SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

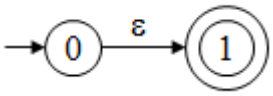
18CSC304J - COMPILER DESIGN

EVENSEM 2023-24

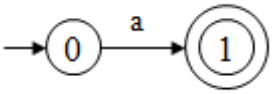
WORKSHEET

I.Convert RE into NFA:-

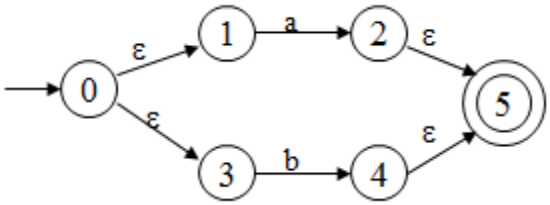
1. ε



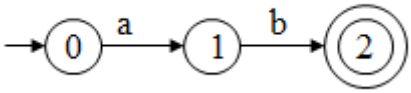
2. a



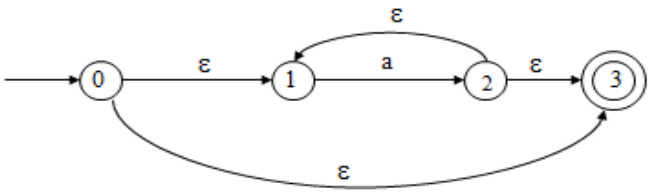
3. a/b



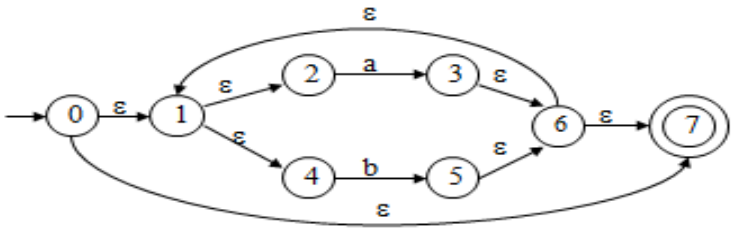
4. ab



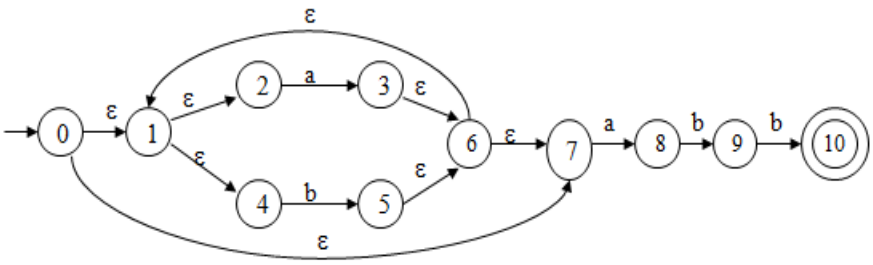
5. a*



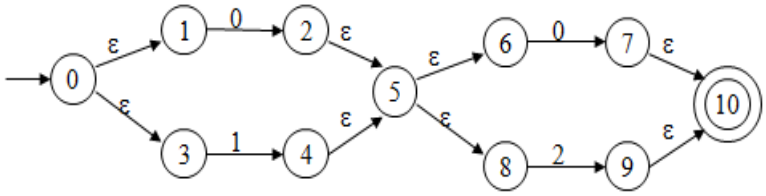
6. (a/b)*



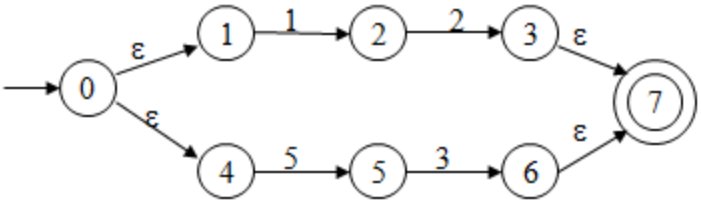
7. (a/b)*abb



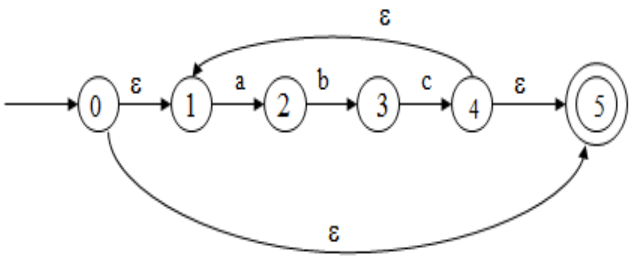
8. (0/1) (0/2)



