

ECE374

Fall 2020

JFLAP Practice

Name: Zhang Yichi

Sept. 26th 2020

1 Introduction

JFLAP is a software that helps us build and manipulate finite automata (FA), deterministic finite-state automata (DFA) and nondeterministic finite-state automaton (NFA). This report is about the introduction to JFLAP.

2 Formal Definition of Finite Automata

An FA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$

Q is a function called the States

Σ is a finite set called the alphabet

$\delta : Q \times \Sigma \rightarrow Q$ is the transition function

q_0 is the start state

F is the set of accept states or final states

3 Construct and Run

3.1 Creating States

By selecting the State Creator on the tool bar, we can create states by we clicking on th canvas.

3.2 Defining Initial and Final States

Select the Attribute Editor on the tool bar. Right clicking the states, we can set the states as initial states or final states. Initial states are denoted by a triangular next to it. Final states are denoted by a ring around it.

3.3 Creating Transitions

Select the Transition Creator on the tool bar. Clicking the state can create transitions to itself. Draging between states with mouse can generate transitions between states. Type the inputs we use. If there you do not type anything, it will automatically create a transition with input empty string.

3.4 Deleting States and Transitions

Select the Deleter on the tool bar. Click and delete anything that is not desired.

3.5 Running the FA on Multiple Strings

Select Input: Multiple Run from the menu bar. Type the input we want to test and click Run Inputs. Then we can get the results whether they are accepted or rejected.

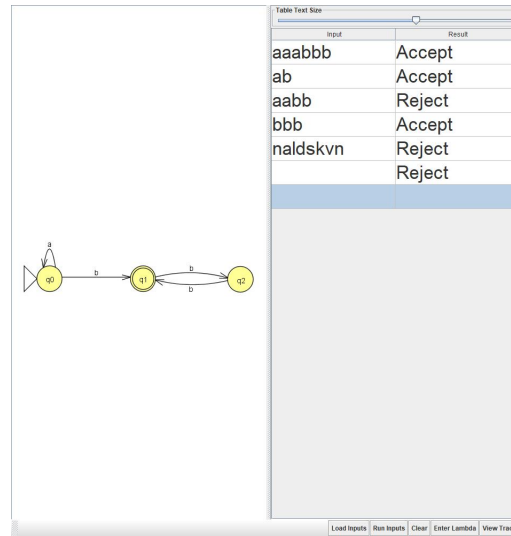


Figure 1: Test FA

As the figure shows, a^* plus an odd number of b is accepted. Nonexpecting, invalid and empty inputs will be rejected.

3.6 Building a Nondeterministic Finite Automaton

When creating transitions, if there is multiple transitions or empty string transitions, the FA will be considered as NFA.

3.7 Highlighting Nondeterministic States

Select Test: Highlight Nondeterminism to highlight the nondeterministic states.

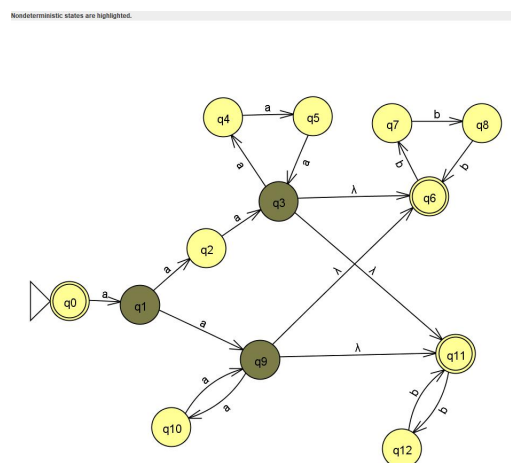


Figure 2: NFA

3.8 Running Input on an NFA

Select Input: Step with Closure and type the input we want to test to run input on the NFA.

