

# SLR Parser

LR(0)

Ex-1  $S \rightarrow AS|b$   
 $A \rightarrow SA|a$

$S \rightarrow AS$  0  
 $S \rightarrow b$  1  
 $A \rightarrow SA$  2  
 $A \rightarrow a$  3

Step 1

Augmented Grammar

$S' \rightarrow S$  } 0  
 $S \rightarrow AS$  } 1  
 $S \rightarrow b$  } 2  
 $A \rightarrow SA$  } 3  
 $A \rightarrow a$  } 4

Step-1: Canonical Collection of calculating of LR(0) items

$I_0$   $S' \rightarrow \cdot S$  0  
 $S \rightarrow \cdot AS$  1  
 $S \rightarrow \cdot b$  2  
 $A \rightarrow \cdot SA$  3  
 $A \rightarrow \cdot a$  4

Goto ( $I_0, S$ )

$S' \rightarrow S \cdot$

$A \rightarrow S \cdot A$   
 $A \rightarrow \cdot SA$   
 $A \rightarrow \cdot a$   
 $S \rightarrow \cdot AS$   
 $S \rightarrow \cdot b$

Goto ( $I_0$ , A)

$I_2$   $S \rightarrow A \cdot S$

$S \rightarrow \cdot b$

$S \rightarrow \cdot AS$

$A \rightarrow \cdot SA$

$A \rightarrow \cdot a$

Goto ( $I_0$ , b)

$I_3$   $S \rightarrow b \cdot$

Goto ( $I_0$ , a)

$I_4$   $A \rightarrow a \cdot$

Goto ( $I_1$ , A)

$I_5$   $A \rightarrow SA \cdot$

$S \rightarrow A \cdot S$

$S \rightarrow \cdot AS$

$S \rightarrow \cdot b$

$A \rightarrow \cdot SA$

$A \rightarrow \cdot a$

Goto ( $I_1$ , S)

$I_6$   $A \rightarrow S \cdot A$

$A \rightarrow \cdot SA$

$A \rightarrow \cdot a$

$S \rightarrow \cdot AS$   $S \rightarrow \cdot b$

Goto ( $I_1$ , a)

~~$I_4$~~   $A \rightarrow a \cdot$   $I_4$

Goto ( $I_1$ , b)

$S \rightarrow b \cdot$   $I_3$

Goto ( $I_2$ , S)

$I_7$   $S \rightarrow AS \cdot$

$A \rightarrow S \cdot A$

$A \rightarrow \cdot SA$

$A \rightarrow \cdot a$

$S \rightarrow \cdot b$

$S \rightarrow \cdot AS$

Goto ( $I_2$ , A)

~~$S \rightarrow AS \cdot$~~   ~~$A \rightarrow S \cdot A$~~   $S \rightarrow A \cdot S$

~~$I_4$~~   $S \rightarrow \cdot b$

$S \rightarrow \cdot AS$

$A \rightarrow \cdot SA$

$A \rightarrow \cdot a$

Goto ( $I_2$ , a)

$A \rightarrow a \cdot$   $I_4$

Goto ( $I_2$ , b)

$S \rightarrow b \cdot$   $I_3$

Goto ( $I_5$ , S)

$S \rightarrow AS \cdot$

$A \rightarrow S \cdot A$

$A \rightarrow \cdot SA$

$A \rightarrow \cdot a$

$I_7$

Goto ( $I_5, A$ )

$S \rightarrow A \cdot S$   
 $S \rightarrow \cdot AS$   
 $S \rightarrow \cdot b$   
 $A \rightarrow \cdot SA$   
 $A \rightarrow \cdot a$

}  $I_2$

Goto ( $I_5, a$ ):

$A \rightarrow a \cdot$   $I_4$

Goto ( $I_5, b$ ):

$S \rightarrow b \cdot$   $I_3$

Goto ( $I_6, A$ )

$A \rightarrow SA \cdot$   
 $S \rightarrow A \cdot S$   
 $S \rightarrow \cdot AS$   
 $S \rightarrow \cdot b$   
 $A \rightarrow \cdot SA$   
 $A \rightarrow \cdot a$

}  $I_5$

Goto ( $I_6, S$ )

$A \rightarrow S \cdot A$   
 $A \rightarrow \cdot SA$   
 $A \rightarrow \cdot a$

}  $I_6$

Goto ( $I_6, a$ ):

$a \cdot$   $I_4$

Goto ( $I_6, b$ ):

$b \cdot$   $I_3$

Goto ( $I_7, A$ )

$A \rightarrow SA \cdot$   
 $S \rightarrow A \cdot S$

}  $I_5$

Goto ( $I_7, S$ )

