

**DEPARTMENT OF CSE & IT**  
**23CS6401 / COMPILER DESIGN**  
**YEAR/ SEMESTER : III/VI**  
**UNIT I- INTRODUCTION TO COMPILER DESIGN**

**PART – B (16-Marks)**

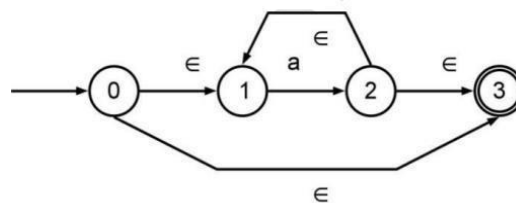
**5. Apply Thompson's construction to convert the regular expression into its equivalent NFA with suitable state transitions. Illustrate regular expression to NFA for the sentence  $(a|b)^*a$**   
Construct the NFA from the  $(a/b)^*a$  using thompson's construction algorithm

Let us assume

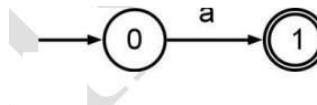
$R1 = (a / b)$  and  $R2 = a$

Thompson's Rules

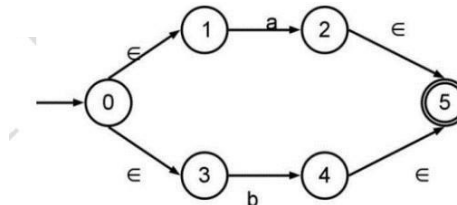
**Rule for  $a^*$**



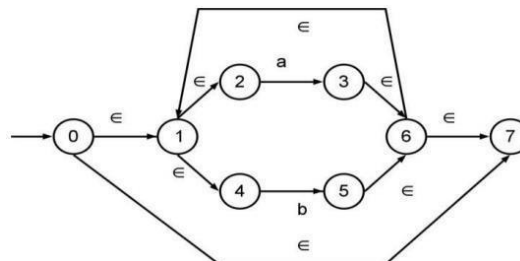
**Rule for  $a$**



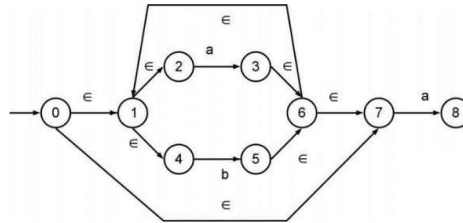
**Rule for  $a / b$**



**$R1: (a / b)^*$**

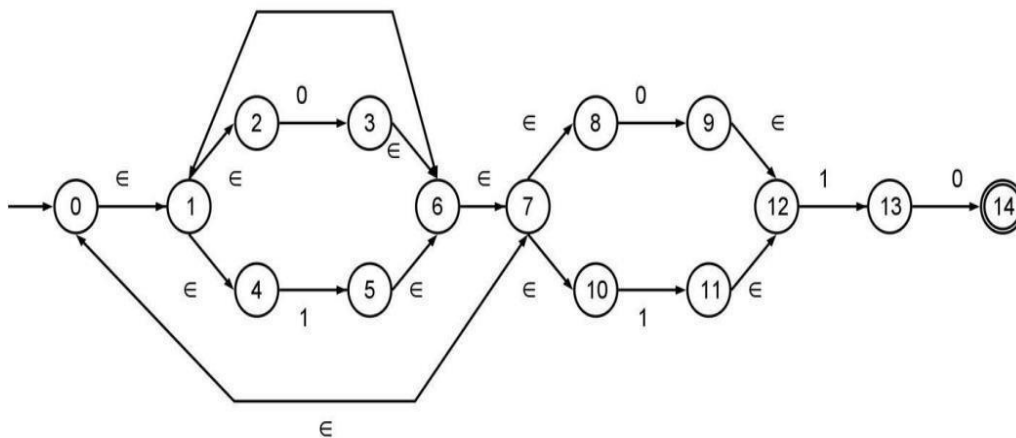


**R1 R2:(a / b)\* a**



**7. Apply DFA construction and minimization methods to derive the minimized DFA for the given regular expression  $(0+1)^*(0+1)$**

