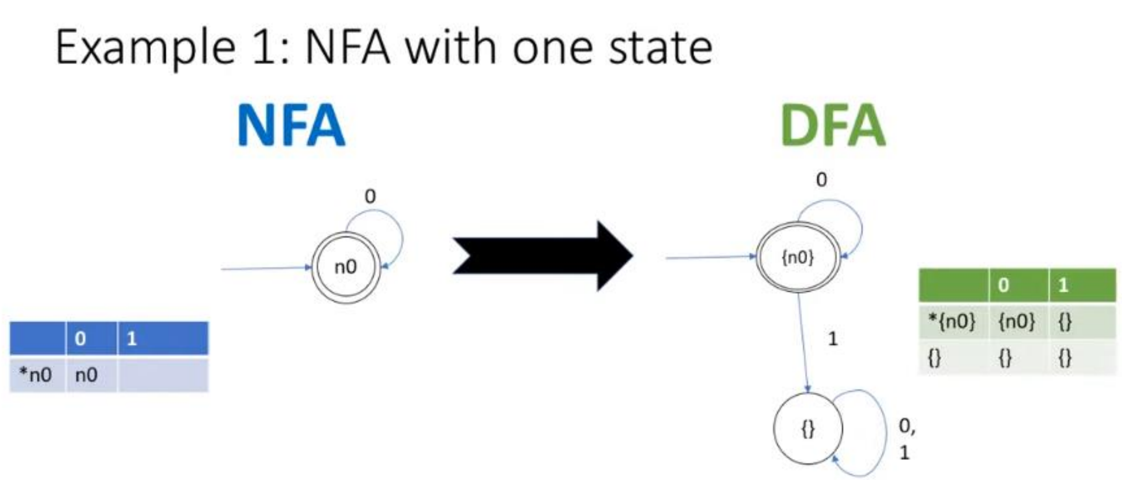
Steps for converting NFA to DFA: Lecture 8

Problem Statement

Let **X = (Q_x, ∑, δ_x, q_0, F_x)** be an NDFA which accepts the language L(X). We have to design an equivalent DFA **Y = (Q_y, ∑, δ_y, q_0, F_y)** such that **L(Y) = L(X)**. The following procedure converts the NDFA to its equivalent DFA −

Algorithm

Input − An NDFA
Output − An equivalent DFA
Step 1 − Create state table from the given NDFA.
Step 2 − Create a blank state table under possible input alphabets for the equivalent DFA.
Step 3 − Mark the start state of the DFA by q0 (**Same as the NDFA**).
Step 4 − Find out the combination of States {Q_0, Q_1,... , Q_n} for each possible input alphabet.
Step 5 − Each time we generate a new DFA state under the input alphabet columns, we have to apply step 4 again, otherwise go to step 6.
Step 6 − The states which contain any of the final states of the NDFA are the final states of the equivalent DFA.

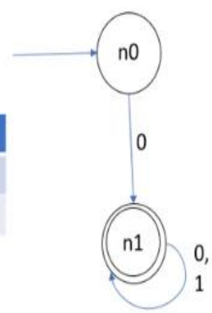


NFA ⟹ DFA ✓
DFA ⟹ NFA ✗
NFA ≅ DFA

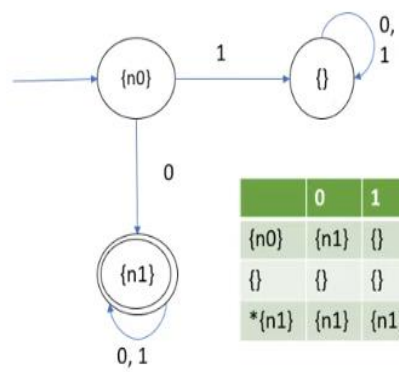
Example2: NFA with two states

NFA

	0	1
n0	n1	
*n1	n1	n1



DFA



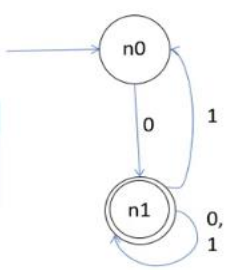
	0	1
{n0}	{n1}	{}
{}	{}	{}
*{n1}	{n1}	{n1}