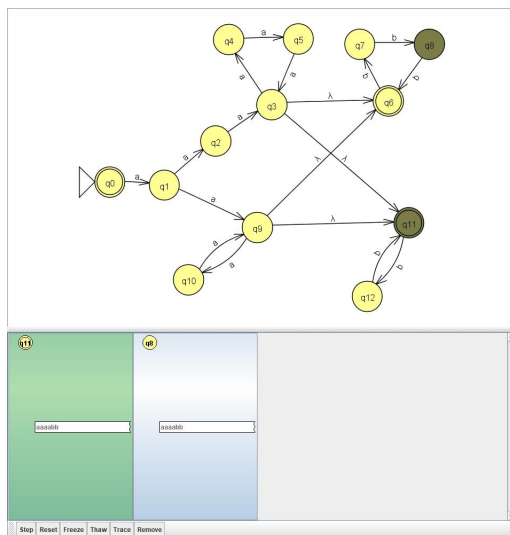


Figure 3: run input "aaaabb"

aaaabb is accepted.

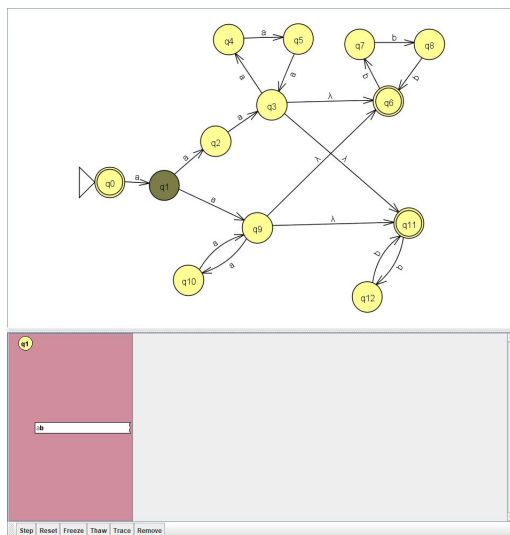


Figure 4: run input "ab"

ab is rejected.

3.9 Producing a Trace

Click Trace to produce a traceback from the start state to the current state.

3.10 Removing Configurations

Click Remove to remove the configurations taht we do not want to test further.

3.11 Freezing and Thawing configurations

Click Freeze to stop certain configurations to go further. Click Thaw to continue.

3.12 Resetting the simulator

Click Reset to start over.

4 Manipulating Transitions

Select the Attribute Editor on the tool bar. Click again to change or add the inputs. Drag to manipulate the transition line.

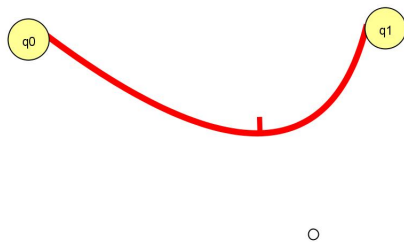


Figure 5: manipulate transition

5 Add a Trap State to DFA

Select Convert: Add Trap State to DFA on the menu bar. Each states have to have outputs for all the inputs but sometimes some inputs are invalid for certain states so we create a trap state to be the output of all these kinds of inputs. Once the DFA enters a trap state, it means the input string is rejected. See the example below.

