**1. Given the alphabet {a,b,c}, construct a DFA to recognize any strings with length equal to 4. Solve the problem using JFLAP. Submit your final .jff file**

Strings of length 3n are supposed to be recognized where 'n' is a positive integer.

It means, the lenght of the strings in the language will be the multiples of which are greater than 0.

a,b,c a,b,c a,b,c P R S a,b,c

The DFA shown above is used for the same.

Here P, Q, T and S are the states in which P is the start state while S is the final state.

Strings which end up in the final state S are considered to be accepted.

If the strins end up in any other states, then they will be simply rejected by the DFA.

For example, if we take "abc" as the input then the string will start from state P and then to Q, then to R and finally to state S. Therefore the string will be accepted.

If we take "abbc". Here the flow will be like :

P\small \rightarrowQ\small \rightarrowR\small \rightarrowS\small \rightarrowQ

The string ends in state Q so the string will not be accepted.

2. Find the minimal dfa for L(((aa + b) ∗ + aba) ∗ ) using JFLAP. Submit the final .jff file and explain the steps. Step by Step

See answer

100% (6 ratings)

. Find the NFA for L(((aa + b)\* + ab\*a)\* ) using JFLAP. Convert the NFA to DFA using JFLAP. Submit the .jff files for both NFA and DFA and explain the steps of using JFAP

1. Enter ((aa+b)\*+aba)\* as the regular expression into the editor.

2. From the toolbar, select Convert > Convert to NFA

3. Click the “Do All” button to complete the conversion of the general transition graph to an NFA.

4. Click “Export” to see the NFA in a new window.

5. In the new window, from the toolbar select Convert > Convert to DFA

6. Click “Complete” to complete the conversion to DFA.

7. Click “Done?” to bring the completed DFA into a new window.

8. In the new window, from the toolbar select Convert > Minimize DFA

9. In the right-half of the window, select the root node of the tree.

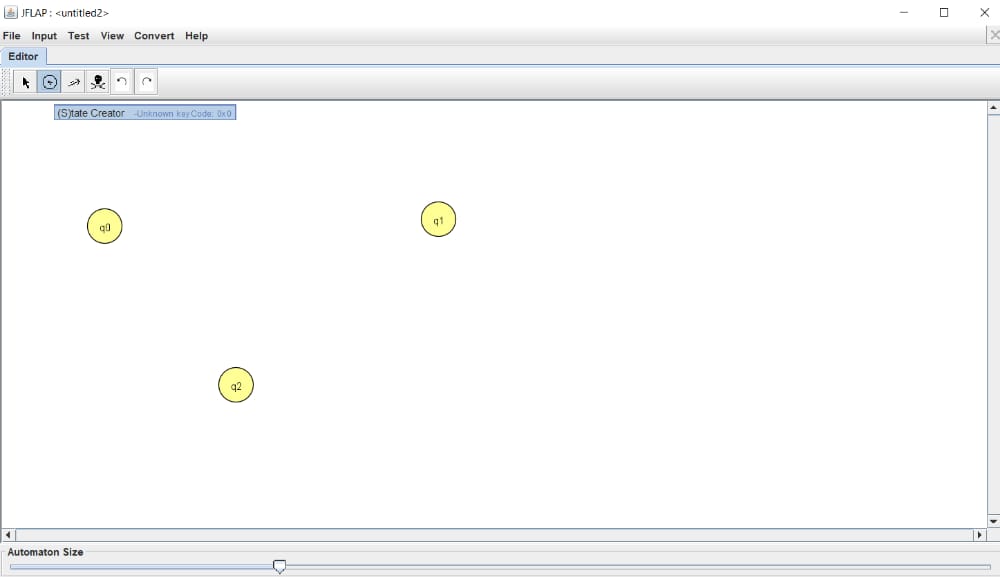
10. Click “Complete Subtree” to see the tree representation of the minimized DFA states.