$A \rightarrow S \cdot A$   
 $A \rightarrow \cdot SA$   
 $A \rightarrow \cdot a$   
 $S \rightarrow \cdot AS$   
 $S \rightarrow \cdot b$

}  $I_6$

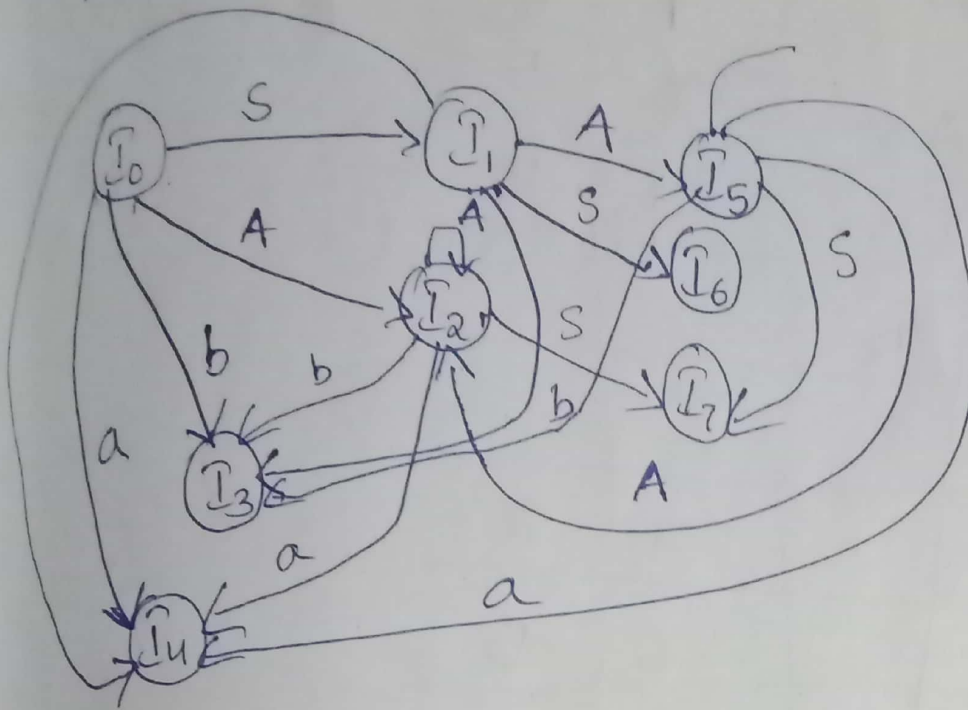
Goto ( $I_7, a$ )

$a \cdot$   $I_4$

Goto ( $I_7, b$ )

$b \cdot$   $I_3$

Step-2:-



Step-3:- Construction of Parse table

$\text{First}(S) = \{b, a\}$

$\text{follow}(S) = \{\$, a, b\}$

$\text{First}(A) = \{a, b\}$

$\text{follow}(A) = \{b, a\}$

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

	Action			Goto	
	a	b	\$	S	A
0	S <sub>4</sub>	S <sub>3</sub>		1	2
→ 1	S <sub>4</sub>	S <sub>3</sub>	Accepted	6	5
2	S <sub>4</sub>	S <sub>3</sub>		7	2
3	<del>S<sub>4</sub></del>	<del>S<sub>3</sub></del>	r <sub>2</sub>		
4	<del>S<sub>4</sub></del>	<del>S<sub>3</sub></del>			
5	S <sub>4</sub> /r <sub>3</sub>	S <sub>3</sub> /r <sub>3</sub>		7	2
6	S <sub>4</sub>	S <sub>3</sub>		6	5
7	S <sub>4</sub> /r <sub>1</sub>	S <sub>3</sub> /r <sub>1</sub>	r <sub>1</sub>	6	5

not SLR(1)

Step-4:

Stack  
\$ 0  
\$ 0 a y  
\$

Input buffer  
a b a b \$  
b a b \$

Action  
S<sub>4</sub> (shift 4)  
reduce A → a



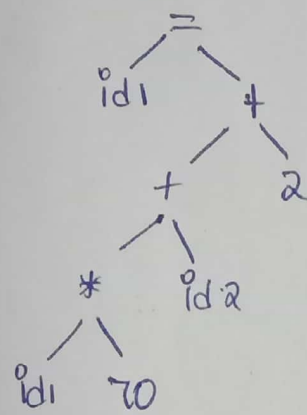
# Phases of Compiler

$$i = i * 70 + j + 2$$

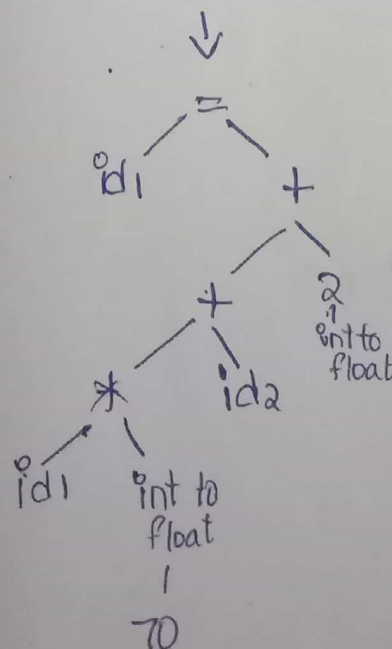
↓  
Lexical Analyser

$$id_1 = id_1 * 70 + id_2 + 2$$

↓  
Syntax Analyser



↓  
Semantic Analyser



Intermediate Code Generator

↓  
 $t_1 = \text{int to float}(70)$   
 $t_2 = id_1 * t_1$   
 $t_3 = t_2 + id_2$   
 $t_4 = \text{int to float}(2)$   
 $t_5 = t_3 + t_4$   
 $id_1 = t_5$

↓  
Code optimisation

↓  
 $t_1 = id_1 * 70.0$   
 $t_2 = t_1 + id_2$   
 $t_3 = t_2 + 2.0$   
 $id_1 = t_3$

↓  
Code Generator

↓  
 LDF R1, id1  
 MULF R1, R1, #70.0  
 LDF R2, id2  
 ADDF R1, R1, R2  
 ADDF R1, R1, #2.0  
 STF id1, R1

Symbol Table

1	id	...
2	id	...
...	...	...

# CLR Parser

## Canonical LR Parser

$$LR(1) = LR(0) + \text{lookahead}$$

$$S \rightarrow CC$$

$$C \rightarrow dd$$

$$S \rightarrow CC$$

$$C \rightarrow cC$$

$$C \rightarrow d$$

Step-1: Augmented grammar

$$S' \rightarrow S$$

$$S \rightarrow CC$$

$$C \rightarrow cC$$

$$C \rightarrow d$$

calculation of  
lookahead part

$$S' \rightarrow \cdot S, \$$$

$$S \rightarrow \cdot S (\epsilon, \$)$$

$$\text{first}(\epsilon, \$)$$

$$\text{first}(\$)$$

$$c, d$$

Step-2: Calculating LR(1) items

$$I_0 \quad S' \rightarrow \cdot S, \$$$

$$S \rightarrow \cdot CC, \$$$

$$C \rightarrow \cdot cC, c/d$$

$$C \rightarrow \cdot d, c/d$$

3

goto ( $I_0, S$ ):

$I_1: S' \rightarrow S., \$$

goto ( $I_0, C$ ):

$I_2: S \rightarrow C.C., \$$

$C \rightarrow .cC, cld \$$

$C \rightarrow .d, cld \$$

goto ( $I_0, c$ ):

$I_3: C \rightarrow c.C, cld$

$C \rightarrow .cC, cld$

$C \rightarrow .d, cld$

goto ( $I_0, d$ ):

$I_4: C \rightarrow d., cld$

goto ( $I_2, C$ )

$I_5: S \rightarrow CC., \$$

goto ( $I_2, c$ )

$I_6: C \rightarrow C.C., \$$

$C \rightarrow .cC, \$$

$C \rightarrow .d, \$$

goto ( $I_2, d$ )

$I_7: C \rightarrow d., \$$

goto ( $I_3, C$ )

$I_8: C \rightarrow cC., cld$

goto ( $I_3, c$ )

$C \rightarrow c.C, cld$

$C \rightarrow .cC, cld$

$C \rightarrow .d, cld$

goto ( $I_3, d$ )

$C \rightarrow .d., cld$

goto ( $I_6, C$ )

$I_9: C \rightarrow cC., \$$

goto ( $I_6, c$ )

$C \rightarrow c.C, \$$

$C \rightarrow .cC, \$$

$C \rightarrow .d, \$$

goto ( $I_6, d$ )

$C \rightarrow d., \$$



# Construction of CLR Parse table

c	Action		goto
	c	d	
0	S <sub>3</sub>	S <sub>4</sub>	S
1		Accepted	
2	S <sub>6</sub>	S <sub>7</sub>	
3	S <sub>3</sub>	S <sub>4</sub>	
4	r <sub>3</sub>	r <sub>3</sub>	
5		r <sub>1</sub>	
6	S <sub>6</sub>	S <sub>7</sub>	
7		r <sub>3</sub>	
8	r <sub>2</sub>	r <sub>2</sub>	
9		r <sub>2</sub>	

Shift  
Reduce  
Accept

S → S. 0  
S → CC. 1  
C → cC. 2  
C → d. 3

Stack	Input String	Action
\$0	<u>d</u> d\$	Shift 4
\$0d4	<u>d</u> \$	Reduce C → d
\$0C2	<u>d</u> \$	Shift 7
\$0C2d7	<u>\$</u>	Reduce C → d
\$0C2C5	<u>\$</u>	Reduce S → CC
\$0S1	<u>\$</u>	Accepted