**(i) Construction of NFA**



**(ii) Find the DFA States**



$\epsilon$ - closure (0) =	{0,1,2,4,7,8,10} = A
Move (A,0) =	{3,9}
Move (A,1) =	{5,11}
$\epsilon$ - Closure of ( Move (A,0) ) =	{3,6,7,1,2,4,8,10,9,12}
=	{1,2,3,4,6,7,8,9,10,12} = B
$\epsilon$ - closure of ( Move (A,1) ) =	{5,6,7,8,10,1,2,4,11,12}
=	{1,2,4,5,6,7,8,10,11,12} = C
Dtran[A,0]=	B
Dtran[A,1]=	C
Move (B,0) =	{3,9}
Move (B,1) =	{5,11,13}
$\epsilon$ - closure ( Move(B,0) ) =	{3,6,7, 8,10,1,2,4,9,12}
=	{1,2,3,4,6,7,8,9,10,12} = B
$\epsilon$ - closure ( Move(B,1) ) =	{1,2,4,5,6,7,8,10,11,12,13} =D
Dtran[B,0]=	B
Dtran[B,1]=	D
Move (C,0) =	{3,9}
Move (C,1) =	{5,11,13}
$\epsilon$ - closure ( Move(C,0) ) =	B
$\epsilon$ - closure ( Move(C,1) ) =	D
Dtran[C,0]=	B

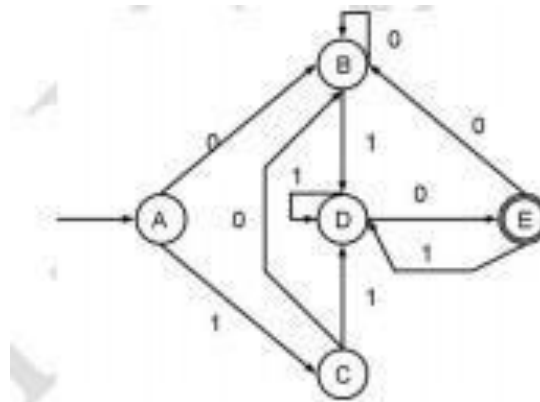


Dtran[C,1]=	D
Move(D,0)=	{3,9,14}
Move(D,1)=	{5,11,13}
$\xi$ -closure(Move(D,0))=	{3,6,7,8,10,1,2,4,9,12,14}
	{1,2,3,4,6,7,8,9,10,12,14}=E
$\xi$ -closure(Move(D,1))=	D
Dtran[D,0]=	E
Dtran[D,1]=	D
Move(E,0)=	{3,9}
Move(E,1)=	{5,11,13}
$\xi$ -closure(Move(E,0))=	B
$\xi$ -closure(Move(E,1))=	D
Dtran[E,0]=	B
Dtran[E,1]=	D

**(iii) Construction of Transition Table**

States	i/p Symbols	
	0	1
->A	B	C
B	B	D
C	B	D
D	E	D
*E	B	D

**(iv) Construction of DFA**



**(v) Minimization of DFA**

= (ABCD) (E)

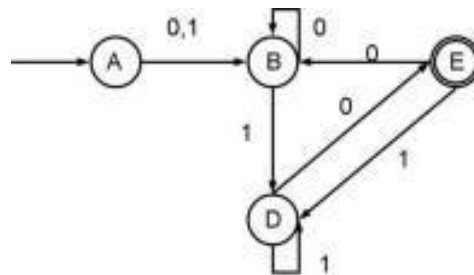
= (ABC) (D) (E)

= (A) (BC) (D) (E)

**(vi) Minimized DTRAN**

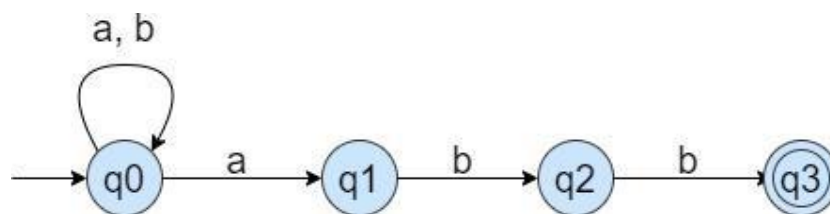
States	i/p Symbols	
	0	1
->A	B	B
B	B	D
D	E	D
*E	B	D

**(vii) Minimized DFA**



**8. Apply the direct DFA construction method to convert the regular expression  $(a+b)^*abb$  into an equivalent DFA.**

The NFA of the regular expression  $(a+b)^*abb$  is given below.



