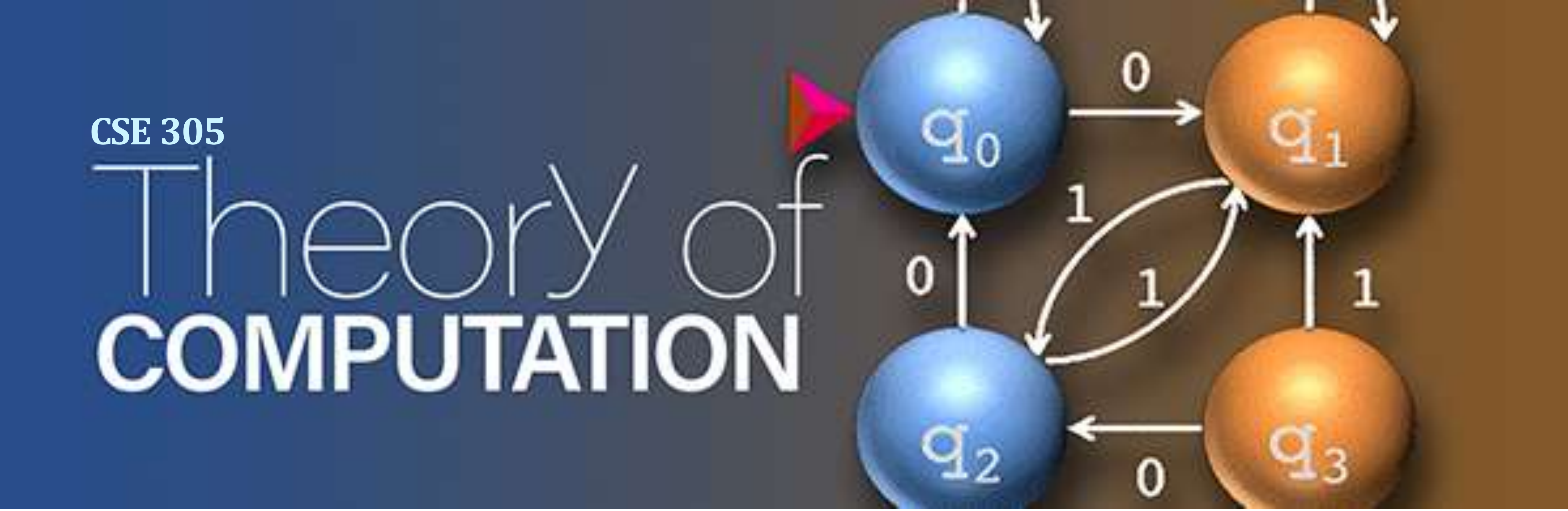


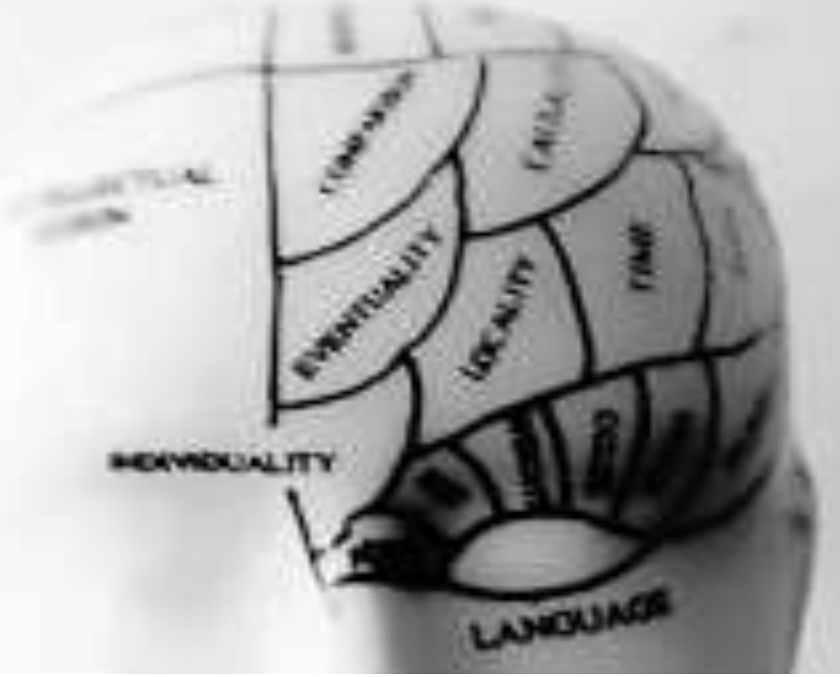

# Lecture 17

# Finite-State Automata (4)

**Md. Mijanur Rahman, Prof. Dr.**

Dept. of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh.

www.mijanrahman.com




# Contents

Finite State Automata

# NFA (Non-Deterministic Finite Automata)

- **NFA stands for non-deterministic finite automata. It is easy to construct an NFA than DFA for a given regular language.**
- The finite automata are called NFA when there exist many paths for specific input from the current state to the next state.
- Every NFA is not DFA, but each NFA can be translated into DFA.
- **NFA is defined in the same way as DFA but with the following two exceptions, it contains multiple next states, and it contains ε transition.**
- In the diagram, we can see that from state q0 for input a, there are two next states q1 and q2, similarly, from q0 for input b, the next states are q0 and q1. Thus it is not fixed or determined that with a particular input where to go next. Hence this FA is called non-deterministic finite automata.

