The Subset Construction Algorithm

The algorithm for constructing a DFA from a given NFA such that it recognizes the same language is called subset construction.
The reason is that each state of the DFA machine corresponds to a set of states of the NFA.
The DFA keeps in a particular state all possible states to which the NFA makes a transition on the given input symbol.
In other words, after processing a sequence of input symbols the DFA is in a state that actually corresponds to a set of states from the NFA reachable from the starting symbol on the same inputs.

There are three operations that can be applied on NFAstates:

Operation	Definition
ε -closure(s)	set of NFA states reachable from state s on ε-transition
ε -closure(T)	set of NFA states reachable from some s in T on ε-transition
move(T,a)	set of NFA states to which there is transition on input a from some state s in the set T

The starting state of the automaton is assumed to be s0. The ε-closure(s) operation computes exactly all the states reachable from a particular state on seeing an input symbol. When such operations are defined the states to which our automaton can make a transition from set T on input a can be simply specified as: ε-closure(move(T,a))

Subset Construction Algorithm
Initialize: Let -closure(s0) be the only state in Dstates (of the DFA)
Repeat: while there are unmarked states T in Dstate s do
mark T
for each symbol ado
U= -closure(move(T,a))if Uis not in Dstatesthenadd Uas unmarked state in DstatesDtran[T, a] = Uendend

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, q0, F}.

Where

1. Q: finite set of states
2. ∑: finite set of the input symbol
3. q0: 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.

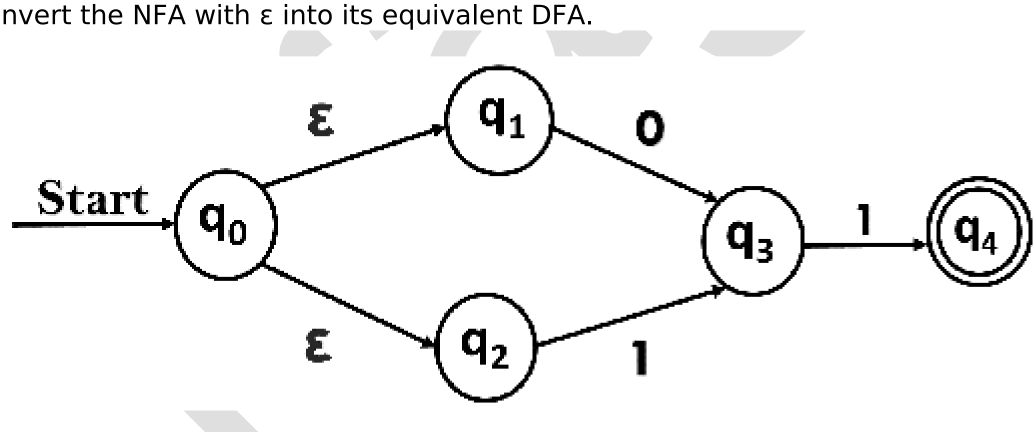
Step 3: If we found a new state, take it as current state and repeat step 2.

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 1:

Convert the NFA with ε into its equivalent DFA.



Solution:

Let us obtain ε-closure of each state.

1. ε-closure {q0} = {**q0, q1, q2}**
2. ε-closure {q1} = {q1}
3. ε-closure {q2} = {q2}
4. ε-closure {q3} = {q3}

5. ε-closure {q4} = {q4}

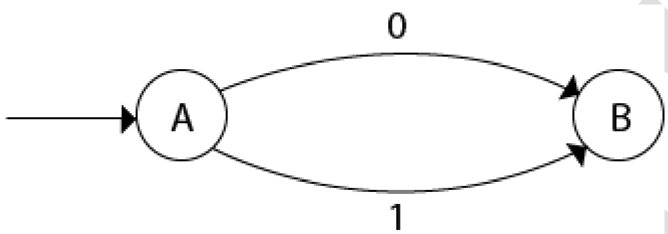
Now, let ε-closure {q0} = {q0, q1, q2} be state A.

Hence

δ'(A, 0) = ε-closure {δ((q0, q1, q2), 0) }
= ε-closure {δ(q0, 0) ∪ δ(q1, 0) ∪ δ(q2, 0) }
= ε-closure {q3}
= {q3} call it as state B.

