Theory of Computation
CS-202
Finite Automata

Finite Automata

-Deterministic Finite Accepters

-Non Deterministic Finite Accepters

Nondeterministic Finite Accepters

A nondeterministic finite accepter or nfa is defined by the quintuple

$$M = (Q, \Sigma, \delta, q_0, F)$$

where Q is a finite set of internal states,

Σ is a finite set of symbols called the input alphabet,

$q_0 \in Q$ is the initial state,

$F \subseteq Q$ is a set of final states.

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

$\forall$ NFA $\exists$ a DFA

$\Rightarrow$ DFA $\subseteq$ NFA

Conversion from NFA to DFA

- In NFA, when a specific input is given to the current state, the machine goes to multiple states. It can have **zero, one or more than one move** on a given input symbol.

- On the other hand, in DFA, when a specific input is given to the current state, the machine goes to only one state. **DFA has only one move on a given input symbol.**

- Let, $M = (Q, \sum, \delta, q_0, F)$ is an NFA which accepts the language L(M). There should be equivalent DFA denoted by $M' = (Q', \sum', q_0', \delta', F')$ such that $L(M) = L(M')$.

Steps for converting NFA to DFA:

Step 1: Initially Q' = ϕ

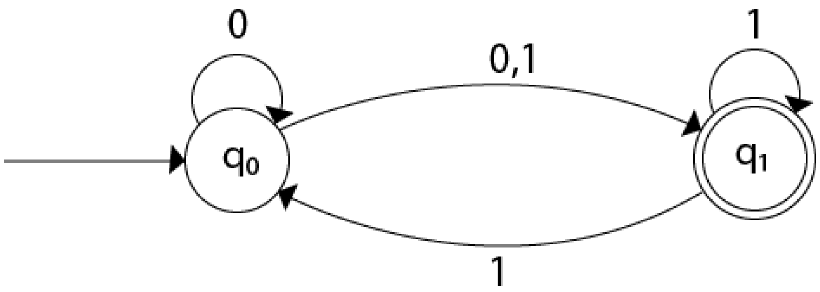
Step 2: Add q_0 of NFA to Q'. Then find the transitions from this start state.

Step 3: In Q', find the possible set of states for each input symbol. If this set of states is not in Q', then add it to Q'.

Step 4: In DFA, the final state will be all the states which contain F(final states of NFA)

Example-1

Convert the given NFA to DFA



Solution: For the given transition diagram we will first construct the transition table.

State	0	1
→q0	{q0, q1}	{q1}
*q1	ϕ	{q0, q1}

Example-1 (Cont..)

Solution:

State	0	1
→q0	{q0, q1}	{q1}
*q1	ϕ	{q0, q1}

Now we will obtain δ' transition for state q_0.

$\delta'([q_0], 0) = \{q_0, q_1\}$

$= [q_0, q_1]$ (new state)

$\delta'([q_0], 1) = \{q_1\} = [q_1]$

Example-1 (Cont..)

Solution:

State	0	1
→q0	{q0, q1}	{q1}
*q1	ϕ	{q0, q1}

Now we will obtain δ' transition for state q_1.

δ'([q1], 0) = ϕ , δ'([q1], 0) = q_2 (new state)

δ'([q1], 1) = [q0, q1]

Example-1 (Cont..)

Solution:

State	0	1
→q0	{q0, q1}	{q1}
*q1	ϕ	{q0, q1}

Note: for $\delta'([q_1], 0) = \phi$, we need to create a new state (called dead state) say q_2
Now we will obtain δ' transition for state q_2.

$\delta'([q_2], 0) = q_2$
$\delta'([q_2], 1) = q_2$

Example-1 (Cont..)

Solution:

State	0	1
→q0	{q0, q1}	{q1}
*q1	ϕ	{q0, q1}

Now we will obtain δ' transition on [q0, q1].

δ'([q0, q1], 0) = δ(q0, 0) ∪ δ(q1, 0)
= {q0, q1} ∪ ϕ
= {q0, q1}
= [q0, q1]

δ'([q0, q1], 1) = δ(q0, 1) ∪ δ(q1, 1)
= {q1} ∪ {q0, q1}
= {q0, q1}
= [q0, q1]