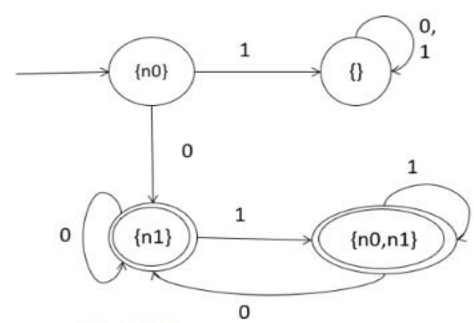
Example3: NFA with 2 states

	0	1
{n0}	{n1}	{}
{}	{}	{}
*{n1}	{n1}	{n1}
*{n0, n1}	{n1}	{n0, n1}

NFA

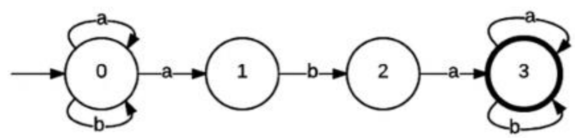
	0	1
n0	n1	
*n1	n1	n0, n1





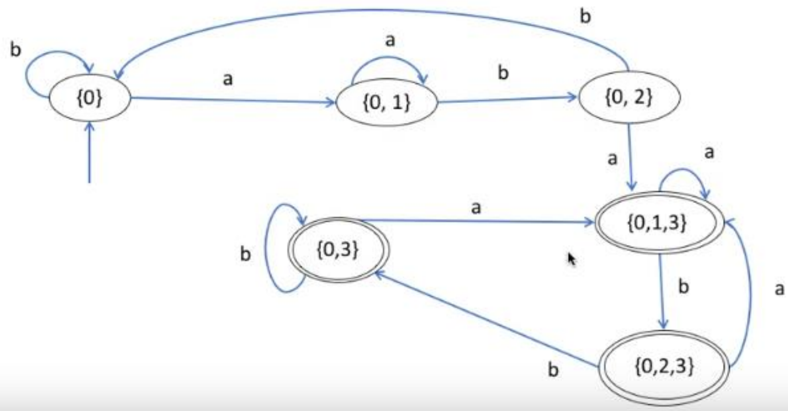
DFA

- Convert this NFA to a DFA



states	a	b
0	0, 1	0
1		2
2	3	
3	3	3

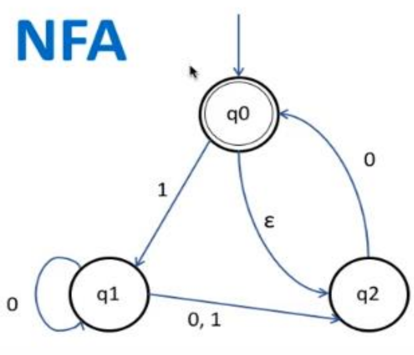
states	a	b
{0}	{0, 1}	{0}
{0, 1}	{0, 1}	{0, 2}
{0, 2}	{0, 1, 3}	{0}
*{0, 1, 3}	{0, 1, 3}	{0, 2, 3}
*{0, 2, 3}	{0, 1, 3}	{0, 3}
*{0, 3}	{0, 1, 3}	{0, 3}



states	0	1	ε
q0		q1	q2
q1	q1,q2	q2	
q2	q0		

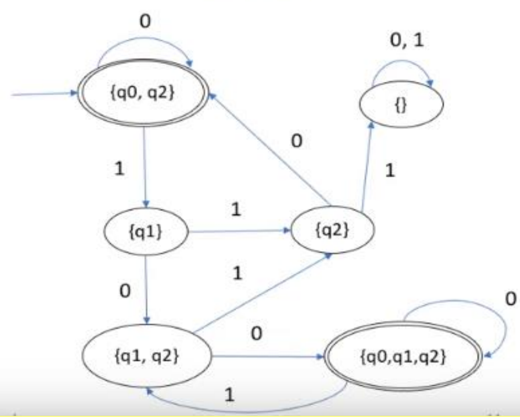
Example 5: convert NFA to DFA

NFA



NFA_4

DFA



Example

Consider the following NFA shown in Figure 1.

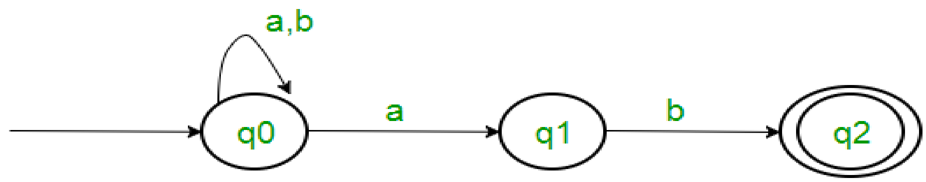


Figure 1

Following are the various parameters for NFA.

Q = { q0, q1, q2 }

∑ = (a, b)

F = { q2 }

δ (Transition Function of NFA)

State	a	b
q0	q0,q1	q0
q1		q2
q2		

Step 1: Q' = ϕ

Step 2: Q' = {q0}

Step 3: For each state in Q', find the states for each input symbol.

Currently, state in Q' is q0, find moves from q0 on input symbol a and b using transition function of NFA and update the transition table of DFA.

δ' (Transition Function of DFA)

State	a	b
q0	{q0,q1}	q0

Now { q0, q1 } will be considered as a single state. As its entry is not in Q', add it to Q'.

So Q' = { q0, { q0, q1 } }

Now, moves from state { q0, q1 } on different input symbols are not present in transition table of DFA, we will calculate it like:

δ' ({ q0, q1 }, a) = δ (q0, a) ∪ δ (q1, a) = { q0, q1 }

δ' ({ q0, q1 }, b) = δ (q0, b) ∪ δ (q1, b) = { q0, q2 }

Now we will update the transition table of DFA.

δ' (Transition Function of DFA)

State	a	B
q0	{q0,q1}	q0
{q0,q1}	{q0,q1}	{q0,q2}

Now { q0, q2 } will be considered as a single state. As its entry is not in Q', add it to Q'.

So Q' = { q0, { q0, q1 }, { q0, q2 } }

Now, moves from state {q0, q2} on different input symbols are not present in transition table of DFA, we will calculate it like:

δ' ({ q0, q2 }, a) = δ (q0, a) ∪ δ (q2, a) = { q0, q1 }

δ' ({ q0, q2 }, b) = δ (q0, b) ∪ δ (q2, b) = { q0 }

Now we will update the transition table of DFA.