The transition table of the NFA is

State	A	b
→{q0}	{q0, q1}	{q0}
{q1}	—	{q2}

{q2}	—	{q3}
*{q3}	—	—

Convert the above NFA to DFA by expanding the transition table. Start by adding new states to the transition table.

State	a	b
$\rightarrow \{q_0\}$	{q0, q1}	{q0}
{q1}	—	{q2}
{q2}	—	{q3}
*{q3}	—	—
{q0, q1}	{q0, q1}	{q0, q2}
{q0, q2}	{q0, q1}	{q0, q3}
*{q0, q3}	{q0, q1}	{q0}

$q_0$  is our start state. It is impossible to reach  $q_1, q_2, q_3$  from  $q_0$ . Remove  $q_1, q_2, q_3$  from the transition table.

State	a	b
$\rightarrow \{q_0\}$	{q0, q1}	{q0}
{q0, q1}	{q0, q1}	{q0, q2}
{q0, q2}	{q0, q1}	{q0, q3}
*{q0, q3}	{q0, q1}	{q0}

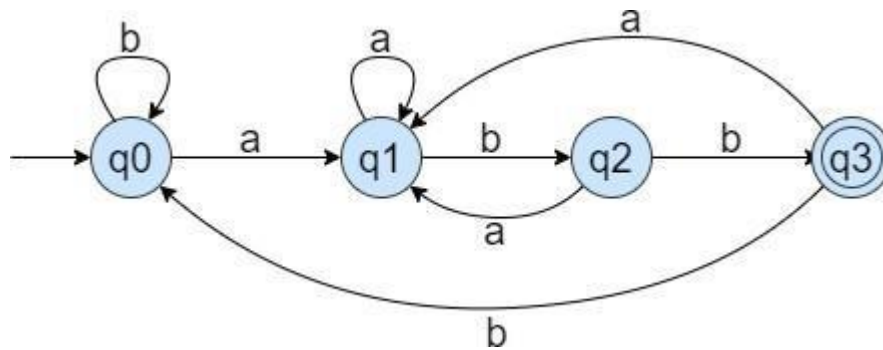
Rename the states to make things simpler.

1. Replace {q0, q1} by {q1}
2. Replace {q0, q2} by {q2}
3. Replace {q0, q3} by {q3}

State	a	B
$\rightarrow \{q_0\}$	{q1}	{q0}

{q1}	{q1}	{q2}
{q2}	{q1}	{q3}
*{q3}	{q1}	{q0}

The above transition table represents a deterministic automata. Construct DFA of  $(a+b)^*abb$  using above transition table.



### 11. Apply Thompson's construction algorithm to construct an NFA for the regular expression $(a/b)^*a(a/b)$

Construct the NFA from the  $(a/b)^*a(a/b)$  using thompson's construction algorithm

