Ques- Conversion from Regular Expression to DFA without NFA.

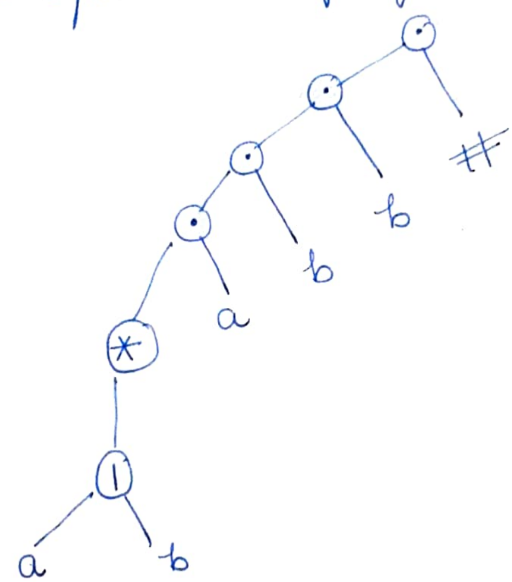
Regular Expression → $(a|b)^*abb$

Answer Convert regular expression $(a|b)^*abb$ to DFA without NFA.

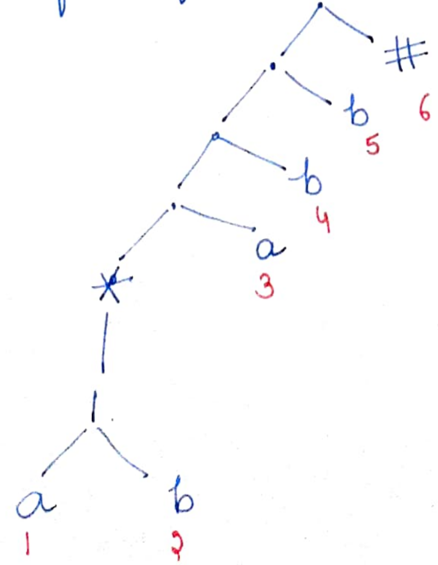
Firstly, we add # at the end of the regular expression.

Augmented RE ⇒ $(a|b)^*abb\#$

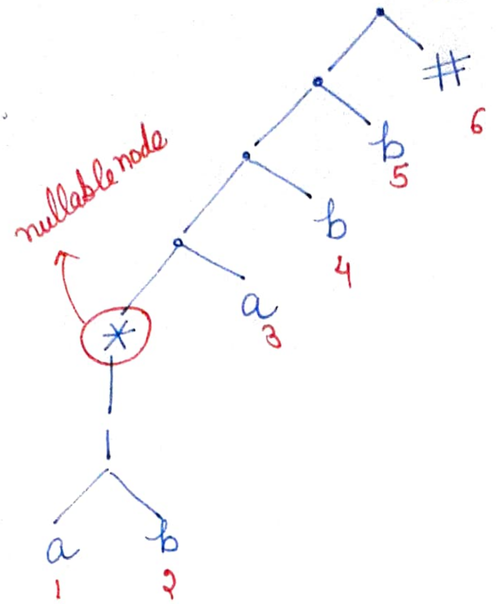
Step-1 Construct Syntax tree of Regular expression $(a|b)^*abb\#$.



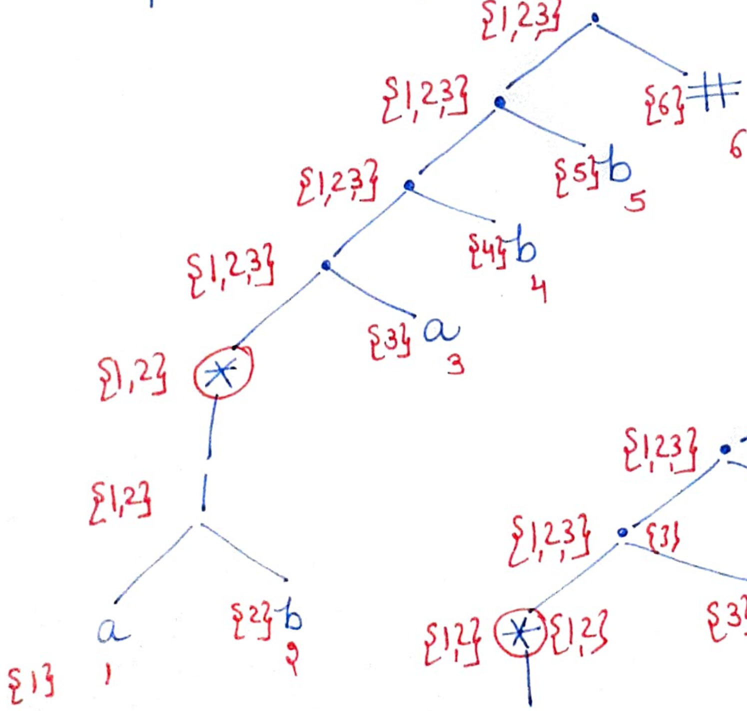
Add lable of leaf nodes.



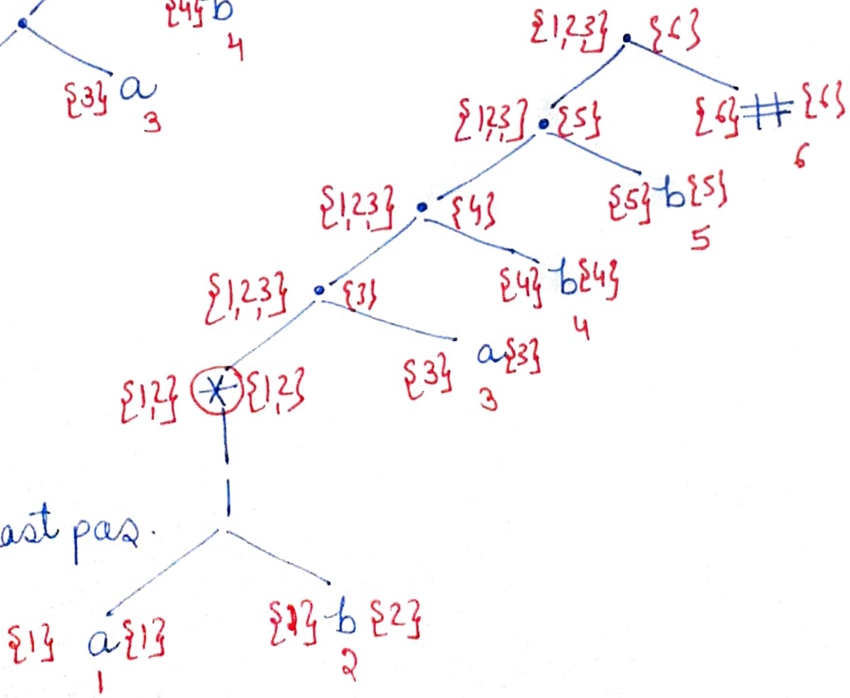
Step 2 Find nullable node from syntax tree.



Step-3 • Calculate first pos.



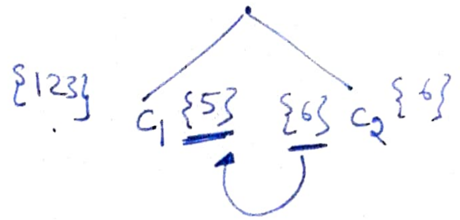
• calculate last pos.



Step-4 Calculate followpos from construct Syntax Tree.

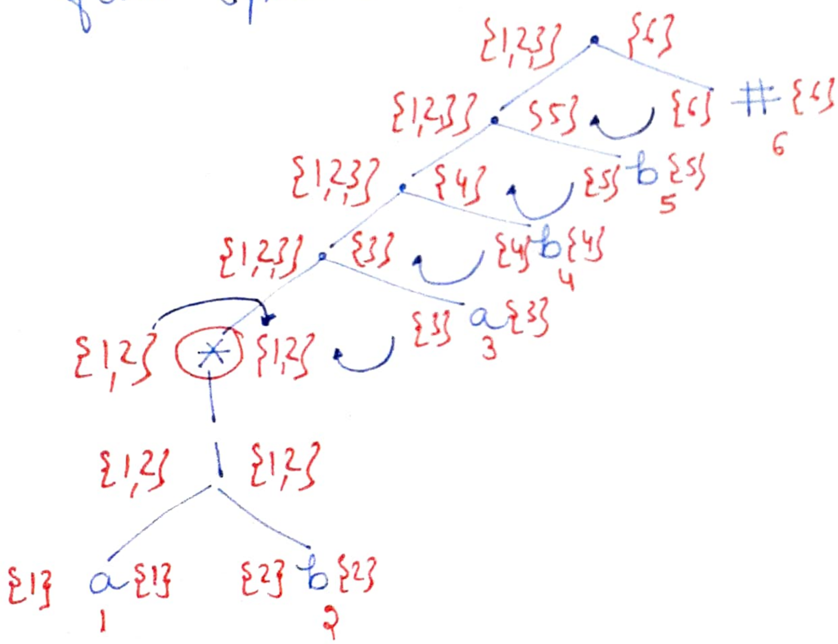
⟹ for find followpos we allways take (• & *) only.

start from top.



position	followpos
1	3, 1, 2
2	3, 1, 2
3	4
4	5
5	6
6	---

So, find all (• & *) followpos of the syntax tree.



Now, construct DFA by using Syntax tree and followpos table.

we always start from top.

Initial state = firstpos of root = {1, 2, 3} --- (A)

$A = \{1, 2, 3\}$

State A :- Transition . $\delta(A, a)$ & $\delta(A, b)$

$\delta(A, a) =$ followpos(1) ∪ followpos(3)

$= \{1, 2, 3\} \cup \{4\}$

$= \{1, 2, 3, 4\}$ — (B)

$$\delta(A, b) = \text{followpos}(2)$$
$$= \{1, 2, 3\} \text{ --- } A$$

State B :- Transition $\delta(B, a)$ & $\delta(B, b)$

$$B = \{1, 2, 3, 4\}$$

$$\delta(B, a) = \text{followpos}(1) \cup \text{followpos}(3)$$
$$= (1, 2, 3) \cup (4)$$
$$= \{1, 2, 3, 4) \text{ --- } B$$

$$\delta(B, b) = \text{followpos}(2) \cup \text{followpos}(4)$$
$$= \{1, 2, 3\} \cup \{5\}$$
$$= \{1, 2, 3, 5\} \text{ --- } C$$

State C: $C = \{1, 2, 3, 5\}$ Transition $\delta(C, a)$ & $\delta(C, b)$

$$\Rightarrow \delta(C, a) = \text{followpos}(1) \cup \text{followpos}(3)$$
$$= (1, 2, 3) \cup (4)$$
$$= (1, 2, 3, 4) \text{ — } B$$

$$\delta(C, b) = \text{followpos}(2) \cup \text{followpos}(5)$$
$$= (1\ 2\ 3) \cup (6)$$
$$= (1, 2, 3, 6) \text{ --- } D$$

State D = $D = \{1, 2, 3, 6\}$ Transition $\delta(D, a)$ & $\delta(D, b)$

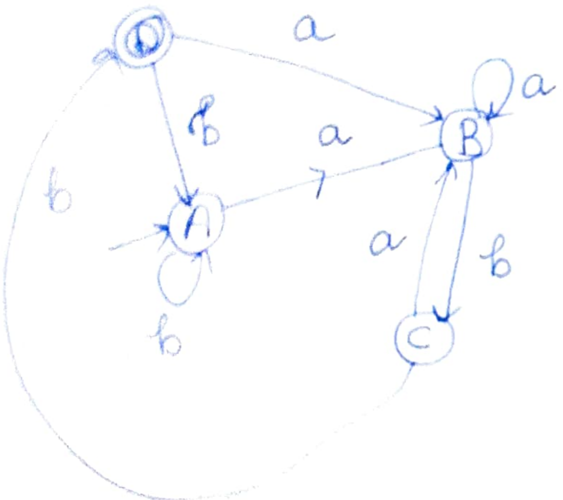
$$\Rightarrow \delta(D, a) = \text{followpos}(1) \cup \text{followpos}(3)$$
$$= (1, 2, 3) \cup (4)$$
$$= (1, 2, 3, 4) \text{ — } B$$

$$\delta(D, b) = \text{followpos}(2)$$
$$= \cancel{\text{followpos}}(1, 2, 3) \text{ — } A$$

Transition - table

State	Input	
Q	a	b
A = (1,2,3)	B	A
B = (1,2,3,4)	B	C
C = (1,2,3,5)	B	D
D = (1,2,3,6)	B	A

• Transition Digram (DFA)

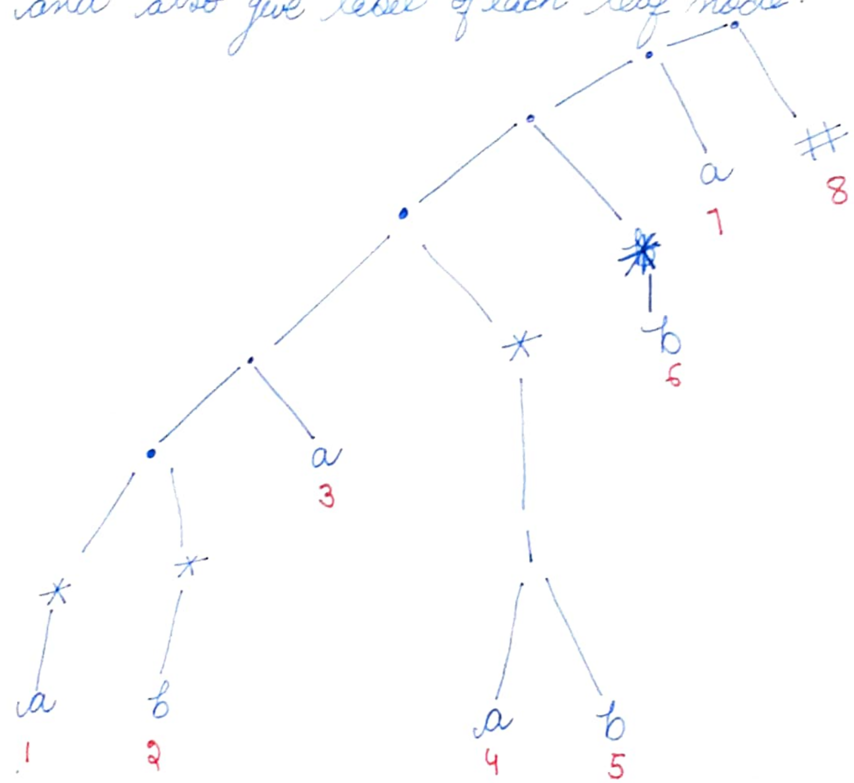


⇒ D is final state because in NFA 6 has a # which is considered as final state.

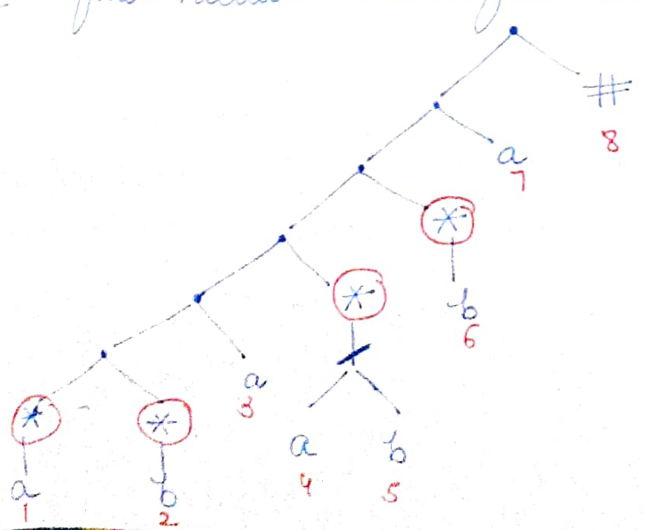
Ques Conversion from Regular Expression to DFA without NFA.
Regular Expression $a^* b^* a(a|b)^* b^* a$.

Answer Convert regular expression $a^* b^* a(a|b)^* b^* a$ to DFA without NFA.
Firstly, we add # at the end of the regular expression.
Augmented RE ⇒ $a^* b^* a(a|b)^* b^* a\#$

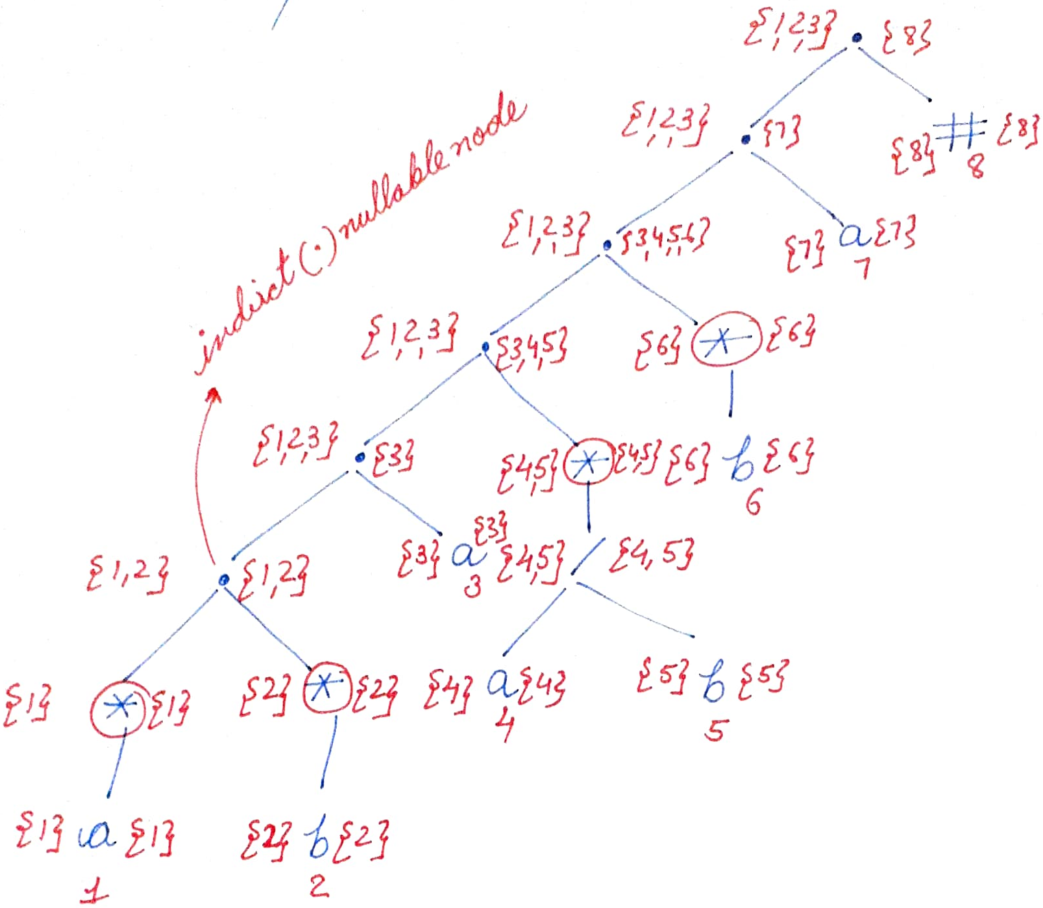
Step 1 Construct syntax tree of regular expression $a^* b^* a(a|b)^* b^* a\#$ and also give label of each leaf node.



Step-2 find nullable nodes from constructed syntax tree.

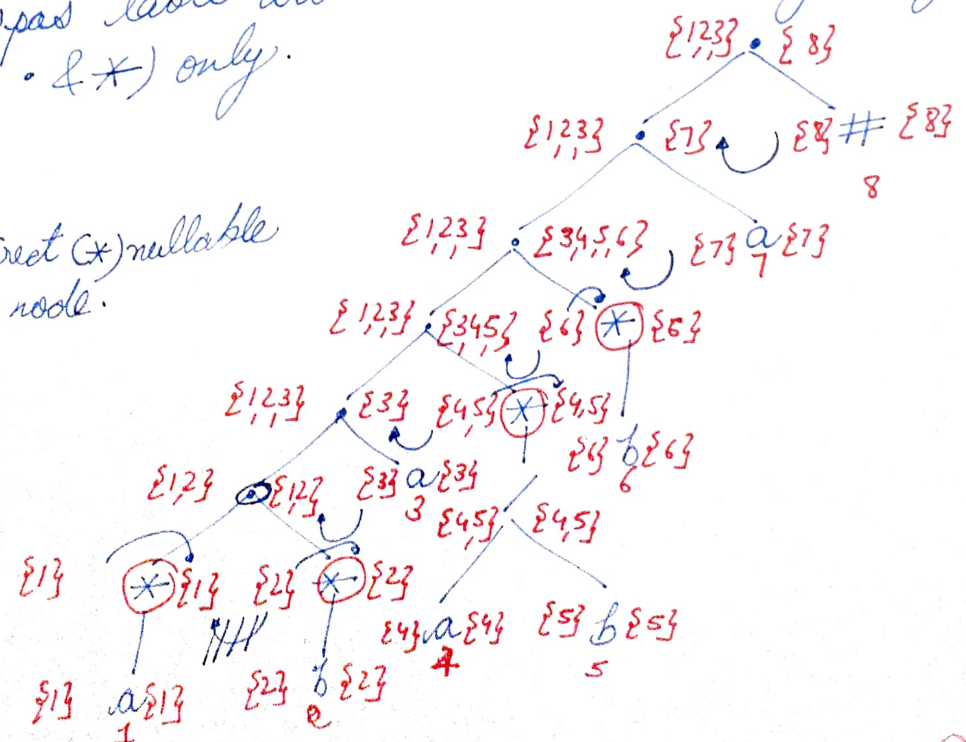


Step-3 Calculate firstpos and lastpos from constructed syntax tree.



Step-4 Calculate followpos from constructed syntax tree
→ followpos table will be construct (create) by using (· & *) only.

⊙ indirect (*) nullable node.



position	followpos
1	$\{1, 3\}$
2	$\{2, 3\}$
3	$\{4, 5, 6, 7\}$
4	$\{4, 5, 6, 7\}$
5	$\{4, 5, 6, 7\}$
6	$\{6, 7\}$
7	$\{8\}$
8	---

Now, construct DFA by using construct syntax tree and followpos table.

Start from top.

So, Initial state $\{1, 2, 3\}$

→ State A = $\{1, 2, 3\}$

- Transition $\delta(A, a)$ and $\delta(A, b)$
- $\delta(A, a)$ = followpos(1) ∪ followpos(3)
  = $\{1, 3\} \cup \{4, 5, 6, 7\}$
  = $\{1, 3, 4, 5, 6, 7\}$ --- B
- $\delta(A, b)$ = followpos(2)
  = $\{2, 3\}$ --- C

→ State B = $\{1, 3, 4, 5, 6, 7\}$

Transition $\delta(B, a)$ and $\delta(B, b)$

- $\delta(B, a)$ = followpos(1) ∪ followpos(3) ∪ followpos(4) ∪ followpos(7)
  = $\{1, 3\} \cup \{4, 5, 6, 7\} \cup \{4, 5, 6, 7\} \cup \{8\}$
  = $\{1, 3, 4, 5, 6, 7, 8\}$ --- D
- $\delta(B, b)$ = followpos(5) ∪ followpos(6)
  ~~= followpos(4, 5, 6, 7) ∪~~
  = $\{4, 5, 6, 7\} \cup \{6, 7\}$
  = $\{4, 5, 6, 7\}$ --- E

→ State C = $\{2, 3\}$

- Transition $\delta(C, a)$ and $\delta(C, b)$.
- ~~$\delta(C, a) = \{4, 5, 6, 7\}$ --- C~~
- $\delta(C, a)$ = followpos(3)
  = $\{4, 5, 6, 7\}$ --- E
- $\delta(C, b)$ = followpos(2)
  = $\{2, 3\}$ --- C

→ State D = $\{1, 3, 4, 5, 6, 7, 8\}$

- Transition $\delta(D, a)$ and $\delta(D, b)$
- $\delta(D, a)$ = followpos(1) ∪ followpos(3) ∪ followpos(4) ∪ followpos(7)
  = $\{1, 3\} \cup \{4, 5, 6, 7\} \cup \{4, 5, 6, 7\} \cup \{8\}$
  = $\{1, 3, 4, 5, 6, 7, 8\}$ --- D
- $\delta(D, b)$ = followpos(5) ∪ followpos(6)
  = $\{4, 5, 6, 7\} \cup \{6, 7\}$
  = $\{4, 5, 6, 7\}$ --- E

→ State E = $\{4, 5, 6, 7\}$

- Transition $\delta(E, a)$ & $\delta(E, b)$

- $\delta(E, a) = followpos(4) \cup followpos(7)$
  $= \{4, 5, 6, 7\} \cup \{8\}$
  $= \{4, 5, 6, 7, 8\}$ --- F

- $\delta(E, b) = followpos(5) \cup followpos(6)$
  $= \{4, 5, 6, 7\} \cup \{6, 7\}$
  $= \{4, 5, 6, 7\}$ --- E

→ State $F = \{4, 5, 6, 7, 8\}$

- Transition $\delta(F, a)$ & $\delta(F, b)$

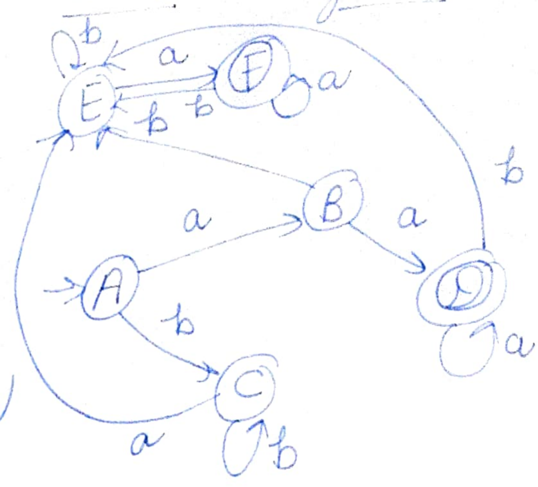
- $\delta(F, a) = followpos(4) \cup followpos(7)$
  $= \{4, 5, 6, 7\} \cup \{8\}$
  $= \{4, 5, 6, 7, 8\}$ --- F

- $\delta(F, b) = followpos(5) \cup followpos(6)$
  $= \{4, 5, 6, 7\} \cup \{6, 7\}$
  $= \{4, 5, 6, 7\}$ --- E

- Transition Table

State Q	Input a	Input b
→ A = {1,2,3}	B	C
B = {1,3,4,5,6,7}	D	E
C = {2,3}	E	C
D = {1,3,4,5,6,7,8}	D	E
E = {4,5,6,7}	F	E
F = {4,5,6,7,8}	F	E

- Transition Diagram



DFA

Final state are D and F because 8 is present in D and F only.

Ques Conversion from Regular Expression to DFA without NFA.
Regular Expression $a(a|b)^*b$

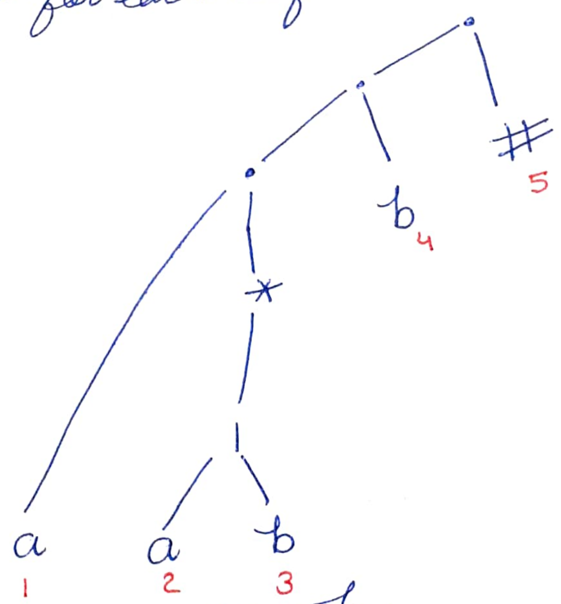
We may convert a regular expression into a DFA (without creating a NFA).

Firstly, we augment the given regular expression by concatenating it with a special symbol #.

$r \rightarrow r(\#)$.

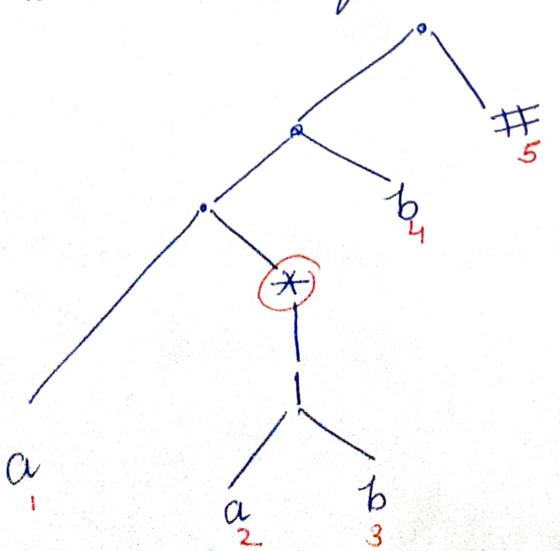
$a(a|b)^*b\#$

Step-1 Convert regular expression $a(a|b)^*b\#$ into ST and also give lablel for each leaf node.

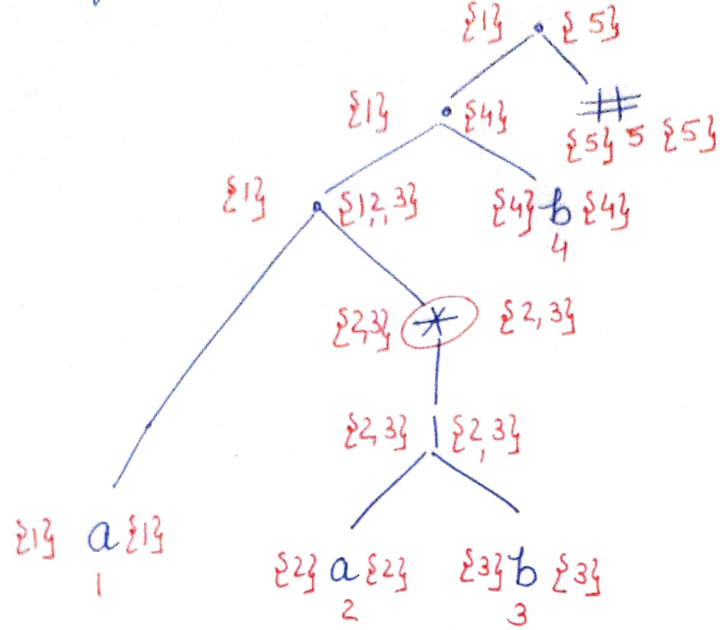


Syntax Tree.

Step-2 find nullable node from constructed syntax tree.

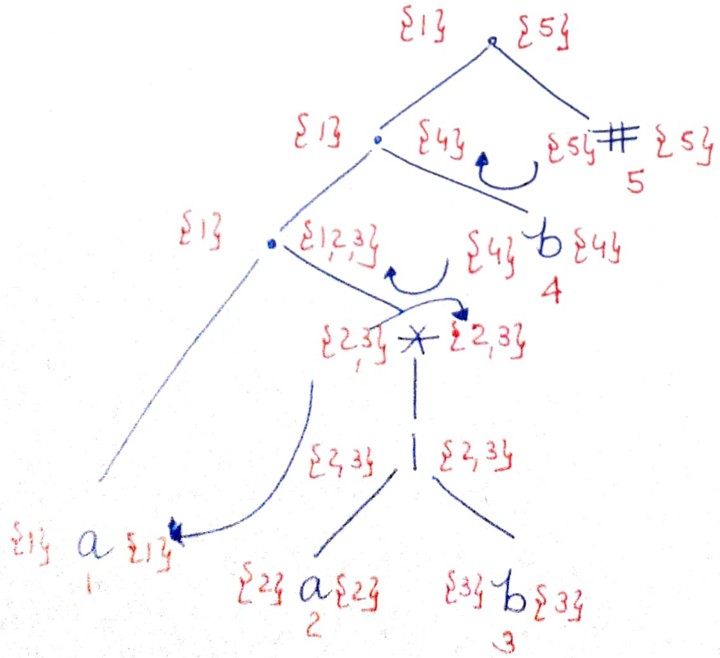


Step-3 Calculate firstpos and ~~po~~ lastpos from constructed syntax tree.



syntax tree

Step-4 Calculate followpos from constructed syntax tree. → followpos table will be ~~constructed~~ (create) by using (• and *) only.



syntax tree

position	followpos
1	2, 3, 4
2	2, 3, 4
3	2, 3, 4
4	5
5	---

followpos table

Now, construct DFA by using constructed syntax tree and followpos table.

Start from top

so initial state $\{1\}$

→ State $A = \{1\}$

Transition $\delta(A, a)$ and $\delta(A, b)$

- $\delta(A, a) = \text{followpos}(1)$
  $= \{2, 3, 4\}$ --- B
- $\delta(A, b) =$ no move by b.

→ State $B = \{2, 3, 4\}$

Transition $\delta(B, a)$ and $\delta(B, b)$

- $\delta(B, a) = \text{followpos}(2)$
  $= \{2, 3, 4\}$ --- B
- $\delta(B, b) = \text{followpos}(3) \cup \text{followpos}(4)$
  $= \{2, 3, 4\} \cup \{5\}$
  $= \{2, 3, 4, 5\}$ — C

→ State $C = \{2, 3, 4, 5\}$

Transition $\delta(C, a)$ and $\delta(C, b)$

- $\delta(C, a) = \text{followpos}(2)$
  $= \{2, 3, 4\}$ — B
- $\delta(C, b) = \text{followpos}(3) \cup \text{followpos}(4)$
  $= \{2, 3, 4\} \cup \{5\}$
  $= \{2, 3, 4, 5\}$ — C

Transition Table

State Q	Input a	b
→$A = \{1\}$	B	ϕ
$B = \{2, 3, 4\}$	B	C
$C = \{2, 3, 4, 5\}$	B	C

Transition Diagram