9. 12/53

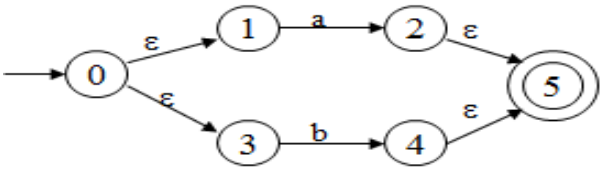


10. (abc)*

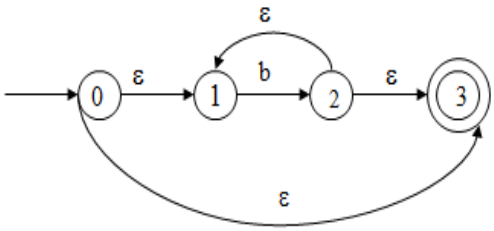


11. ((a/b) b*)*

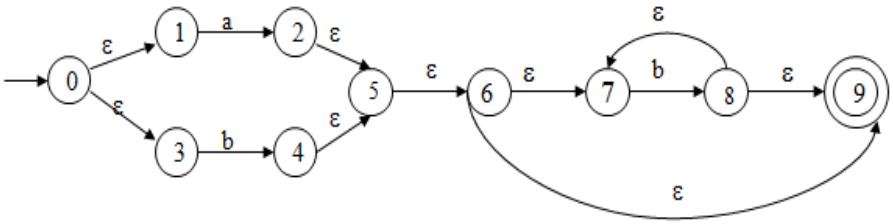
(a/b)



