CD: COMPILER DESIGN
CD: UNIT-1 20/09/2022

20 SEPTEMBER 2022 / IT-3rd year, Vth semester
FALL SEMESTER, YEAR (Vth, 3rd)
FALL SESSION (2022-23)
(CD)
MS. SHWETA TIWARI
Published: SEPTEMBER 20, 2022

PREPARED FOR

Engineering Students

All Engineering College

CD: COMPILER DESIGN

TOPIC On : UNIT-1 LEXICAL ANALYZER, lastpos, firstpos, followpos, syntax tree method DFA

By SHWETA TIWARI

Under On: INTRODUCTION TO COMPILER

TOPIC On : UNIT-1 LEXICAL ANALYZER, lastpos, firstpos, followpos, syntax tree method DFA

Converting Regular Expressions Directly to DFAs
Explanation with examples

Converting Regular Expressions Directly to DFAs

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

- First we augment the given regular expression by concatenating it with a special symbol #.

  r ➔ (r)# augmented regular expression

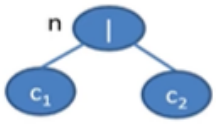
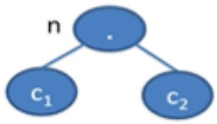
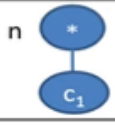
Steps for Converting a RE Directly to a DFA

- Construct a syntax tree for $(r)\#$
- Traverse the tree to construct functions ***nullable***, ***firstpos***, ***lastpos***, **and** ***followpos***
- Computing *followpos*
- Converting **a RE Directly to a DFA**

❑ Function computed from the syntax tree

- *nullable(n)*
  - The subtree at node n generates languages including the empty string.
- *firstpos(n)*
  - The set of positions that can match the first symbol of a string generated by the subtree at node n.
- *lastpos(n)*
  - The set of positions that can match the last symbol of a string generated be the subtree at node n.
- *followpos(i)*
  - The set of positions that can follow position i in the tree.

❑ Rules to compute nullable, firstpos, lastpos

Node *n*	*nullable*(*n*)	*firstpos*(*n*)	*lastpos*(*n*)
A leaf labeled by ε	**true**	∅	∅
A leaf with position i	**false**	$\{i\}$	$\{i\}$
	$nullable(c_1)$ **or** $nullable(c_2)$	$firstpos(c_1)$ ∪ $firstpos(c_2)$	$lastpos(c_1)$ ∪ $lastpos(c_2)$
	$nullable(c_1)$ **and** $nullable(c_2)$	**if** $(nullable(c_1))$ **then** $firstpos(c_1) \cup firstpos(c_2)$ **else** $firstpos(c_1)$	**if** $(nullable(c_2))$ **then** $lastpos(c_1) \cup lastpos(c_2)$ **else** $lastpos(c_2)$
	***true***	$firstpos(c_1)$	$lastpos(c_1)$

❑ Rules to compute followpos

1. If n is **concatenation** node with left child c1 and right child c2 and ***i*** is a position in lastpos(c1), then all position in firstpos(c2) are in followpos(i)

2. If n is * node and ***i*** is position in lastpos(n), then all position in firstpos(n) are in followpos(i)

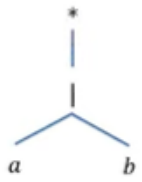
❑ Conversion from regular expression to DFA without constructing NFA