δ'(A, 1) = ε-closure {δ((q0, q1, q2), 1) }
= ε-closure {δ((q0, 1) ∪ δ(q1, 1) ∪ δ(q2, 1) }
= ε-closure {q3}
= {q3} = B.

The partial DFA will be



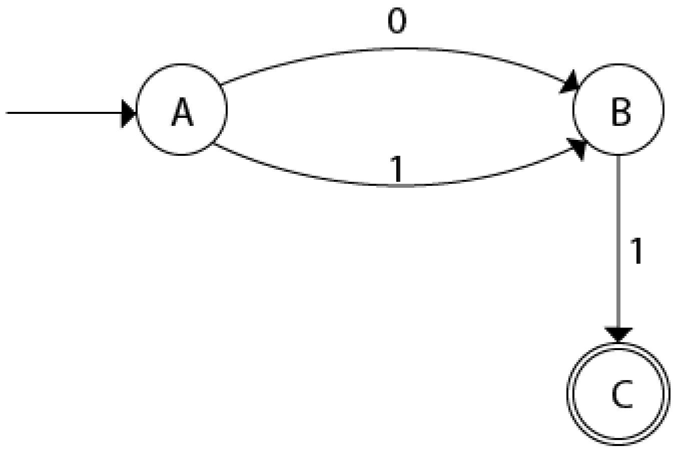
Now, q3

δ'(B, 0) = ε-closure {δ(q3, 0) }
= ϕ
δ'(B, 1) = ε-closure {δ(q3, 1) }
= ε-closure {q4}
= {q4} i.e. state C

For state C:

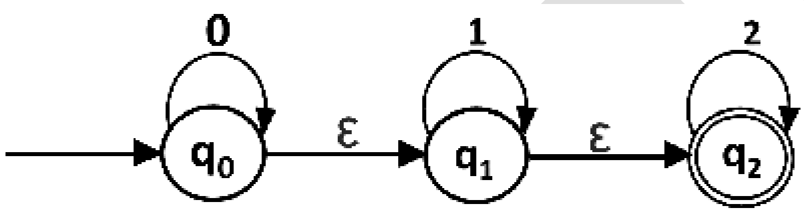
1. δ'(C, 0) = ε-closure {δ(q4, 0) }
2. = ϕ
3. δ'(C, 1) = ε-closure {δ(q4, 1) }
4. = ϕ

The DFA will be,



Example 2:

Convert the given NFA into its equivalent DFA.



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

1. ε-closure(q0) = {q0, q1, q2}

Now we will obtain δ' transition. Let ε-closure(q0) = {q0, q1, q2} 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} =A

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

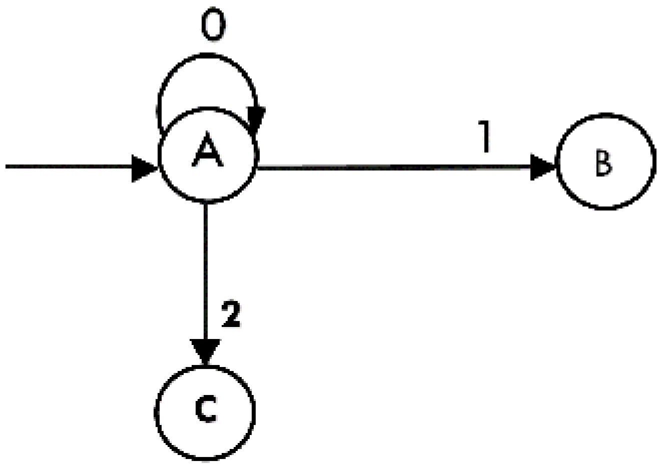
Thus we have obtained

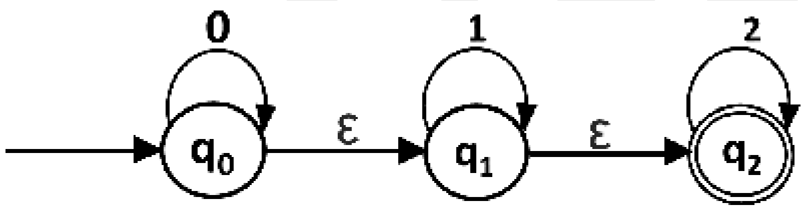
1. δ'(A, 0) = A

2. δ'(A, 1) = B

3. δ'(A, 2) = C

The partial DFA will be:





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

Hence {q1, q2} B

δ'(B, 0) = ε-closure{δ((q1, q2), 0)}
= ε-closure{δ(q1, 0) ∪ δ(q2, 0)}
= ε-closure{ϕ}
= ϕ

δ'(B, 1) = ε-closure{δ((q1, q2), 1)}
= ε-closure{δ(q1, 1) ∪ δ(q2, 1)}
= ε-closure{q1}

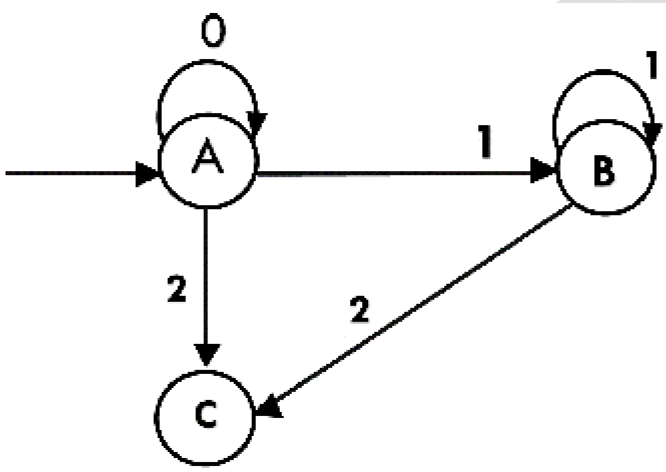
$= \{q1, q2\}$ i.e. state B itself

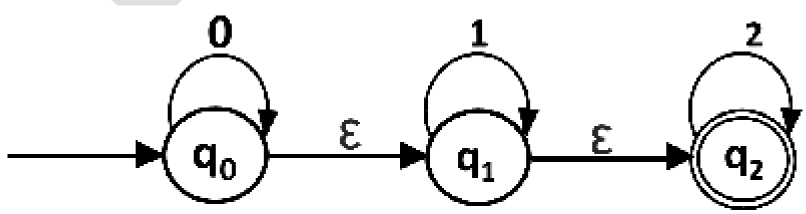
$\delta'(B, 2) = \varepsilon\text{-closure}\{\delta((q1, q2), 2)\}$
$= \varepsilon\text{-closure}\{\delta(q1, 2) \cup \delta(q2, 2)\}$
$= \varepsilon\text{-closure}\{q2\}$
$= \{q2\}$ i.e. state C itself

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





{q2} C

Now we will obtain transitions for C:

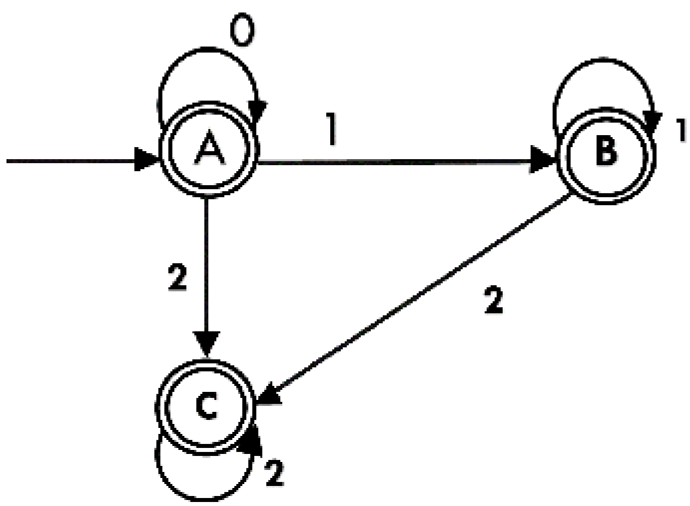
$\delta'(C, 0) = \varepsilon\text{-closure}\{\delta(q2, 0)\}$

= ε-closure{ϕ}
= ϕ

δ'(C, 1) = ε-closure{δ(q2, 1)}
= ε-closure{ϕ}
= ϕ

δ'(C, 2) = ε-closure{δ(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.

