**Theme : Finite Automata**

1. Present the automaton in form of graph. Is this automaton deterministic or no? Why?

2. Convert a Finite Automaton to the Regular Grammar.

3. Transform nondeterministic finite automaton (NFA ) into a deterministic automaton (DFA). Present the DFA in form of graph.

4. Present 1 string based on the input alphabet ∑ that will be not accepted by the automaton. For showing use the Multiple Run option from menu Input from JFLAP.

5. For the finite automaton AF=(Q, ∑, δ, q0, F) present five strings that will be accepted. The length of strings should be more then n+2, where n is the number of stares from Q. For showing use the Multiple Run option from menu Input from JFLAP. Elaborate the program that for the obtained DFA will verify if string is accepted or no.

6. For each string x, present the configuration sequence for the acceptance of this string. To show the acceptance use the option Step with Closure and Fast Run from menu Input from JFLAP.

7. For each obrtained 5 strings build the decompozition x=uvw , using the pumping lemma.

8.Obtained DFA minimize.

**AF=(Q, ∑, δ, q0, F),**

**Q = { q0, q1, q2, q3 },**

**∑ = { a, b, c }, F = { q3}.**

**δ (q0, a ) = q0 ,**

**δ (q0, a ) = q1 ,**

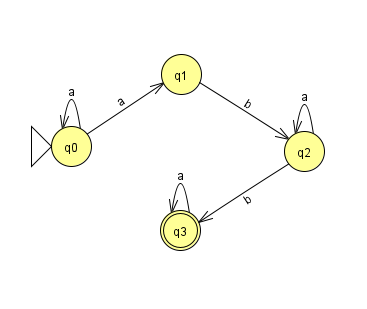
**δ (q1, b ) = q2 ,**

**δ (q2, a ) = q2,**

**δ (q3, a ) = q3,**

**δ (q2, b) = q3.**

**1)**



This automaton is not deterministic because we have the node q0 which has two paths with ’a’, one to q0 and another to q1.

**2)**

q0 = S, q1 = A, q2 = B, q3 = C

S -> aA | aS

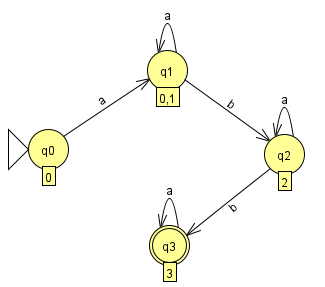
A -> bB

B -> aB | bC

C -> aC | empty

**3)**

**DFA**

****