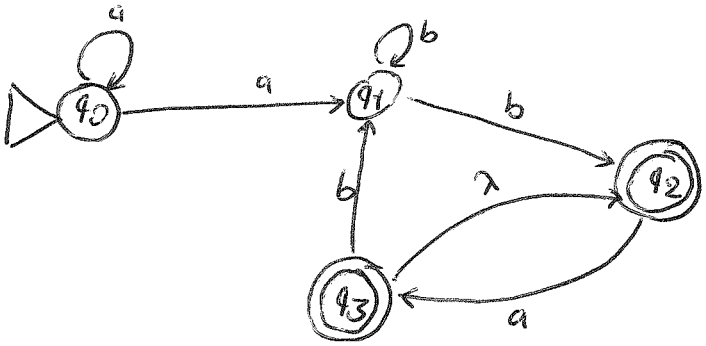
Introduction to JFLAP

- Introduction: download the program from. www.jflap.org

To run the program

- on windows, Mac: Double-click on JFLAP.jar
- On Linux, windows, Mac: java -jar JFLAP.jar

1) a) select the 'Finite Automaton' button and make the following NFA.



Tips:
- For making the initial/final state right-click on the state and select the appropriate option.
- To have a λ transition leave the transition without any lable.
- Save the automaton using 'Save As' from 'Input' menu.

b) Use each of the options in the 'Input' menu to check if following strings are accepted by the automaton:
a, ab, abbab

c) Convert the automaton to DFA.

d) Minimize the DFA.

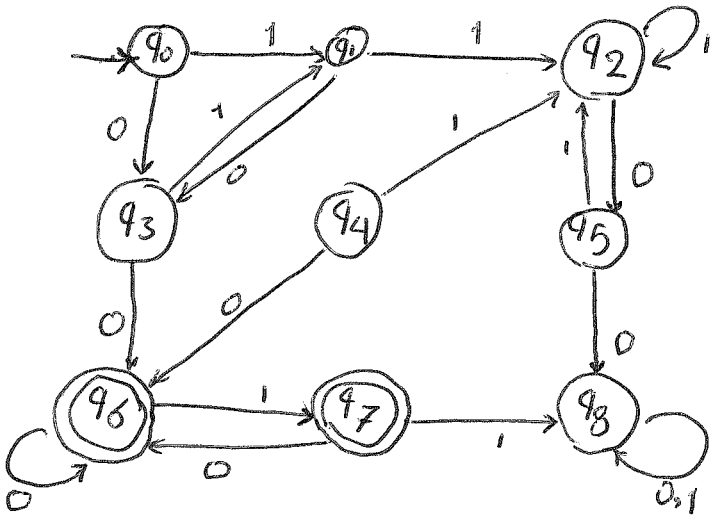
e) Check if the DFA is equivalent to the first NFA.

f) make the 'RE' which accepts the same language as the automaton in a) and d)

2) Convert the RE $(\lambda+a)b+b^*+cd$ to a NFA
use '!' instead of λ in the editor.

3) Create the following Automaton.

a)



b) Save it with two names A1 and A2. In A2 Change the initial state to q2. Check if two automaton are equal. In A1 Chang the Initial State to q5 and q8 and check if two automaton are equal? then what can you guess about the states of the minimized automata?

c) Use 'Multiple Run' from 'Input' Menu to check if A1 and A2 are really accepting the same set of strings. You can use following strings: {0, 000, 000100, 1110}

d) Minimize the automaton and check if the resulting DFA is equal to the first DFA.

e) Compute the equivalent regular expression.
:: attention: In this Program there should be only one final state which is non-initial. Moreover you should first add all the ∅ transitions between each two states which don't have a transition to the other (make the graph Complete!).

4) Convert $(a+\lambda)^*c + (dc) + d$ to a DFA