δ' (Transition Function of DFA)

State	a	B
q0	{q0,q1}	q0
{q0,q1}	{q0,q1}	{q0,q2}
{q0,q2}	{q0,q1}	q0

As there is no new state generated, we are done with the conversion. Final state of DFA will be state which has q2 as its component i.e., { q0, q2 }

Following are the various parameters for DFA.

Q' = { q0, { q0, q1 }, { q0, q2 } }

∑ = (a, b)

F = { { q0, q2 } } and transition function δ' as shown above. The final DFA for above NFA has been shown in Figure 2.

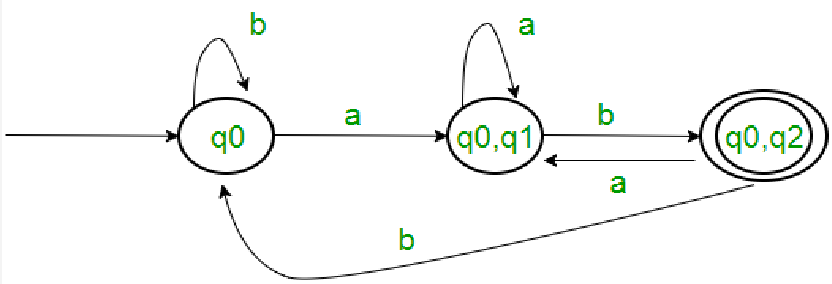


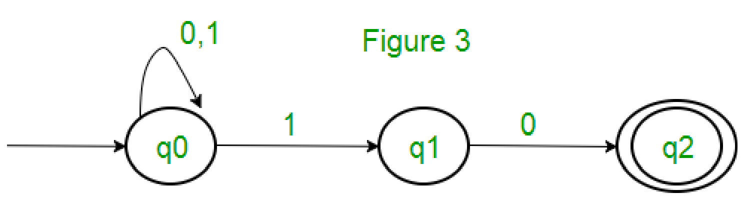
Figure 2

Note : Sometimes, it is not easy to convert regular expression to DFA. First you can convert regular expression to NFA and then NFA to DFA.

Question : The number of states in the minimal deterministic finite automaton corresponding to the regular expression (0 + 1)* (10) is ___________.

Solution : First, we will make an NFA for the above expression. To make an NFA for (0 + 1)*, NFA will be in same state q0 on input symbol 0 or 1. Then for concatenation, we will add two moves (q0 to q1 for 1 and q1 to q2 for 0) as shown in Figure 3.

Figure 3



Using above algorithm, we can convert NFA to DFA as shown in Figure 4

State	0	1
q0	q0	q0,q1
q0,q1	q0,q2	q0,q1
q0,q2	q0	q0,q1

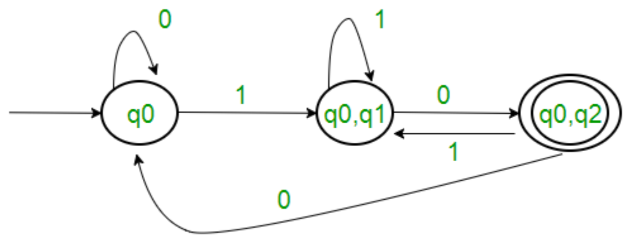
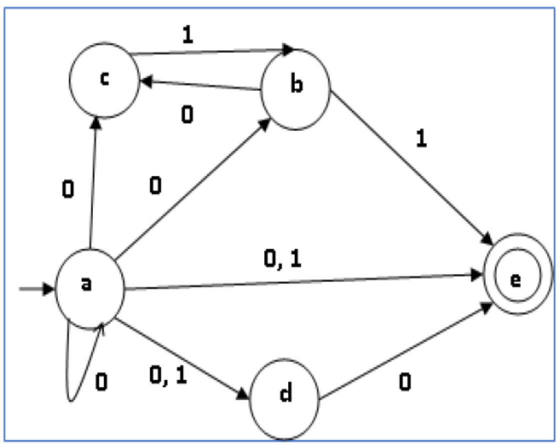


Figure 4

Example

Let us consider the NDFA shown in the figure below.

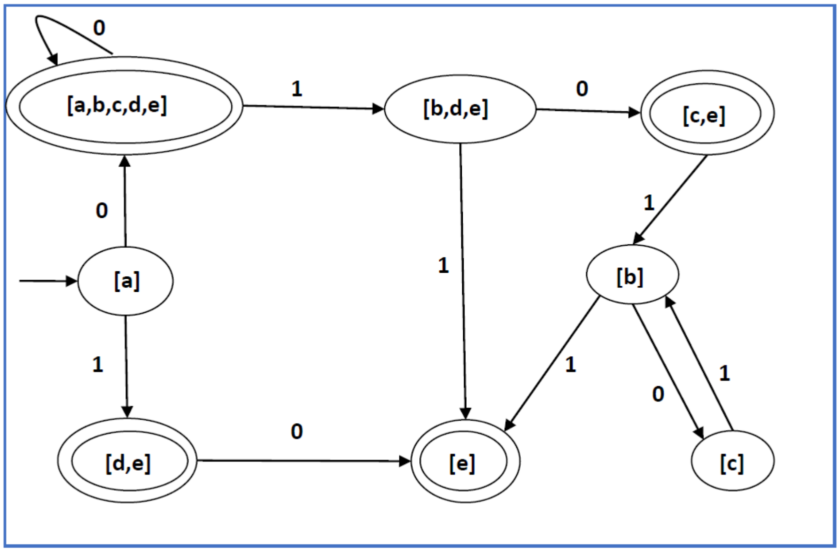


q	δ(q,0)	δ(q,1)
a	{a,b,c,d,e}	{d,e}
b	{c}	{e}
c	∅	{b}
d	{e}	∅
e	∅	∅

Using the above algorithm, we find its equivalent DFA. The state table of the DFA is shown in below.

