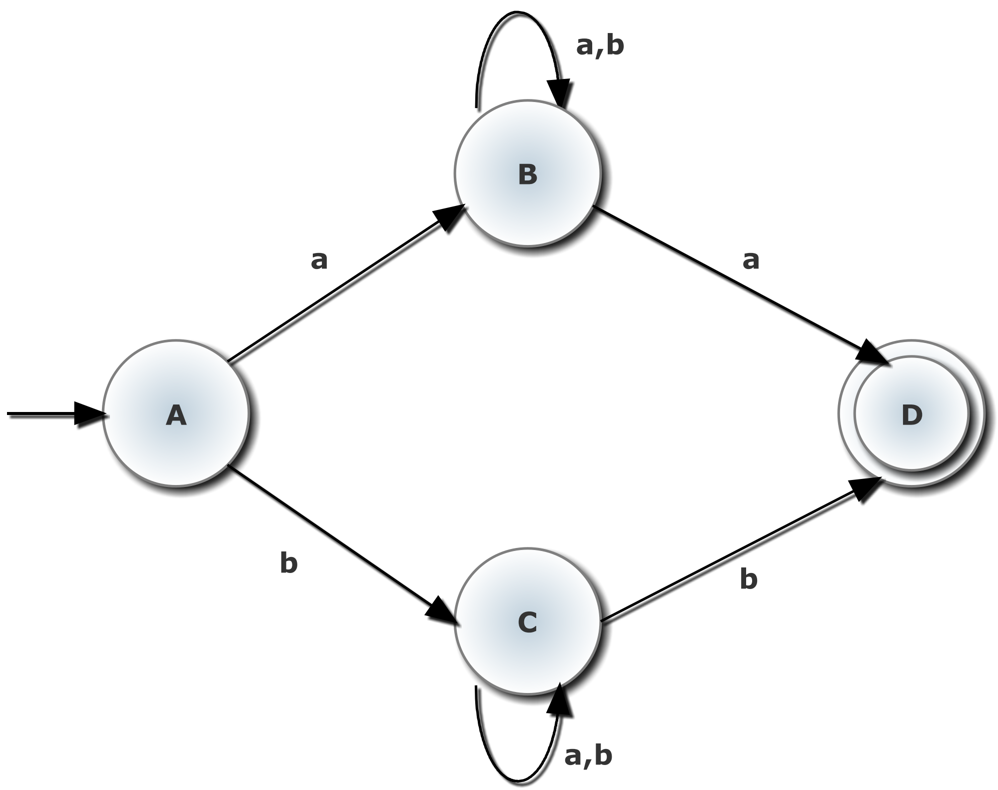
BITS Pilani
Pilani Campus

NFA with Epsilon Transitions

Shashank Gupta
Assistant Professor
Department of Computer Science and Information Systems

More NFA Designing Examples

- Construct NFA for $\{wxw^R \mid x \in (a,b)^*, w \in (a,b)^+$ & w^R is the reverse of the string w$\}$



Equivalence of NFA and DFA

For every NFA, there is an equivalent DFA.

Theorem: Let $N = (Q, \Sigma, \delta, q_0, F)$ be a NFA.

- There exists a DFA $D = (Q`, \Sigma, \delta`, q_0`, F`)$ s.t. $L(N) = L(D)$

Construction of DFA from NFA

$$Q` = 2^Q$$

$$Let\ A \subseteq Q\ then\ \delta`(A,a) = \bigcup_{r \in A} \delta(r,a)$$

$$q`_0 = \{q_0\}$$

- $F` = \{A \subseteq Q \mid A \cap F \neq \phi\}$

More Examples

Convert the following NFA to DFA over the given input alphabet Σ ={0,1}

I/P Symbol / State Symbol	0	1
→ P	{P,Q}	{P}
Q	{R,S}	{T}
R	{P,R}	{T}
s	Φ	Φ
t	Φ	Φ

I/P Symbol / STATE	0	1
→ P	PQ	P
PQ	PQRS	PT
PQRS	PQRS	PT
PT	PQ	P

Few Key Points

While converting from NFA to DFA, the number of states may increases or decreases .