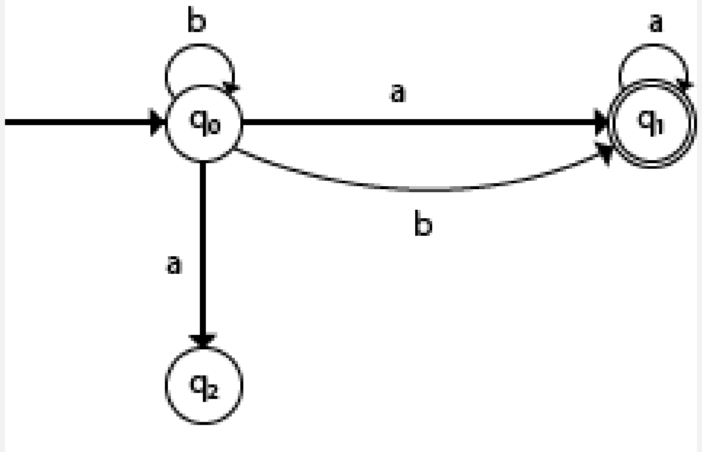


Fig:- NDFA

# Formal Definition of NFA

- **NFA also has five states same as DFA, but with different transition function,** as shown follows:

$$\delta: Q \times \Sigma \rightarrow 2^Q$$

where,

Q: finite set of states

∑: finite set of the input symbol

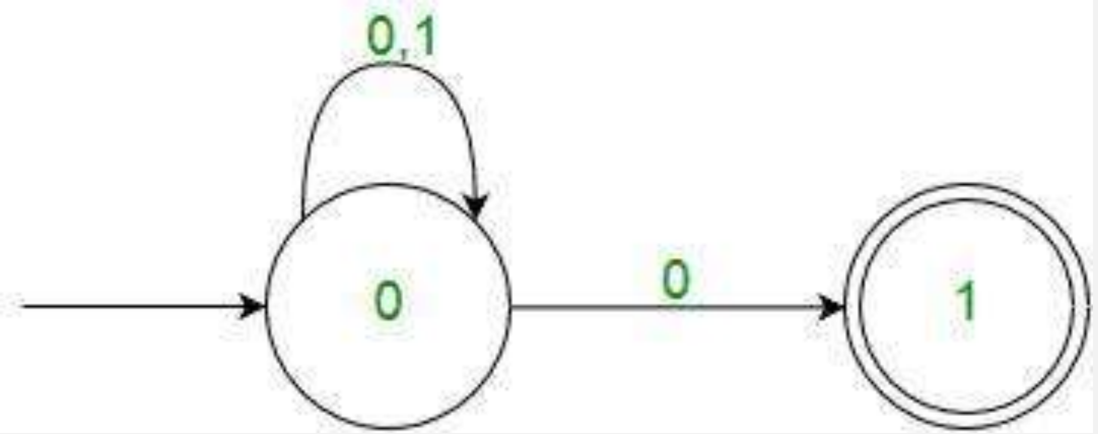
q0: initial state

F: **final** state

δ: Transition function

# Graphical Representation of an NFA

- **An NFA can be represented by digraphs called state diagram.** In which:
  - The state is represented by vertices.
  - The arc labeled with an input character show the transitions.
  - The initial state is marked with an arrow.
  - The final state is denoted by the double circle.



# Graphical Representation of an NFA

**Example-1: An NFA shown in the figure is defined by:**

Q = {q0, q1, q2}

∑ = {0, 1}

q0 = {q0}

F = {q2}

Transition diagram:

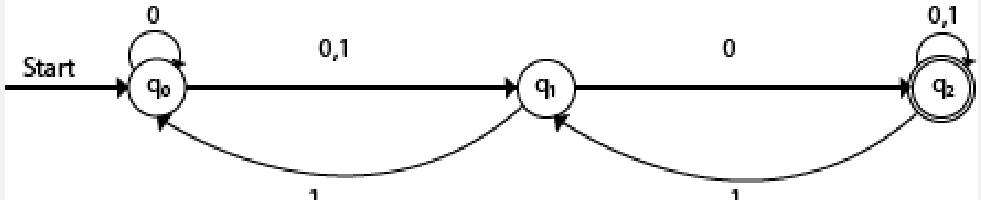



Fig: NFA

Transition Table:

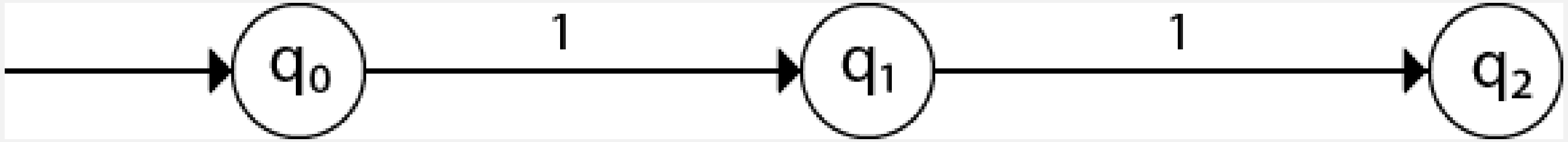
| Present State | Next state for Input 0 | Next State of Input 1 |
|---|---|---|
| →q0 | q0, q1 | q1 |
| q1 | q2 | q0 |
| *q2 | q2 | q1, q2 |

- In the above diagram, we can see that when the current state is q0, on input 0, the next state will be q0 or q1, and on 1 input the next state will be q1. When the current state is q1, on input 0 the next state will be q2 and on 1 input, the next state will be q0. When the current state is q2, on 0 input the next state is q2, and on 1 input the next state will be q1 or q2.

