

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Innovations in Teaching Learning Process

Name	M. Banu Priya, D.Vidhya & S.Saranya
Subject code & Title	20CS410 – Automata and Compiler Design
Academic year & Semester	2023-2024 V

Online Platforms and Tools

JFLAP Tool using for Finite Automata and Regular Expression

Objectives: Our aim is to create a JFLAP tool that empowers users by JFLAP program makes it possible to create and simulate automata.

Description:

JFLAP program makes it possible to create and simulate automata. Learning about automata with pen and paper can be difficult, time consuming and error-prone. With JFLAP we can create automata of different types and it is easy to change them as we want. JFLAP supports creation of DFA and NFA, Regular Expressions, PDA, Turing Machines, Grammars and more.

Setup

JFLAP is available from the homepage: (www.JFLAP.org). From there press "Get FLAP" and follow the instructions. You will notice that JFLAP have a .JAR extension. This means that you need Java to run JFLAP. With Java correctly installed you can simply select the program to run it. You can also use a command console run it from the files current directory with, *Java -jar JFLAP.jar*.

Using JFLAP

When you first start JFLAP you will see a small menu with a selection of eleven different automata and rule sets. Choosing one of them will open the editor where

you create chosen type of automata. Usually you can create automata containing states and transitions but there is also creation of Grammar and Regular Expression which is made with a text editor.

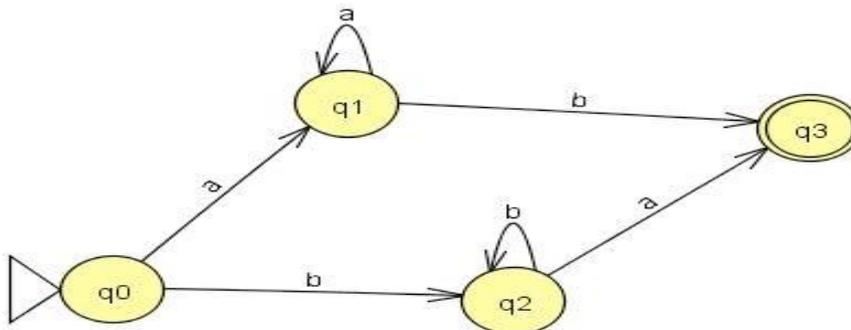
Outcomes:

- Provides users with JFLAP is an automata simulator that supports DFA/NFA, PDA, Turing machines and more.
- JFLAP also supports Regular Expressions and Grammar which can be converted to an automaton and back. JFLAP has been used worldwide on many automata theory courses and is still under development.

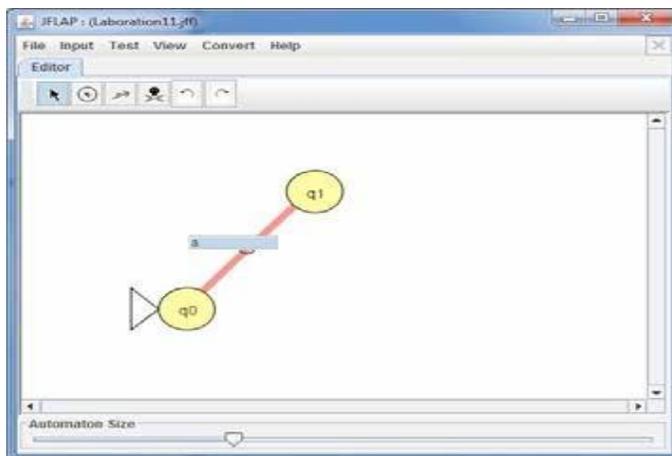
Sample:

REGULAR LANGUAGES AND FINITE STATE AUTOMATA

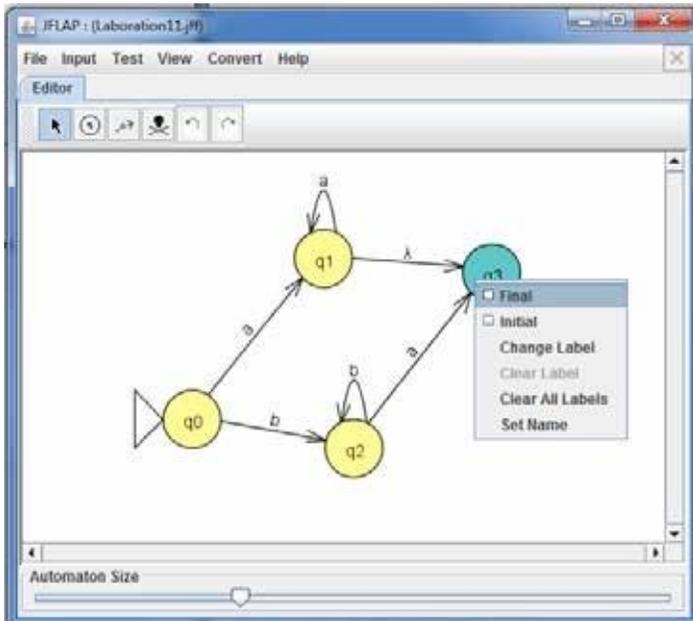
Create this Deterministic Finite Automaton. First start JFLAP and select “Finite Automaton” as a new automaton. Use the toolbar to drag and drop states and transitions.