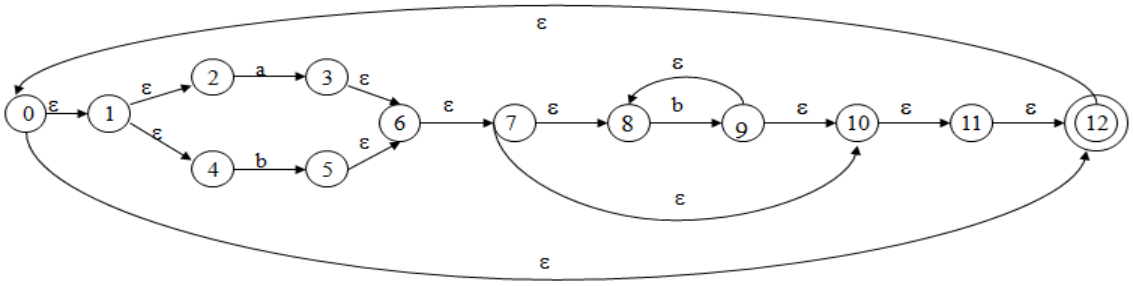
b*



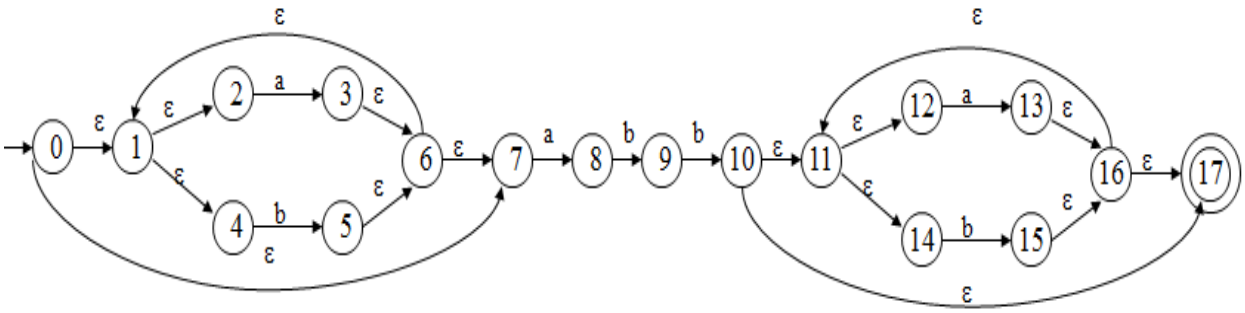
(a/b)b*



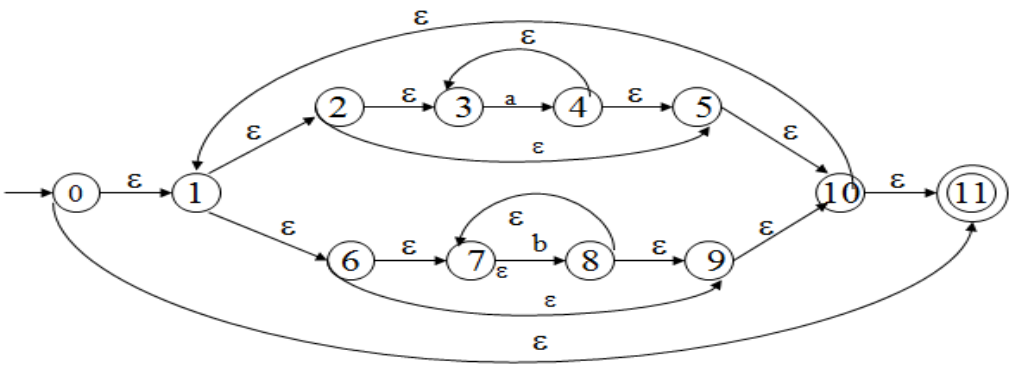
((a/b)b*)*



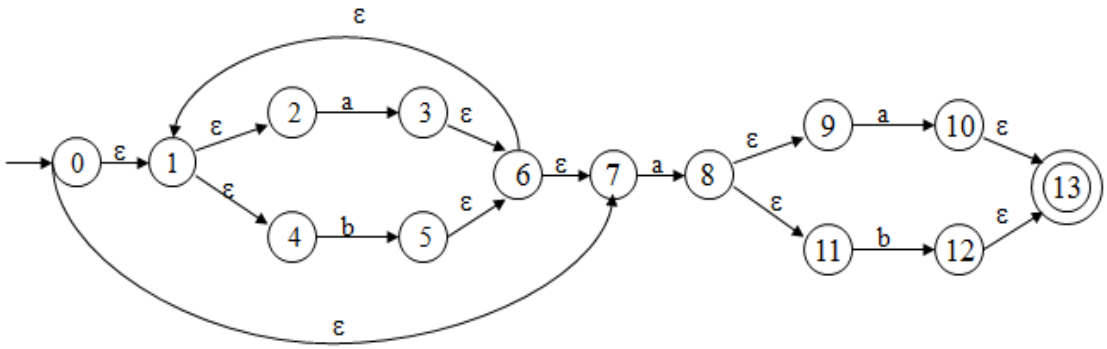
12. (a/b)*abb (a/b)*



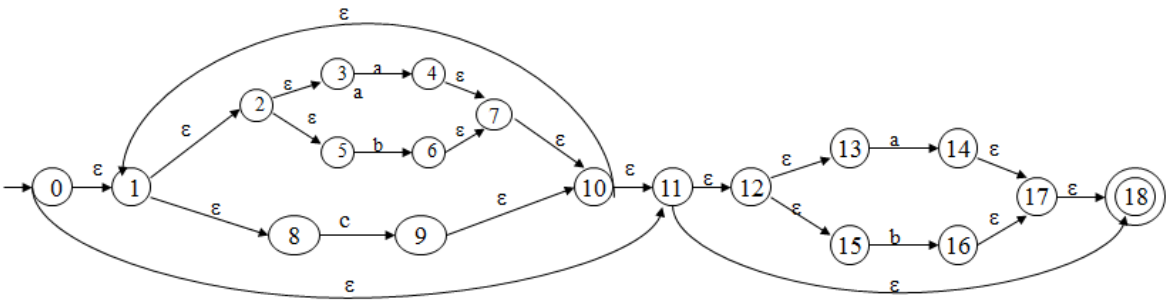
13. (a*/b*)*



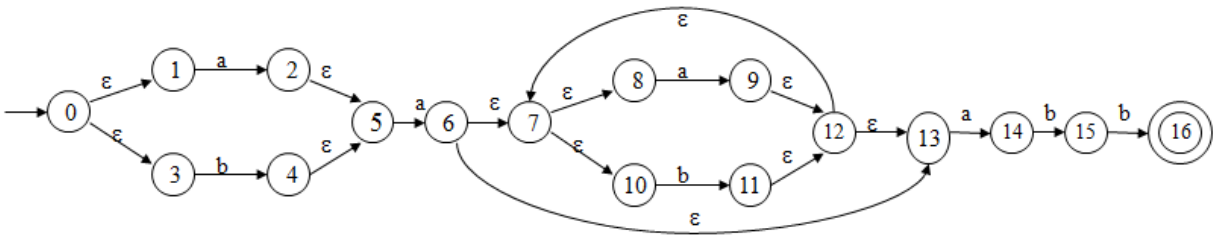
