## Construct an Equivalent DFA from NFA.

Every DFA is an NFA, it is clear that the class of languages accepted by NFA's includes the languages accepted by DFA's.

To convert NFA to DFA we can carry out the following algorithm:

1- begin with **{q$_0$},** start state, and calculate $\delta(\{q0\},a)$ for all **a** in Σ this gives a number of new states **Q⁻.**

2- For each new states **Q⁻**, we again calculate $\delta(Q-,a)$ for all **a** in Σ and introduce new states if necessary.

3- Repeat step2 until there are no new states.

4- Final states of new DFA are the states that contain any final state of the previous NFA.

<u>**Example:**</u>

Convert the following NFSA into equivalent FSA
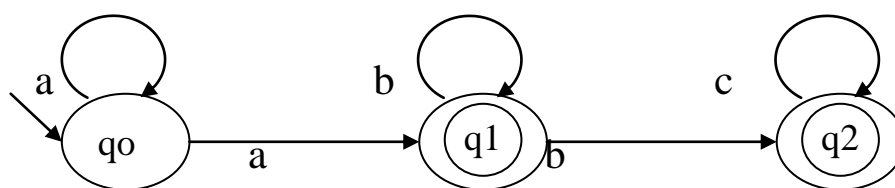
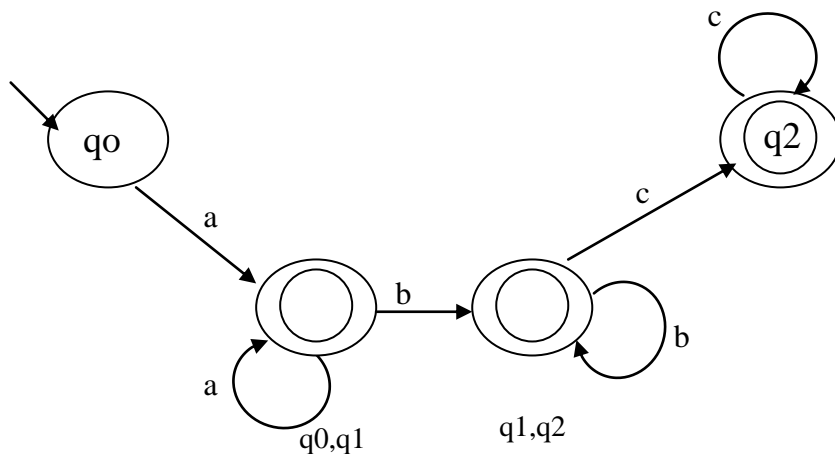$M = (Q, \Sigma, q_0, \delta, F)$

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b, c\}$

$q_0 = \{q_0\}$

$F = \{q_1, q_2\}$

$M⁻ = (Q⁻, \Sigma, q_0, \delta⁻, F⁻)$

1- δ(q0,a)→{q0,q1}

2- δ ({q0,q1} , a)→ {q0,q1}

3- δ ({q0,q1} , b) → {q1,q2}

4- δ ({q1,q2} , b) → {q1,q2}

5- δ ({q1,q2} , c) → {q2}

6- δ ({q2} , c) → {q2}

- **Convert NFSA with empty move into NFSA without empty move.**

If there is NFSA with empty move $M = (Q, \sum, q_0 , t , F)$ then there is SA without empty move $M^- = (Q^- , \sum, q_0 , t^- , F^-)$

Define $t : Q \times (\sum \cup \{\varepsilon\}) \to Q$

By
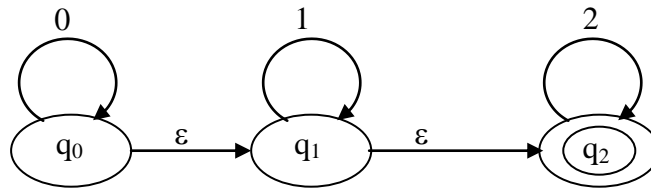
t(K,ε)=R(K) or ε-closure(K)

t(K,a)=R(t(R(K),a))

The set of final states F^- is $F \cup \{q\}$ if $R(q) \in F$

Example:

Convert the NFSA with empty move into without empty move.

$t(q_0,0)=R(t(R(q_0),0))=R(t(\{q_0,q_1,q_2\},0))=\{q_0,q_1,q_2\}$

$t(q_0,1)=R(t(R(q_0),1))= R(t(\{q_0,q_1,q_2\},1))=\{q_1,q_2\}$

$t(q_0,2)= R(t(R(q_0),2))= R(t(\{q_0,q_1,q_2\},2))=\{q_2\}$

$t(q_1,0)=R(t(R(q_1),0))=R(t(\{q_1,q_2\},0))=\{\}$

$t(q_1,1)=R(t(R(q_1),1))= R(t(\{q_1,q_2\},1))=\{q_1,q_2\}$

$t(q_1,2)= R(t(R(q_1),2))= R(t(\{q_1,q_2\},2))=\{q_2\}$

$t(q_2,0)=R(t(R(q_2),0))=R(t(\{q_2\},0))=\{\}$

$t(q_2,1)=R(t(R(q_2),1))= R(t(\{q_2\},1))=\{\}$

$t(q_2,2)= R(t(R(q_2),2))= R(t(\{q_2\},2))=\{q_2\}$

$F^-=\{q_0,q_1,q_2\}$