q	δ(q,0)	δ(q,1)
[a]	[a,b,c,d,e]	[d,e]
[a,b,c,d,e]	[a,b,c,d,e]	[b,d,e]
[d,e]	[e]	Ø
[b,d,e]	[c,e]	[e]
[e]	Ø	Ø
[c, e]	Ø	[b]
[b]	[c]	[e]
[c]	Ø	[b]

The state diagram of the DFA is as follows −



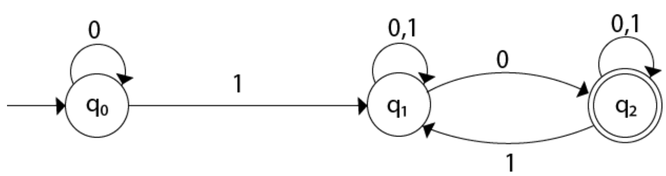
State diagram of DFA

Simple Algorithm

Step 1: Initially Q' = ϕ
Step 2: Add q0 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

Convert the given NFA to DFA.



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

State	0	1
$\rightarrow q_0$	q_0	q_1
q_1	$\{q_1, q_2\}$	q_1
$*q_2$	q_2	$\{q_1, q_2\}$

Now we will obtain δ' transition for state q0.

$$\delta'([q_0], 0) = [q_0]$$
$$\delta'([q_0], 1) = [q_1]$$

The δ' transition for state q1 is obtained as:

$$\delta'([q_1], 0) = [q_1, q_2] \quad \text{(new state generated)}$$
$$\delta'([q_1], 1) = [q_1]$$

The δ' transition for state q_2 is obtained as:

$$\delta'([q_2], 0) = [q_2]$$
$$\delta'([q_2], 1) = [q_1, q_2]$$

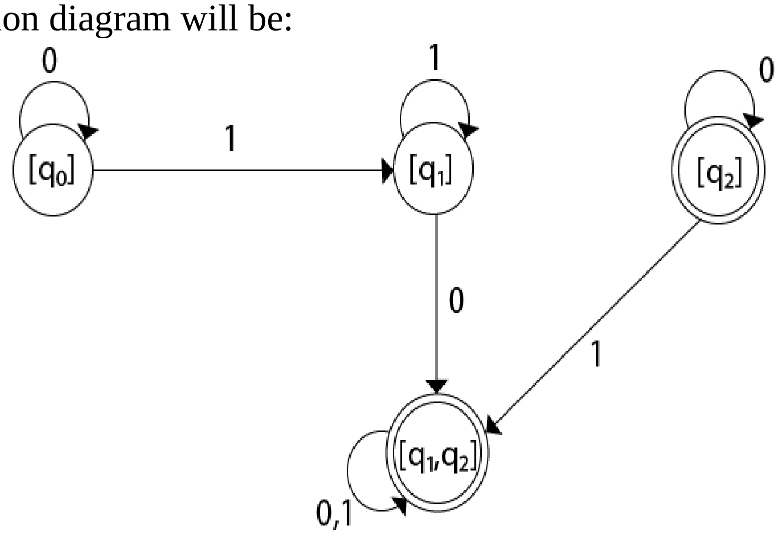
Now we will obtain δ' transition on $[q_1, q_2]$.

$$\delta'([q_1, q_2], 0) = \delta(q_1, 0) \cup \delta(q_2, 0)$$
$$= \{q_1, q_2\} \cup \{q_2\}$$
$$= [q_1, q_2]$$
$$\delta'([q_1, q_2], 1) = \delta(q_1, 1) \cup \delta(q_2, 1)$$
$$= \{q_1\} \cup \{q_1, q_2\}$$
$$= \{q_1, q_2\}$$
$$= [q_1, q_2]$$

The state $[q_1, q_2]$ is the final state as well because it contains a final state q_2. The transition table for the constructed DFA will be:

State	**0**	**1**
$\rightarrow[q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1, q_2]$	$[q_1]$
$*[q_2]$	$[q_2]$	$[q_1, q_2]$
$*[q_1, q_2]$	$[q_1, q_2]$	$[q_1, q_2]$

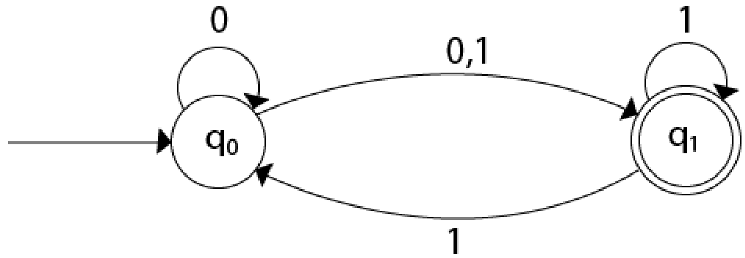
The Transition diagram will be:



The state **q_2** can be eliminated because **q_2** is an unreachable state.