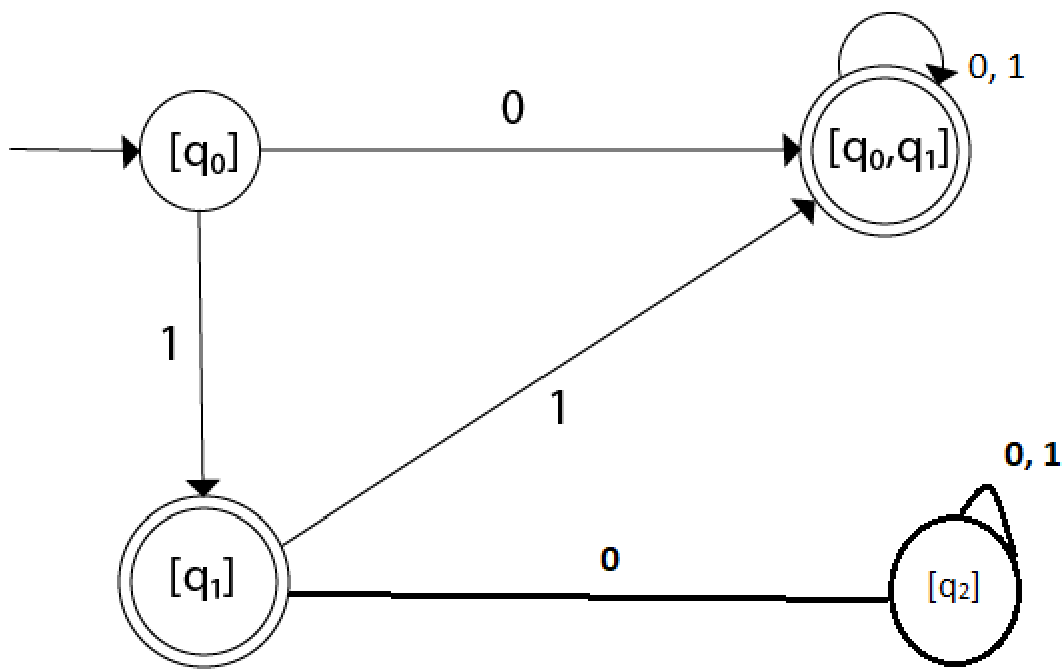
- As in the given NFA, q_1 is a final state, then in DFA wherever, q_1 exists that state becomes a final state. Hence in the DFA, final states are $[q_1]$ and $[q_0, q_1]$.

- Therefore set of final states $F = \{[q_1], [q_0, q_1]\}$.

- The transition table for the constructed DFA will be:

State	0	1
→[q0]	[q0, q1]	[q1]
*[q1]	[q_2]	[q0, q1]
*[q0, q1]	[q0, q1]	[q0, q1]
q_2	q_2	[q_2]

The Transition diagram for DFA will be:



ε –NFA or Epsilon NFA

ε –NFA or Epsilon NFA

It is an extended version of NFA, which makes the designing of NFA much easier.

An ε –NFA is defined using the quintuple

$$M=(Q, \Sigma, q0, \delta, F)$$

where Q is a finite set of internal states,

Σ is a finite set of symbols called the input alphabet,

$q_0 \in Q$ is the initial state,

$F \subseteq Q$ is a set of final states.

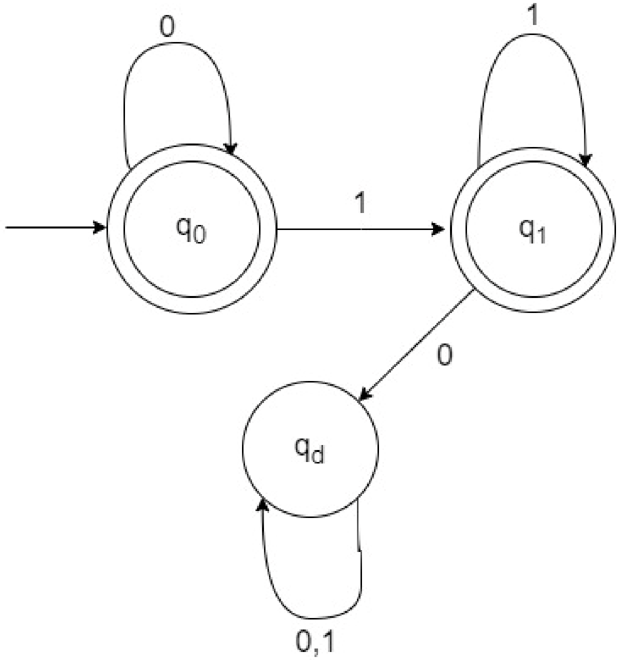
$\delta: Q \times (\Sigma \cup \varepsilon) \rightarrow 2^Q$