- You may also come across some new states which are not reachable from start state.

Special Category of NFA

NFA with $\in$ Transitions

$\in$-NFA

We extend the class of NFAs by allowing $\in$ 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 $\in$.

- Note that this does not mean that $\in$ has become an input symbol.
- On the contrary, we assume that the symbol ε does not belong to any input alphabet.

$\in$-NFA

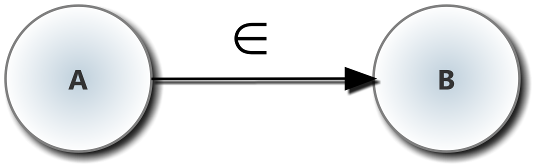
$\in$-NFAs add a convenient feature but (in a sense) they bring us nothing new: they do not extend the class of languages that can be represented.

Both NFAs and $\in$-NFAs recognize exactly the same languages.

$\in$-NFA

A special case of NFA in which transitions are also defined over Epsilon($\in$) moves. It is also a collection of 5 tuples (Q, Σ, δ, q_0, F) where

- $\delta: Q \times \Sigma \cup \{\in\} \rightarrow 2^Q$
- OR
- $\delta: Q \times \Sigma_{\in} \rightarrow 2^Q$



- **If there is a $\in$ transition, then the automaton moves to state A and B without reading the next input bit.**
- **An input is accepted if there is some computation path that leads to an accept state.**

Acceptance Mechanism

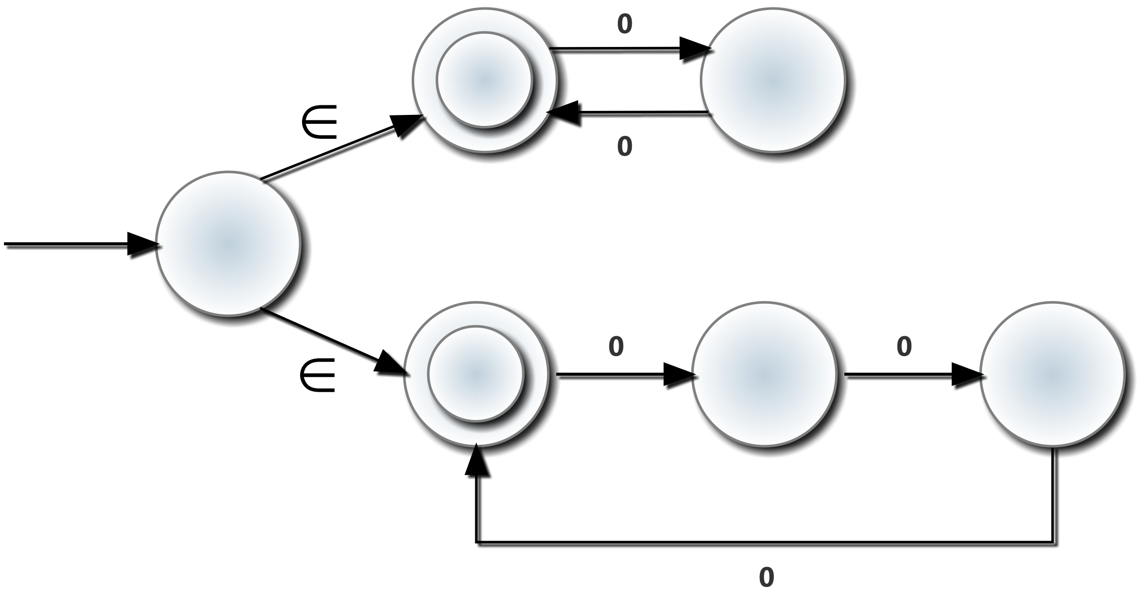
ϵ-NFA accepts $w = a_1 a_2 \ldots a_n$ if we can write w as $w = b_1 b_2 \ldots b_m$, where each $b_i \in \Sigma_\epsilon$ and there exists a sequence of states $r_0, r_1, \ldots, r_m \in Q$ (not necessarily distinct), such that

- $r_0 = q_0$, **(Initial State Condition)**
- $r_i \in \delta(r_{i-1}, b_i)$ for $i = 1, 2, \ldots m$ **(Transition Function Condition)**
- $r_m \in F$. **(Acceptance Condition)**

Then, $L(N) = \{w \in \Sigma^* \mid N \text{ accepts } w\}$.