Example

Convert the given NFA to DFA.



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

State	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_1\}$
$*q_1$	ϕ	$\{q_0, q_1\}$

Now we will obtain δ' transition for state q_0.

$$\delta'([q_0], 0) = \{q_0, q_1\}$$
$$= [q_0, q_1] \quad \text{(new state generated)}$$
$$\delta'([q_0], 1) = \{q_1\} = [q_1]$$

The δ' transition for state q_1 is obtained as:

$$\delta'([q_1], 0) = \phi$$
$$\delta'([q_1], 1) = [q_0, q_1]$$

Now we will obtain δ' transition on $[q_0, q_1]$.

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

Similarly,

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

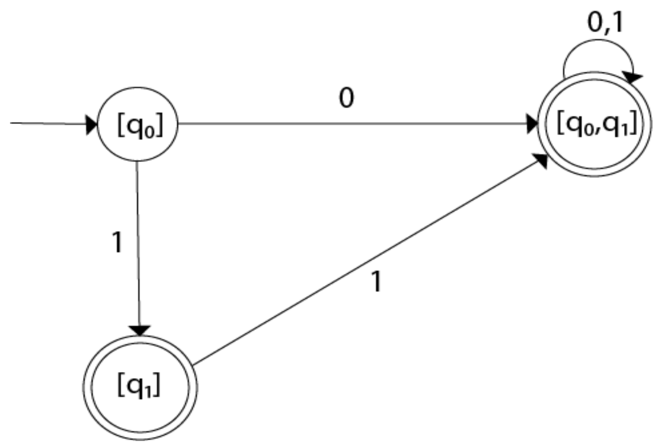
As in the given NFA, q1 is a final state, then in DFA wherever, q1 exists that state becomes a final state. Hence in the DFA, final states are [q1] and [q0, q1]. Therefore set of final states F = {[q1], [q0, q1]}.

The transition table for the constructed DFA will be:

State	0	1
$\rightarrow[q_0]$	$[q_0, q_1]$	$[q_1]$

*[q_1]	ϕ	[q_0, q_1]
*[q_0, q_1]	[q_0, q_1]	[q_0, q_1]

The Transition diagram will be:

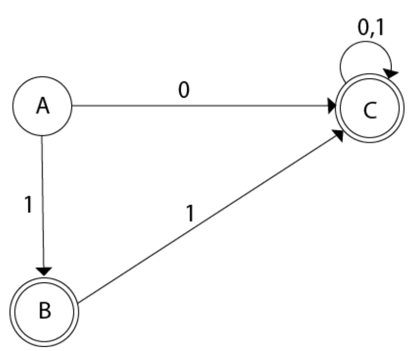


Even we can change the name of the states of DFA.

Suppose

A = [q_0]
B = [q_1]
C = [q_0, q_1]

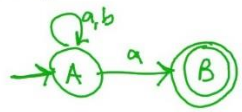
With these new names the DFA will be as follows:



Example

$L = \{$ ends with an "a" $\}$

NFA

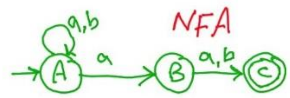


	a	b
→A	{A,B}	{A}
*B	{}	{}

DFA

	a	b
[A]	[AB]	[A]
[AB]	[AB]	[A]

Example



NFA

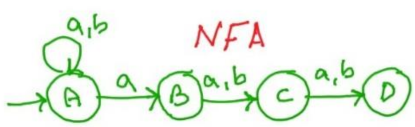
	a	b
-A	{A,B}	{A}
B	{C}	{C}
*C	{}	{}

NFA table

	a	b
-[A]	[AB]	[A]
[AB]	[ABC]	[AC]
[AC]	[AB]	[A]
[ABC]	[ABC]	[AC]

DFA table

Example



NFA

	a	b
-A	{A,B}	{A}
B	{C}	{C}
C	{D}	{D}
*D	{}	{}

	a	b
-[A]	[AB]	[A]
[AB]	[ABC]	[AC]
[AC]	[ABD]	[AD]
*[AD]	[AB]	[A]
[ABC]	[ABCD]	[ACD]
*[ABD]	[ABC]	[AC]
*[ACD]	[ABD]	[AD]
*[ABCD]	[ABD]	[ACD]

DFA

Conversion from NFA with ε to DFA

Non-deterministic finite automata(NFA) is a finite automata where for some cases when a specific input is given to the current state, the machine goes to multiple states or more than 1 states. It can contain ε move. It can be represented as M = { Q, ∑, δ, q_0, F}.

Where

1. Q: finite set of states
2. ∑: finite set of the input symbol
3. q_0: initial state
4. F: final state
5. δ: Transition function

NFA with ∈ move: If any FA contains ε transaction or move, the finite automata is called NFA with ∈ move.

ε-closure: ε-closure for a given state A means a set of states which can be reached from the state A with only ε(null) move including the state A itself.