Let us assume

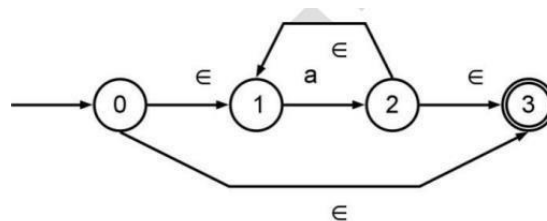
$R1 = (a / b)$

$R2 = a$

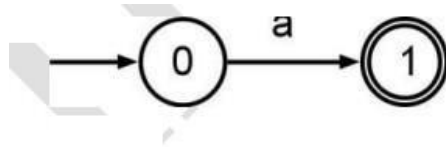
$R3 = (a / b)$

Thompson's Rules

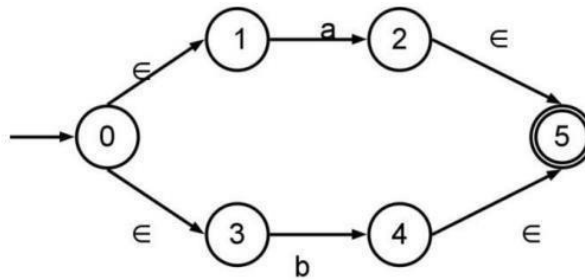
**Rule for  $a^*$**



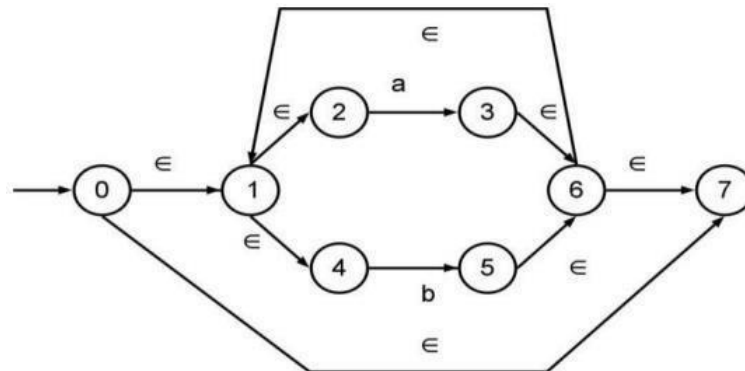
**Rule for a**



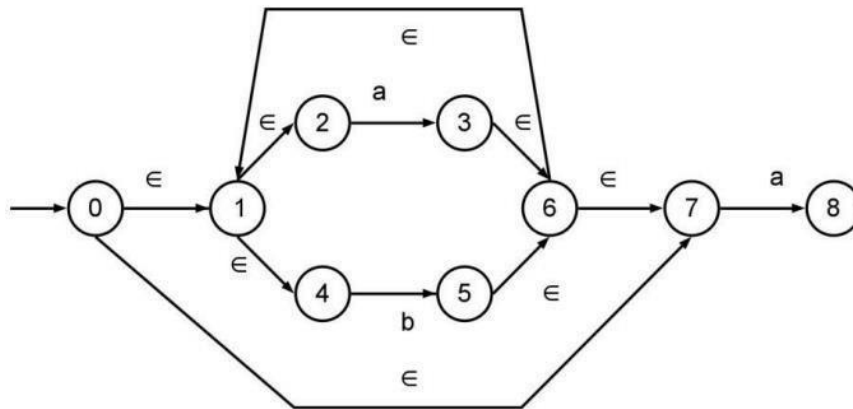
**Rule for a / b**



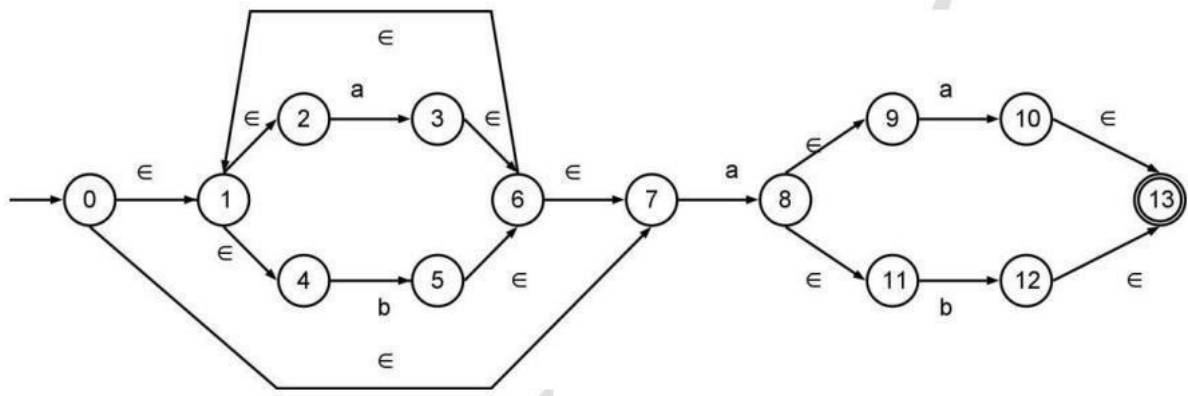
**R1: (a / b)\***



**R1 R2:  $(a / b)^* a$**



**R1 R2 R3:  $(a / b)^* a (a / b)$**



**ii) Apply the principles of transition diagram construction to model relational operators and unsigned numbers in Pascal**

The first diagram (top) shows a sequence of states 12 through 19. State 12 is the start state. Transitions are: 12 to 13 on 'digit'; 13 has a self-loop on 'digit' and goes to 14 on '.'; 14 to 15 on 'digit'; 15 has a self-loop on 'digit' and goes to 16 on 'E'; 16 to 17 on '+ or -'; 17 to 18 on 'digit'; 18 has a self-loop on 'digit' and goes to 19 on 'other'; 19 is the accept state. There are also direct transitions from 13 to 16 and 15 to 18 labeled 'E'.

