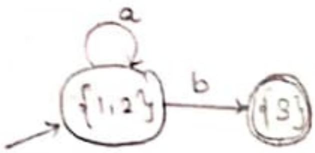
Step 4: find the starting position of NDFA

$S = \{1,2\}$.

Step 5: construct minimal DFA from the table.

minimal DFA diagram.



(2) $a^* bb$

method-1:

Step 1: Partion the states into accepting and non-accepting states

$P_0 = \underset{G_1}{(ABC)} \quad \underset{G_2}{(D)}$

Step 2: find the successor of state A, B, C, D. & split, no. split operation.

$A \xrightarrow{a} B$, $B \xrightarrow{a} B$, $C \xrightarrow{a} -$, $D \xrightarrow{a} -$ } NO split

$A \xrightarrow{b} C$, $B \xrightarrow{b} C$ } no. split

$C \xrightarrow{b} D$ ⟶ split

$D \xrightarrow{b} -$

Step 3: There contain split. operation.

1 - Equivalent

$P_1 = \underset{G_1}{(AB)} \quad \underset{G_2}{(C)} \quad \underset{G_3}{(D)}$

$A \xrightarrow{a} B$, $B \xrightarrow{a} B$ } no split $\quad$ $A \xrightarrow{b} C$, $B \xrightarrow{b} C$ } no-split

2 - equivalent

$P_2 = (AB)(C)(D)$

$\vdots$

$P_n = (AB)(C)(D)$

(State A = State B)

DFA Table

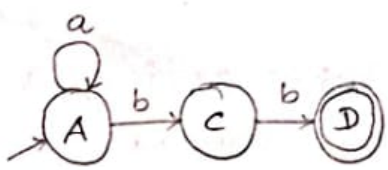
ip / State	a	b
A	~~B~~ A	C
~~B/A~~	~~B/A~~	~~C~~
C	–	D
D	–	–

duplicate

modified DFA table

ip / State	a	b
A	A	C
C	–	D
D	–	–

minimal DFA.



(3) (a|b)ab.

Set of states that can be reached from states through ε-transition.

I/p	Starting node	ending node
a	1,5	2,6
b	3,6	4,7

ε-closure (0) = {0,1,3} → A

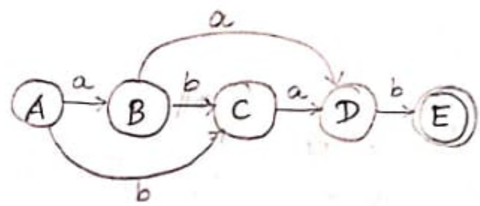
Apply input 'a' on state (A). ε-closure (2) = {2,5} → B
Apply input 'b' on state (A), ε-closure (4) = {4,5} → C
Apply input 'a' on state (B), ε-closure (6) = {6} → D
Apply input 'b' on state (B), ε-closure (-) = —
Apply input 'a' on state (C), ε-closure (6) = D
Apply input 'b' on state (C), ε-closure (-) = —

a-D = (-) b(D) = (7) - (7) → E
a.E = (-) b(E) = —

DFA/ Table

State \ I/p	a	b
A	B	C
B	D	—
C	D	—
D	—	E
E	—	—

DFA diagram.



(4) ab(a|b)

Set of states that can be reached from states through ε-transition.

i/p	Starting node	ending node
a	0,3	1,4
b	1,5	2,6

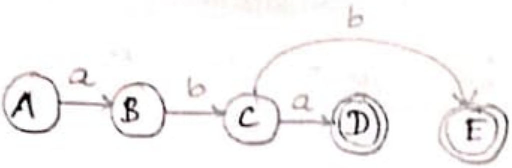
ε-closure (0) = {0} → A

Apply input 'a' on state (A). ε-closure (1) = {1} → B
Apply input 'b' on state (A), ε-closure (-) = —
Apply input 'a' on state (B), ε-closure (-) = —
Apply input 'b' on state (B), ε-closure (2) = {2,3,5} → C
Apply input 'a' on state (C), ε-closure (4) = {4,5,7} → D
Apply input 'b' on state (C), ε-closure (6) = {6,7} → E
Apply input 'a' on state (D), ε-closure (-) = —
Apply input 'b' on state (D), ε-closure (-) = —
Apply input 'a' on state (E), ε-closure (-) = —
Apply input 'b' on state (E). ε-closure (-) = —

(4) DFA Table

State \ ip	a	b
A	B	-
B	-	C
C	D	F
D	-	-
E	-	-

DFA diagram.