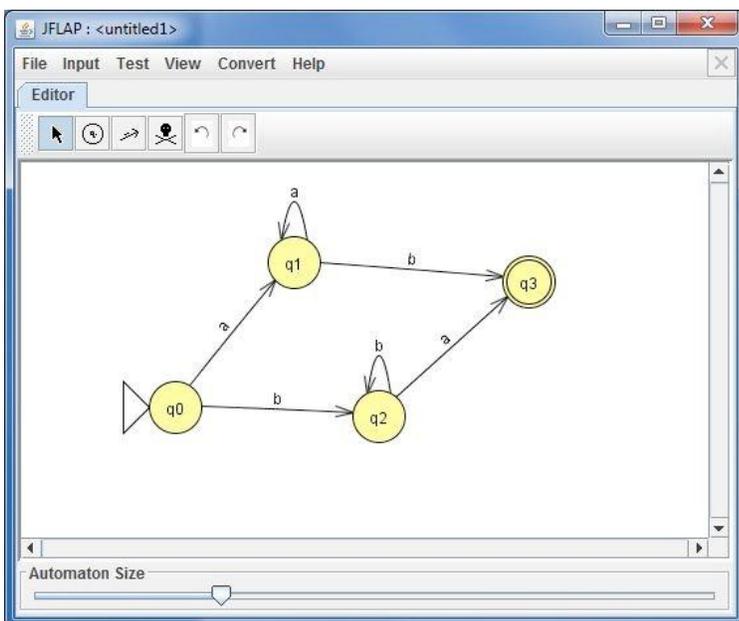
Select the transitions with the selection tool. Then type the input character for each transition.



Remember that initial and final states must be set to make the automaton run. State q_0 is set as the initial state which means the starting state. State q_3 will be the final state which is a state that should be active once the string is accepted and consumed. There can be more than one final state.

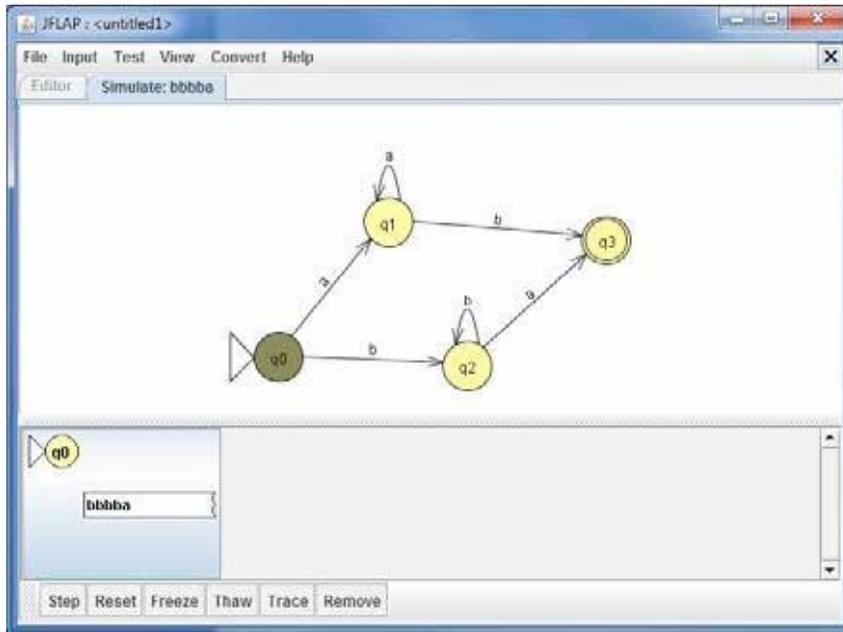


When you are done, the automaton should look like this.



To run this DFA select *Input* \rightarrow *Step By State*, then choose an input string to run the automata with You will see JFLAP's simulation view where you can debug the automaton

while it consumes the input string. Use the Step button to step through the string. Here the automaton has moved the active state and has consumed all the b characters.



Once the whole string is consumed the automata should park at the final state and the stack turns green. The string is accepted.