14. (a/b)*a (a/b)



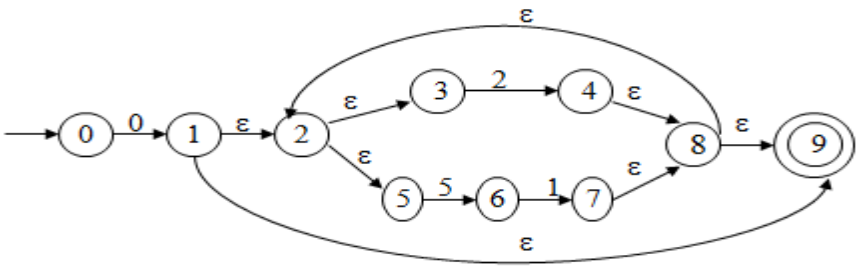
15. (a/b/c)* (a/b)*



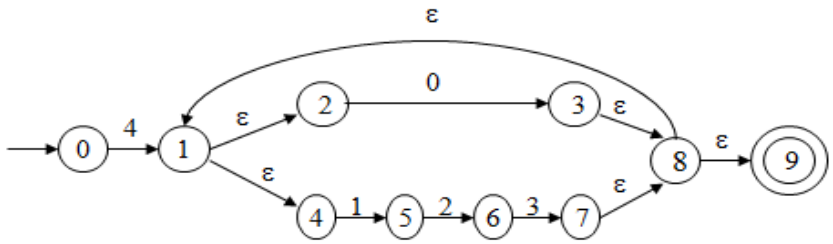
16. (a/b)a(a/b)*abb



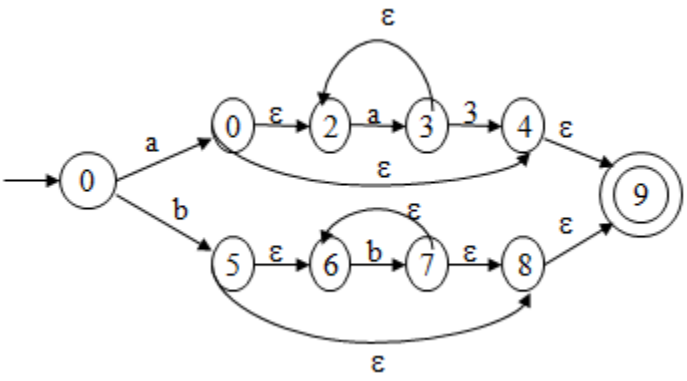
17. 0(2/51)*



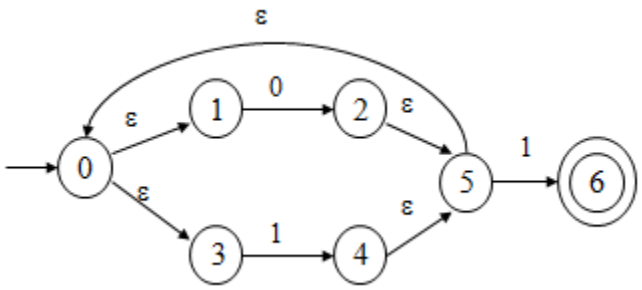
18. 4(0/123)*



19. (a,a*/bb*)