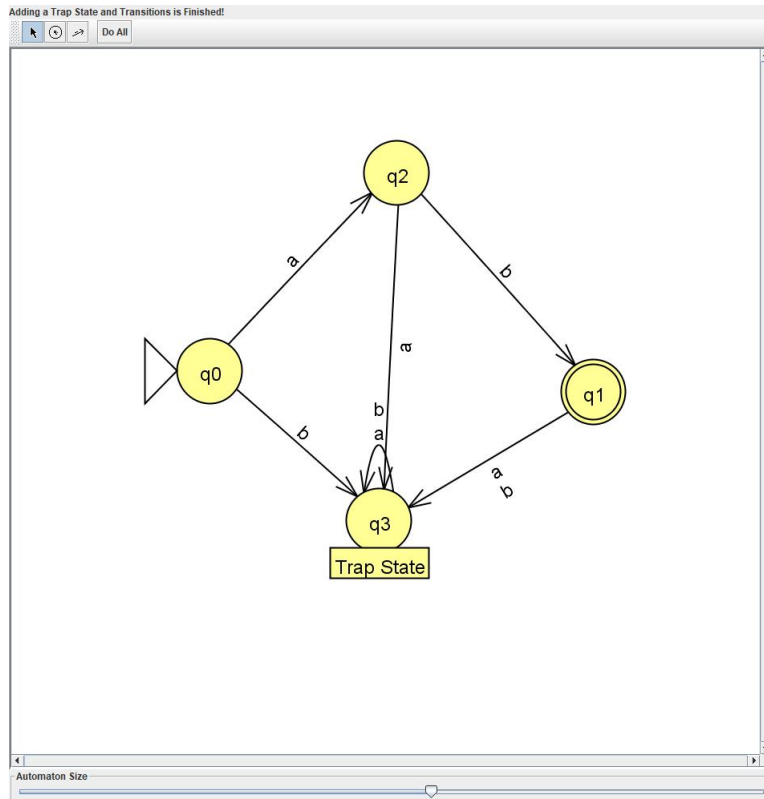


Figure 6: add a trap state

6 Converting an NFA to a DFA

Select Convert: Convert to DFA on the menu bar to convert an NFA to a DFA. The software automatically combines transitions with same inputs but different outputs of a state in an NFA and combines the states connected by transitions with an empty string. After the operations, the software will re-name all the states and generate a new DFA. See the example below.

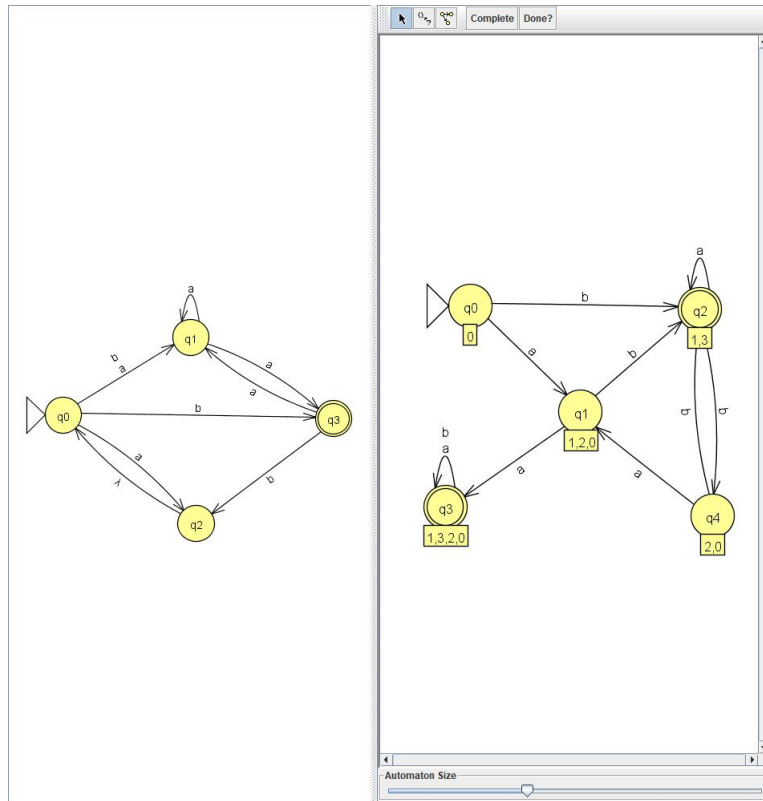


Figure 7: NFA to DFA

7 Converting a DFA to a Minimal State DFA

Select Convert: Minimize DFA on the menu bar to minimize a DFA. The software will automatically generate a tree of state from the state diagram. Select the Nonfinal node of the tree and click Complete Subtree to get a complete tree with which we can build a simplified state diagram. Then click Finish to generate a new re-named state diagram. See the example below.