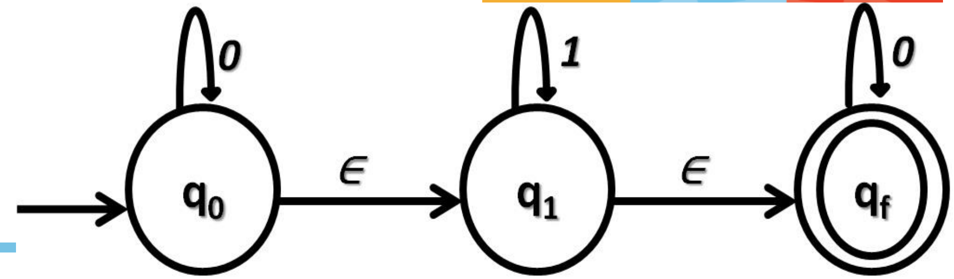
Example

- Construct a $\in$-NFA for the following language
  $L = \{w \in \{0\}^* \mid |w| \text{ is divisible by 2 or 3}\}$



Equivalence between $\in$-NFA and NFA

Computation of $\in$-Closure

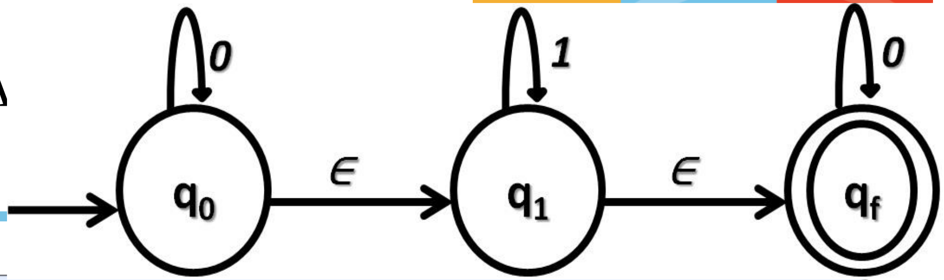


Let R ⊆ Q. $\in$-closure of R is defined as

$$E(R) = \bigcup_{r \in R} \{q \mid q \text{ can be reached from } r \text{ by using } 0 \text{ or more } \in - \text{transitions}\}$$

The $\in$-closure (q) is the set that contains q, together with all states that can be reached starting at q by following only ε-transitions.

- $\in$-closure $(q_0) = \{q_0, q_1, q_f\}$
- $\in$-closure $(q_1) = \{q_1, q_f\}$
- $\in$-closure $(q_f) = \{q_f\}$

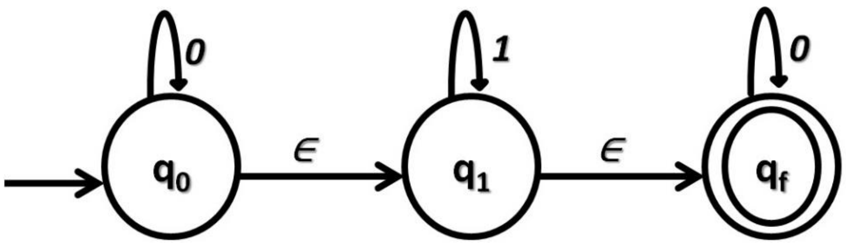


Transitions are defined based on the following formulae:

- $\delta`(q, a) = \in\text{-closure}(\delta(\in\text{-closure}(q),a))$

Equivalence between $\in$ -NFA and NFA

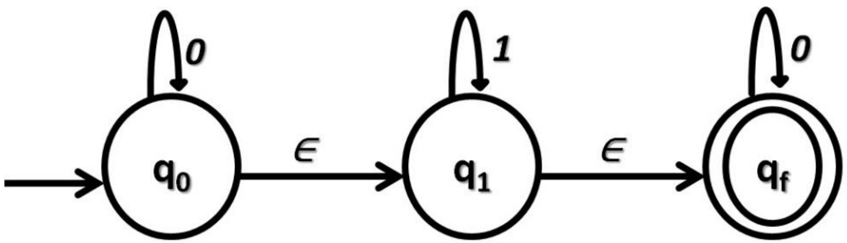
$\in$ -NFA



$\in$ -Closure

Q	ε-closure
q_0	$\{q_0, q_1, q_f\}$
q_1	$\{q_1, q_f\}$
q_f	$\{q_f\}$

Equivalence between $\in$-NFA and NFA (Continued….)



	0	1
→ q_0	$\{q_0,q_1,q_f\}$	
q_1		
q_f		

Initially we calculate for $\delta(q_0,0)$

$$\delta(q_0,0) = \in-closure(\delta(\in-closure(q_0),0))$$

$$\delta(q_0,0) = \in-closure(\delta((q_0,q_1,q_f),0)) \quad [\text{Since} \in-closure(q_0) = q_0,q_1,q_f)]$$

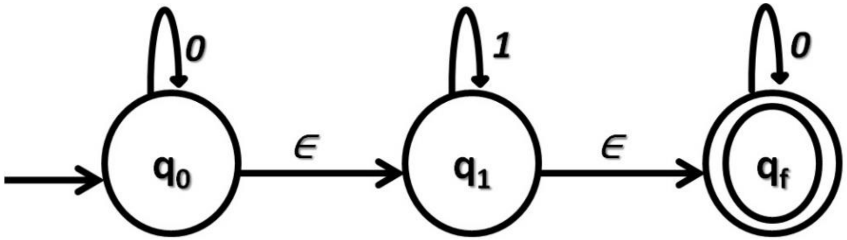
$$\delta(q_0,0) = \in-closure(\delta(q_0,0) \cup \delta(q_1,0) \cup \delta(q_f,0))$$

$$\delta(q_0,0) = \in-closure(q_0 \cup \phi \cup q_f)$$

$$\delta(q_0,0) = \in-closure(q_0,q_f)$$

$$\delta(q_0,0) = \{q_0,q_1,q_f\}$$

Equivalence between $\in$ -NFA and NFA (Continued….)



Similarly, we calculate for $\delta(q_0,1)$

	0	1
→q_0	$\{q_0,q_1,q_f\}$	$\{q_1,q_f\}$
q_1		
q_f		

$$\delta(q_0,1) = \in -closure(\delta(\in -closure(q_0),1))$$

$$\delta(q_0,1) = \in -closure(\delta((q_0,q_1,q_f),1)) \text{ [Since } \in -closure(q_0) = q_0,q_1,q_f \text{]}$$

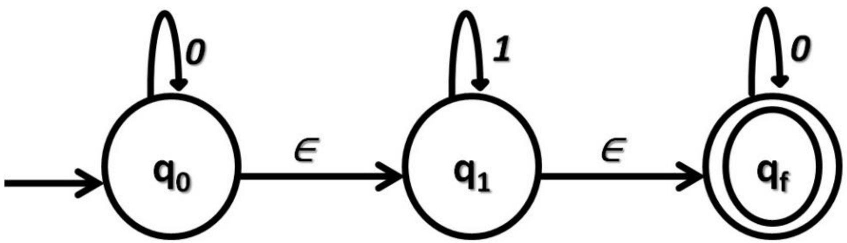
$$\delta(q_0,1) = \in -closure(\delta(q_0,1) \cup \delta(q_1,1) \cup \delta(q_f,1))$$

$$\delta(q_0,1) = \in -closure(\phi \cup q_1 \cup \phi)$$

$$\delta(q_0,1) = \in -closure(q_1)$$

$$\delta(q_0,1) = q_1, q_f$$

Equivalence between $\in$-NFA and NFA (Continued….)