R.E: (a|b)* abb #

❑ Conversion from regular expression to DFA without constructing NFA

R.E: (a|b)* abb #

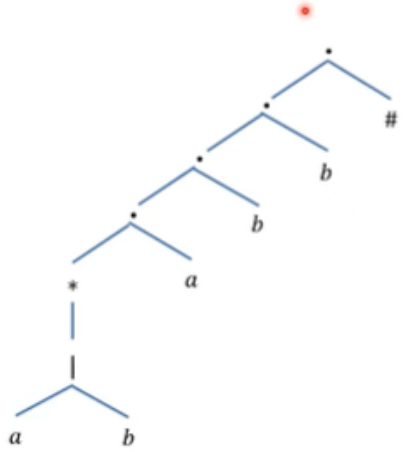
Step 1: Construct Syntax Tree



❑ Conversion from regular expression to DFA without constructing NFA

R.E: (a|b)*abb #

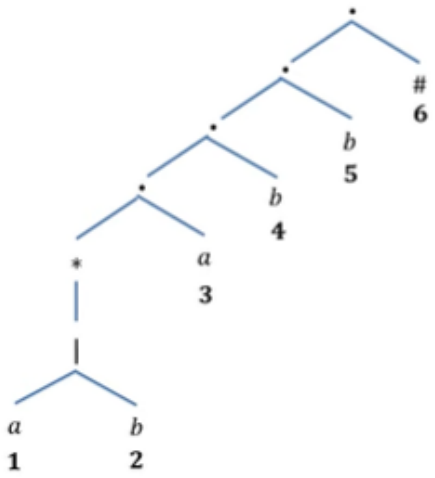
Step 1: Construct Syntax Tree



❑ Conversion from regular expression to DFA without constructing NFA

R.E: (a|b)*abb #

Step 1: Construct Syntax Tree

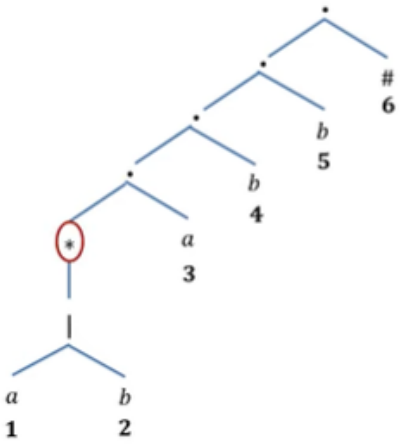


❑ Conversion from regular expression to DFA without constructing NFA

R.E: (a|b)*abb #

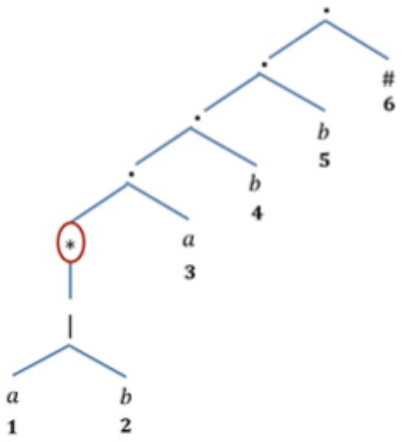
Step 1: Construct Syntax Tree

Step 2: Nullable node



❑ Conversion from regular expression to DFA without constructing NFA

Step 3: Calculate firstpos

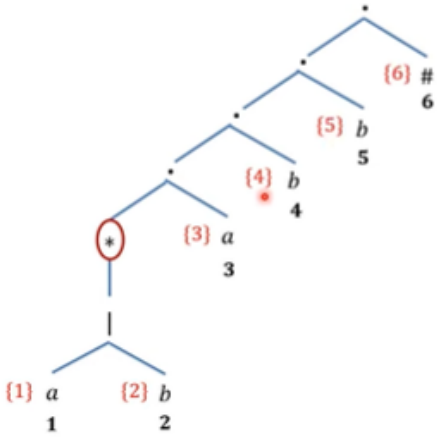


❑ Conversion from regular expression to DFA without constructing NFA

Step 3: Calculate firstpos

Firstpos —

A leaf with position $i = \{i\}$

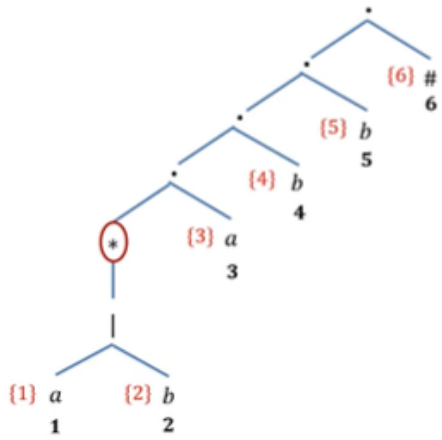


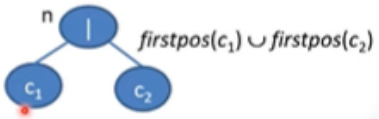
❑ Conversion from regular expression to DFA without constructing NFA

Step 3: Calculate firstpos

Firstpos —

A leaf with position $i = \{i\}$

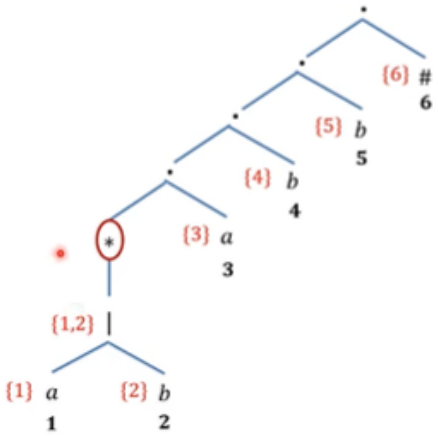




❑ Conversion from regular expression to DFA without constructing NFA

Step 3: Calculate firstpos

Firstpos —



A leaf with position $i = \{i\}$

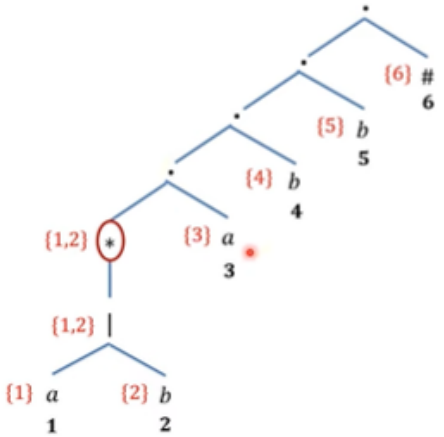
n | c_1 c_2 $firstpos(c_1) \cup firstpos(c_2)$

n * c_1 $firstpos(c_1)$

❑ Conversion from regular expression to DFA without constructing NFA

Step 3: Calculate firstpos

Firstpos —



A leaf with position $i = \{i\}$

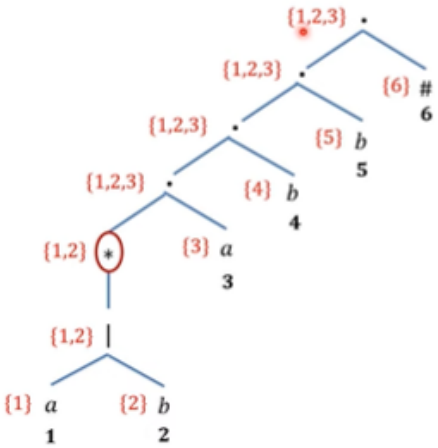
n | c_1 c_2 $firstpos(c_1) \cup firstpos(c_2)$

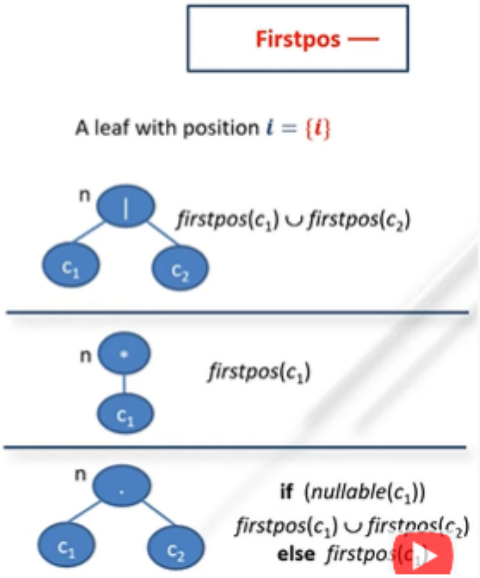
n * c_1 $firstpos(c_1)$

n . c_1 c_2 if $(nullable(c_1))$ $firstpos(c_1) \cup firstpos(c_2)$ else $firstpos(c_1)$

❑ Conversion from regular expression to DFA without constructing NFA

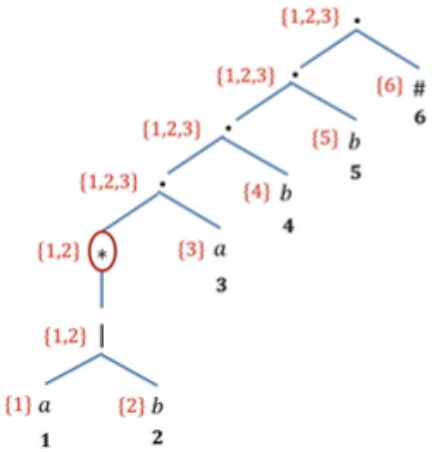
Step 3: Calculate firstpos





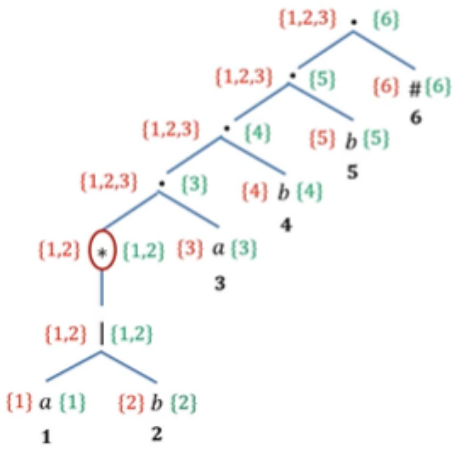
❑ Conversion from regular expression to DFA without constructing NFA

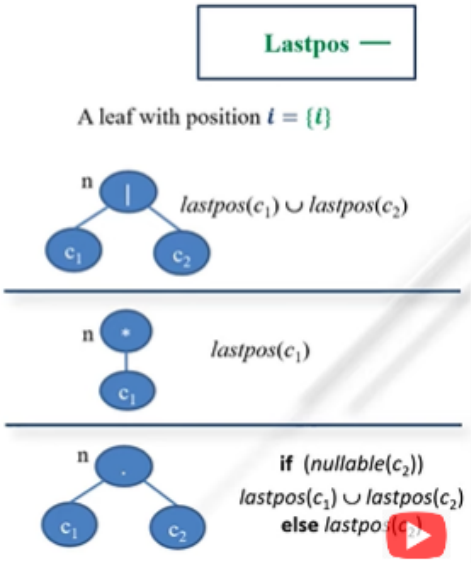
Step 3: Calculate lastpos



❑ Conversion from regular expression to DFA without constructing NFA

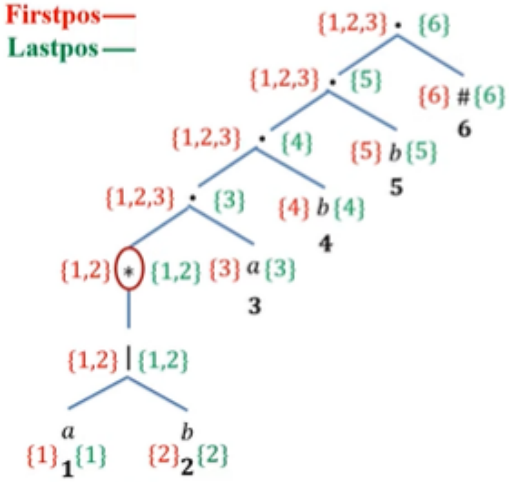
Step 3: Calculate lastpos





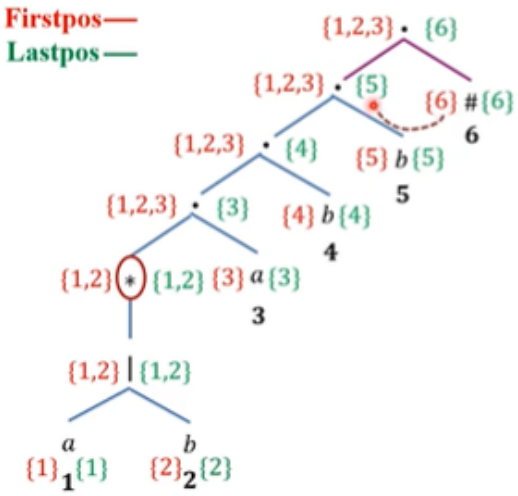
❑ Conversion from regular expression to DFA without constructing NFA

Step 4: Calculate followpos



❑ Conversion from regular expression to DFA without constructing NFA

Step 4: Calculate followpos

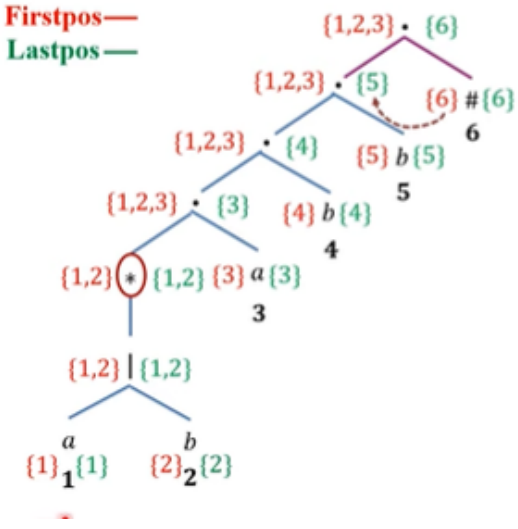


$i = lastpos(c_1) = \{5\}$
$firstpos(c_2) = \{6\}$

❑ Conversion from regular expression to DFA without constructing NFA

Step 4: Calculate followpos

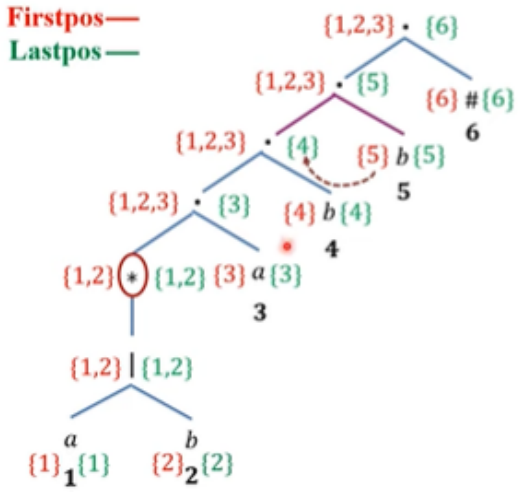
Position	followpos
5	6



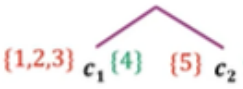
$i = lastpos(c_1) = \{5\}$
$firstpos(c_2) = \{6\}$
$followpos(5) = \{6\}$

❑ Conversion from regular expression to DFA without constructing NFA

Step 4: Calculate followpos



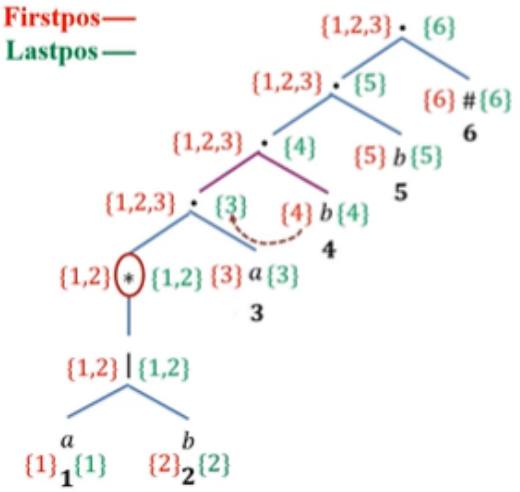
Position	followpos
5	6
4	5



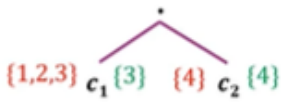
$i = lastpos(c_1) = \{4\}$
$firstpos(c_2) = \{5\}$
$followpos(4) = \{5\}$

❑ Conversion from regular expression to DFA without constructing NFA

Step 4: Calculate followpos



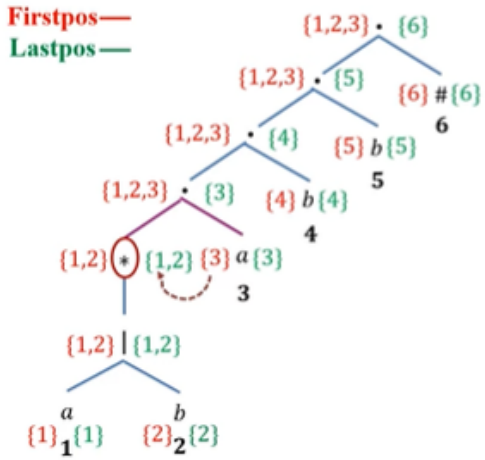
Position	followpos
5	6
4	5
3	4



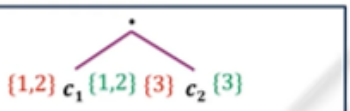
$i = lastpos(c_1) = \{3\}$
$firstpos(c_2) = \{4\}$
$followpos(3) = \{4\}$

❑ Conversion from regular expression to DFA without constructing NFA

Step 4: Calculate followpos



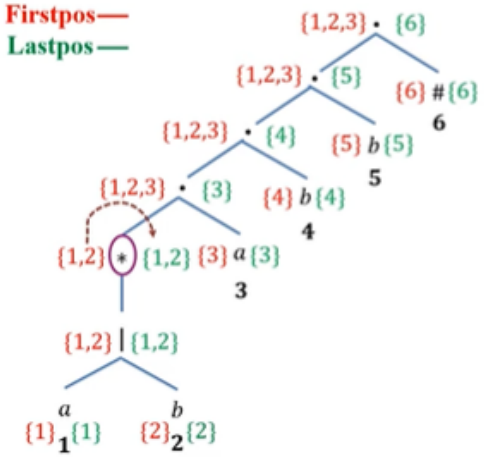
Position	followpos
5	6
4	5
3	4
2	3
1	3



$i = lastpos(c_1) = \{1,2\}$
$firstpos(c_2) = \{3\}$
$followpos(1) = \{3\}$
$followpos(2) = \{3\}$

❑ Conversion from regular expression to DFA without constructing NFA

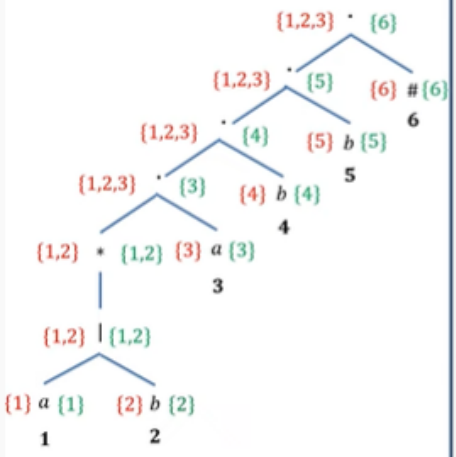
Step 4: Calculate followpos



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

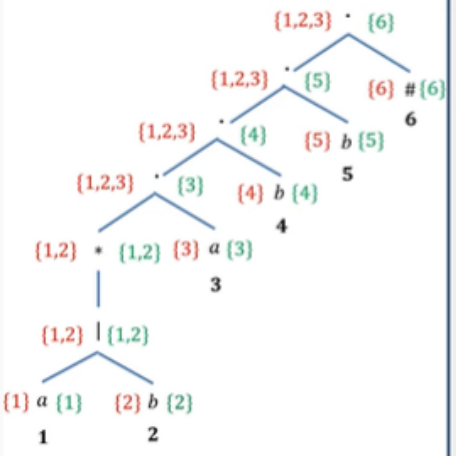
$i = lastpos(n) = \{1,2\}$
$firstpos(n) = \{1,2\}$
$followpos(1) = \{1,2\}$
$followpos(2) = \{1,2\}$

❑ Constructing DFA



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

❑ Constructing DFA



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

State A

δ((A,a) = followpos(1) U followpos(3)

=(1,2,3) U (4) = {1,2,3,4} ----- B

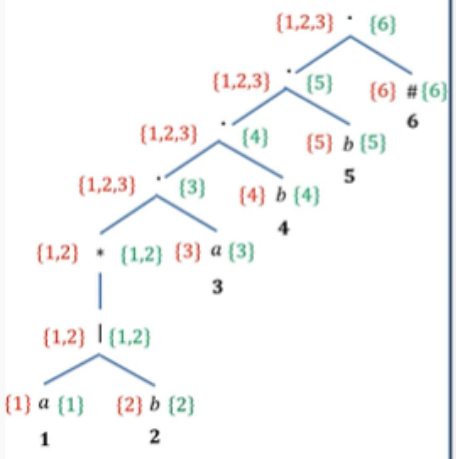
δ((A,b) = followpos(2)

=(1,2,3) ----- A

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

States	a	b
A={1,2,3}	B	A
B={1,2,3,4}		

❑ Constructing DFA



State B

δ((B,a) = followpos(1) U followpos(3)

=(1,2,3) U (4) = {1,2,3,4} ----- B

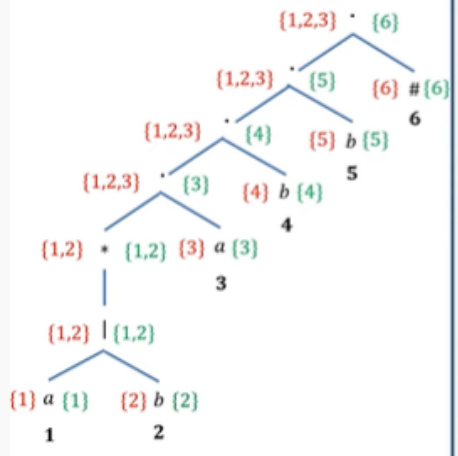
δ((B,b) = followpos(2) U followpos(4)

=(1,2,3) U (5) = {1,2,3,5} ----- C

Position	followpos
5	6
4	5
3	4
2	1,2,3
1	1,2,3

States	a	b
A={1,2,3}	B	A
B={1,2,3,4}	B	C
C={1,2,3,5}		

❑ Constructing DFA



State B

δ((B,a) = followpos(1) U followpos(3)

=(1,2,3) U (4) = {1,2,3,4} ----- B

δ((B,b) = followpos(2) U followpos(4)

=(1,2,3) U (5) = {1,2,3,5} ----- C

State C

δ((C,a) = followpos(1) U followpos(3)

=(1,2,3) U (4) = {1,2,3,4} ----- B

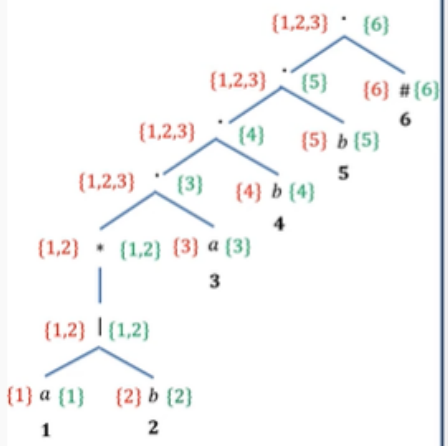
δ((C,b) = followpos(2) U followpos(5)

=(1,2,3) U (6) = {1,2,3,6} ----- D

Position	followpos
5	6
4	5
3	4
2	1,2,3
1	1,2,3

States	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}		

❑ Constructing DFA



State D

δ(D,a) = followpos(1) U followpos(3)

=(1,2,3) U (4) = {1,2,3,4} ----- B

δ(D,b) = followpos(2)

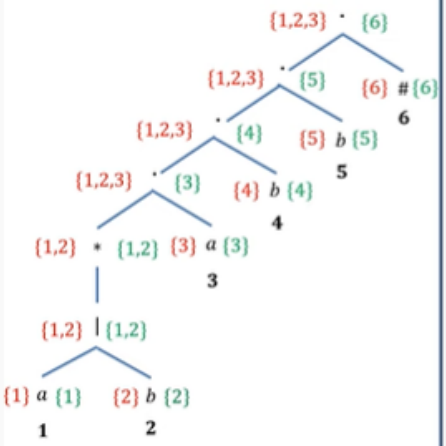
=(1,2,3) ----- A

Position	followpos
6	--
5	{6}
4	{5}
3	{4}
2	{1,2,3}
1	{1,2,3}

States	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 Table for DFA

❑ Constructing DFA



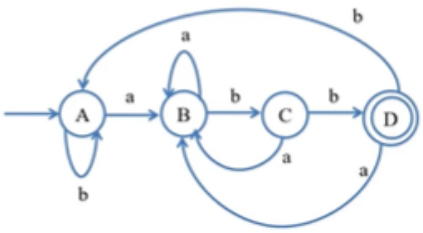
State D

δ(D,a) = followpos(1) U followpos(3)

=(1,2,3) U (4) = {1,2,3,4} ----- B

δ(D,b) = followpos(2)

=(1,2,3) ----- A



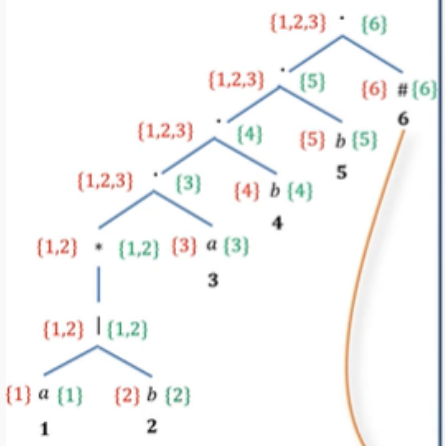
DFA for the R.E: (a|b)*abb

Position	followpos
6	--
5	{6}
4	{5}
3	{4}
2	{1,2,3}
1	{1,2,3}

States	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 Table for DFA

❑ Constructing DFA



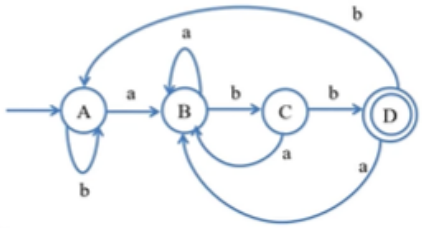
State D

δ(D,a) = followpos(1) U followpos(3)

=(1,2,3) U (4) = {1,2,3,4} ----- B

δ(D,b) = followpos(2)

=(1,2,3) ----- A



DFA for the R.E: (a|b)*abb

Position	followpos
6	--
5	{6}
4	{5}
3	{4}
2	{1,2,3}
1	{1,2,3}

States	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 Table for DFA

❑ Conversion from regular expression to DFA without constructing NFA

R.E: $a^*b^*a(a \mid b)^*b^*a\#$

(Nov-2016) (7 marks)