11. Click “Finish” to see the minimized DFA without transitions.

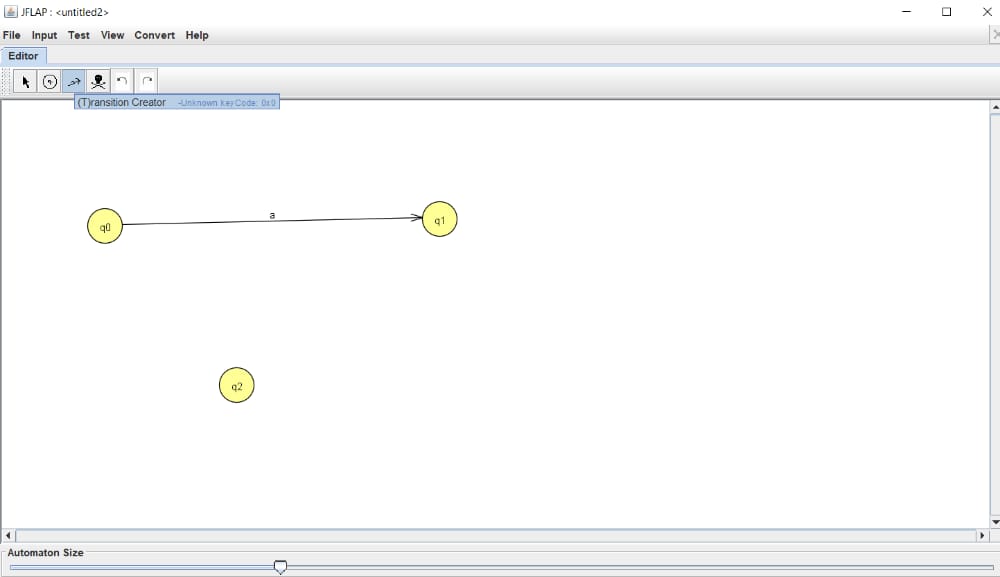
12. Click “Complete” to have JFLAP add the transitions to the DFA.

13. Click “Done?” and the complete minimized DFA for ((aa+b)\*+aba)\*) will be placed in a new window. 14. Finished!

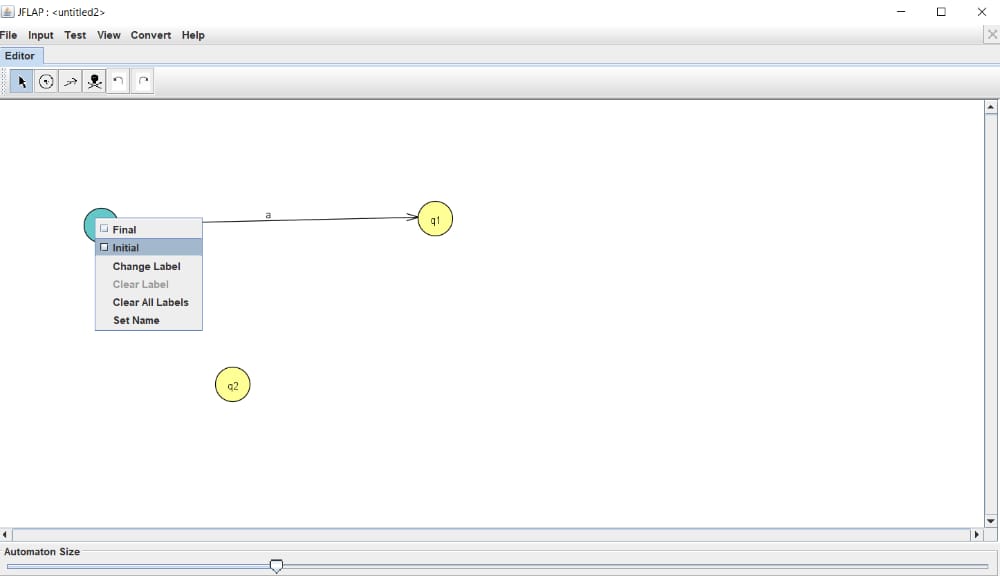
1.Click on create state and create 3 states