(5) (a|b) aaa (a|b).

Set of states that can be reached from States through ε transition.

I/p	Starting node	ending node.
a	1,5,6,7,9	2,4,6,7,8,10
b	3,11	4,12

ε. closure (0) = {0,1,3} → (A)

Apply input 'a' on State (A), ε-closure (2) = {2,5} → (B)
Apply input 'b' on State (A), ε-closure (4) = {4,5} → (C)
Apply input 'a' on State (B), ε-Closure (6) = {6} → (D)
Apply input 'b' on State (B), ε-closure (-) = —
Apply input 'a' on state (C), ε-closure (6) = → (D)
Apply input 'b' on state (C), ε-closure (-) = —
Apply inpute 'a' on state (D), ε-closure (7) = {7} → (E)
Apply input 'b' on State (D), ε-closure (-) = —
Apply input 'a' on state (E), ε-closure (8) = {8,9,11} → (F)
Apply input 'b' on state (E), ε-closure (-) = —
Apply input 'a' on state (F), ε-closure (10) = {10,13} → (G)
Apply input 'b' on state (F), ε-closure (12) = {12,13} → (H)
Apply input 'a' on state (G), ε-closure (-) = —
Apply input 'b' on state (G), ε-closure (-) = —
Apply input 'a' on State (H), ε-closure (-) = —
Apply input 'b' on state (H), ε-closure (-) = —

(5) (a|b)ab.

NFA:

(a|b)ab = {a,b} ab

= {

(6) (a|b)*

Set of states that can be reached from states through ε-transition.

ip	starting node	ending node.
a	2	3
b	4	5

ε-closure (0) = {0,1,2,4,7} → (A)

Apply input 'a' on state (A), ε-closure (3) = {1,2,3,4,6,7} → (B)

Apply input 'b' on state (A), ε-closure (5) = {1,2,4,5,6,7} → (C)

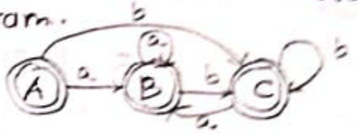
Apply input 'a' on state (B), ε-closure (3) → (B)

Apply input 'b' on state (B), ε-closure (5) → (C)

Apply input 'a' on state (C), ε-closure (3) → (B)

Apply input 'b' on state (C), ε-closure (5) → (C)

DFA diagram.



DFA table.

State \ ip	a	b
A	B	C
B	B	C
C	B	C

(7) (a|b)*ab.

Set of States that can be reached from state through ε-transition.

ip	starting node	ending node.
a	2, 7	3, 8
b	4, 8	5, 9

ε-closure (0) = {0, 1, 2, 4, 7} → (A)

Apply input 'a' on state (A), ε-closure (3,8) = {1,2,4,3,6,7,8} → (B)

Apply input 'b' on state (A), ε-closure (5) = {1,2,4,5,6,7} → (C)

Apply input 'a' on state (B), ε-closure (3,8) = (B) → (B)

Apply input 'b' on state (B), ε-closure (5,9) = {1,2,4,5,6,7,9} → (D)

Apply input 'a' on state (C), ε-closure (3,8) → (B)

Apply input 'b' on state (C), ε-closure (5) = (C)

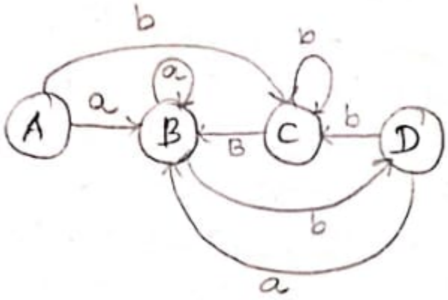
Apply

Apply input 'a' on state (D), ε-closure (3,8) = (B)
Apply input 'b' on state (D), ε-closure (9) = {9} → (E)
Apply input 'a' on state (E), ε-closure (F) = —
Apply input 'b' on state (E), ε-closure (G) = —

DFA table

state \ i/p	a	b
A	B	C
B	B	D
C	B	C
D	B	C

DFA diagram.