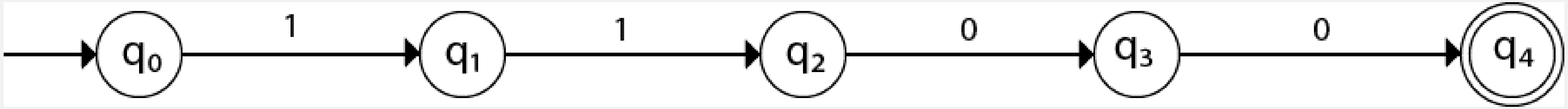
# Examples of NFA

**Example-2:** Design an NFA with ∑ = {0, 1} in which double '1' is followed by double '0'.

- The FA with double 1 is as follows:

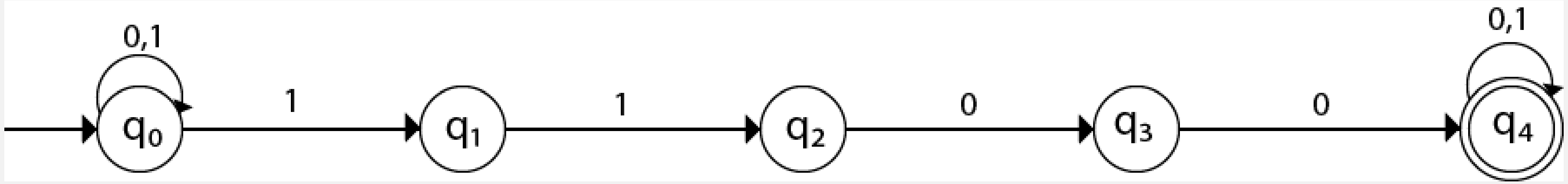



- It should be immediately followed by double 0.
- Then,



# Examples of NFA

- Now before double 1, there can be any string of 0 and 1.
- Similarly, after double 0, there can be any string of 0 and 1.
- Hence the NFA becomes:



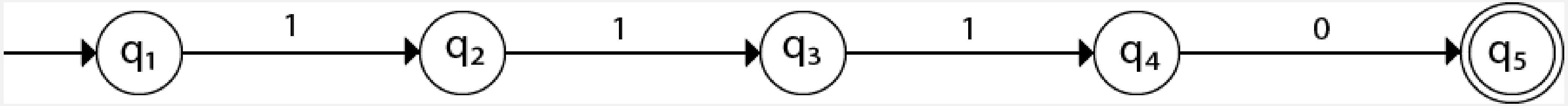

- Now considering the string 01100011

q0 → q1 → q2 → q3 → q4 → q4 → q4 → q4

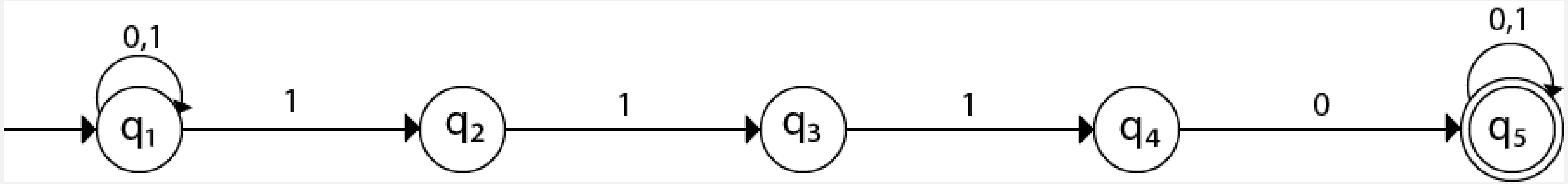
# Examples of NFA

**Example-3:** Design an NFA in which all the string contain a substring 1110.

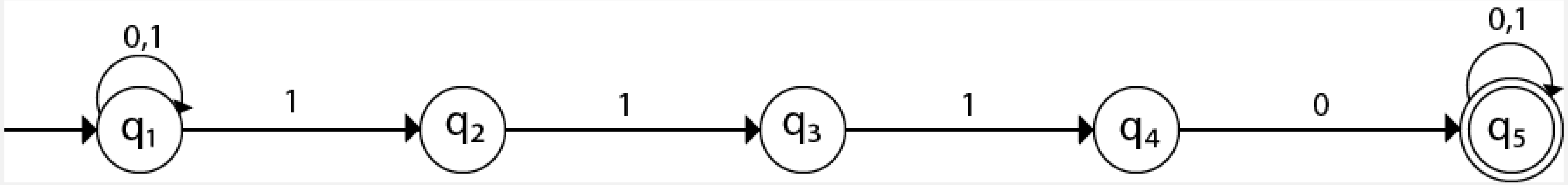
- The language consists of all the string containing substring 1010. The partial transition diagram can be:




- Now as 1010 could be the substring. Hence we will add the inputs 0's and 1's so that the substring 1010 of the language can be maintained. Hence the NFA becomes:



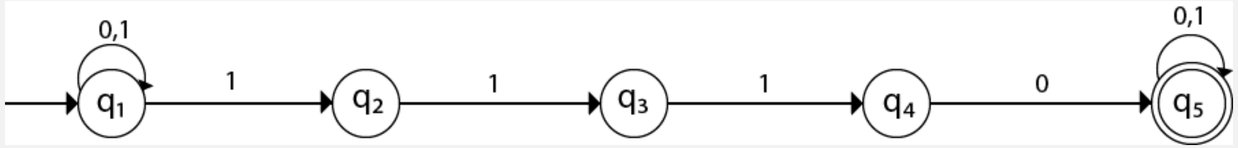
# Examples of NFA




- Transition table for the above transition diagram can be given below:

| Present State | 0 | 1 |
|---|---|---|
| →q1 | q1 | q1, q2 |
| q2 | | q3 |
| q3 | | q4 |
| q4 | q5 | |
| *q5 | q5 | q5 |

# Examples of NFA




- **Consider a string 111010, to be processed by this NFA.**

δ(q1, 111010) = δ(q1, 1100)
= δ(q1, 100)
= δ(q2, 00)

- Got stuck! As there is no path from q2 for input symbol 0. We can process string 111010 in another way.

δ(q1, 111010) = δ(q2, 1100)
= δ(q3, 100)
= δ(q4, 00)
= δ(q5, 0)
= δ(q5, ε)

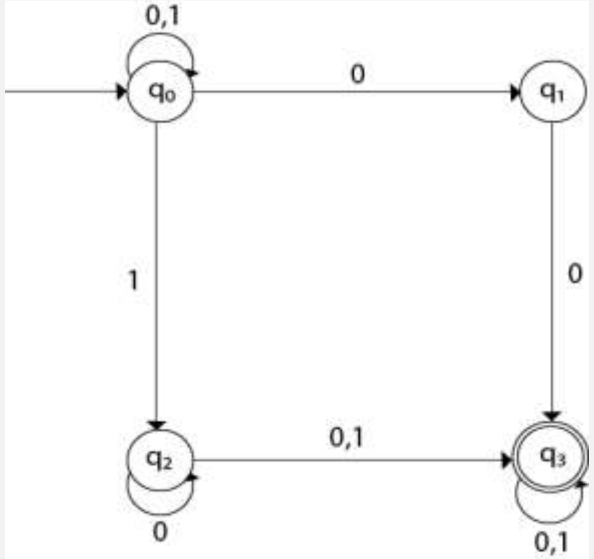
- **As state q5 is the accept state. We get the complete scanned, and we reached to the final state.**

# Examples of NFA

**Example-4:** Design a NFA for the transition table as given below:

| Present State | 0 | 1 |
|---|---|---|
| →q0 | q0, q1 | q0, q2 |
| q1 | q3 | ε |
| q2 | q2, q3 | q3 |
| →q3 | q3 | q3 |

- The transition diagram can be drawn by using the mapping function as given in the table.




Here,

δ(q0, 0) = {q0, q1}

δ(q0, 1) = {q0, q2}

Then, δ(q1, 0) = {q3}

Then, δ(q2, 0) = {q2, q3}

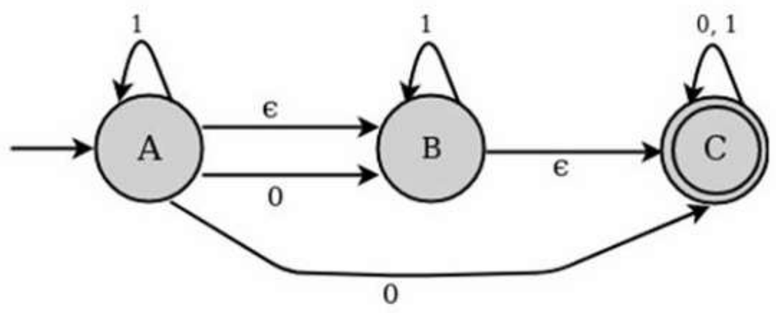
δ(q2, 1) = {q3}

Then, δ(q3, 0) = {q3}

δ(q3, 1) = {q3}

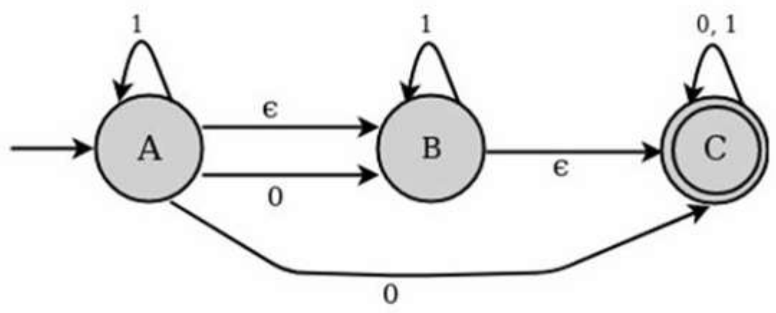
# NFA with epsilon transition (ε Moves)

- **Nondeterministic finite automaton with ε-moves (NFA-ε) is a further generalization to NFA.** In this kind of automaton, the transition function is additionally defined on the empty string ε.
- **A transition without consuming an input symbol is called an ε-transition and is represented in state diagrams by an arrow labeled "ε".**
- **ε-transitions provide a convenient way of modeling systems whose current states are not precisely known:** i.e., if we are modeling a system and it is not clear whether the current state (after processing some input string) should be q or q', then we can add an ε-transition between these two states, thus putting the automaton in both states simultaneously.