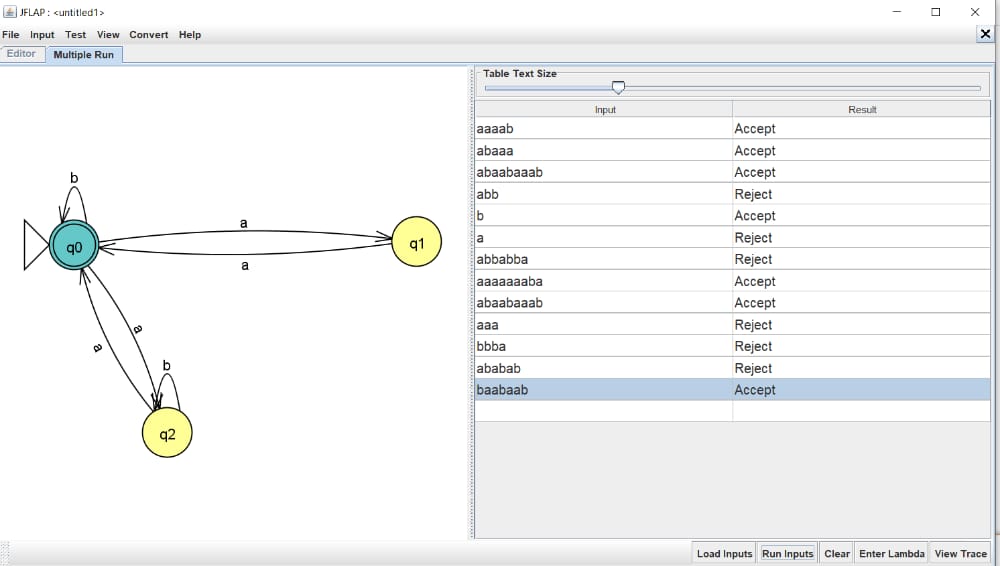
2.Click on create transition and add appropriate transition