Steps for converting NFA with ε to DFA:

Step 1: We will take the ε-closure for the starting state of NFA as a starting state of DFA.

Step 2: Find the states for each input symbol that can be traversed from the present. That means the union of transition value and their closures for each state of NFA present in the current state of DFA.

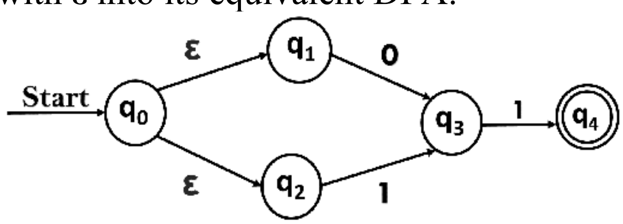
Step 3: If we found a new state, take it as current state and repeat step 2.

Step 4: Repeat Step 2 and Step 3 until there is no new state present in the transition table of DFA.

Step 5: Mark the states of DFA as a final state which contains the final state of NFA.

Example

Convert the NFA with ε into its equivalent DFA.



Solution:

Let us obtain ε-closure of each state.

ε-closure $\{q_0\} = \{q_0, q_1, q_2\}$
ε-closure $\{q_1\} = \{q_1\}$
ε-closure $\{q_2\} = \{q_2\}$
ε-closure $\{q_3\} = \{q_3\}$
ε-closure $\{q_4\} = \{q_4\}$

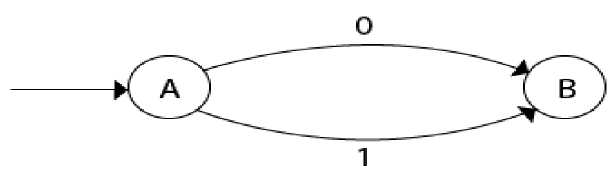
Now, let ε-closure $\{q_0\} = \{q_0, q_1, q_2\}$ be state A.

Hence

$\delta'(A, 0) = \varepsilon\text{-closure}\ \{\delta((q0, q1, q2), 0)\ \}$
$= \varepsilon\text{-closure}\ \{\delta(q0, 0) \cup \delta(q1, 0) \cup \delta(q2, 0)\ \}$
$= \varepsilon\text{-closure}\ \{q3\}$
$= \{q3\}$ **call it as state B**.

$\delta'(A, 1) = \varepsilon\text{-closure}\ \{\delta((q0, q1, q2), 1)\ \}$
$= \varepsilon\text{-closure}\ \{\delta((q0, 1) \cup \delta(q1, 1) \cup \delta(q2, 1)\ \}$
$= \varepsilon\text{-closure}\ \{q3\}$
$= \{q3\} = B$.

The partial DFA will be



Now,
$\delta'(B, 0) = \varepsilon\text{-closure}\ \{\delta(q3, 0)\ \}$
$= \phi$
$\delta'(B, 1) = \varepsilon\text{-closure}\ \{\delta(q3, 1)\ \}$
$= \varepsilon\text{-closure}\ \{q4\}$
$= \{q4\}$ **i.e. state C**

For state C:

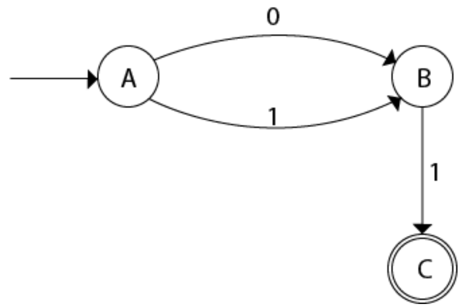
1. $\delta'(C, 0) = \varepsilon\text{-closure}\ \{\delta(q4, 0)\ \}$

   $= \phi$

2. $\delta'(C, 1) = \varepsilon\text{-closure}\ \{\delta(q4, 1)\ \}$

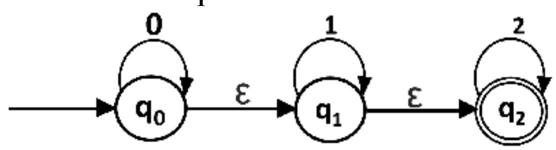
   $= \phi$

The DFA will be,



Example 2:

Convert the given NFA into its equivalent DFA.



Solution: Let us obtain the ε-closure of each state.

ε-closure$(q_0) = \{q_0, q_1, q_2\}$
ε-closure$(q_1) = \{q_1, q_2\}$
ε-closure$(q_2) = \{q_2\}$

Now we will obtain δ' transition.

Let ε-closure(q0) = $\{q_0, q_1, q_2\}$ call it as **state A**.