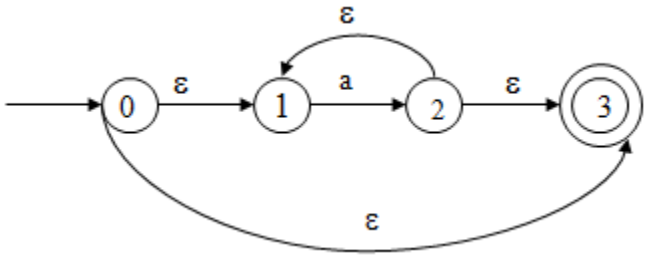
20. (0/1)*1



II. CONVERT RE to DFA (INDIRECT METHOD)

1. RE = a*

NFA



Finding the states

ε closure {0} = {0,1,3}

$$\boxed{A = \{0,1,3\}}$$

ε closure[move(A,a)] = {2}

= {2,1,3}

ε closure[move(B,a)] = {2} $\boxed{B = \{1,2,3\}}$

= B

Transition table:

	a
A	B
B	B

Minimizing DFA:

= {A B}

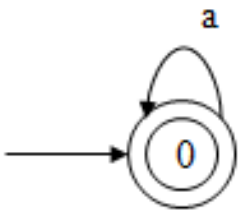
= {(A B)}

= { A }

Minimized Transition table:

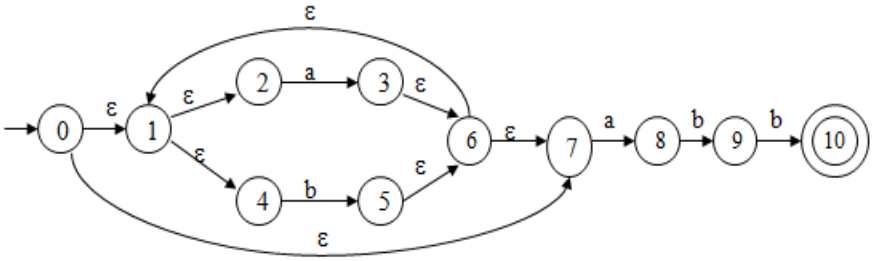
State	a
A	A

DFA:



2. RE = (a/b)*abb Draw the NFA and Convert it into DFA

NFA



ε closure {0} = {0,1,2,4,7} = A

mov(A,a) = ε closure {3,8} = {3,8,6,1,2,4,7} = B

mov(A,b) = ε closure {5} = {5,6,1,2,4,7} = C

mov(B,a) = ε closure {3,8} = B

mov(B,b) = ε closure {9,5} = {9,5,6,7,1,2,4} = D

mov(C,a) = ε closure {3,8} = B

mov(C,b) = ε closure {5} = C

mov(D,a) = ε closure {8,3} = B

mov(D,b) = ε closure {10,5} = {10,5,6,7,1,2,4} = E

mov(E,a) = ε closure {8,3} = B

mov(E,b) = ε closure {5} = C

Transition table:

	a	b
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C

Minimizing DFA:

π = { (A ,B,C,D) (E) }

= { (A,B,C) (D) (E)}

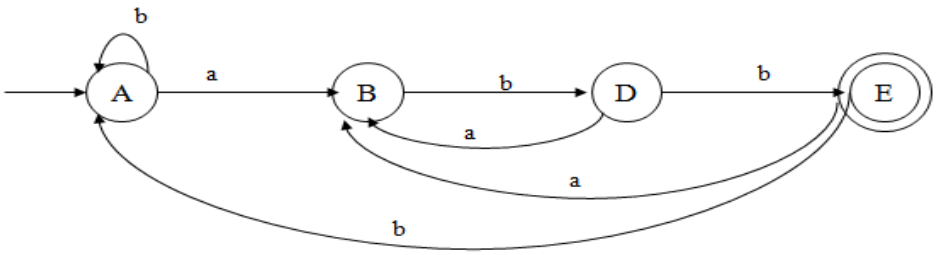
= { (AC) (B) (D) (E)}

π = { (A) (B) (C) (D) (E) }

Minimized Transition table:

State	a	b
A	B	A
B	B	D
D	B	E
E	B	A

DFA:

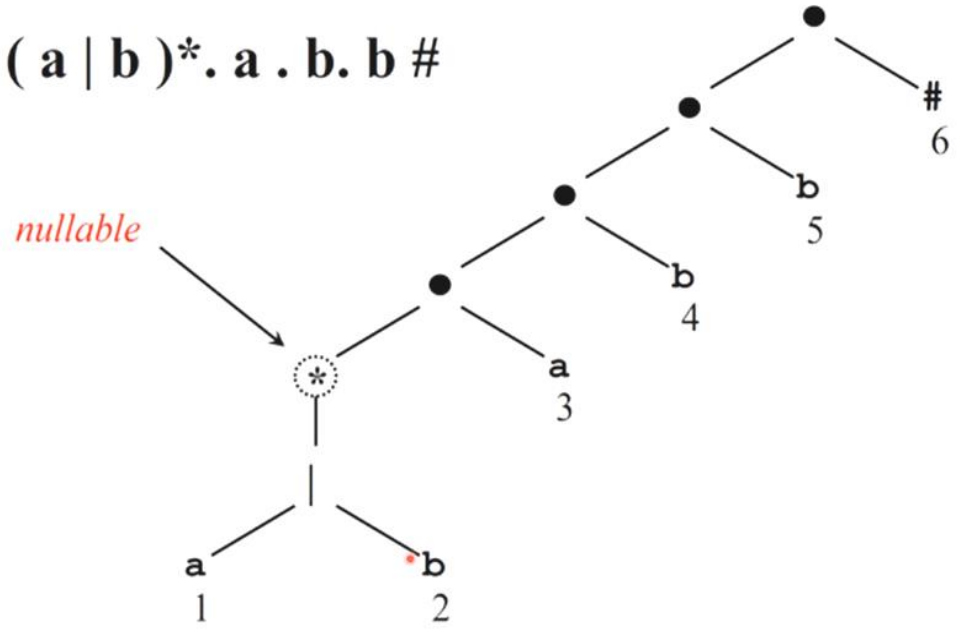


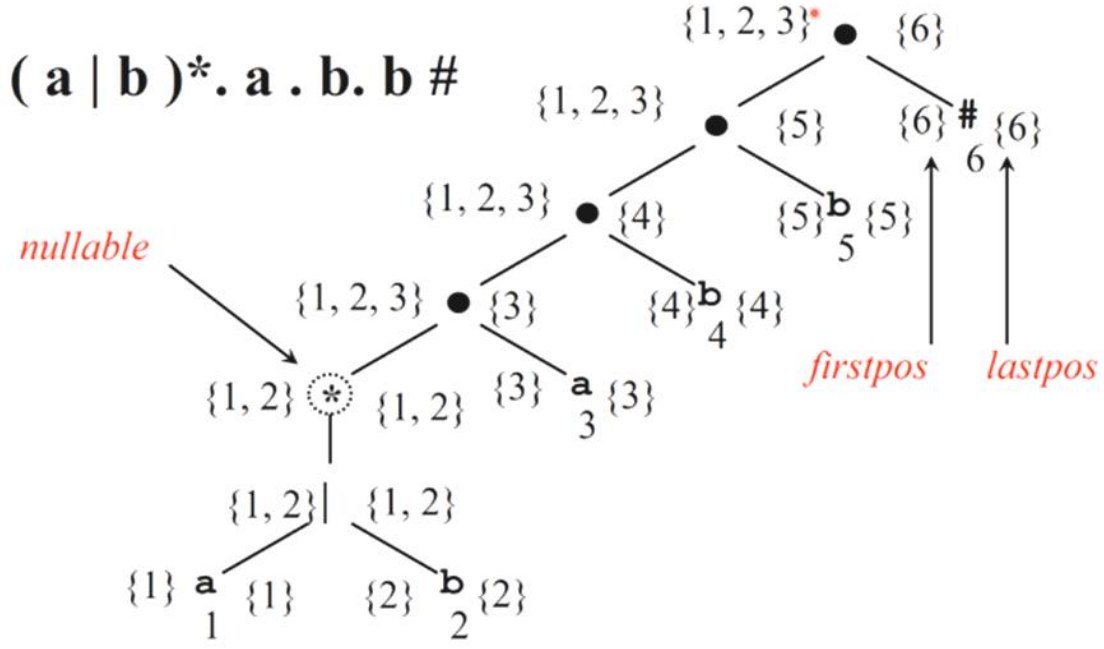
II. CONVERT RE to DFA (DIRECT METHOD)