# NFA with epsilon transition (ε Moves)

- Thus, We extend the class of NFAs by allowing instantaneous ε transitions:
  - **The automaton may be allowed to change its state without reading the input symbol.**
- In diagrams, such transitions are depicted by labeling the appropriate arcs with ε.
- Note that this does not mean that E has become an input symbol. On the contrary, we assume that the symbol E does not belong to any alphabet.
  - **ε -NFAs add a convenient feature but (in a sense) they bring us nothing new.**
  - **They do not extend the class of languages that can be represented.**



# Formal Definition of *NFA-ε*

- **An *NFA-ε* is defined as-**

An *NFA-ε* is represented formally by a 5-tuple, $(Q, \Sigma, \delta, q_0, F)$, consisting of

- a finite set of states $Q$
- a finite set of input symbols called the alphabet $\Sigma$
- a transition function $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow \mathcal{P}(Q)$
- an *initial* (or *start*) state $q_0 \in Q$
- a set of states $F$ distinguished as *accepting* (or *final*) *states* $F \subseteq Q$.

Here, $\mathcal{P}(Q)$ denotes the power set of $Q$ and ε denotes empty string.

# Epsilon (ε) - Closure

- **Epsilon closure for a given state X is a set of states which can be reached from the states X with only (null) or E moves including the state X itself.**
- In other words, £-closure for a state can be obtained by union operation of the £-closure of the states which can be reached from X with a single E move in a recursive manner.

**ε-closure of a state or set of states**

For a state $q \in Q$, let $E(q)$ denote the set of states that are reachable from $q$ by following ε-transitions in the transition function $\delta$, i.e., $p \in E(q)$ if there is a sequence of states $q_1, \ldots, q_k$ such that

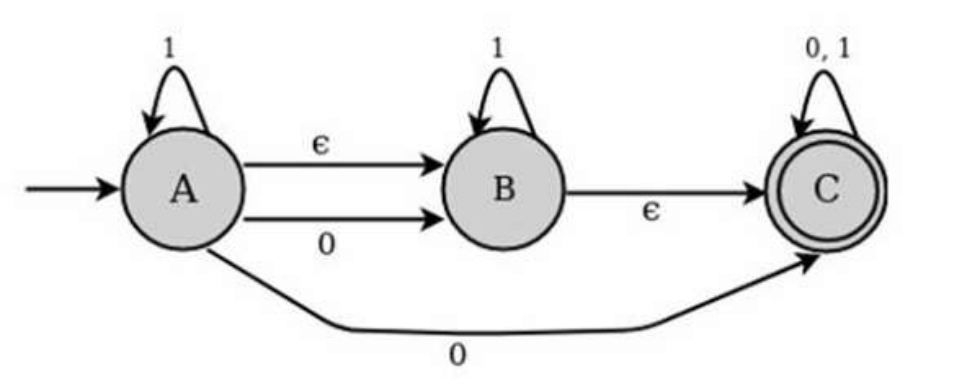
- $q_1 = q$,
- $q_{i+1} \in \delta(q_i, \varepsilon)$ for each $1 \leq i < k$, and
- $q_k = p$.

$E(q)$ is known as the **epsilon closure**, (also **ε-closure**) of $q$.

The ε-closure of a set $P$ of states of an NFA is defined as the set of states reachable from any state in $P$ following ε-transitions. Formally, for $P \subseteq Q$, define $E(P) = \bigcup_{q \in P} E(q)$.

# Example: ε - NFA

- Consider the following figure of NFA with ε move:




- The transition state table for the above NFA is as follows:

| State | 0 | 1 | epsilon |
|---|---|---|---|
| A | B,C | A | B |
| B | - | B | C |
| C | C | C | - |

- For the above example, ε closure are as follows:

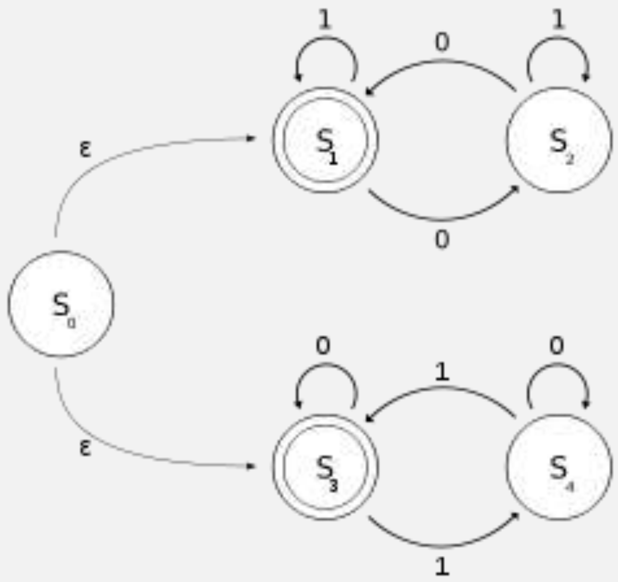
  E-closure( A) : {A, B,C}

  E-closure( B) :{B,C}

  E-closure( C) : {C}

# Example: ε - NFA

- Consider the following figure of a NFA-ε, with a binary alphabet, that determines if the input contains an even number of 0s or an even number of 1s.