The second diagram (middle) shows states 20 through 24. State 20 is the start state. Transitions are: 20 to 21 on 'digit'; 21 has a self-loop on 'digit' and goes to 22 on '.'; 22 to 23 on 'digit'; 23 has a self-loop on 'digit' and goes to 24 on 'other'; 24 is the accept state.

The third diagram (bottom) shows states 25 through 27. State 25 is the start state. Transitions are: 25 to 26 on 'digit'; 26 has a self-loop on 'digit' and goes to 27 on 'other'; 27 is the accept state.

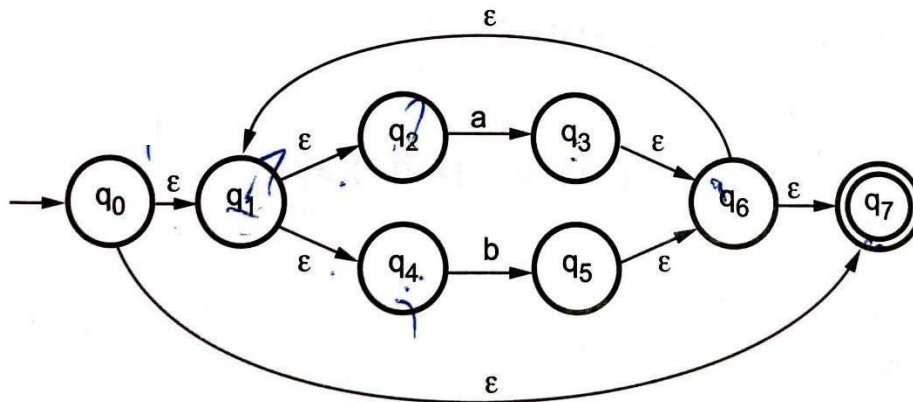
10

**13. Apply DFA minimization techniques to demonstrate that the two given regular expressions are equivalent**

a)  $(a \mid b)^*$

b)  $(a^* \mid b^*)^*$

The NFA with  $\epsilon$  for  $(a \mid b)^*$  will be



We will eliminate  $\epsilon$  moves and convert it to DFA

$$\epsilon\text{-closure } \{q_0\} = \{q_0, q_1, q_2, q_4, q_7\}$$

$$\epsilon\text{-closure } \{q_1\} = \{q_1, q_2, q_4\}$$

$$\epsilon\text{-closure } \{q_2\} = \{q_2\}$$

$$\epsilon\text{-closure } \{q_3\} = \{q_3, q_6, q_1, q_2, q_4, q_7\} = \{q_1, q_2, q_3, q_4, q_6, q_7\}$$

$$\epsilon\text{-closure } \{q_4\} = \{q_4\}$$

$$\epsilon\text{-closure } \{q_5\} = \{q_1, q_2, q_4, q_5, q_6, q_7\}$$

$$\epsilon\text{-closure } \{q_6\} = \{q_1, q_2, q_4, q_6, q_7\}$$

$$\epsilon\text{-closure } \{q_7\} = \{q_7\}$$

consider  $\epsilon\text{-closure } \{q_0\} = \{q_0, q_1, q_2, q_4, q_7\} \longrightarrow$  call it as state A

$$\delta'(A, a) = \epsilon\text{-closure } \{ \delta(A, a) \}$$

$$= \epsilon\text{-closure } \{ \delta(q_0, a), \delta(q_1, a), \delta(q_2, a), \delta(q_4, a), \delta(q_7, a) \}$$

$$= \epsilon\text{-closure } \{ \delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_2, a) \cup \delta(q_4, a) \cup \delta(q_7, a) \}$$

$$= \epsilon\text{-closure } \{q_3\}$$

$$= \{q_1, q_2, q_3, q_4, q_6, q_7\} \longrightarrow$$
 call it as state B

$$\delta'(A, a) = B$$

$$\delta'(A, b) = \epsilon\text{-closure } \{ \delta(q_0, b), \delta(q_1, b), \delta(q_2, b), \delta(q_4, b), \delta(q_7, b) \}$$

$$= \epsilon\text{-closure } \{q_5\}$$

$$= \{q_1, q_2, q_4, q_5, q_6, q_7\} \longrightarrow$$
 call it as state C

$$\delta'(A, b) = C$$

Now consider state B and C for input transitions.