Examples

1. Design an ε NFA for the language

$$L=\{0^n1^m,\ n,m\geq 0\}$$

DFA

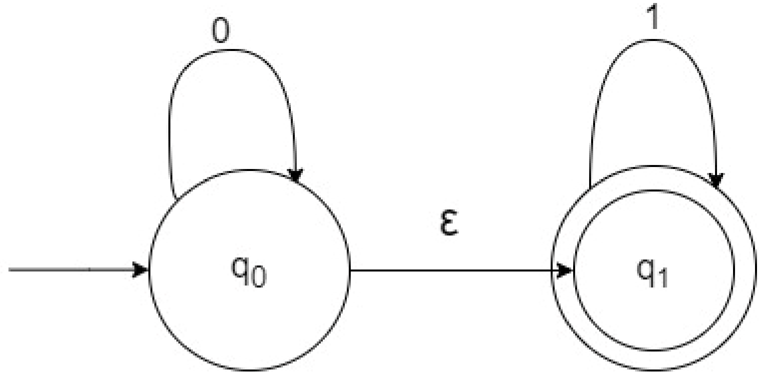


Examples (Cont..)

1. Design an ε NFA for the language

$L=\{0^n1^m, n,m\geq0\}$

ε NFA



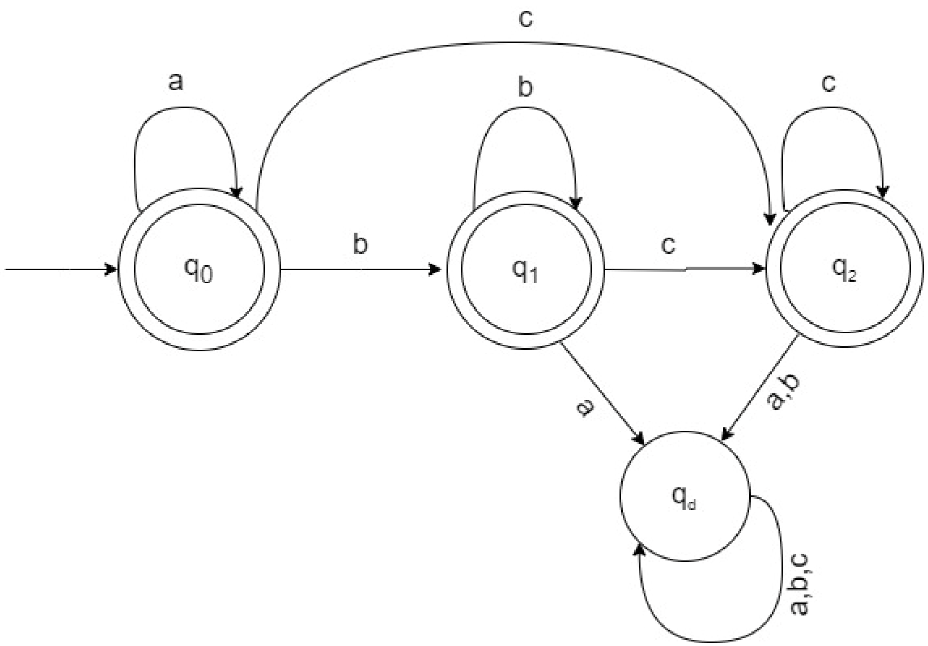
Examples (Cont..)

2. Design an ε NFA for the language

$L=\{a^m b^n c^p,\ m,n,p \geq 0\}$

first design DFA

DFA

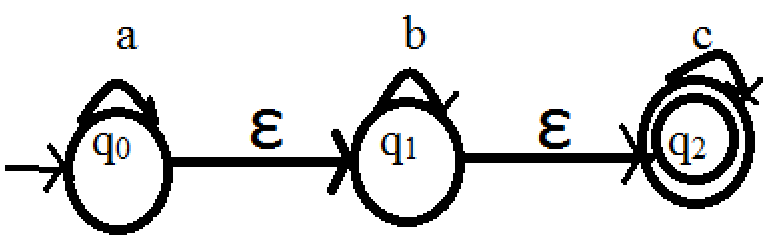


Examples (Cont..)

2. Design an ε NFA over ∑ ={a,b,c}for the language

$$L=\{a^m b^n c^p,\ m,n,p \geq 0\}$$

ε NFA



Examples (Cont..)

3. Design an ε NFA for the language

$L=\{0^m1^n, m+n=odd\}$

Occurrence of 0's and 1's

0	1	sum
Even	even	even
Even	odd	odd
Odd	even	odd
Odd	odd	even

3. Design an ε NFA for the language

$L=\{0^m1^n, m+n=odd\}$

Input Strings	0	0	1	1		Rejected
	0	0	1	1	1	Accepted