LALR

↓  
~~LR(1)~~

.. Augmented Grammar

$$S' \rightarrow S$$

$$S \rightarrow L = R$$

$$S \rightarrow R$$

$$L \rightarrow * R$$

$$L \rightarrow id$$

$$R \rightarrow L$$

$$S \rightarrow L = R / R$$

$$L \rightarrow * R / id$$

$$R \rightarrow L$$

goto ( $I_0, *$ ):

$$I_4 \cdot L \rightarrow * \cdot R, = / \$$$

$$R \rightarrow \cdot L, = / \$$$

$$L \rightarrow \cdot * R, = / \$$$

$$L \rightarrow \cdot id, = / \$$$

LR(1) items

$$I_0 \ S' \rightarrow \cdot S, \$$$

$$S \rightarrow \cdot L = R, \$$$

$$S \rightarrow \cdot R, \$$$

$$L \rightarrow \cdot * R, = / \$$$

$$L \rightarrow \cdot id, = / \$$$

$$R \rightarrow \cdot L, \$$$

goto ( $I_0, S$ ):

$$I_1 \ S' \rightarrow S \cdot, \$$$

goto ( $I_0, L$ ):

$$I_2 \ S \rightarrow L \cdot = R, \$$$

$$R \rightarrow L \cdot, \$$$

goto ( $I_0, R$ )

$$I_3 \ S \rightarrow R \cdot, \$$$

goto ( $I_0, id$ ):

$$I_5 \ L \rightarrow id \cdot, = / \$$$

goto ( $I_2, =$ )

$$I_6 \ S \rightarrow L = \cdot R, \$$$

$$R \rightarrow \cdot L, \$$$

$$L \rightarrow \cdot * R, \$$$

$$L \rightarrow \cdot id, \$$$

goto ( $I_4, R$ ):

$$I_7 \ L \rightarrow * R \cdot, = / \$$$

goto ( $I_4, L$ ):

$$I_8 \ R \rightarrow L \cdot, = / \$$$

goto ( $I_4, *$ )

$$L \rightarrow * \cdot R, = / \$ \quad I_4$$

$$R \rightarrow \cdot L, = / \$$$

$$L \rightarrow \cdot * R$$

goto ( $I_4, id$ )

$L \rightarrow id \cdot, \$ \} I_5$

goto ( $I_6, R$ )

$I_9 \rightarrow L = R \cdot, \$$

goto ( $I_6, L$ )

$I_{10} R \rightarrow L \cdot, \$$

goto ( $I_6, *$ )

$I_{11} L \rightarrow * \cdot R, \$$

$R \rightarrow \cdot L, \$$

$L \rightarrow \cdot * R, \$$

$L \rightarrow \cdot id, \$$

goto ( $I_6, id$ ):

$I_{12} L \rightarrow id \cdot, \$$

goto ( $I_{11}, R$ ):

$I_{13} L \rightarrow * \cdot R \cdot, \$$

goto ( $I_{11}, L$ ):

~~$I_{10}$~~   $R \rightarrow L \cdot, \$ \} I_{10}$

goto ( $I_{11}, *$ )

$L \rightarrow * \cdot R, \$ ?$

$R \rightarrow \cdot L \} I_{11}$

goto ( $I_{11}, id$ )

$L \rightarrow id \cdot, \$ \} I_{12}$

$I_4 \& I_{11} \rightarrow I_{11}$

$I_5 \& I_{12} \rightarrow I_{12}$

$I_7 \& I_{13} \rightarrow I_{13}$

$I_8 \& I_{10} \rightarrow I_{10}$

# Construction of LALR Parse table

Action*			Go to		
	id	= * \$	S	L	R
0	S512	S411	1	2	3
1		Accepted			
2	S6	r5			
3		r2			
411	S512	S411	810	713	
512	r4	r4			
6	S512	S411	810	9	
713	r3	r3			
810	r5	r5			
9		r1			

id = id

Stack	Alp string	Action
\$ 0	id = id \$	Shift 512
\$ 0 <u>id</u> 512	= id \$	Reduce 4 L → R
\$ 0 <u>L</u> 2	= id \$	Shift 6
\$ 0 <u>L</u> 2 = 6	id \$	Shift 512
\$ 0 <u>L</u> 2 = 6 <u>id</u> 512	\$	Reduce 4 L → R
\$ 0 <u>L</u> 2 = 6 <u>L</u> 810	\$	Reduce 5 R → L
\$ 0 <u>L</u> 2 = 6 <u>R</u> 9	\$	Reduce 1 S → L = R