3.Then make the initial state and the final state by right clicking on the state

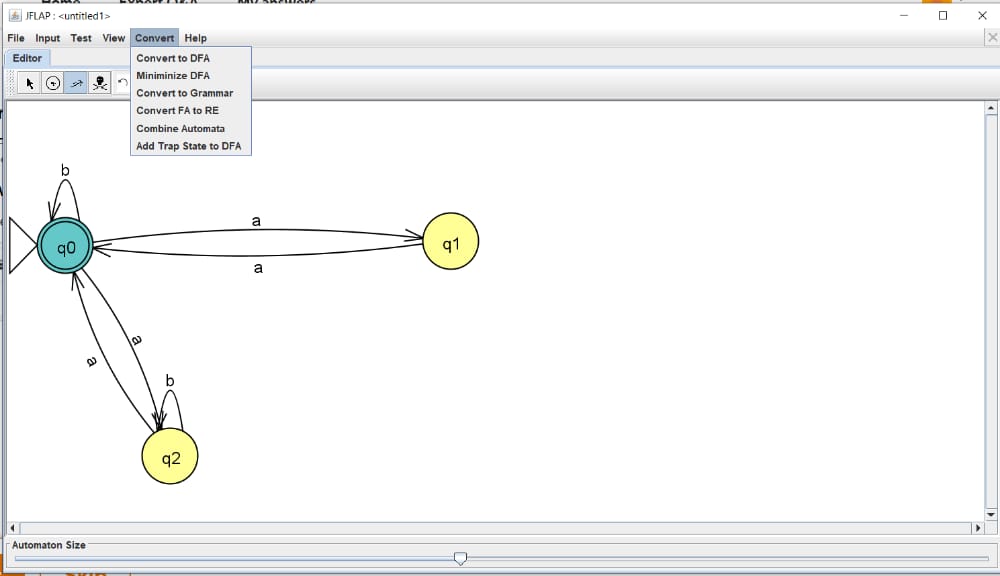


4.Here are the test cases to check whether our NFA is correct

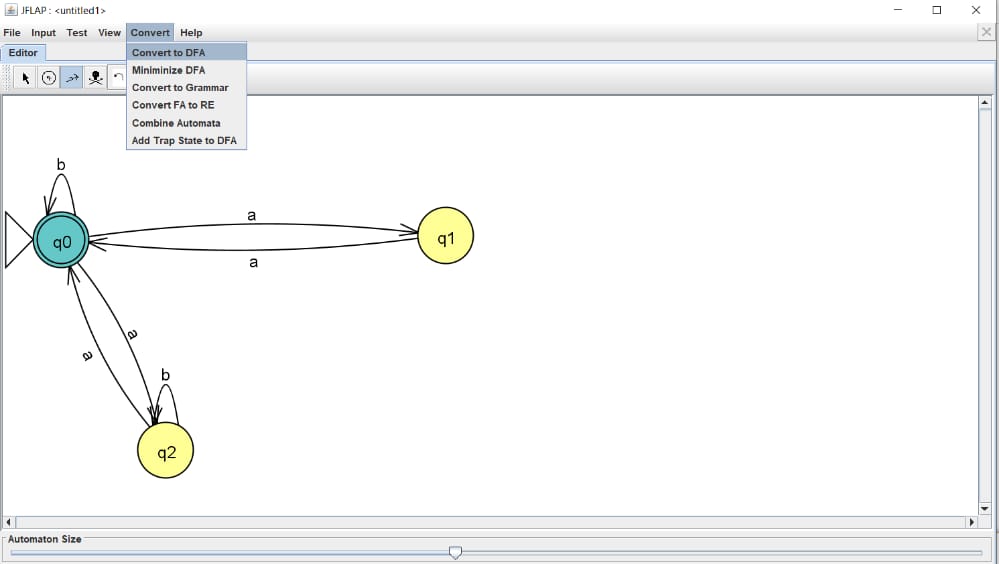