δ'(A, 0) = ε-closure{δ((q0, q1, q2), 0)}
= ε-closure{δ(q0, 0) ∪ δ(q1, 0) ∪ δ(q2, 0)}
= ε-closure{q0}
= {q0, q1, q2}

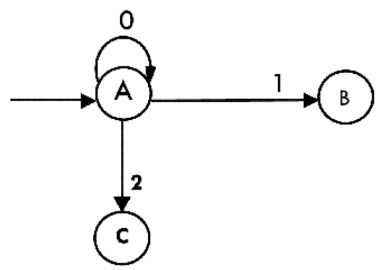
δ'(A, 1) = ε-closure{δ((q0, q1, q2), 1)}
= ε-closure{δ(q0, 1) ∪ δ(q1, 1) ∪ δ(q2, 1)}
= ε-closure{q1}
= {q1, q2} **call it as state B**

δ'(A, 2) = ε-closure{δ((q0, q1, q2), 2)}
= ε-closure{δ(q0, 2) ∪ δ(q1, 2) ∪ δ(q2, 2)}
= ε-closure{q2}
= {q2} **call it state C**

Thus, we have obtained

1. $\delta'(A, 0) = A$
2. $\delta'(A, 1) = B$
3. $\delta'(A, 2) = C$

The partial DFA will be:



Now we will find the transitions on states B and C for each input.

Hence

$\delta'(B, 0) = \varepsilon\text{-closure}\{\delta((q1, q2), 0)\}$
$= \varepsilon\text{-closure}\{\delta(q1, 0) \cup \delta(q2, 0)\}$
$= \varepsilon\text{-closure}\{\phi\}$
$= \phi$

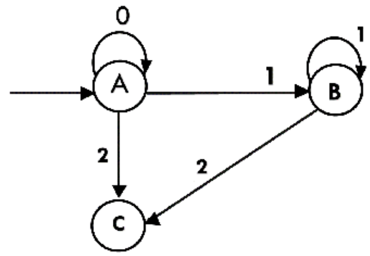
$\delta'(B, 1) = \varepsilon\text{-closure}\{\delta((q1, q2), 1)\}$
$= \varepsilon\text{-closure}\{\delta(q1, 1) \cup \delta(q2, 1)\}$
$= \varepsilon\text{-closure}\{q1\}$
$= \{q1, q2\}$ **i.e. state B itself**

$\delta'(B, 2) = \varepsilon\text{-closure}\{\delta((q1, q2), 2)\}$
$= \varepsilon\text{-closure}\{\delta(q1, 2) \cup \delta(q2, 2)\}$
$= \varepsilon\text{-closure}\{q2\}$
$= \{q2\}$ **i.e. state C itself**

Thus, we have obtained

1. $\delta'(B, 0) = \phi$
2. $\delta'(B, 1) = B$
3. $\delta'(B, 2) = C$

The partial transition diagram will be



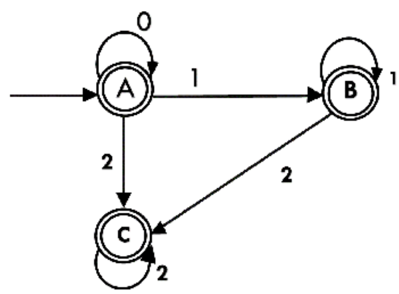
Now we will obtain transitions for C:

$\delta'(C, 0) = \varepsilon\text{-closure}\{\delta(q2, 0)\}$
$= \varepsilon\text{-closure}\{\phi\}$
$= \phi$

$\delta'(C, 1) = \varepsilon\text{-closure}\{\delta(q2, 1)\}$
$= \varepsilon\text{-closure}\{\phi\}$
$= \phi$

$\delta'(C, 2) = \varepsilon\text{-closure}\{\delta(q2, 2)\}$
$= \{q2\}$

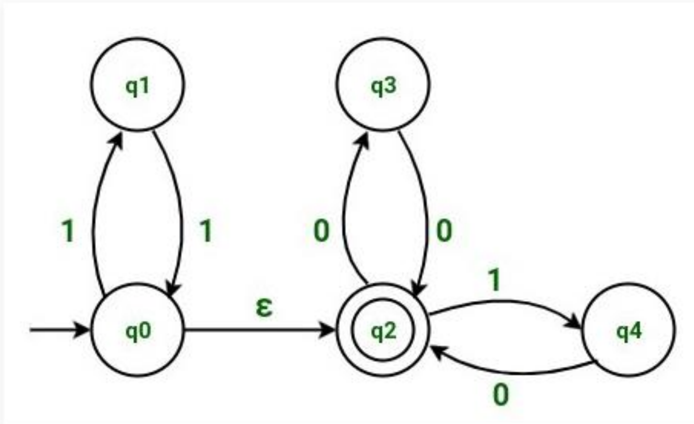
Hence the DFA is



As A = {q0, q1, q2} in which final state q2 lies hence A is final state. B = {q1, q2} in which the state q2 lies hence B is also final state. C = {q2}, the state q2 lies hence C is also a final state.

Example: Convert epsilon-NFA to NFA.

Consider the example having states q0, q1, q2, q3, and q4.



In the above example, we have 5 states named as q0, q1, q2, q3 and q4. Initially, we have q0 as start state and q2 as final state. We have q1, q3 and q4 as intermediate states.

Transition table for the above NFA is:

STATES/INPUT	INPUT 0	INPUT 1	INPUT EPSILON
q0	–	q1	q2
q1	–	q0	–
q2	q3	q4	–
q3	q2	–	–
q4	q2	–	–

According to the transition table above,

- state q0 on getting input 1 goes to state q1.
- State q0 on getting input as a null move *(i.e. an epsilon move)* goes to state q2.
- State q1 on getting input 1 goes to state q0.
- Similarly, state q2 on getting input 0 goes to state q3, state q2 on getting input 1 goes to state q4.
- Similarly, state q3 on getting input 0 goes to state q2.
- Similarly, state q4 on getting input 0 goes to state q2.