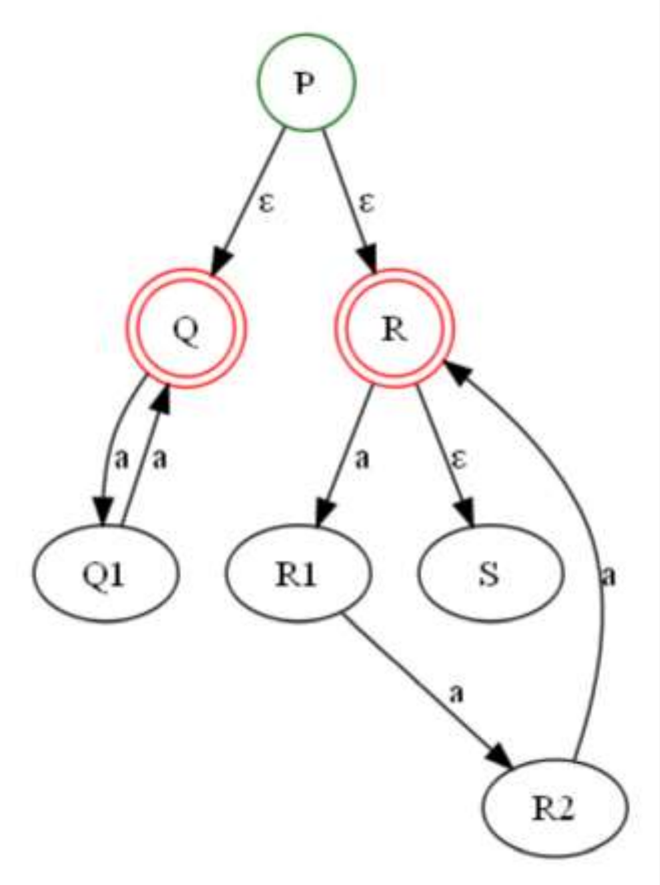

- The transition state table for the above NFA is as follows:

| Input / State | 0 | 1 | ε |
|---|---|---|---|
| $S_0$ | {} | {} | $\{S_1, S_3\}$ |
| $S_1$ | $\{S_2\}$ | $\{S_1\}$ | {} |
| $S_2$ | $\{S_1\}$ | $\{S_2\}$ | {} |
| $S_3$ | $\{S_3\}$ | $\{S_4\}$ | {} |
| $S_4$ | $\{S_4\}$ | $\{S_3\}$ | {} |

# Example: Computing ε - Closure

**Consider the given NFA-ε :**

- Compute e-close by adding new states until no new states can be added.
  - Start with [P]
  - Add Q and R to get [P,Q,R]
  - Add S to get [P,Q,R S]
  - No new states can be added



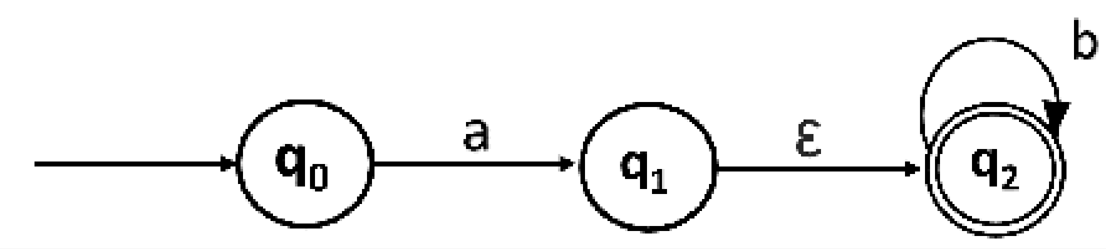

- Thus, for the above example:
  - ECLOSE(P) ={P,Q,R,S}
  - ECLOSE(R)={R,S}

# Eliminating ε Transitions

- **NFA with ε can be converted to NFA without ε, and this NFA without ε can be converted to DFA.**
- To do this, we will use a method, which can remove all the ε transition from given NFA. The method will be:
  1. Find out all the ε transitions from each state from Q. That will be called as ε-closure{qi} where qi ∈ Q.
  2. Then δ' transitions can be obtained. The δ' transitions mean a ε-closure on δ moves.
  3. Repeat Step-2 for each input symbol and each state of given NFA.
  4. Using the resultant states, the transition table for equivalent NFA without ε can be built.

