Direct RE to DFA Construction

DFA (Deterministic Finite Automata)

- A deterministic finite automata(DFA) is a special case of NFA where:
 - 1. There are no moves on input ε
 - 2. for each state, and for each input symbol there is only one state

Direct Regular Expression to DFA

Input: A regular expression r

Output: A DFA D that recognizes L(r)

Step 1: Construct Augmented Regular Expression from the given expression.

Step 2: Construct the syntax tree for the augmented regular expression.

Step 3: Construct the functions nullable, firstpos, lastpos and followpos for syntax tree.

Step 4: Construct Dstates -> set if states of DFA D

Construct Dtrans -> transitions functions for DFA D

Step 5: Draw the DFA for the transition trable.

In the Syntax tree:

- The Leaf node contains alphabet symbols or ε .
- Each non-ε leaf is associated with a unique number- position of the leaf and position of the symbol
- The Internal node contains operators(also known as cat-node, or-node or star-node)

How to evaluate firstpos, lastpos, nullable

<u>n</u>	nullable(n)	firstpos(n)	lastpos(n)
leaf labeled ε	true	Φ	Φ
leaf labeled with position i	false	{i}	{i}
C ₁ C ₂	nullable(c ₁) or nullable(c ₂)	firstpos(c ₁) U firstpos(c ₂)	lastpos(c₁) ∪ lastpos(c₂)
c ₁ c ₂	nullable(c ₁) and nullable(c ₂)	if (nullable(c₁)) firstpos(c₁) ∪ firstpos(c₂) else firstpos(c₁)	if (nullable(c₂)) lastpos(c₁) ∪ lastpos(c₂) else lastpos(c₂)
* C ₁	true	firstpos(c ₁)	lastpos(c ₁)

Example 01

Step 1: Construct an Augmented regular expression

Suppose, given regular expression is:

$$r = (a \mid b)^* abb$$

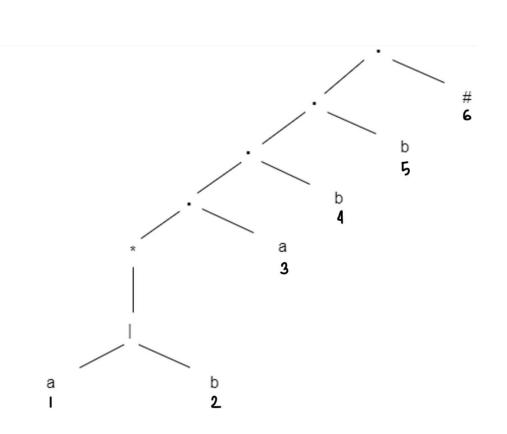
Concat a unique right end marker # to a regular expression r:

$$r# = (a | b)^* abb#$$

Step 2: Construct the syntax tree for the augmented regular expression.

Syntax tree for r# = (a | b)* abb# Here,

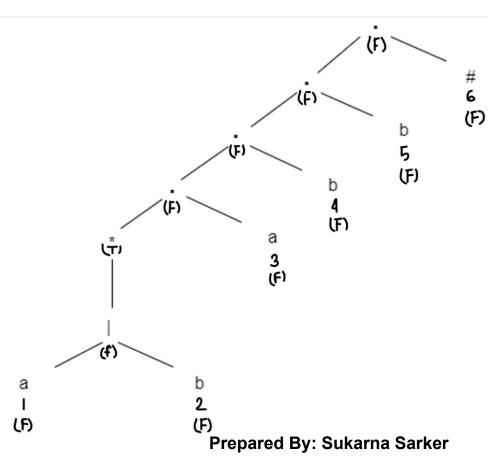
- 1. **a**, **b**, # are in leaf nodes
- 2. |, * , . are in internal nodes
- 3. Each leaf node has **unique numbers**.



1. Nullable

True = Nullable(can produce empty string ε)
False = Not Nullable(can not produce empty string ε)

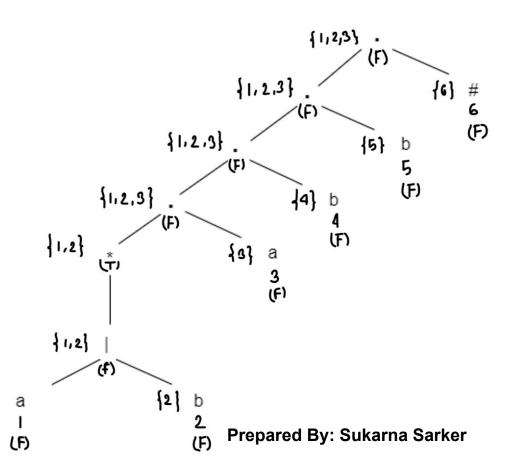
- If n is a leaf labeled by ε then nullable (n) = true
- If n is a leaf labeled with position i then nullable (n) = false
- If n is an or-node (|) with children c1 and c2 then nullable(n) = nullable(c1) or nullable(c2)
- If n is an cat-node (·) with children c1 and c2
 then nullable(n) = nullable(c1) and nullable(c2)
- If n is an star-node (*) with children c1 then nullable (n) = true



{1}

2. Firstpos(n)

- If n is a leaf labeled by ε then firstpos (n) = ∅
- If n is a leaf labeled with position i firstpos (n) = {i}
- If n is an or-node (|) with children c1 and c2 then firstpos(n) = firstpos(c1) ∪ firstpos(c2)
- If n is a cat-node (·) with children c1 and c2 then firstpos(n) = If nullable(c1) then
 - firstpos(c1) ∪ firstpos (c2) else firstpos(c1)
- If n is an star-node (*) with children c1 firstpos (n) = firstpos(c1)

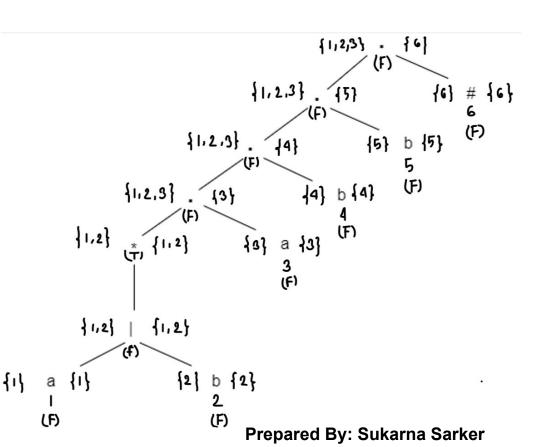


3. Lastpos(n)

- If n is a leaf labeled by ε then lastpos (n) = Ø
- If n is a leaf labeled with position i lastpos (n) = {i}
- If n is an or-node (|) with children c1 and c2
 lastpos (n) = lastpos(c1) ∪ lastpos (c2)
- If n is an cat-node (·) with children c1 and c2 lastpos(n) = If nullable (c2) then lastpos(c1) ∪ lastpos (c2) else

lastpos(c2)

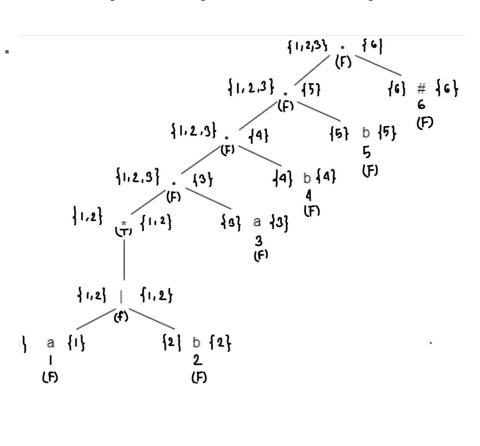
If n is an star-node (*) with children c1
 lastpos (n) = lastpos(c1)



4. Followpos(n)

- If n is a cat-node with left child c1 and right child c2 and i is a position in lastpos (c1), then all positions in firstpos(c2) are in followpos(i)
- If n is a star node, and i is a position in lastpos(n),
 then all positions in firstpos(n) are in followpos(i)

Node	Followpos
1	{1,2,3}
2	{1,2,3}
3	{4}
4	{5}
5	{6}
6	{}



```
initialize Dstates to contain only the unmarked state firstpos(n<sub>0</sub>), where n<sub>0</sub> is the root of syntax tree T for (r)#;

while ( there is an unmarked state S in Dstates ) {
    mark S;
    for ( each input symbol a ) {
        let U be the union of followpos(p) for all p
            in S that correspond to a;
        if ( U is not in Dstates )
            add U as an unmarked state to Dstates;
```

}	if (<i>U</i> i	s not in <i>Dst</i>	tates) unmarked state to Dstates;
State	а	b	firstpos(n0) = $\{1,2,3\}$ ->A
Α			

{1,2,3}. {1,2,3}. {1,2}. {1,2}. {1,2}. {1,2}.	(F) 5 6 5 5 6 6 6 6 6 6	{6} # {6} 6 (F)
11,2} {1,2}	Node	Followpos
(f) {2} b {2}	1	{1,2,3}
(F) (F)	2	{1,2,3}

(F) 2 {1,2,3}
3 {4}
4 {5}
5 {6}
6 {}
Prepared By: Sukarna Sarker

```
initialize Dstates to contain only the unmarked state firstpos(n_0),
      where n_0 is the root of syntax tree T for (r)\#;
while (there is an unmarked state S in Dstates) {
      \max S;
      for (each input symbol a) {
             let U be the union of followpos(p) for all p
                   in S that correspond to a;
             if ( U is not in Dstates )
                   add U as an unmarked state to Dstates;
             Dtran[S, a] = U;
```

Dtran[A,a] = followpos(1) U

followpos(3)

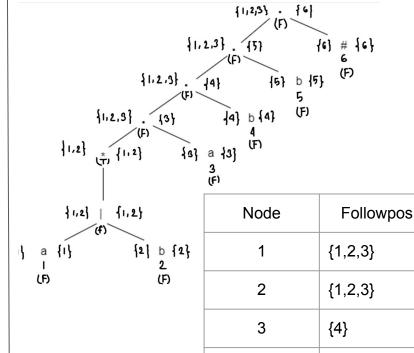
 $= \{1,2,3\} \cup \{4\}$ $= \{1,2,3,4\} -> B$

State	а	b

а	b

D	
_ D	

Α



4

5

Prepared By: Sukarna Sarker

{5}

{6}

```
initialize Dstates to contain only the unmarked state firstpos(n_0), where n_0 is the root of syntax tree T for (r)\#; while ( there is an unmarked state S in Dstates ) { mark S; for ( each input symbol a ) { let U be the union of followpos(p) for all p in S that correspond to a; if ( U is not in Dstates ) add U as an unmarked state to Dstates; Dtran[S, a] = U; }
```

}	if (<i>U</i> i	s not in Ds	tates) unmarked state to Dstates;
State	а	b	Dtran[A,b] = followpos(2)
Α	В	Α	$= \{1,2,3\} -> A$
В			

, , ,	{1,2,3} . {6} (F) (F) (F) (F) (F) (F) (F) (F) (F) (F)	{6} # {6}
{1,2} {1,2}	Node	Followpos
(f) {2} b {2} (F) (F)	1	{1,2,3}
	2	{1,2,3}
	3	{4}
	4	{5}
	5	{6}
	6	{}
Prepa	red By: Sukar	na Sarker

```
initialize Dstates to contain only the unmarked state firstpos(n_0), where n_0 is the root of syntax tree T for (r)\#;

while ( there is an unmarked state S in Dstates ) {
    mark S;
    for ( each input symbol a ) {
        let U be the union of followpos(p) for all p
        in S that correspond to a;
        if ( U is not in Dstates )
        add U as an unmarked state to Dstates;
    Dtran[S, a] = U;
```

J			
State	а	b	Dtran[B,a] = followpos(1) U
Α	В	Α	followpos(3) = {1,2,3} U {4}
В	В		= {1,2,3,4} -> B

		{1,2,3} . (F)	161
	{1,:	2,3} (F)	1
	{1,2,3} (F)	/	b {5] 5
	2.3} (5)	44} b 44} a 43} 3 (F)	(F)
	{1,2}	Node	
(f)	{2} b {2}	1	
(F)	{2 b {2} 2 (F)	2	
		3	
		4	
		5	

Followpos

{1,2,3}

{1,2,3}

{4}

{5}

{6}

{}

6

```
initialize Dstates to contain only the unmarked state firstpos(n_0),
      where n_0 is the root of syntax tree T for (r)#;
while (there is an unmarked state S in Dstates) {
      \max S;
      for (each input symbol a) {
            let U be the union of followpos(p) for all p
                   in S that correspond to a;
            if ( U is not in Dstates )
                   add U as an unmarked state to Dstates;
```

Dtran[S, a] = U;

b а

В Α

C

В В

State

Α

Dtran[B,b] = followpos(2) Ufollowpos(4) $= \{1,2,3\} \cup \{5\}$ $= \{1,2,3,5\} -> C$

Construct L	erans		
		{1,2,3} . {6}	
	{1, :	1,3} . (5)	(6) # {6} 6
	{1,2,3}	4} 45} b 45	/E)
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	(£) /	5	
{1,2,3}	. {3} (F)	ردان ز د ا 4	
1,2}	} {3}	a {3} (F)	
		3 (F)	
{1,2} {1,2	}	Node	Followpos
	2 b {2} 2	1	{1,2,3}
(F)	(F)	2	{1,2,3}
		3	{4}

{5}

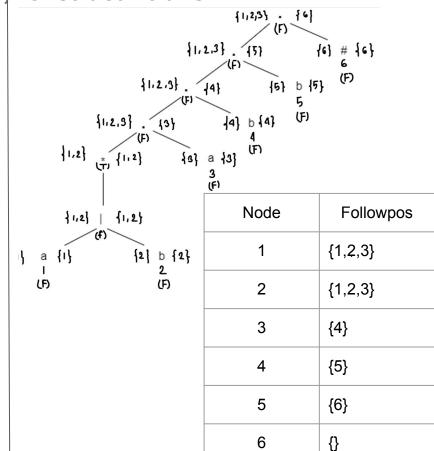
{6}

{}

5

Dtran[C,a] = followpos(1) U
followpos(3)
= {1,2,3} U {4}
= {1,2,3,4} -> B

initialize	Dstates to	contain only	y the unmarked state $firstpos(n_0)$,		
_			vntax tree T for $(r)\#$;		
1000 1000 20		5	tate S in D states) $\{$		
	$\operatorname{ark} S;$				
fo	r (each inp	ut symbol a	a) {		
			of $followpos(p)$ for all p		
			rrespond to a;		
		s not in Ds			
	NO. 030 1000 100		unmarked state to <i>Dstates</i> ;		
		[S,a]=U;	,		
}	•	, , ,			
}					
,					
Ctata	_	L			
State	а	b	Dtran[C,a] = followpos(1) U		
Λ	П	Λ	followpos(3)		
Α	В	Α	$= \{1,2,3\} \cup \{4\}$		
			$= \{1,2,3,4\} \rightarrow B$		
В	В	С	(1,-,0,1)		
С	В				



initialize *Dstates* to contain only the unmarked state $firstpos(n_0)$, where n_0 is the root of syntax tree T for (r)#; **while** (there is an unmarked state S in Dstates) { $\max S$; for (each input symbol a) { let U be the union of followpos(p) for all p in S that correspond to a; **if** (U is not in Dstates)

add U as an unmarked state to Dstates; Dtran[S, a] = U;

b State а

В

В

В

Α

В

Α

C

D

 $= \{1,2,3\} \cup \{6\}$ $= \{1,2,3,6\} \rightarrow D$

Dtran[C,b] = followpos(2) U

followpos(5)

		{1,2,3} . (F)	10
	{1,:	(F) (F)	
	{1,2,3} (F)	[4] [5]	b + 5 (F)
{۱،2,3 ا ابع) ج	(F)	44} b 44} a 431	(F)
(m) '	, ,,	3 (F)	
{1,2} {1,	2}	Node	
(f) a {1}	{2 b {2}	1	
(F)	2 (F)	2	
		3	
		4	
		5	

/(r)						
{1,2,3}. {5} {6} # {6}						
(F) (F)	[4] {5} b {5} 5	(F)				
3}	44} b (4) (F) 4 (F)					
{0}	a {3} (F) 3 (F)					
	(F)					
	Node	Followpos				
{2}	1	{1,2,3}				
- -)	2	{1,2,3}				
	3	{4}				
	4	{5}				
	5	{6}				
	6	{}				

```
initialize Dstates to contain only the unmarked state firstpos(n_0),
      where n_0 is the root of syntax tree T for (r)\#;
while (there is an unmarked state S in Dstates) {
      \max S;
      for (each input symbol a) {
            let U be the union of followpos(p) for all p
                   in S that correspond to a;
             if ( U is not in Dstates )
                   add U as an unmarked state to Dstates;
             Dtran[S, a] = U;
```

b	Dtran[D,a] = followpos(1) U
Α	followpos(3) = {1,2,3} U {4}
С	= {1,2,3,4} -> B

- 7				
			{1,2}	رث، رث
	13	a I (F)	{1,2}	(f)

\$1,2,3

1,2}

1113					
{1,2,3} . {6} {1,2,3} . {5} {6} # {6} 6 (F) (F) (F) (F) (F) (F) (F) (F)					
	Node	Followpos			
[2]	1	{1,2,3}			
	2	{1,2,3}			
	3	{4}			
	4	{5}			
	5	{6}			

Prepared By: Sukarna Sarker

В В В D

а

В

State

Α

В D

```
initialize Dstates to contain only the unmarked state firstpos(n_0),
      where n_0 is the root of syntax tree T for (r)\#;
while (there is an unmarked state S in Dstates) {
      \max S;
      for (each input symbol a) {
            let U be the union of followpos(p) for all p
                   in S that correspond to a;
             if ( U is not in Dstates )
                   add U as an unmarked state to Dstates;
             Dtran[S, a] = U;
```

State	а	b
Α	В	Α

В

В

В

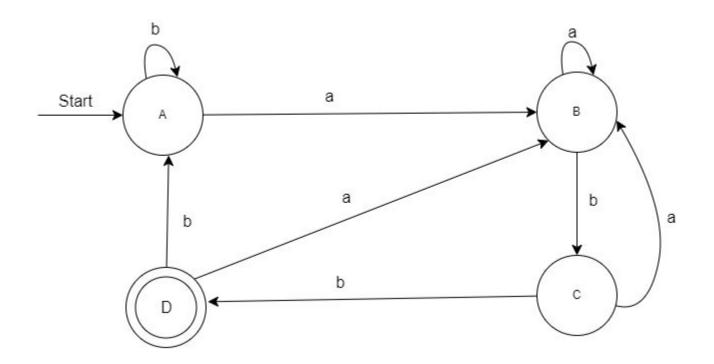
D

Α

Dtran[D,b] = followpos(2)
$= \{1,2,3\} \rightarrow A$

Э,	Constituct Ditails	1	
; 3 ,	\$1,2,3}. \$1,2,3}. \$1,2,3}. \$1,2,3}.	{1,2,3} . {6} (F) (F) 1 (4) (5) b (5) (4) b (4) (F)	(6) # {6} 6 (F)
		a {3} 3 (F)	
	{1,2} {1,2}	Node	Followpos
	(f) {2} b {2}	1	{1,2,3}
	LF) (F)	2	{1,2,3}
		3	{4}
		4	{5}
		5	{6}
		6	{}
	Prepai	red By: Sukarr	na Sarker

Step 5: Draw the DFA for the transition trable.



Example 02

Step 1: Construct an Augmented regular expression

Suppose, given regular expression is:

$$r = (a+b)^* + (ac)^*$$

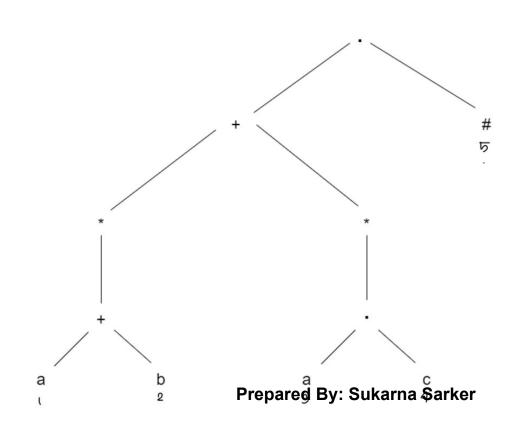
Concat a unique right end marker # to a regular expression r:

$$r# = (a+b)^* + (ac)^*#$$

Step 2: Construct the syntax tree for the augmented regular expression.

Syntax tree for r# = (a+b)* + (ac)*# Here,

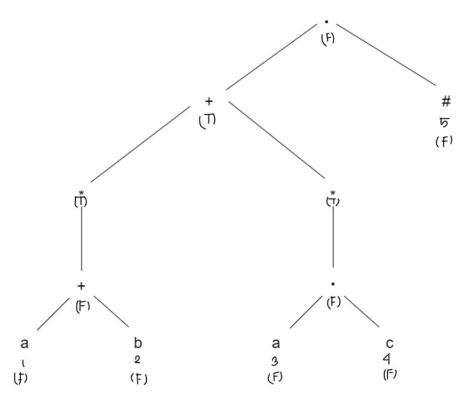
- 1. **a**, **b**, **c**, # are in leaf nodes
- 2. +, *, . are in internal nodes
- 3. Each leaf node has **unique numbers**.



1. Nullable

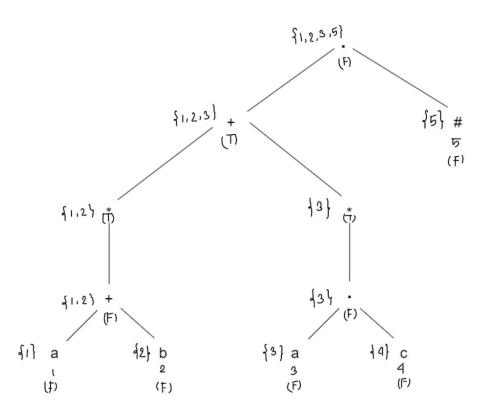
True = Nullable(can produce empty string ε) False = Not Nullable(can not produce empty string ε)

- If n is a leaf labeled by ε then nullable (n) = true
- If n is a leaf labeled with position i then nullable (n) = false
- If n is an or-node (|) with children c1 and c2 then nullable(n) = nullable(c1) or nullable(c2)
- If n is an cat-node (·) with children c1 and c2 then nullable(n) = nullable(c1) and nullable(c2)
- If n is an star-node (*) with children c1 then nullable (n) = true



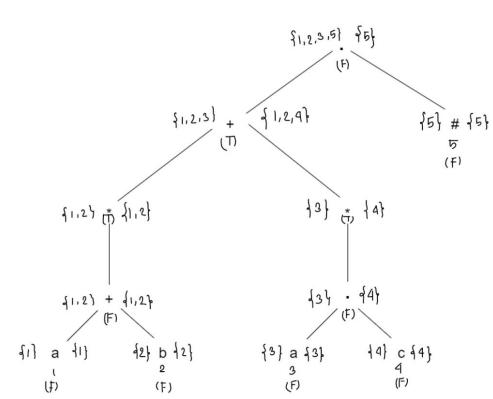
2. Firstpos(n)

- If n is a leaf labeled by ε then firstpos (n) = Ø
- If n is a leaf labeled with position i firstpos (n) = {i}
- If n is an or-node (|) with children c1 and c2 then firstpos(n) = firstpos(c1) ∪ firstpos(c2)
- If n is a cat-node (·) with children c1 and c2 then firstpos(n) = If nullable(c1) then
 - firstpos(c1) ∪ firstpos (c2) else
 - firstpos(c1)
- If n is an star-node (*) with children c1 firstpos (n) = firstpos(c1)



3. Lastpos(n)

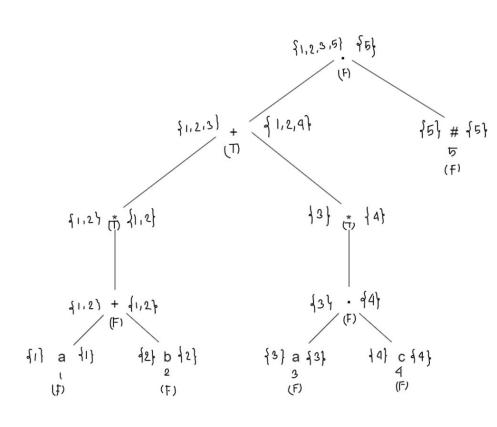
- If n is a leaf labeled by ε then lastpos (n) = ∅
- If n is a leaf labeled with position i lastpos (n) = {i}
- If n is an or-node (|) with children c1 and c2
 lastpos (n) = lastpos(c1) ∪ lastpos (c2)
- If n is an cat-node (·) with children c1 and c2 lastpos(n) = If nullable (c2) then lastpos(c1) ∪ lastpos (c2) else lastpos(c2)
- If n is an star-node (*) with children c1
 lastpos (n) = lastpos(c1)



4. Followpos(n)

- If n is a cat-node with left child c1 and right child c2 and i is a position in lastpos (c1), then all positions in firstpos(c2) are in followpos(i)
- If n is a star node, and i is a position in lastpos(n),
 then all positions in firstpos(n) are in followpos(i)

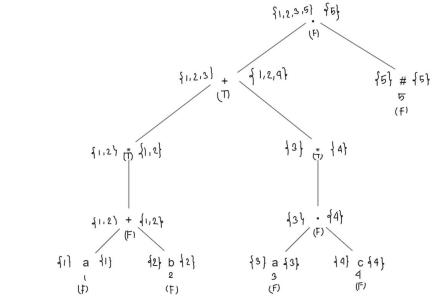
Node	Followpos	
1	{1,2,5}	
2	{1,2,5}	
3	{4}	
4	{3,5}	
5	{}	



initialize Dstates to contain only the unmarked state $firstpos(n_0)$, where n_0 is the root of syntax tree T for (r)#; while (there is an unmarked state S in Dstates) { mark S; for (each input symbol a) { let U be the union of followpos(p) for all p in S that correspond to a; if (U is not in Dstates) add U as an unmarked state to Dstates; Dtran[S, a] = U; }

 $firstpos(n0) = \{1,2,3,5\} -> A$

}		
State	0	1
Α		



Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	^{{4} } Prep	ared By: Sukaı	na Sarker

J			
State	а	b	С
Α	В		
В			

Dtran[A,a] = followpos(1) U followpos(3) = $\{1,2,5\}$ U $\{4\}$ = $\{1,2,5,4\}$ -> B

\{1,2,9,15\}.\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	
{1,2,3} + {1,2,9} (T) (F)	
11.27 (†) {11.2} 13} (†) {4}	
41,2) + 41,27 (F) (F)	
$\{1\}$ $\{2\}$ $\{4\}$	

Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	^{4} Prep	ared By: Sukaı	rna Sarker

```
initialize Dstates to contain only the unmarked state firstpos(n_0), where n_0 is the root of syntax tree T for (r)\#; while ( there is an unmarked state S in Dstates ) { mark S;
    for ( each input symbol a ) {
        let U be the union of followpos(p) for all p in S that correspond to a;
        if ( U is not in Dstates )
            add U as an unmarked state to Dstates;
        Dtran[S, a] = U;
}
```

}		tran[S,	<i>a</i>] =
State	а	b	С
Α	В	С	
В			
С			

Dtran[A,b] = followpos(2) $= \{1,2,5\} -> C$

2002	\(\frac{\xi_1, \gamma, \frac{\xi_5}{\xi_5}}{\xi_5}\)
-	\(\frac{1}{1}\)\(\frac{1}{2}\)\(\frac{1}{1}\)\(\frac{1}\)\(\fra
	11.27 # {11.2} 13} # {4}
	41.2) + 41,27 434 · 447
	fi] a fi] f2 b f2 } {3} a f3} f4 c f4} (F) (F) (F)

Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	^{4} Prep	ared By: Sukaı	rna Sarker

initialize Dstates to contain only the unmarked state $firstpos(n_0)$, where n_0 is the root of syntax tree T for (r)#;while (there is an unmarked state S in Dstates) {
 mark S;
 for (each input symbol a) {
 let U be the union of followpos(p) for all p in S that correspond to a;
 if (U is not in Dstates)
 add U as an unmarked state to Dstates;
 Dtran[S, a] = U;}

State	а	b	(
Α	В	С	

ВС

B

C

Dtran[A,c] = c is not present in node 1,2,3 or 5

	\{1,2,3\} + (T)	\$1,2,5,5} \$5} (F)	र्वड} # र्वड} ज
	f1127 (* {112}	13} * 10	(+)
t	$\begin{cases} 1 & 2 \\ + & 1 \\ 2 \\ (F) \end{cases}$ $\begin{cases} 1 \\ 1 \\ 3 \\ 1 \end{cases}$ $\begin{cases} 1 \\ 2 \\ 42 \end{cases}$ $\begin{cases} 2 \\ 5 \\ (F) \end{cases}$	{3} a {3}	1 1 1 2 4 (F)

Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	^{4} Prep	ared By: Sukaı	rna Sarker

```
initialize Dstates to contain only the unmarked state firstpos(n_0), where n_0 is the root of syntax tree T for (r)\#; while ( there is an unmarked state S in Dstates ) { mark S; for ( each input symbol a ) { let U be the union of followpos(p) for all p in S that correspond to a; if ( U is not in Dstates ) add U as an unmarked state to Dstates; Dtran[S, a] = U; }
```

}		Dtran[S, a]		
State	а	b	(
Α	В	С	-	
В	С			
С				

Dtran[B,a] = followpos(1) $= \{1,2,5\}$

 \$1,2,3} + {1,2,4}
11,27 th {1,2} {3} th
$\{1,2\} + \{1,2\}$ $\{3\} \cdot \{4\}$ $\{5\} = \{1\}$ $\{1\} = \{1\}$ $\{2\} = \{2\}$ $\{3\} = \{3\}$ $\{4\}$ $\{5\} = \{4\}$ $\{5\} = \{5\}$ $\{6\} = \{6\}$ $\{6\}$ $\{6\} = \{6\}$ $\{6\}$

Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	^{4} Prep	ared By: Sukaı	rna Sarker

```
initialize Dstates to contain only the unmarked state firstpos(n_0), where n_0 is the root of syntax tree T for (r)\#; while ( there is an unmarked state S in Dstates ) { mark S; for ( each input symbol a ) { let U be the union of followpos(p) for all p in S that correspond to a; if ( U is not in Dstates ) add U as an unmarked state to Dstates; Dtran[S, a] = U; }
```

}			
State	а	b	c
Α	В	С	-
В	С	С	
С			

Dtran[B,b] = followpos(2) $= \{1,2,5\}$

 \$1,2,3} + {1,2,4}
11,27 th {1,2} {3} th
$\{1,2\} + \{1,2\}$ $\{3\} \cdot \{4\}$ $\{5\} = \{1\}$ $\{1\} = \{1\}$ $\{2\} = \{2\}$ $\{3\} = \{3\}$ $\{4\}$ $\{5\} = \{4\}$ $\{5\} = \{5\}$ $\{6\} = \{6\}$ $\{6\}$ $\{6\} = \{6\}$ $\{6\}$

Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	^{4} Prep	ared By: Sukaı	rna Sarker

```
initialize Dstates to contain only the unmarked state firstpos(n_0), where n_0 is the root of syntax tree T for (r)\#; while ( there is an unmarked state S in Dstates ) { mark S;
    for ( each input symbol a ) {
        let U be the union of followpos(p) for all p in S that correspond to a;
        if ( U is not in Dstates )
            add U as an unmarked state to Dstates;
        Dtran[S, a] = U;
}
```

}			
State	а	b	С
Α	В	С	-
В	С	С	D
С			
D			

Dtran[B,c] = followpos(4) $= \{3,5\} -> D$

\$1,2,2,5} {5}
{1,2,3} + {1,2,4} {5} # {5}
11.27 th 11.23
(F) (F) (3) - 44)
(F) {2} b {2} {3} a {3} {4} c {4} {4} {5} (F) (F)

Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	{4} Pror	nared By: Suka	rna Sarkor

}			
State	а	b	С
Α	В	С	-
В	С	С	D
С	С		
D			

Dtran[C,a] = followpos(1) $= \{1,2,5\}$

\$1,2,2,5} {5}
{1,2,3} + {1,2,4} {5} # {5}
11.27 th 11.23
(F) (F) (3) - 44)
(F) {2} b {2} {3} a {3} {4} c {4} {4} {5} (F) (F)

Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	^{4} Prep	ared Bv: Sukaı	rna Sarker

}					
State	а	b	С		
Α	В	С	-		
В	С	С	D		
С	С	С	-		
D					

Dtran[C,c] = c is not present in node 1,2 or 5

	\$1,2,3,5} \$5}
1	(T) {1,2,4} {5} # {5}
f1,2} ₩ {1,2}	13} * 14}
41,27 (F)	(5) (F)
41) a 41) 42) b 4	2} {3} a {3}

Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	{4} Prop	arad By: Suka	rna Sarkar

initialize *Dstates* to contain only the unmarked state $firstpos(n_0)$, where n_0 is the root of syntax tree T for (r)#; while (there is an unmarked state S in Dstates) { ymbol a) { e union of followpos(p) for all pthat correspond to a; in Dstates) U as an unmarked state to *Dstates*; =U;

$egin{array}{l} \mathrm{mark} \ S; \\ \mathbf{for} \ (\ \mathrm{each} \ \mathrm{input} \ s \ \\ \mathrm{let} \ U \ \mathrm{be} \ \mathrm{th} \\ \mathrm{in} \ S \ \end{array}$				
<u>,</u>	D	t (U is a at S)	dd U	
State	а	b	C	
Α	В	С	_	
В	С	С	С	
С	С	С	_	
D	Е			

Ε

Dtran[D,a] = followpos(3)={4} ->E

,			f1,2,3,5} f5}		
			(F)		
٠		{1,2,3} +	﴿ ا, 2, 4}	{5} # {5}	
		(1)	り (f)	
	٠,٠	14 * 51.2}	13} (T) {a	•	
	f1,27 t (1,2}		, , (a)		
			(0) = 0(1)		
	41.1	2) + d1,27p	434 · 944		
	र्गो a री	} 42} b 12}	{3} a {3}	c 44}	
	(t)	(F)	(F)	(F)	
	Node	Followpos	Node	Followpos	
	1	{1,2,5}	4	{3,5}	

Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}

Prepared By: Sukarna Sarker

{4}

initialize *Dstates* to contain only the unmarked state $firstpos(n_0)$, where n_0 is the root of syntax tree T for (r)#; **while** (there is an unmarked state S in Dstates) { $\max S$; for (each input symbol a) { let U be the union of followpos(p) for all p in S that correspond to a; **if** (U is not in Dstates) add U as an unmarked state to Dstates; Dtran[S, a] = U;

а	b	
В	С	

С

	ш	L	I.

В	С	С	
С	С	С	
D	Е	-	

State

Α

Ε

Dtran[D,b] = b is not preser in node 3 or 5

25555		\$1,2,3,5} \$5}
	\$11.2.3} + (T)	{ 1,2,4} { 5} # { 5}
	f1,27 t {1,2}	13} * {4}
	\$1,2) + \$1,2} (F)	434 · 444
		{3} a {3}

nt	ا، : إ إا a إا إلى	(F) 11,27 (F) 12}	{3} a {3} 44	c {4} 4 (F)
	Node	Followpos	Node	Followpo
	1	{1,2,5}	4	{3,5}

Q'	117	ζ,	
Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	^{4} Prep	ared By: Sukaı	rna Sarker

initialize Dstates to contain only the unmarked state $firstpos(n_0)$, where n_0 is the root of syntax tree T for (r)#; while (there is an unmarked state S in Dstates) { mark S; for (each input symbol a) { let U be the union of followpos(p) for all p in S that correspond to a; if (U is not in Dstates) add U as an unmarked state to Dstates; Dtran[S, a] = U; }

}		t (U) is a $tran[S,$	$\mathrm{dd}\ U$
tate	а	b	С
Α	В	С	-
В	С	С	D

Dtran[D,c] = c is not present In node 3 or 5

,		\(\frac{1,2,3,5}{(F)}\)				
		{1,2,3} +	1,2,9}	{5} # {5} 5 (f)		
	4 11.	2) + {1,2} (F)	{3} * {4}			
t	ના! a ના ((‡)	12} b 12}	13/ a 43} 14 3 (F)	1 C 14} 4 (F)		
	Node	Followpos	Node	Followpos		
	1	{1,2,5}	4	{3,5}		
	2	{1,2,5}	5	{}		
	3	^{4} Prep	ared By: Suka	rna Sarker		

В	С	С	
С	С	С	
D	Е	-	
E			

initialize *Dstates* to contain only the unmarked state $firstpos(n_0)$, where n_0 is the root of syntax tree T for (r)#; **while** (there is an unmarked state S in Dstates) { $\max S$; for (each input symbol a) { let U be the union of followpos(p) for all p in S that correspond to a; **if** (U is not in Dstates) U as an unmarked state to Dstates; =U;

)		tran[S,	$\mathrm{dd}\ U$
state	а	b	С
Α	В	С	-
В	С	С	D

Dtran[Fa] = a is not prese In

ent	
	No

41,2) + d1,2%

{3} a {3} **Followpos**

Prepared By: Sukarna Sarker

S1,2,3,5} 85}

rııanı∟,aj –	a is	HOL	hiese	711
n node 4				

ı				
	Node	Followpos	Node	
	1	{1,2,5}	4	{;

{4}

1	{1,2,5}	4	{3,5}	
2	{1,2,5}	5	{}	

Ε D Ε

initialize Dstates to contain only the unmarked state $firstpos(n_0)$, where n_0 is the root of syntax tree T for (r)#; while (there is an unmarked state S in Dstates) { mark S; for (each input symbol a) { let U be the union of followpos(p) for all p in S that correspond to a; if (U is not in Dstates) add U as an unmarked state to Dstates; Dtran[S, a] = U; }

, }		C(U) is action $C(S, T)$	dd l
State	а	b	(
Α	В	С	
В	С	С	Г

Dtran[E,b] = b is not present In node 4

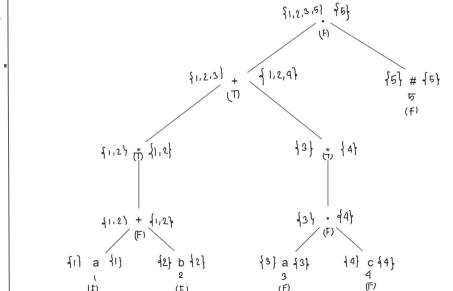
	\$1,2,3} + (T) (T) (E) + {1,2}	{3} * {4}	
	12} b 12} (F)	{3} a {3}	
Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	^{4} Prep	ared By: Suka	rna Sarker

{1,2,3,5} \$5}

B C C
C
C C
D E E -

initialize *Dstates* to contain only the unmarked state $firstpos(n_0)$, where n_0 is the root of syntax tree T for (r)#; while (there is an unmarked state S in Dstates) { mont C.

$\max S$;							
f	for (each input symbol a) {						
	ıe			p ion of $followpos(p)$ for all p			
	: c			t correspond to a ;			
	П	0.00		Dstates)			
	7			s an unmarked state to <i>Dstates</i> ;			
1		tran[S,	[a] = U	';			
}	•						
state	а	b	С	*			
Α	В	С	-	Dtran[E,c] = followpos(4)			
В	С	С	D	= {3,5}			
С	С	С	-				
D	Е	-	-				
E	-	-	D				
				•			



4113 413 a 41 (t)	2) + {1,2} (F) {2} b {2} (F)	{3} · 44} (F) {3} a {3} (F)	c {4} 4 (F)
Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}

(1)	(1)	(1)	V 2
Node	Followpos	Node	Followpos
1	{1,2,5}	4	{3,5}
2	{1,2,5}	5	{}
3	{4} Pren	ared By: Sukai	rna Sarker

Step 5: Draw the DFA for the transition trable.