Conversion from NFA to DFA

In this section, we will discuss the method of converting NFA to its equivalent 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, ∑, δ, q0, F) is an NFA which accepts the language L(M). There should be equivalent DFA denoted by M' = (Q', ∑', q0', δ', F') such that L(M) = L(M').

Steps for converting NFA to DFA:

Step 1: Initially Q' = ϕ

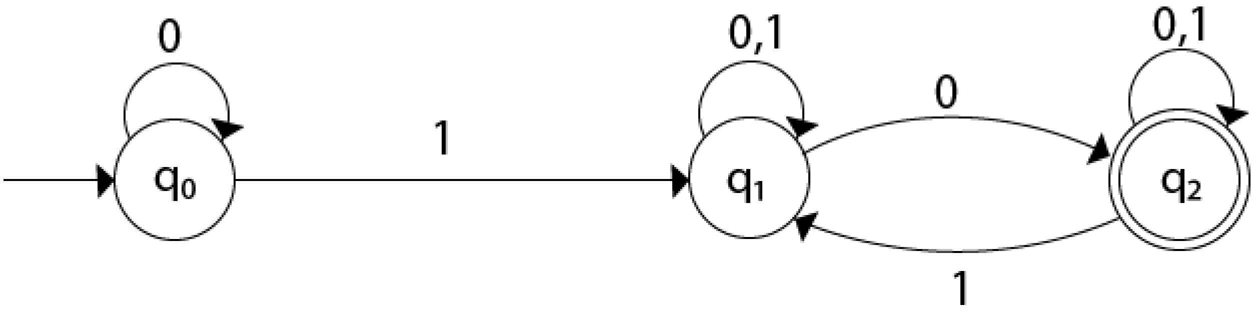
Step 2: Add q0 of NFA to Q'. Then find the transitions from this start state.

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 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, q2}	q1
*q2	q2	{q1, q2}

Now we will obtain δ' transition for state q0.

1. δ'([q0], 0) = [q0]
2. δ'([q0], 1) = [q1]

The δ' transition for state q1 is obtained as:

1. δ'([q1], 0) = [q1, q2]   (new state generated)
2. δ'([q1], 1) = [q1]

The δ' transition for state q2 is obtained as:

1. δ'([q2], 0) = [q2]
2. δ'([q2], 1) = [q1, q2]

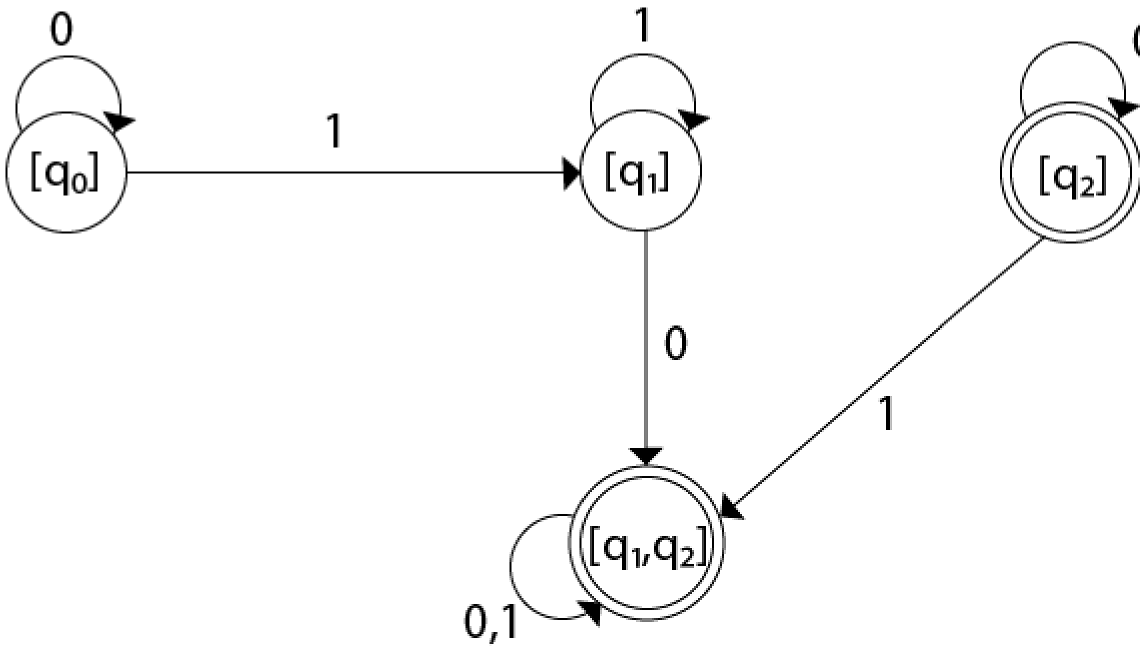
Now we will obtain δ' transition on [q1, q2].

