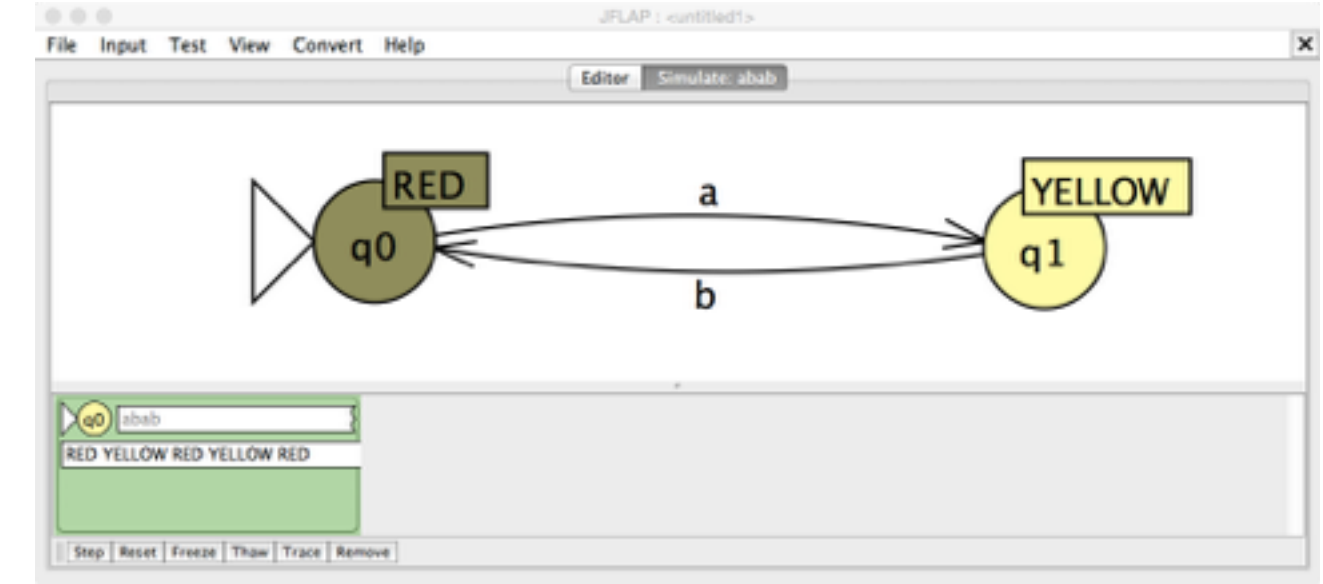
Click on the Step button at the bottom left of the screen to see what is output as the first of those those actions are completed.

The sequence of outputs are then shown in the window at the bottom



The shaded state shows the state that the machine is in. In our case the machine started in state, q0, outputting RED and moved to state, q1, where it output YELLOW, so the output sequence it shows so far after doing an action a (the first in the list we gave it as the input sequence) is RED YELLOW.

Every time you click STEP it takes another input from your sequence and then changes state. When the sequence has finished (if it was a valid sequence of inputs) the box at the bottom turns green to show it has finished.



To see what happens for different input sequences you can choose Input…Step again to try a different sequence. Click on the X (top right corner) once for each simulation to close the windows and return to editing.

Experiment, create and simulate your own models in JFLAP

Now you have the basics, play around with JFLAP. Try running different input sequences including ones that won't work like **aa** for the above and see what happens.

Try extending the state machine with more states and transitions and see what happens.

Once you have the hang of it, create a finite state machine of one of the things we've talked about

- The Knights Tour Puzzle
- The Tour Guide underground
- The Bridges of Königsberg
- The hexahexaflexagon
- An alarm clock

Once you have created it you will have a model of it - a model of a puzzle or of something in the world. For example, a model of the Tour Guide Map can be used to plot other routes from the hotel around the city. Graph models like these are essentially what sit underneath things like the Transport for London website's route planner program.

Perhaps then create models of other simple gadgets from around your house that involve moving between states, like a digital watch or of websites or other programs you use like the states in a game program.

With more sophisticated finite state machine programs like **PVSio-web** (http://www.pvsioweb.org) you can actually go a step further and create working prototypes from the finite state machines by linking them to pictures of the gadget or map being modelled. It was created to help design and test medical devices like infusion pumps that give people drugs in hospital or diabetes monitors.

Use of this booklet
This booklet was created by Paul Curzon of Queen Mary University of London, cs4fn (Computer Science for Fun www.cs4fn.org) and Teaching London Computing (teachinglondoncomputing.org) for use to support a workshop about graphs and finite state machines with support from EPSRC via CHI+MED (EPSRC EP/G059063/1).

See the Teaching London Computing activity sheets in the Resources for Teachers Section of our website (http://teachinglondoncomputing.org/resources/inspiring-unplugged-classroom-activities/) for linked activities including The Knight's Tour and The Tour Guide Activities.