$$\begin{aligned}\delta'(B, a) &= \varepsilon\text{-closure} \{ \delta(q_1, q_2, q_3, q_4, q_5, q_6, q_7), a \} \\ &= \varepsilon\text{-closure} \{ q_3 \} \quad \text{i.e. state B.}\end{aligned}$$

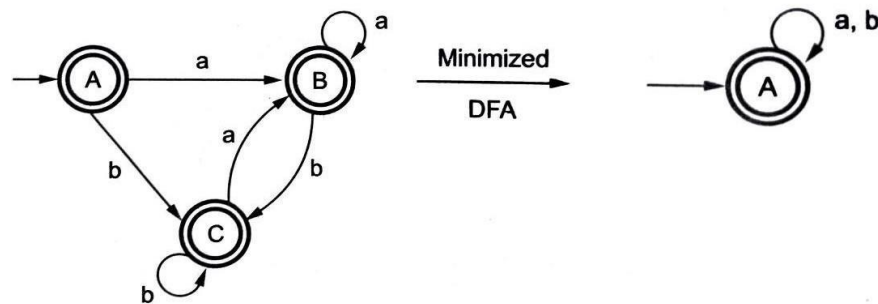
$$\begin{aligned}\delta'(B, b) &= \varepsilon\text{-closure} \{ \delta(q_1, q_2, q_3, q_4, q_6, q_7), b \} \\ &= \varepsilon\text{-closure} \{ q_5 \} \quad \text{i.e. state C.}\end{aligned}$$

**Similarly,**

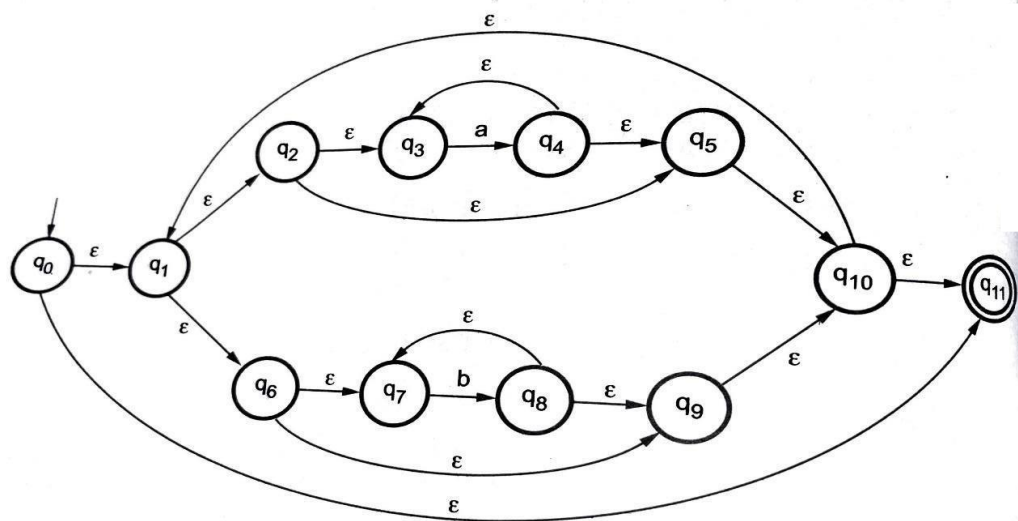
$$\begin{aligned}\delta'(C, a) &= \varepsilon\text{-closure} \{ \delta(q_1, q_2, q_4, q_5, q_6, q_7), a \} \\ &= \text{state B}\end{aligned}$$

$$\delta'(C, b) = \text{state C}$$

The DFA will be



The NFA with  $\varepsilon$  for  $(a^* | b^*)^*$  will be



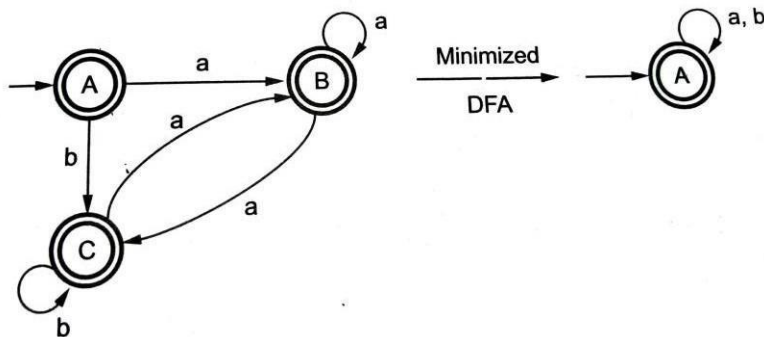
$$\varepsilon\text{-closure} \{ q_0 \} = \{ q_0, q_1, q_2, q_3, q_5, q_6, q_7, q_9, q_{10}, q_{11} \} \longrightarrow \text{call it as state A}$$

$$\delta'(A, a) = \varepsilon\text{-closure} \{ \delta(A, a) \}$$

**Vision** To produce demand driven, quality conscious and globally recognized computer professionals through education, innovation and collaborative research.