	0	1
→q_0	$\{q_0,q_1,q_f\}$	$\{q_1,q_f\}$
q_1	$\{ q_f \}$	
q_f		

Likewise, we calculate for $\delta(q_1,0)$

$$\delta(q_1,0) = \in -closure(\delta(\in -closure(q_1),0))$$

$$\delta(q_1,0) = \in -closure(\delta(q_1,q_f),0)\ \text{[Since} \in -closure(q_1) = q_1,q_f\text{]}$$

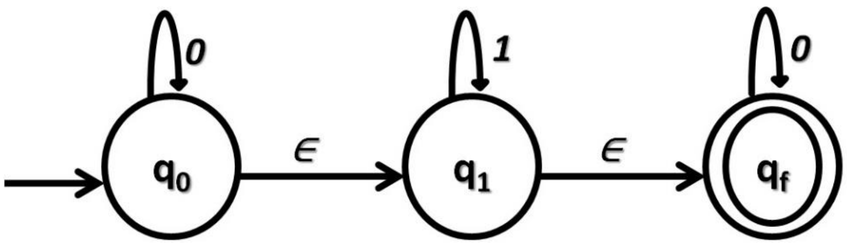
$$\delta(q_1,0) = \in -closure(\delta(q_1,0) \cup \delta(q_f,0))$$

$$\delta(q_1,0) = \in -closure(\phi \cup q_f)$$

$$\delta(q_1,0) = \in -closure(q_f)$$

$$\delta(q_1,0) = q_f$$

Equivalence between $\in$-NFA and NFA (Continued….)



	0	1
→q_0	$\{q_0,q_1,q_f\}$	$\{q_1,q_f\}$
q_1	$\{ q_f \}$	$\{q_1,q_f\}$
q_f		

$$\delta(q_1,1)=\in-closure(\delta(\in-closure(q_1),1))$$

$$\delta(q_1,1)=\in-closure(\delta(q_1,q_f),1)\ [\text{Since} \in-closure(q_1)=q_1,q_f)]$$

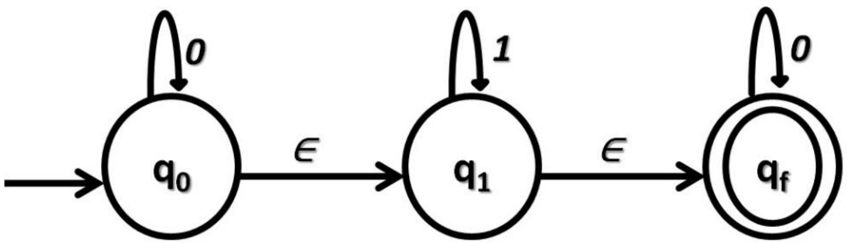
$$\delta(q_1,1)=\in-closure(\delta(q_1,1)\cup\delta(q_f,1))$$

$$\delta(q_1,1)=\in-closure(q_1\cup\phi)$$

$$\delta(q_1,1)=\in-closure(q_1)$$

$$\delta(q_1,1)=q_1,q_f$$

Equivalence between ∈-NFA and NFA (Continued….)



	0	1
→ q_0	$\{q_0,q_1,q_f\}$	$\{q_1,q_f\}$
q_1	$\{ q_f \}$	$\{q_1,q_f\}$
q_f	$\{ q_f \}$	

$$\delta(q_f,0)=\in-closure(\delta(\in-closure(q_f),0))$$

$$\delta(q_f,0)=\in-closure(\delta(q_f),0)\ [\text{Since} \in-closure(q_f)=q_f\ ]$$

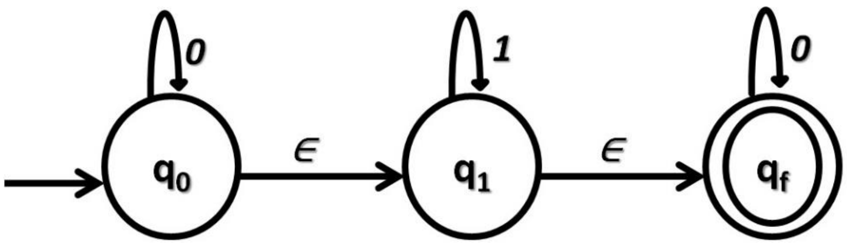
$$\delta(q_f,0)=\in-closure(\delta(q_f,0))$$

$$\delta(q_f,0)=\in-closure(q_f)$$

$$\delta(q_f,0)=q_f$$

Equivalence between $\in$-NFA and NFA (Continued….)

	0	1
$\rightarrow q_0$	$\{q_0,q_1,q_f\}$	$\{q_1,q_f\}$
q_1	$\{q_f\}$	$\{q_1,q_f\}$
q_f	$\{q_f\}$	ϕ



$$\delta(q_f,1) = \in\!-closure(\delta(\in\!-closure(q_f),1))$$

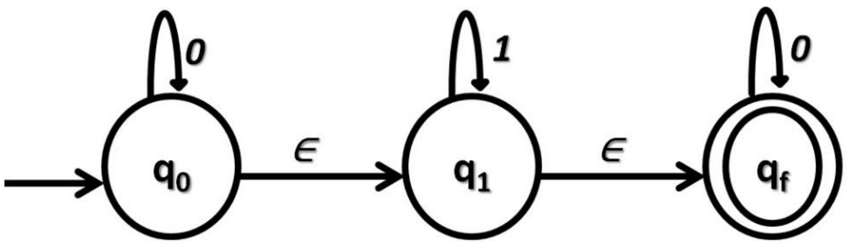
$$\delta(q_f,1) = \in\!-closure(\delta(q_f),1) \quad [\text{Since} \in\!-closure(q_f) = q_f\,]$$

$$\delta(q_f,1) = \in\!-closure(\delta(q_f,1))$$

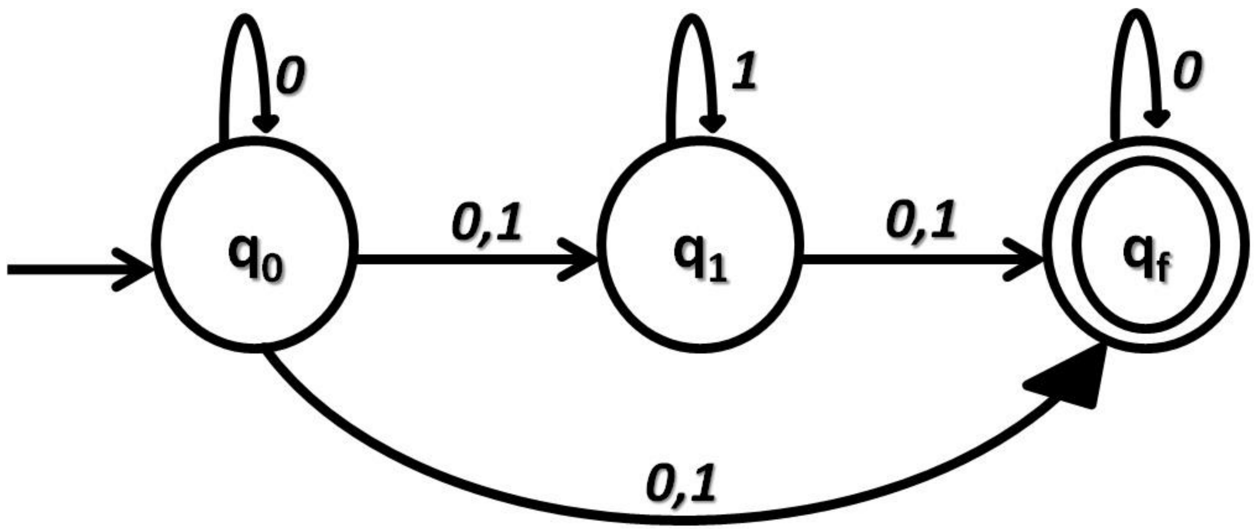
$$\delta(q_f,1) = \in\!-closure(\phi)$$

$$\delta(q_f,1) = \phi$$

Equivalence between ∈-NFA and NFA (Continued….)



	0	1
→ q_0	$\{q_0,q_1,q_f\}$	$\{q_1,q_f\}$
q_1	$\{ q_f \}$	$\{q_1,q_f\}$
q_f	$\{ q_f \}$	ϕ



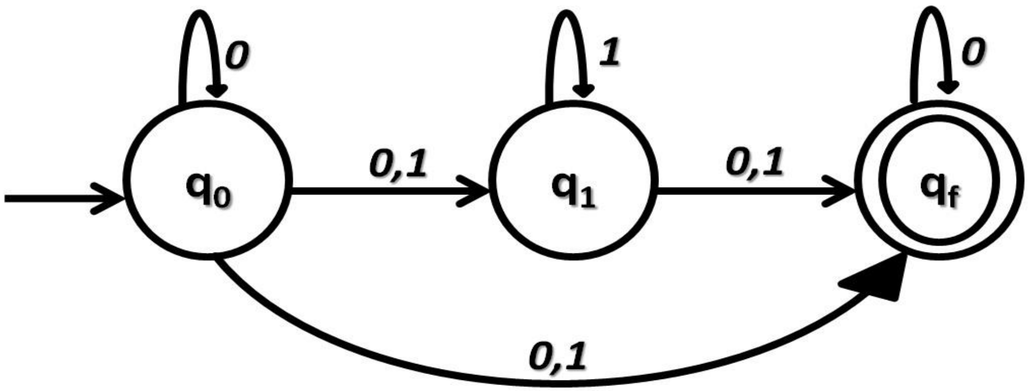
Equivalence between $\in$-NFA and NFA

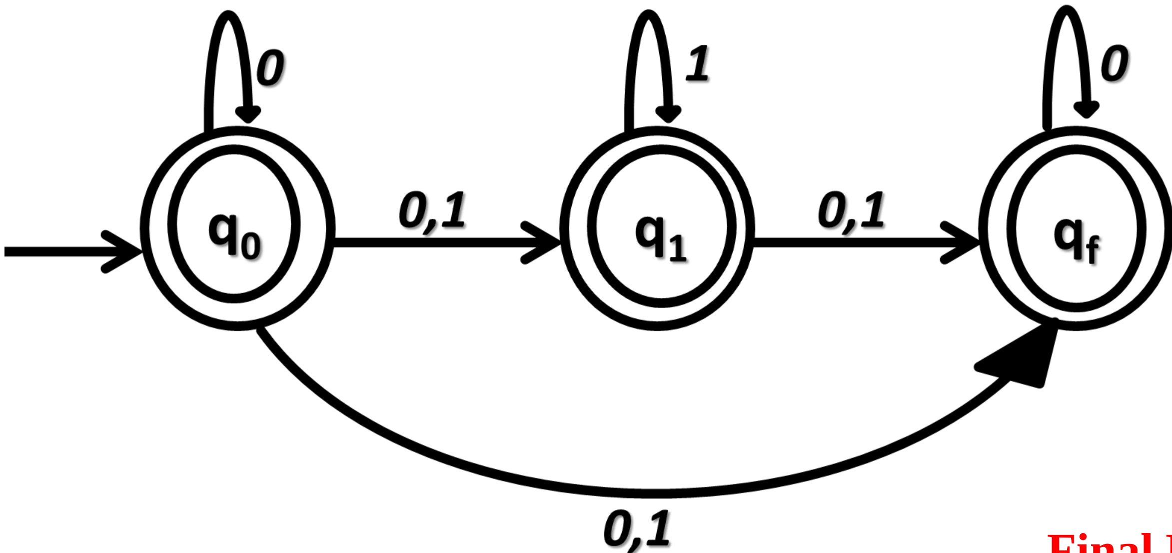
Number of states of NFA will remain same as of $\in$-NFA.

Number of final states may increases

- i.e. which includes all final states of $\in$- NFA and from any state by reading $\in$ if it reaches to final state make that state as final too.

Equivalence between ∈-NFA and NFA (Continued….)





Final Resultant NFA