1. **RE = (a/b)*abb Convert it into DFA in direct method**

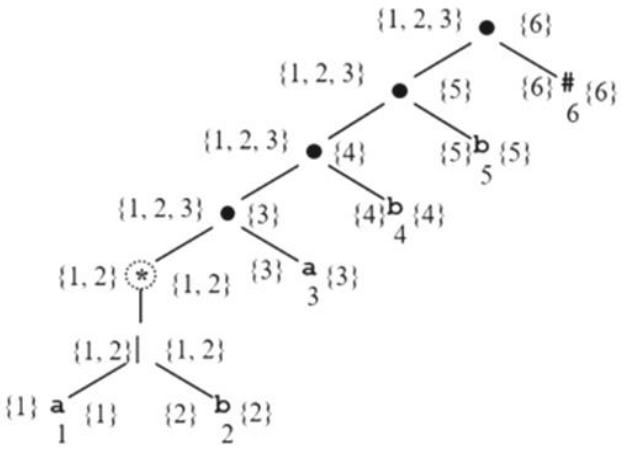
Augmented RE : (a/b)*.a.b.b#

Syntax Tree


(a | b)*. a . b. b #
nullable
#
6
b
5
b
4
a
3
*
a
1
b
2


(a | b)*. a . b. b #
nullable
{1, 2, 3}
{6}
{1, 2, 3}
{5}
{6}
#
6
{6}
{1, 2, 3}
{4}
{5}
b
5
{5}
{1, 2, 3}
{3}
{4}
b
4
{4}
{1, 2}
*
{1, 2}
{3}
a
3
{3}
{1, 2}
{1, 2}
{1}
a
1
{1}
{2}
b
2
{2}
firstpos
lastpos

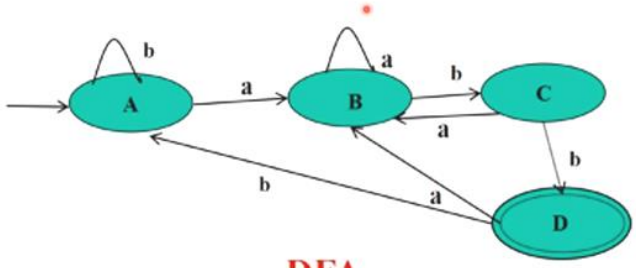
(a | b)*. a . b. b #



Node	*followpos*
1	1,2,3
2	1,2,3
3	4
4	5
5	6
6	-

	Node	*followpos*
a	1	1,2,3
b	2	1,2,3
a	3	4
b	4	5
b	5	6
#	6	-

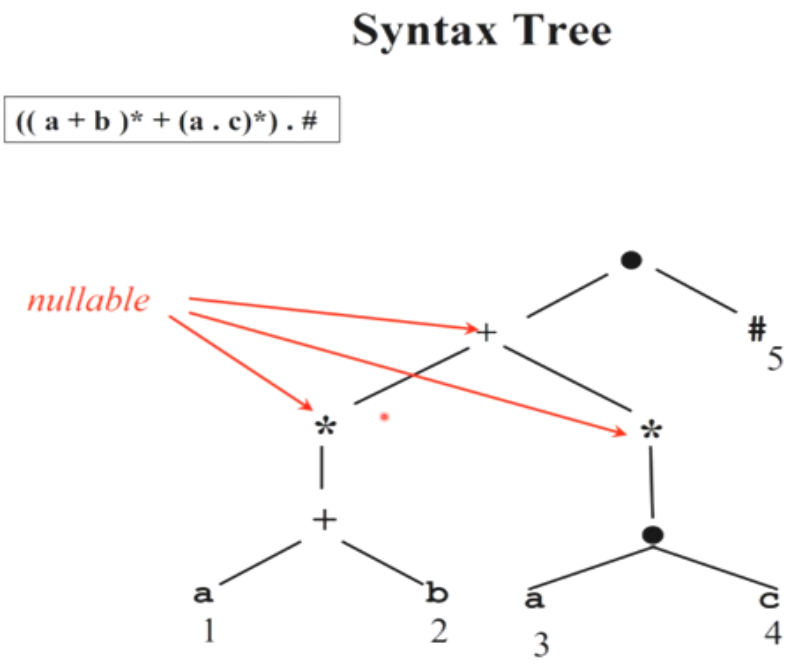
(a | b)*. a . b. b #

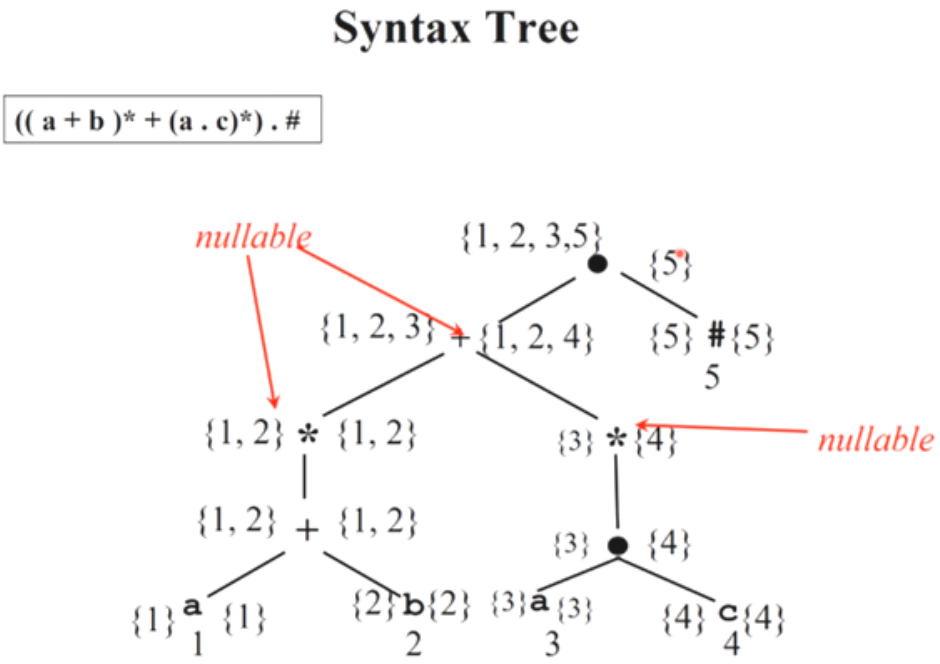


DFA

2. RE = (a+b)*+(a.c)* Convert it into DFA in direct method

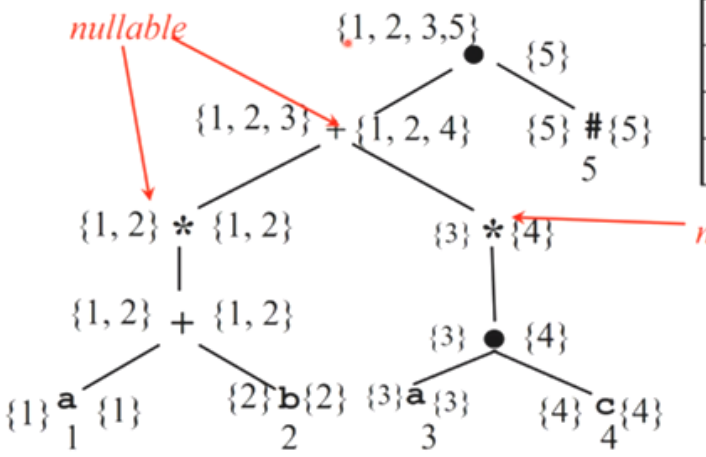
Augmented RE : **((a+b)*+(a.c)*)**#





Followpos

((a + b)* + (a . c)*) . #



Node	*followpos*
1	1,2,5
2	1,2,5
3	4
4	3,5
5	-

((a + b)* + (a . c)*) . #

	Node	*followpos*
a	1	1,2,5
b	2	1,2,5
a	3	4
c	4	3,5
#	5	-