(8) (a|b)* abb

Sets of states that can be reached from states through ε-transition

I/P	starting node	ending node
a	2,7	3,8
b	4,8,9	5,9,10

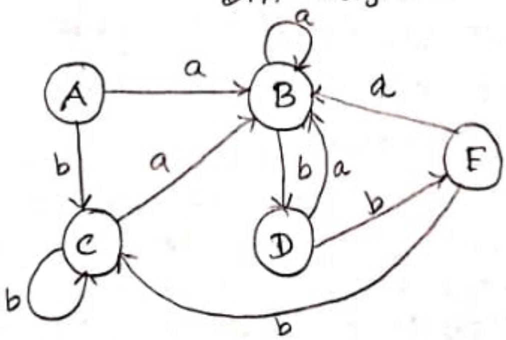
ε-closure (0) = {0,1,2,4,7} → (A)

Apply i/p 'a' on state (A), ε-closure (3,8) = {1,2,3,4,6,7,8} → (B)
Apply input 'b' on state (B), ε-closure (5) = {1,2,4,5,6,7} → (C)
Apply input 'a' on state (B), ε-closure (3,8) → (B)
Apply input 'b' on state (B), ε-closure (5,9) = {1,2,4,5,6,7,9} → (D)
Apply input 'a' on state (C), ε-closure (3,8) → (B)
Apply input 'b' on state (C), ε-closure (5) → (C)
Apply input 'b' on state (D), ε-closure (5,10) = {1,2,4,5,6,7,9,10} → (E)
Apply input 'a' on state (D), ε-closure (3,8) → (B)
Apply input 'a' on state (E), ε-closure (3,8) → (B)
Apply input 'b' on state (E), ε-closure (5) → (C)

DFA Table

state \ i/p	a	b
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C

DFA diagram.



(9) abb(a|b)*

Sol of states that can be reached from states through ε-transition

ip	starting node	ending node
a	0, 5	1, 6
b	1, 2, 7	2, 3, 8

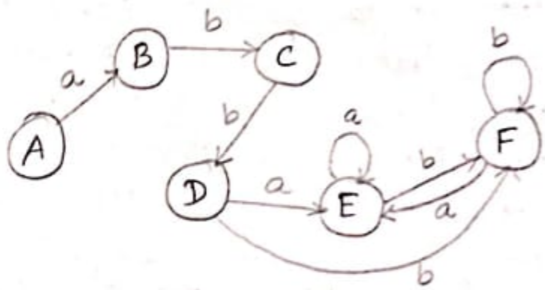
ε-closure (0) = {0} → (A)

Apply input 'a' on state (A), ε-closure (1) = {1} → (B)
Apply input 'b' on state (A), ε-closure (–) = —
Apply input 'a' on state (B), ε-closure (–) = —
Apply input 'b' on state (B), ε-closure (2) = {2} → (C)
Apply input 'a' on state (C), ε-closure (–) = —
Apply input 'b' on state (C), ε-closure (3) = {3, 4, 5, 7, 10} → (D)
Apply input 'a' on state (D), ε-closure (6) = {4, 5, 6, 7, 9, 10} → (E)
Apply input 'b' on state (D), ε-closure (8) = {4, 5, 7, 8, 9, 10} → (F)
Apply input 'a' on state (E), ε-closure (6) → (E)
Apply input 'b' on state (E), ε-closure (8) → (F)
Apply input 'a' on state (F), ε-closure (6) → (E)
Apply input 'b' on state (F), ε-closure (8) → (F).

DFA table

ip / State	a	b
A	B	–
B	~~e~~	C
C	–	D
D	E	F
E	E	F
F	E	F

DFA diagram.



(10) a*|b*

Set of states that can be reached from states through ε-transition

ip	Starting node	ending node.
a	2	3
b	6	7

ε-closure (0) = {0, 1, 2, 4, 5, 6, 8, 9} → (A)

Apply input 'a' on state (A), ε-closure (3) = {2, 3, 4, 9} → (B)
Apply input 'b' on state (A), ε-closure (7) = {6, 7, 8, 9} → (C)
Apply input 'a' on state (B), ε-closure (3) → (B)
Apply input 'b' on state (B), ε-closure (–) = —

(6)

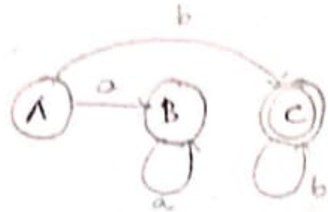
Apply input 'a' on state (C), ε-closure (-) = —

Apply input 'b' on state (C), ε-closure (4) = (C)

DFA table

States \ i/p	a	b
A	B	C
B	B	—
C	—	C

DFA diagram.