1. δ'([q1, q2], 0) = δ(q1, 0) ∪ δ(q2, 0)
2.   = {q1, q2} ∪ {q2}
3.   = [q1, q2]
4. δ'([q1, q2], 1) = δ(q1, 1) ∪ δ(q2, 1)
5.   = {q1} ∪ {q1, q2}
6.   = {q1, q2}
7.   = [q1, q2]

The state [q1, q2] is the final state as well because it contains a final state q2. The transition table for the constructed DFA will be:

State	0	1
→[q0]	[q0]	[q1]
[q1]	[q1, q2]	[q1]
*[q2]	[q2]	[q1, q2]
*[q1, q2]	[q1, q2]	[q1, q2]

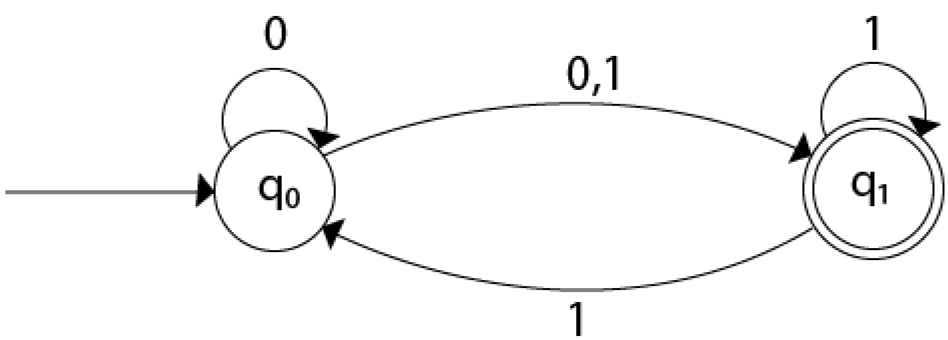
The Transition diagram will be:



The state q2 can be eliminated because q2 is an unreachable state.

Example 2:

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}

Now we will obtain δ' transition for state q0.

1. δ'([q0], 0) = {q0, q1}
2. = [q0, q1] (new state generated)
3. δ'([q0], 1) = {q1} = [q1]

The δ' transition for state q1 is obtained as:

1. δ'([q1], 0) = ϕ
2. δ'([q1], 1) = [q0, q1]

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

1. δ'([q0, q1], 0) = δ(q0, 0) ∪ δ(q1, 0)
2. = {q0, q1} ∪ ϕ
3. = {q0, q1}
4. = [q0, q1]

Similarly,

1. δ'([q0, q1], 1) = δ(q0, 1) ∪ δ(q1, 1)
2. = {q1} ∪ {q0, q1}
3. = {q0, q1}
4. = [q0, q1]