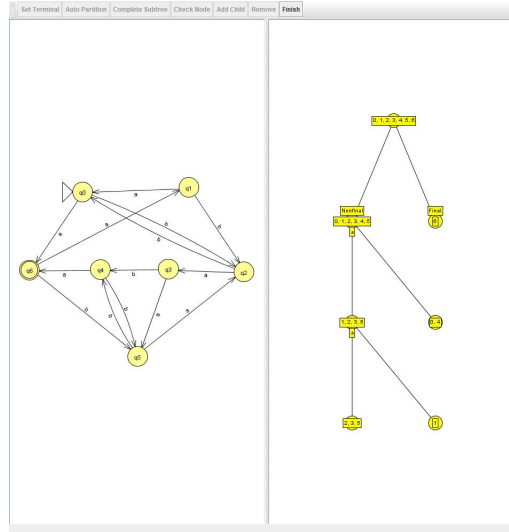


Figure 8: get a complete tree

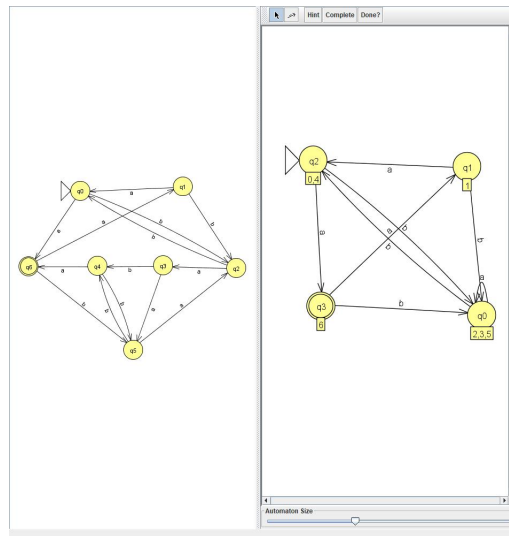


Figure 9: minimize a DFA

8 Convert FA to Grammar

Select Convert: Convert to Grammar on the menu bar to convert the DFA into a regular grammar. With this function, the software will automatically generate all the transition function. See the example below. $S \rightarrow bA$ means state S will turn to state A with input b.

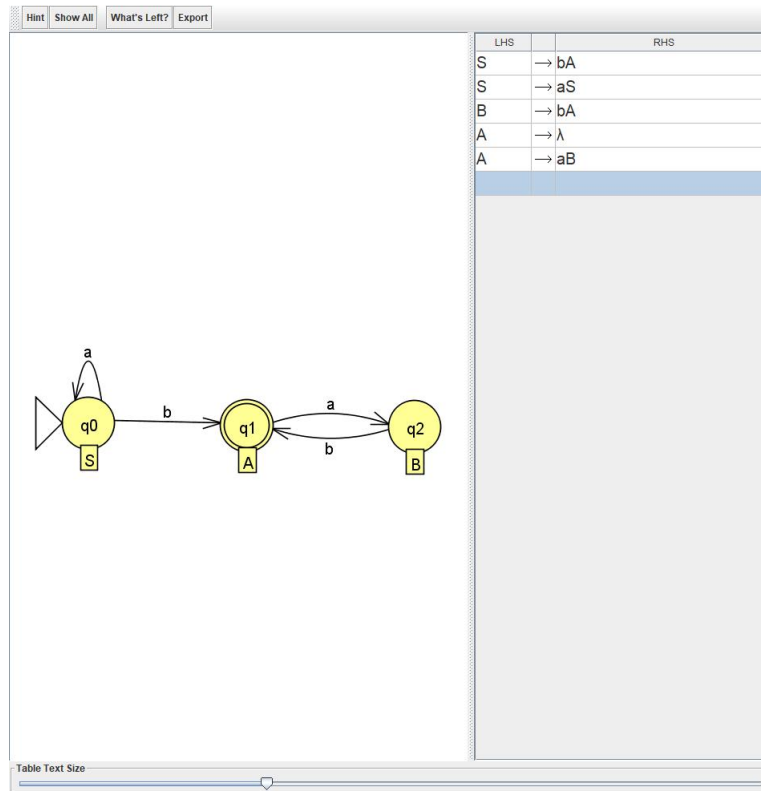


Figure 10: convert to grammar

9 Conclusion

This software can help us create DFA and NFA easily and highly increase our efficiency. I have gotten familiar with the functions introduced and am able to build DFA and NFA with JFLAP.