**These are the steps of using JFLAP to convert NFA to DFA**

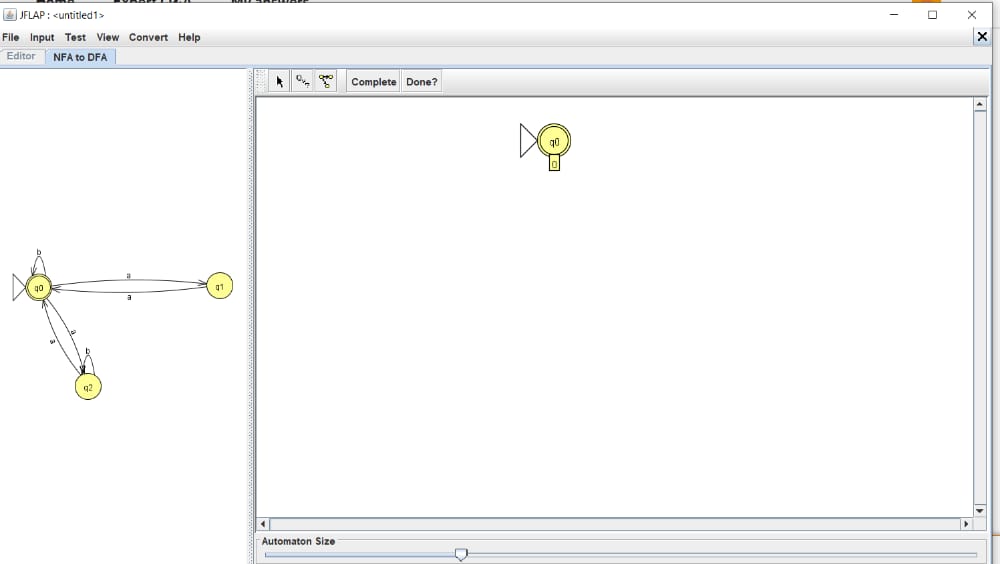
1.First click on convert



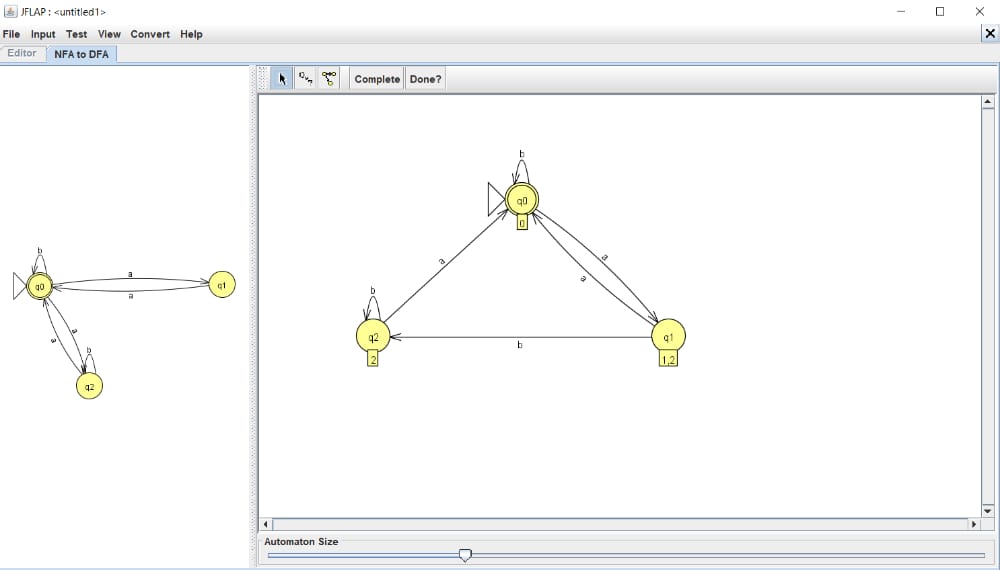
2.Then click to convert to DFA



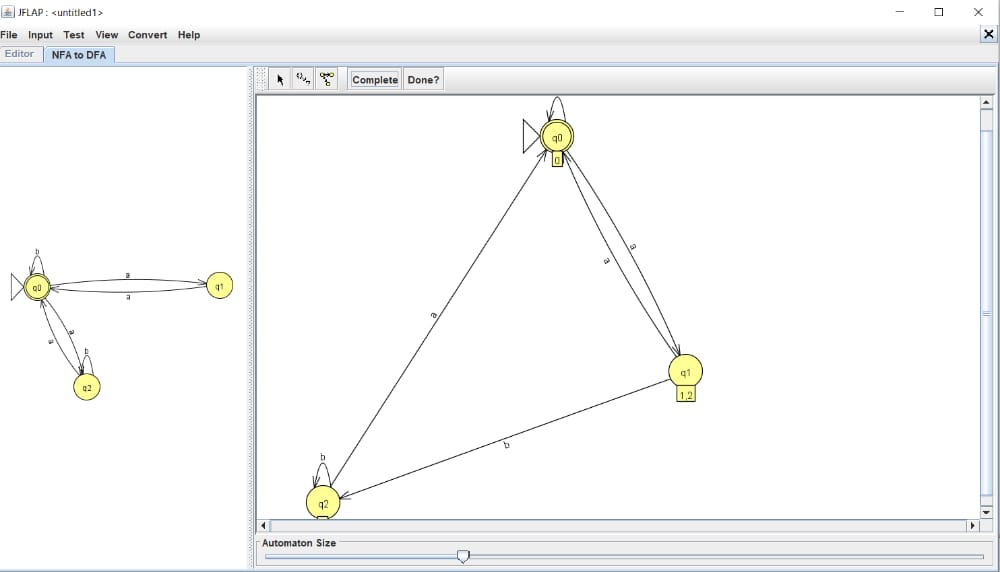
