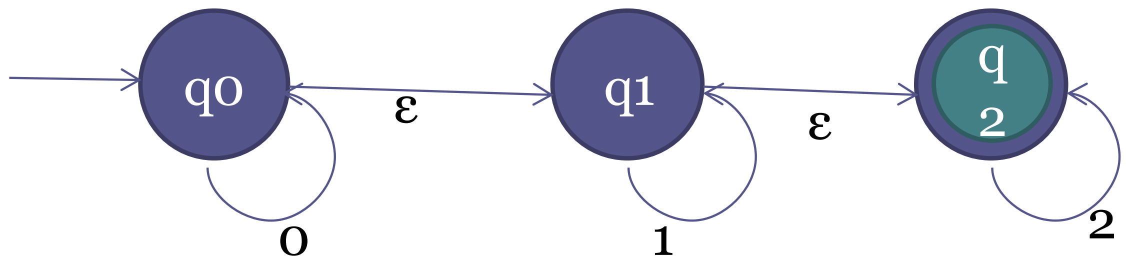
Epsilon NFA – NFA with ε-moves

- ε transition is a state transition of NFA without reading any input.
- Epsilon Non Deterministic Finite State Automata M is a 5 tuple.
- $M=(Q,\sum,\delta,q_o,F)$
- Where
- Q – is a finite set of states
- ∑ – is a finite set of input alphabets
- q_o- initial state. It belongs to set Q.
- F – set of final states/ accept states. It is subset of Q.
- δ – It is transition function or mapping from set **Q and ∑ U {ε} to power set of Q (2^Q).**
- It maps a given state p and input symbol a to zero or more next state i.e subset of Q. **It has e-moves or transitions.**
- **Symbolically δ : Q x ∑ U {ε}->2^Q**

ε-NFA Example

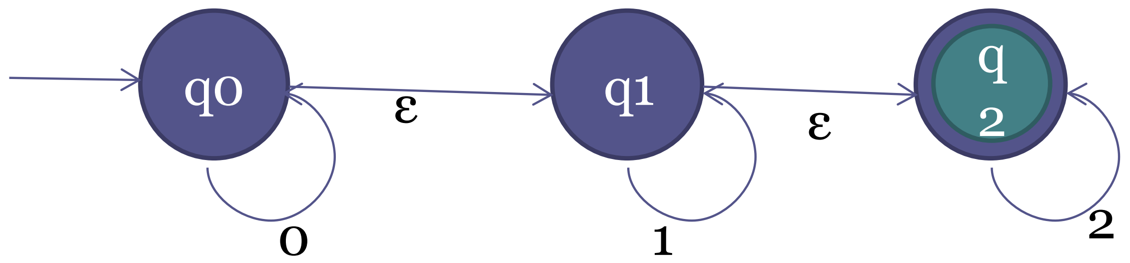
State Transition Diagram



$L(M)=\{0^n 1^m 2^k \mid n,m,k>=0\}$

ε-closure of a state ε*(q)

State Transition Diagram



ε -closure of a state q is defines as set of all states reachable from q through ε – moves including itself.

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

Equivalence of ε-NFA and NFA

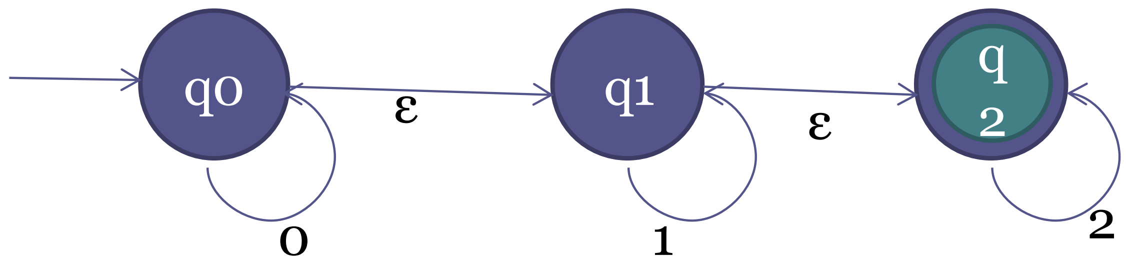
- ε-NFA is equivalent to NFA
- Both ε-NFA and NFA accepts regular language
- Language accepting power/computation power of ε-NFA and NFA is same
- How?
- Convert ε-NFA to NFA

ε-NFA to NFA Conversion

- Theorem
- Let $M=(Q,\sum,\delta,q_0,F)$ be a NFA with ε-moves then an NFA M' without ε-moves can be constructed such that $L(M)=L(M')$.
- **Proof by construction –**
- Given ε-NFA $M=(Q,\sum,\delta,q_0,F)$
- Construct NFA without ε-moves $M'=(Q,\sum,\delta',q_0,F')$
- $F'=F$, if ε-closure of q_0 does not contain a state from F
- $F'=F \cup \{q_0\}$, if ε-closure of q_0 contains a state from F
- δ': $\delta'(q,a)=$ ε-closure(δ(ε-closure(q),a)) for all $q \in Q$ and $a \in \sum$

ε-NFA to NFA

ε*(q0)={q0,q1,q2}, ε*(q1)={q1,q2}, ε*(q2)={q2}



δ'	ε*	1	ε*
q0	q0 q1 q2	Φ q1 Φ	{q1,q2 }
q1	q1 q2	q1 Φ	{q1,q2 }
q2	q2	Φ	Φ

δ'	ε*	0	ε*
q0	q0 q1 q2	q0 Φ Φ	{q0,q1, q2}
q1	q1 q2	Φ Φ	Φ
q2	q2	Φ	Φ

δ'	ε*	2	ε*
q0	q0 q1 q2	Φ Φ q2	{q2}
q1	q1 q2	Φ q2	{q2}
q2	q2	{q2}	{q2}

δ': δ'(q,a)= ε-closure(δ(ε-closure(q),a)) δ'(q,0) 0-moves

δ'	ε*	0	ε*
q0	q0 q1 q2	q0 Φ Φ	{q0,q1, q2}
q1	q1 q2	Φ Φ	Φ
q2	q2	Φ	Φ

δ'(q,1) 1-moves

δ'	ε*	1	ε*
q0	q0 q1 q2	Φ q1 Φ	{q1,q2 }
q1	q1 q2	q1 Φ	{q1,q2 }
q2	q2	Φ	Φ

δ'(q,2) 2-moves

δ'	ε*	2	ε*
q0	q0 q1 q2	Φ Φ q2	{q2}
q1	q1 q2	Φ q2	{q2}
q2	q2	{q2}	{q2}

δ'	0	1	2
q0	{q0,q1, q2}	{q1.q2}	{q2}
q1	Φ	{q1,q2}	{q2}
q2	Φ	Φ	{q2}