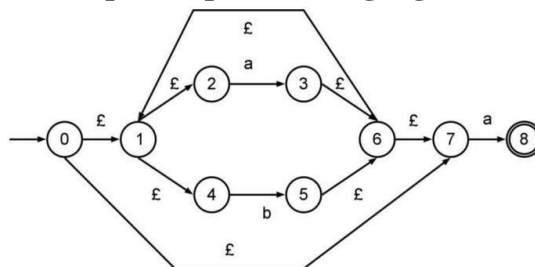
$= \epsilon\text{-closure} \{ (q_0, q_1, q_2, q_3, q_5, q_6, q_7, q_9, q_{10}, q_{11}), a \}$   
 $= \epsilon\text{-closure} \{ q_4 \}$   
 $= \{ q_4, q_5, q_{10}, q_1, q_2, q_3, q_6, q_7, q_9, q_{11} \} \longrightarrow \text{call it as state B}$   
 $\delta^*(A, a) = B$   
 $\delta^*(A, b) = \epsilon\text{-closure} \{ \delta(A, b) \}$   
 $= \epsilon\text{-closure} \{ (q_0, q_1, q_2, q_3, q_5, q_6, q_7, q_9, q_{10}, q_{11}), b \}$   
 $= \epsilon\text{-closure} \{ q_8 \}$   
 $= \{ q_8, q_9, q_{10}, q_1, q_2, q_3, q_5, q_6, q_7, q_9, q_{11} \} \longrightarrow \text{call it as state C}$   
 $\delta^*(A, b) = C$   
 $\delta^*(B, a) = \epsilon\text{-closure} \{ \delta(B, a) \}$   
 $= \epsilon\text{-closure} \{ (q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_9, q_{10}, q_{11}), a \}$   
 $= \epsilon\text{-closure} \{ q_4 \} = B$   
 Similarly  $\delta^*(B, b) = \epsilon\text{-closure} \{ q_8 \}$   
 $\delta^*(B, b) = C$   
 $\delta^*(C, a) = B$   
 $\delta^*(C, b) = C$

After eliminating  $\epsilon$  moves we get following DFA



Thus the DFA obtained in (i) and (ii) are the same. Hence  $(a | b)^* = (a^* | b^*)^*$  is proved.

#### 14. Design an appropriate DFA to accept the specified language $(a|b)^* a$



#### Find the States of DFA

**Vision** To produce demand driven, quality conscious and globally recognized computer professionals through education, innovation and collaborative research.



$\epsilon$ - closure (0) =	$\{0,1,2,4,7\} = A$
Move (A, a) =	$\{3,8\}$
Move (A, b) =	$\{5\}$
$\epsilon$ - Closure of ( Move (A, a) ) =	$\{3,6,7,1,2,4,8\}$
=	$\{1,2,3,4,6,7,8\} = B$
$\epsilon$ - closure of ( Move (A, b) ) =	$\{5,6,7,1,2,4\}$
=	$\{1,2,4,5,6,7\} = C$
Move (B, a) =	$\{3,8\}$
Move (B, b) =	$\{5\}$
$\epsilon$ - closure ( Move(B, a) ) =	B
$\epsilon$ - closure ( Move(B, b) ) =	C
Move (C, a) =	$\{3,8\}$
Move (C, b) =	$\{5\}$
$\epsilon$ - closure ( Move(C, a) ) =	B
$\epsilon$ - closure ( Move(C, b) ) =	C

### Construction of DTRAN of DFA

States	i/p Symbols	
	a	b
->A	B	C
*B	B	C
C	B	C

### Minimized DFA

= (AC) (B)

= A

### MinDTRAN of DFA

States	i/p Symbols	
	a	b
->A	B	A
*B	B	A

### DFA