3.Then click on complete



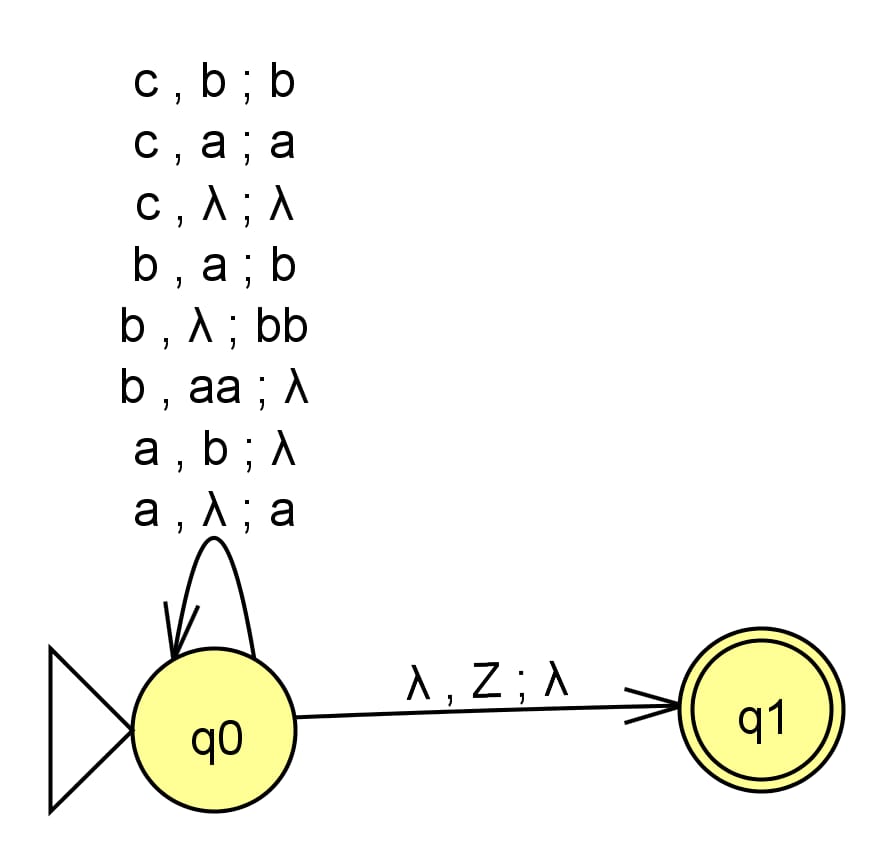
4.After clicking the complete your DFA will build using jflap

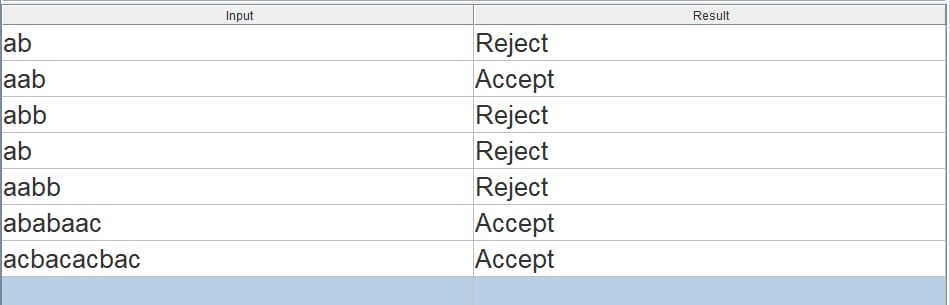


5.You can resize the image of the automata



3.





4.