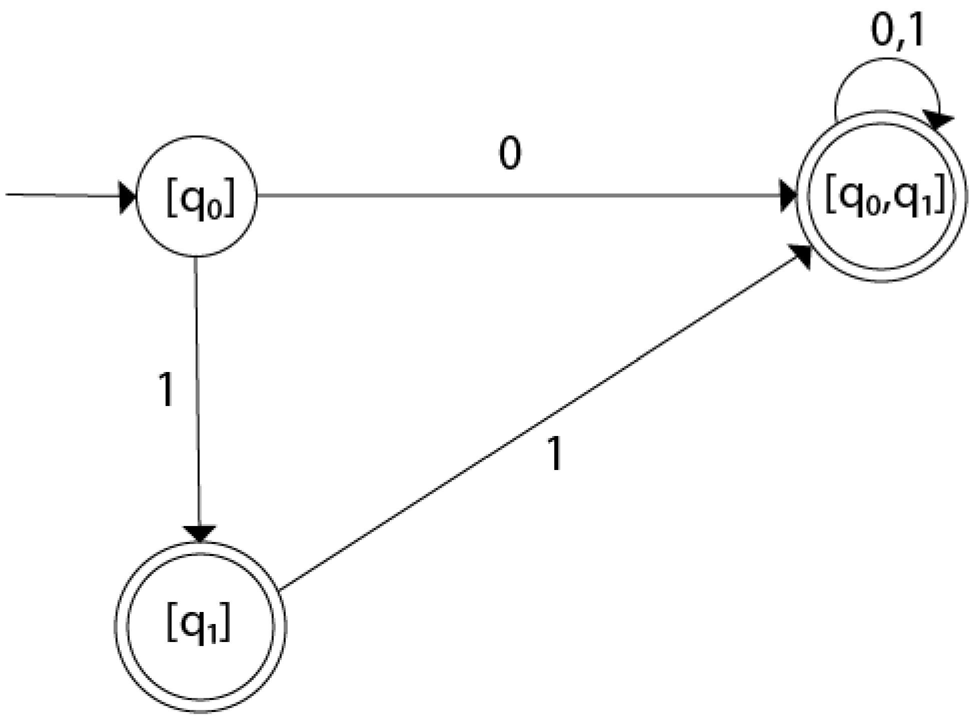
As in the given NFA, q1 is a final state, then in DFA wherever, q1 exists that state becomes a final state. Hence in the DFA, final states are [q1] and [q0, q1]. Therefore set of final states F = {[q1], [q0, q1]}.

The transition table for the constructed DFA will be:

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

*[q0, q1]	[q0, q1]	[q0, q1]

The Transition diagram will be:

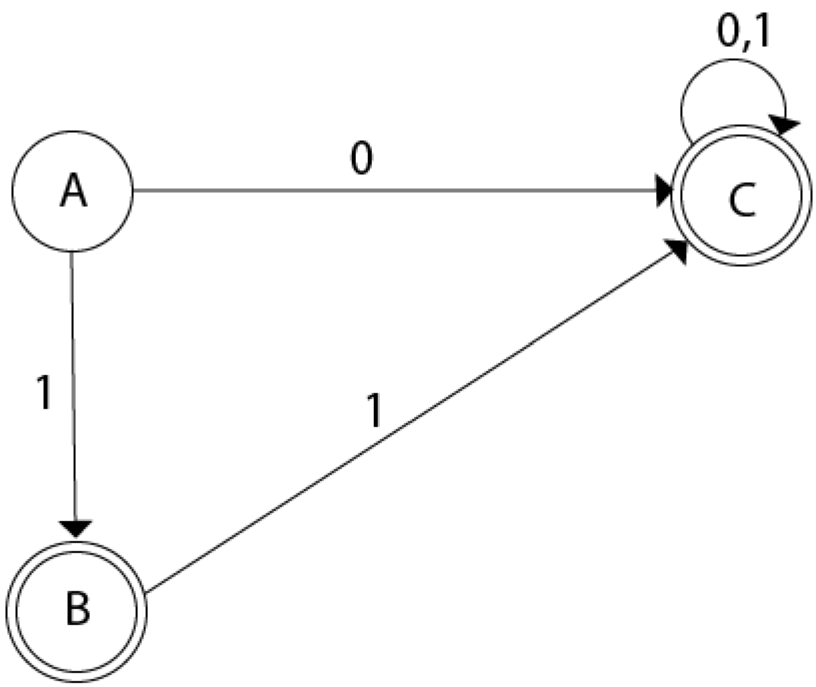


Even we can change the name of the states of DFA.

Suppose

1. A = [q0]
2. B = [q1]
3. C = [q0, q1]

With these new names the DFA will be as follows:


0,1
0
A
C
1
1
B