(11) $(a^*)^* = a^*$

Set of states that can be reached from states through ε-transition

i/p	starting node	ending node
a	2	3

ε-closure (0) = {0, 1, 2, 4, 5} → (A)

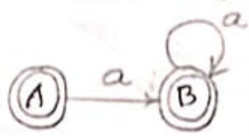
Apply input 'a' on state (A), ε-closure (3) = {1, 2, 3, 4, 5} → (B)

Apply input 'a' on state (B), ε-closure (3) → (B)

DFA Table

State	Input (a)
A	B
B	B

DFA diagram.



(12) $a^+ = aa^* = a^*a$

Set of states that can be reached from states through ε-transition

ε-closure (0) = {0, 1, 3} → (A)

I/p	Starting node	ending node
a	1, 3	2, 4

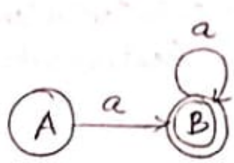
Apply input 'a' on state (A), ε-closure (2, 4) = {1, 2, 3, 4} → (B)

Apply input 'a' on state (B), ε-closure (2, 4) → (B)

DFA Table

State	a
A	B
B	B

DFA diagram.



(13) $(a^*|b^*)abb$

Set of States that can be reached from States through ε-transition.

I/p	Starting node	ending node
a	2, 9	3, 10
b	6, 10, 11	7, 11, 12

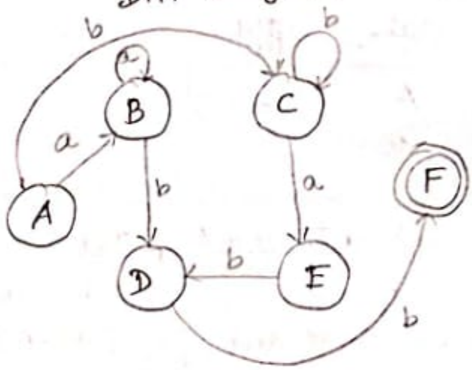
ε-closure(0) = {0, 1, 2, 4, 5, 6, 8, 9} → (A)

Apply input 'a' on State (A), ε-closure (3,10) = {2, 3, 4, 9, 10} → (B)
Apply input 'b' on State (A), ε-closure (7) = {6, 7, 8, 9} → (C)
Apply input 'a' on State (B), ε closure (3,10) = → (B)
Apply input 'b' on State (B), ε-closure (11) = {11} → (D)
Apply input 'a' on State (C), ε closure (10) = {10} → (E)
Apply input 'b' on State (C), ε closure (7) = → (C)
Apply input 'a' on State (D), ε-closure (-) = —
Apply input 'b' on State (D), ε closure (12) = {12} → (F)
Apply input 'a' on State (E), ε closure (-) = —
Apply input 'b' on State (E), ε closure (11) → (D)
Apply input 'a' on State (F), ε-closure (-) = —
Apply input 'b' on State (F), ε closure (-) = —

DFA table

State \ ip	a	b
A	B	C
B	B	D
C	E	C
D	—	F
E	—	D
F	—	—

DFA diagram.



(14) $(a^*|b^*)^*$

Set of States that can be reached from States through ε-transition.

I/p	Starting node	ending node.
a	3	4
b	7	8

ε-closure (0) = {0, 1, 2, 3, 5, 6, 7, 9, 10, 11} → (A)

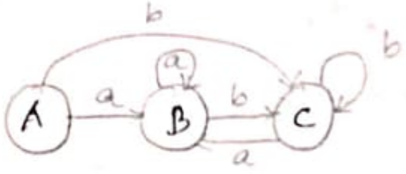
Apply input 'a' on State (A), ε-closure (4) = {1, 2, 3, 4, 5, 6, 7, 9, 10, 11} → (B)
Apply input 'b' on State (A), ε-closure (8) = {1, 2, 3, 5, 6, 7, 8, 9, 10, 11} → (C)
Apply input 'a' on State (B), ε-closure (4) = (B)
Apply input 'b' on State (B), ε-closure (8) = (C).

Apply input 'a' on State (c), ε-closure (4) → (B)
Apply input 'b' on State (c), ε-closure (8) → (C)

DFA Table

State \ i/p	a	b
A	B	C
B	B	C
C	B	C

DFA diagram



(15) (a* b*)

Set of states that can be reached from States through ε-transition.