# Eliminating ε Transitions

- **NFA with ε can be converted to NFA without ε, and this NFA without ε can be converted to DFA.**
- **Example:** Eliminate ε transition from the NFA with ε:




- **Solutions:** We will first obtain ε-closures of q0, q1 and q2 as follows:

ε-closure(q0) = {q0}

ε-closure(q1) = {q1, q2}

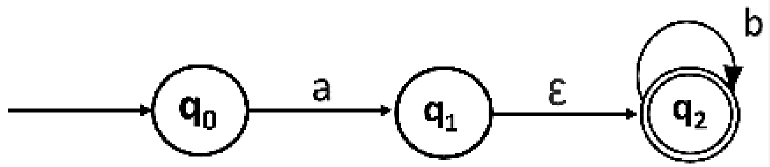
ε-closure(q2) = {q2}

# Eliminating ε Transitions

ε-closure(q0) = {q0}

ε-closure(q1) = {q1, q2}

ε-closure(q2) = {q2}



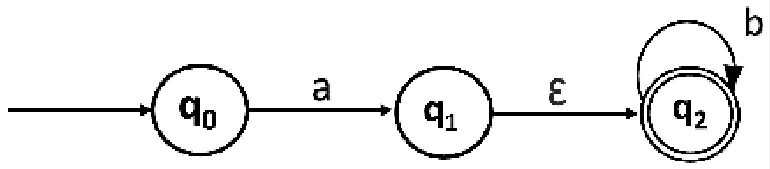

- Compute the δ' transition on each input symbol.
- Now the δ' transition on q0 is obtained as:

$$\begin{aligned}\delta'(q0, a) &= \varepsilon\text{-closure}(\delta(\delta\text{\textasciicircum}(q0, \varepsilon), a)) \\ &= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q0), a)) \\ &= \varepsilon\text{-closure}(\delta(q0, a)) \\ &= \varepsilon\text{-closure}(q1) \\ &= \{q1, q2\}\end{aligned}$$

$$\begin{aligned}\delta'(q0, b) &= \varepsilon\text{-closure}(\delta(\delta\text{\textasciicircum}(q0, \varepsilon), b)) \\ &= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q0), b)) \\ &= \varepsilon\text{-closure}(\delta(q0, b)) \\ &= \Phi\end{aligned}$$

# Eliminating ε Transitions

ε-closure(q0) = {q0}
ε-closure(q1) = {q1, q2}
ε-closure(q2) = {q2}




- The δ' transition on q1 is obtained as:

δ'(q1, a) = ε-closure(δ(δ^(q1, ε),a))
= ε-closure(δ(ε-closure(q1),a))
= ε-closure(δ(q1, q2), a)
= ε-closure(δ(q1, a) ∪ δ(q2, a))
= ε-closure(Φ ∪ Φ)
= Φ

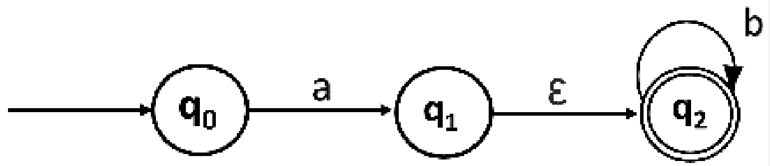
δ'(q1, b) = ε-closure(δ(δ^(q1, ε),b))
= ε-closure(δ(ε-closure(q1),b))
= ε-closure(δ(q1, q2), b)
= ε-closure(δ(q1, b) ∪ δ(q2, b))
= ε-closure(Φ ∪ q2)
= {q2}

# Eliminating ε Transitions

ε-closure(q0) = {q0}

ε-closure(q1) = {q1, q2}

ε-closure(q2) = {q2}




- The δ' transition on q2 is obtained as:

δ'(q2, a) = ε-closure(δ(δ^(q2, ε),a))
= ε-closure(δ(ε-closure(q2),a))
= ε-closure(δ(q2, a))
= ε-closure(Φ)
= Φ

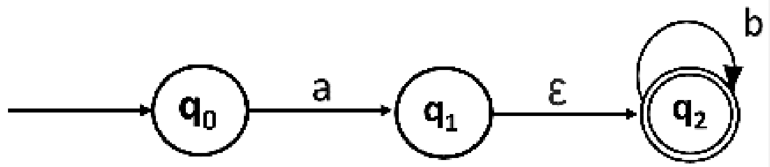
δ'(q2, b) = ε-closure(δ(δ^(q2, ε),b))
= ε-closure(δ(ε-closure(q2),b))
= ε-closure(δ(q2, b))
= ε-closure(q2)
= {q2}

# Eliminating ε Transitions

ε-closure(q0) = {q0}

ε-closure(q1) = {q1, q2}

ε-closure(q2) = {q2}




- Now we will summarize all the computed δ' transitions:

δ'(q0, a) = {q0, q1}

δ'(q0, b) = Φ

δ'(q1, a) = Φ

δ'(q1, b) = {q2}

δ'(q2, a) = Φ

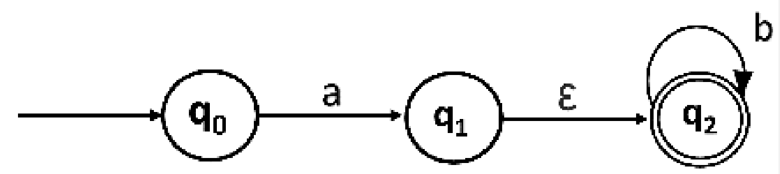
δ'(q2, b) = {q2}

- The transition table can be:

| States | a | b |
|---|---|---|
| →q0 | {q1, q2} | Φ |
| *q1 | Φ | {q2} |
| *q2 | Φ | {q2} |

# Eliminating ε Transitions

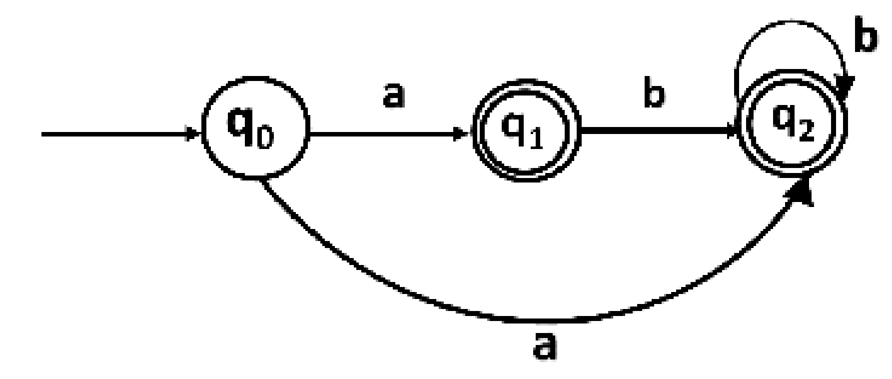
ε-closure(q0) = {q0}
ε-closure(q1) = {q1, q2}
ε-closure(q2) = {q2}




- The transition table can be:

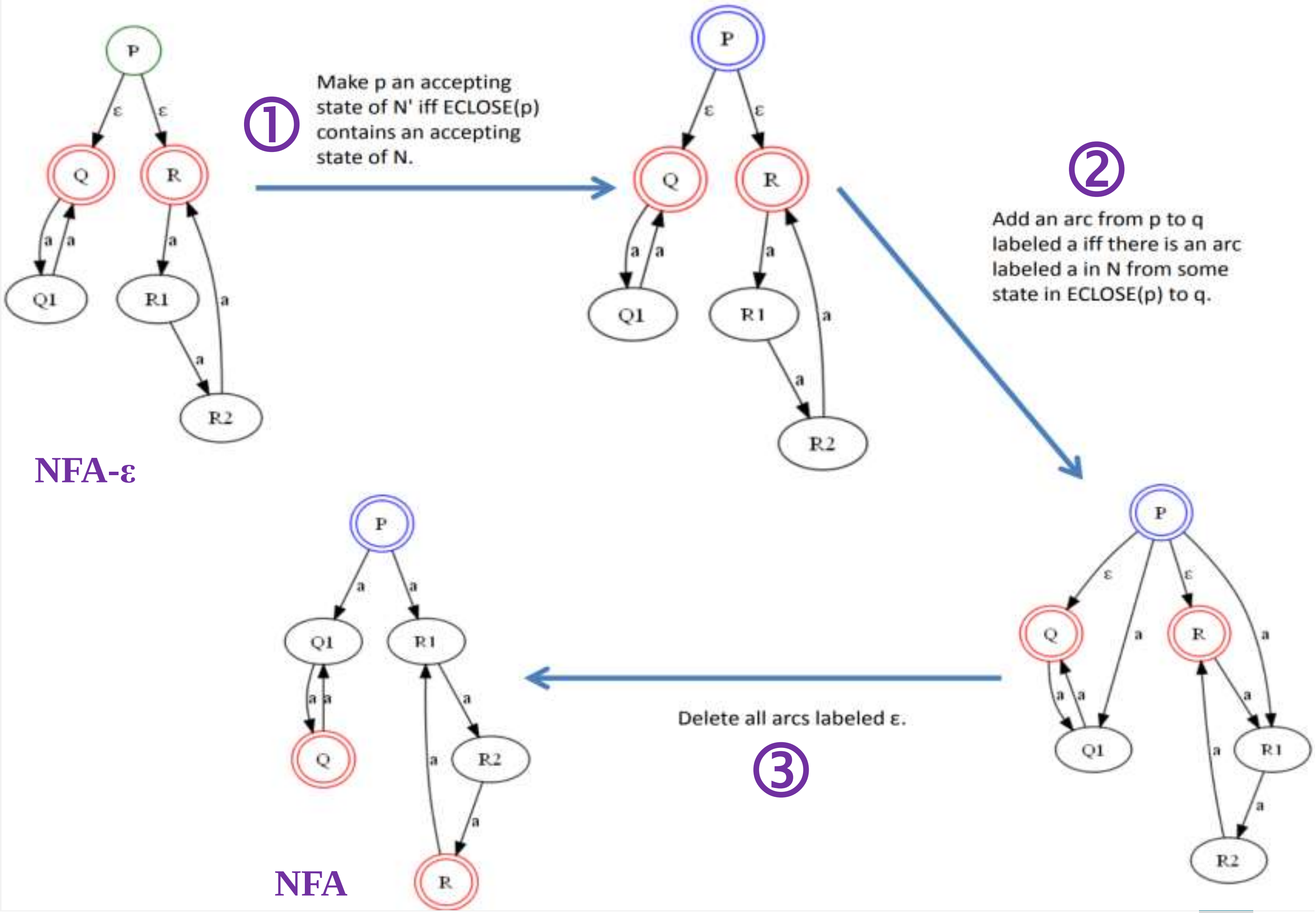
| States | a | b |
|---|---|---|
| →q0 | {q1, q2} | Φ |
| *q1 | Φ | {q2} |
| *q2 | Φ | {q2} |

- **State q1 and q2 become the final state as ε-closure of q1 and q2 contain the final state q2.**
- The NFA can be shown by the following transition diagram:



# Example: Eliminating ε Transitions

- Converting NFA with ε into a NFA without ε:



# ? THE END