We can see that we have an epsilon move from state q0 to state q2, which is to be removed.

To remove epsilon move from state q0 to state q1, we will follow the steps mentioned below.

Step-1:

Considering the epsilon move from state q0 to state q2. Consider the state q0 as vertex v1 and state q2 as vertex v2.

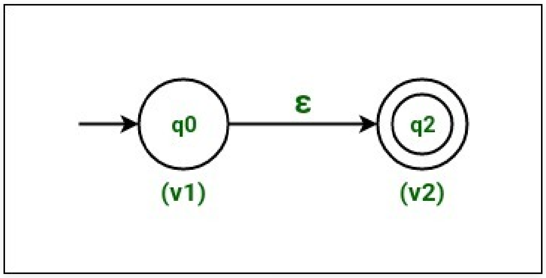


Figure – State q0 as vertex v1 and state q2 as vertex v2

Step-2:

Now find all the moves that starts from vertex v2 *(i.e. state q2)*.

After finding the moves, duplicate all the moves that start from vertex v2 *(i.e state q2)* with the same input to start from vertex v1 *(i.e. state q0)* and remove the epsilon move from vertex v1 *(i.e. state q0)* to vertex v2 *(i.e. state q2)*.

Since state q2 on getting input 0 goes to state q3.
Hence on duplicating the move, we will have state q0 on getting input 0 also to go to state q3.

Similarly state q2 on getting input 1 goes to state q4.
Hence on duplicating the move, we will have state q0 on getting input 1 also to go to state q4.

So, NFA after duplicating the moves is:

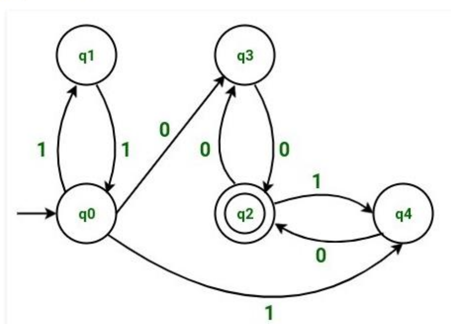


Figure – NFA on duplicating moves

Step-3:

Since vertex v1 *(i.e. state q0)* is a start state. Hence we will also make vertex v2 *(i.e. state q2)* as a start state.

Note that state q2 will also remain as a final state as we had initially.

NFA after making state q2 also as a start state is:

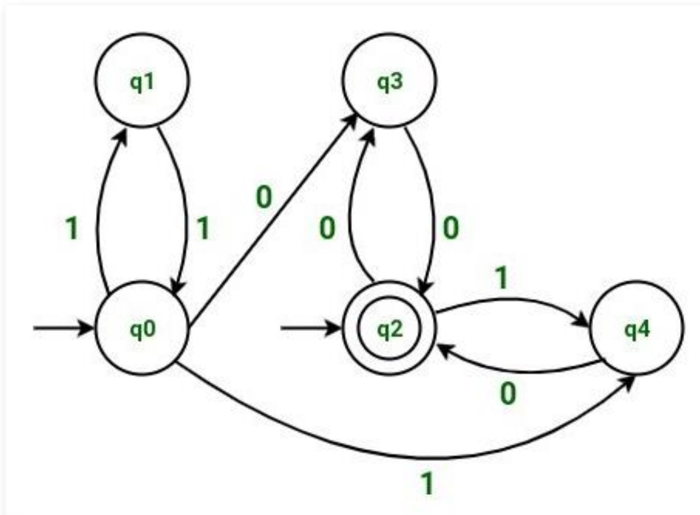


Figure – NFA after making state q2 as a start state

Step-4:

Since vertex v2 *(i.e. state q2)* is a final state. Hence we will also make vertex v1 *(i.e. state q0)* as a final state.

Note that state q0 will also remain as a start state as we had initially.

After making state q0 also as a final state, the resulting NFA is:

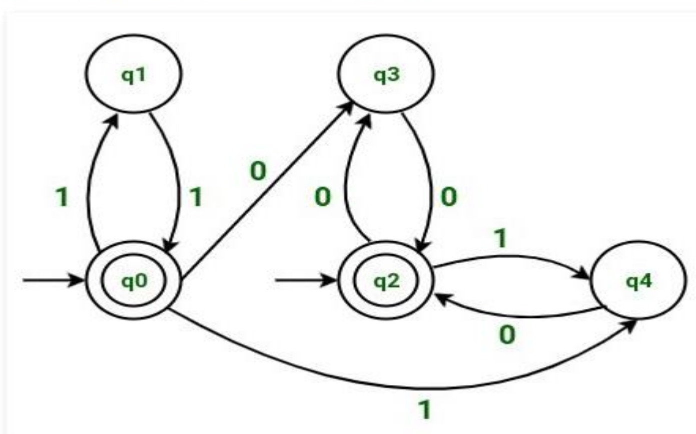


Figure – Resulting NFA (state q0 as a final state)

The transition table for the above resulting NFA is:

STATES/INPUT	INPUT 0	INPUT 1
q0	q3	q1,q4
q1	–	q0
q2	q3	q4
q3	q2	–
q4	q2	–