I/p	Starting node	Ending node
a	1	2
b	4	5

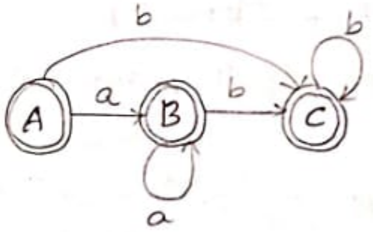
ε-closure (0) = {0,1,3,4,6} → (A)

Apply input 'a' on State (A), ε-closure (2) = {1,2,3,4,6} → (B)
Apply input 'b' on State (A), ε-closure (5) = {4,5,6} → (C)
Apply input 'a' on State (B), ε-closure (2) = → (B)
Apply input 'b' on State (B), ε-closure (5) → (C)
Apply input 'a' on State (c), ε-closure (-) = —
Apply input 'b' on State (c), ε-closure (5) → (C)

DFA Table

State \ i/p	a	b
A	B	C
B	B	C
C	—	C

DFA diagram.



Construction of minimal DFA (or) Reduced DFA

Method - 1.

(1) $a^* b$

Step 1: partition the states into accepting and non-accepting states.

$P_0 = \begin{pmatrix} \text{accepting} \\ \text{states} \end{pmatrix} \cdot \begin{pmatrix} \text{non-accepting} \\ \text{state} \end{pmatrix}$

$P_0 = (AB)\ (C)$

group1 group2.

Step 2: Find successor of state A, B, C when inputs a & b are applied.

$\left.\begin{array}{l} A \xrightarrow{a} B \\ B \xrightarrow{a} B \\ C \xrightarrow{a} - \end{array}\right\}$ no split $\left.\begin{array}{l} A \xrightarrow{b} C \\ B \xrightarrow{b} C \\ C \xrightarrow{b} - \end{array}\right\}$ no split

Step 2: there is used no split operation

$\therefore P_n = (AB)\ (C)$

(State A = State B).

DFA Table

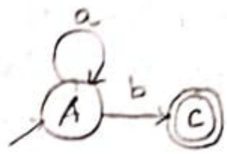
state \ i/p	a	b
A	~~B~~ A	C
~~B~~ A	~~B~~ A	C
C	-	-

duplicate

modified DFA table

state \ i/p	a	b
A	A	C
C	-	-

minimal DFA



method - 2 (directly).

Step 1: Add # symbol at end of regular expression

$a^* b\ \#$.

Step 2: Assign the position including # symbol.

a = 1
b = 2

$\underset{1}{a}{}^* \ \underset{2}{b}\ \underset{3}{\#}$

Step 3: construct position and follow position using table.

$a^* b \# = \{\varepsilon, a, aa, aaa, aaaa \ldots\} b \#$

$= \{b, ab, aab, aaab, aaaab \ldots\} \#$

$= \{23, 123, 1123, 11123, 111123 \ldots\}$

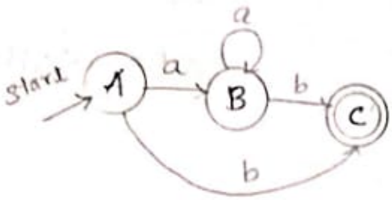
position	follow position.
1	{1, 2}
2	{3}
3	-

(3)

Apply input 'a' on state (C), ε-closure (-) = —
Apply input 'b' on state (C), ε-closure (-) = —
DFA Table:

State \ i/p	a	b
A	B	C
B	B	C
C	–	–

DFA diagram:



(2) a*bb

Set of states that can be reached from states through ε-transition.

I/p	starting node	ending node.
a	1	2
b	3, 4	4, 5

ε-closure (0) = {0,1,3} → (A)
Apply input 'a' on state (A), ε-closure (2) = {1,2,3} → (B)
Apply input 'b' on state (A), ε-closure (4) = {4} → (C)
Apply input 'a' on state (B), ε-closure (2) = (B).
Apply input 'b' on state (B), ε-closure (4) = (C).
Apply input 'a' on state (C), ε-closure (-) = —
Apply input 'b' on state (C), ε-closure (5) = {5} → (D)
Apply input 'a' on state (D), ε-closure (-) = —
Apply input 'b' on state (D), ε-closure (-) = —

DFA Table:

State \ i/p	a	b
A	B	C
B	B	C
C	–	D
D	–	–

DFA diagram: