

LL(1)

①

Ques: Grammar:

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \epsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \epsilon$$

$$F \rightarrow (E) \mid id$$

ⓧ Follow (এর এর) - start variable is always \$
২৬৮।

Step 1

Ans:

First

$$\text{first}(E) : \{ (, id \}$$

$$\text{first}(E') : \{ +, \epsilon \}$$

$$\text{first}(T) : \{ (, id \}$$

$$\text{first}(T') : \{ *, \epsilon \}$$

$$\text{first}(F) : \{ (, id \}$$

Follow

$$\text{follow}(E) : \{ \$,) \}$$

$$\text{follow}(E') : \{ \$,) \}$$

$$\text{follow}(T) : \{ +, \$,) \}$$

$$\text{follow}(T') : \{ +, \$,) \}$$

$$\text{follow}(F) : \{ *, +, \$,) \}$$

step 2

(2)

LL(1) Parsing table :

Upper case / Production	terminal	+	*	()	id	\$
E				$E \rightarrow TE'$		$E \rightarrow TE'$	
E'		$E' \rightarrow +TE'$			$E' \rightarrow E$		$E' \rightarrow E$
T				$T \rightarrow FT'$		$T \rightarrow FT'$	
T'		$T' \rightarrow E$	$T' \rightarrow *FT'$		$T' \rightarrow E$		$T' \rightarrow E$
F				$F \rightarrow (E)$		$F \rightarrow id$	

step 3

Stack	Input	Action
\$ E	(id * id) + id	$E \rightarrow TE'$
\$ E' T	(id * id) + id	$T \rightarrow FT'$
\$ E' T' F	(id * id) + id	$F \rightarrow (E)$
\$ E' T') E (id * id) + id	Match & pop
\$ E' T') E	id * id) + id	$E \rightarrow TE'$
\$ E' T') E' T	id * id) + id	$T \rightarrow FT'$
\$ E' T') E' T' F	id * id) + id	$F \rightarrow id$
\$ E' T') E' T' id	* id) + id	Match & pop
\$ E' T' E' T'	* id) + id	$T' \rightarrow *FT'$
\$ E' T' E' T' F *	* id) + id	Match & pop
\$ E' T' E' T' F	id) + id	$F \rightarrow id$

④

Tree

Input (id * id) + id

Grammer

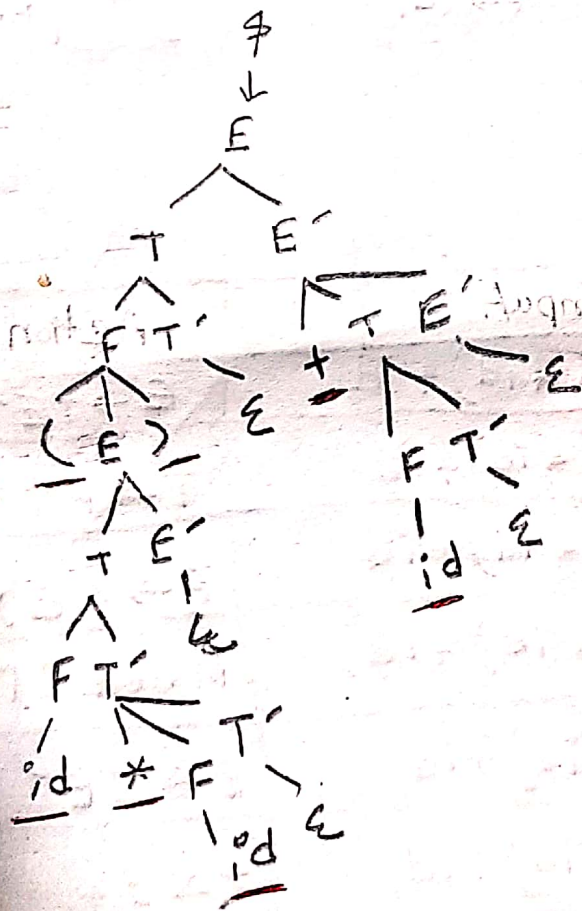
$E \rightarrow TE'$

$E' \rightarrow +TE' / \epsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' / \epsilon$

$F \rightarrow (E) / id$



LR(0) = Closure < Goto

Grammar, $S \rightarrow AA$
 $A \rightarrow aA | b$

CFG = context free grammar
Left recursion

$A \rightarrow Ad | B$

✓ No CFG

✓ no ambiguity

Step 1: add argument, $S' \rightarrow S$

$S' \rightarrow S$

$S \rightarrow AA$

$A \rightarrow aA | b$

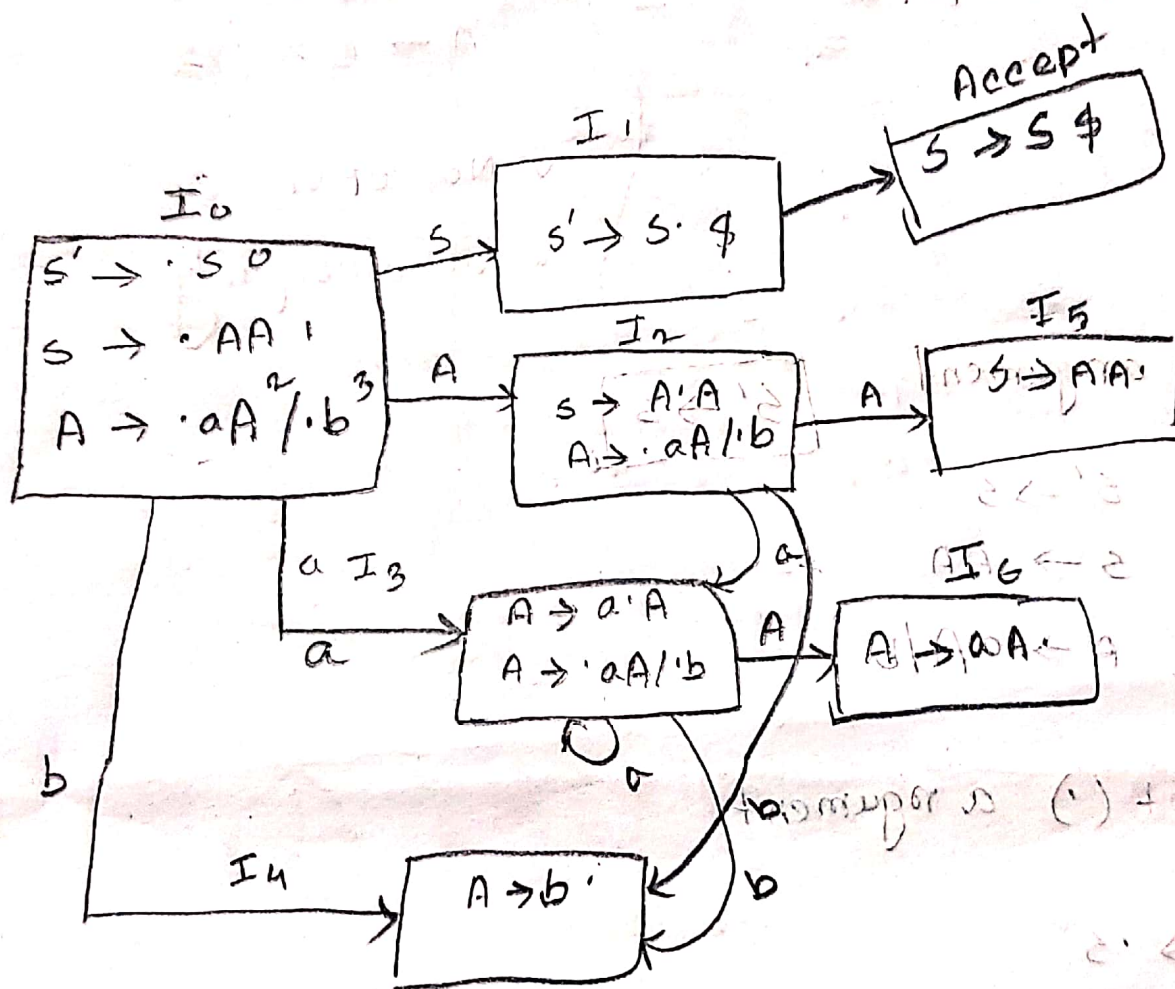
Step 2: add dot (.) argument

$S' \rightarrow \cdot S^0$

$S \rightarrow \cdot AA^1$

$A \rightarrow \cdot aA | b$
 2 3

step 3: DFA Draw.



LR(0) table

terminal (Lower case)

non Terminal / upper case

Stack		Action			Goto	
		a	b	\$	A	S
I ₀	0	S ₃	S ₄		2	1
I ₁	1			Accept		
I ₂	2	S ₃	S ₄		5	
I ₃	3	S ₃	S ₄		6	
I ₄	4	r ₃	r ₃	r ₃		
I ₅	5	r ₁	r ₁	r ₁		
I ₆	6	r ₂	r ₂	r ₂		

if input aabb\$

Stack	input	Action
0	aabb\$	shift 3
0a3	abb\$	shift 3
0a3a3	bb\$	shift 4
0a3a3b4	bb\$	reduce by A → b
0a3a3A6	b\$	reduce by A → aA
0Ab2	b\$	shift 4
0Ab2b4	\$	reduce by A → b
0A2A5	\$	reduce by S → AA
0S1	\$	Accept

Murtash
min

SLR (1) -

given, $S \rightarrow cAd$ input =
 $A \rightarrow ab|e$

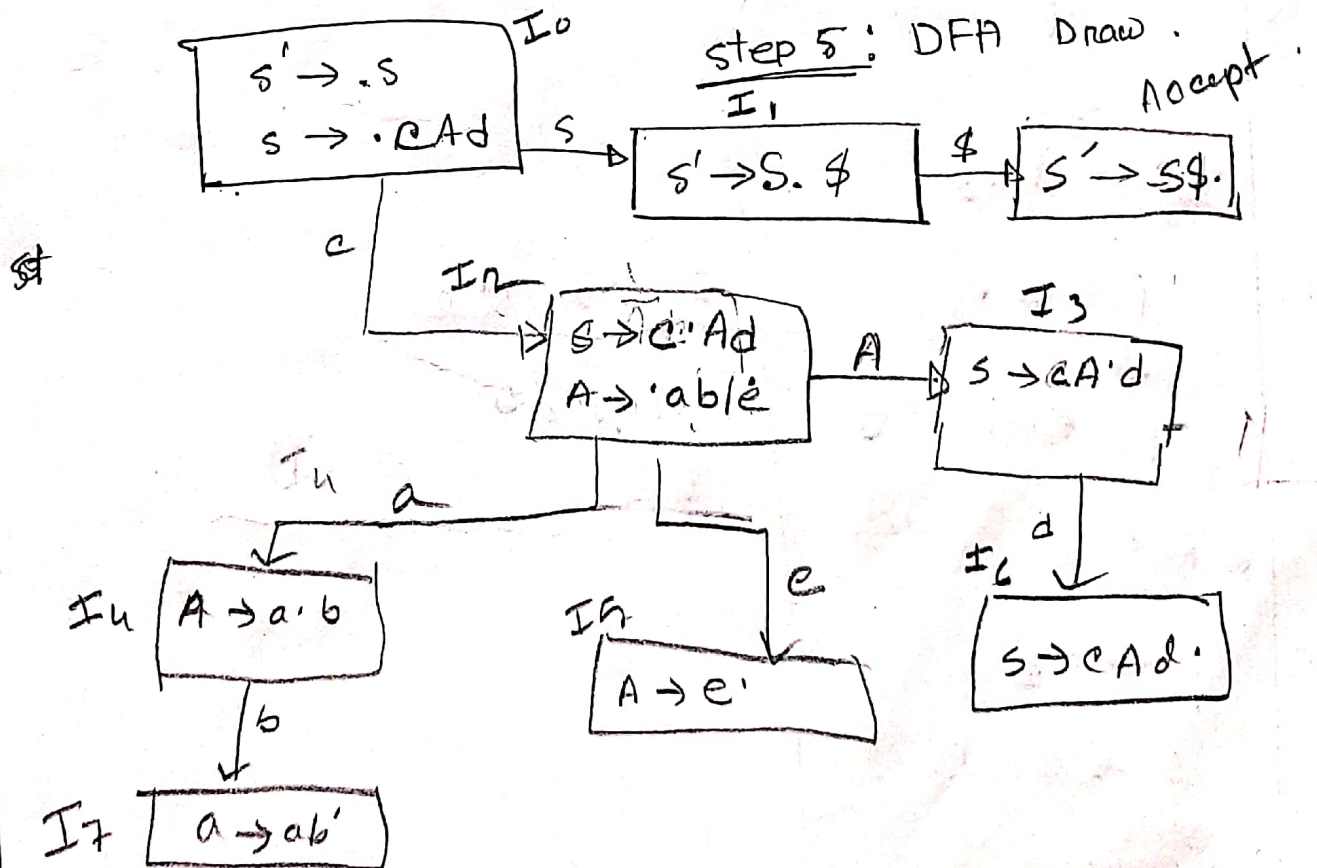
step 1: $s' \rightarrow s$ augment add
 $S \rightarrow cAd$ numbering
 $A \rightarrow ab|e$

step 2: number it

step 3: First & Follow

	First	Follow
S	{c}	{ \$ }
A	{a, e}	{d}

step 4: augment w/ grammar (step 1) \rightarrow (.) (dot) add



SAG

step 6: SLR(1) parsing table

	stack	Action				Goto	
		a	b	\$		A	S
I ₀	0			S ₁			1
I ₁	1			Accepted			
I ₂	2	S ₄ S ₅				3	
I ₃	3						
I ₄	4			S ₆			
I ₅	5			r ₃			
I ₆	6			A r ₁ r ₂			
I ₇	7			r ₂			

I₅ → r₃

A → e

Follow(A) = d

d num w/o r₃ off

I₆ → r₁

S → cAd

follow \$

I₇ → r₂

A → ab

↓

LR(1)
a b c
R: R3 R3
R1 R1 R1

SLR(1)
a b c
5 | MAB
6 | R1

CLR(1)
a b c
3 | R3
6 | R1

LALR(1)

step 7: if input, then generate SIA table

Stack	Input	Action
0	ced\$	shift 2
0c2	ed\$	shift 5
0c2e5	d\$	reduce by A → e.
0c2A3	d\$	shift 6
0c2A3d6	\$	reduce by s → Ad
0s1	\$	Accept

2A
0
stop
num
3
5
12