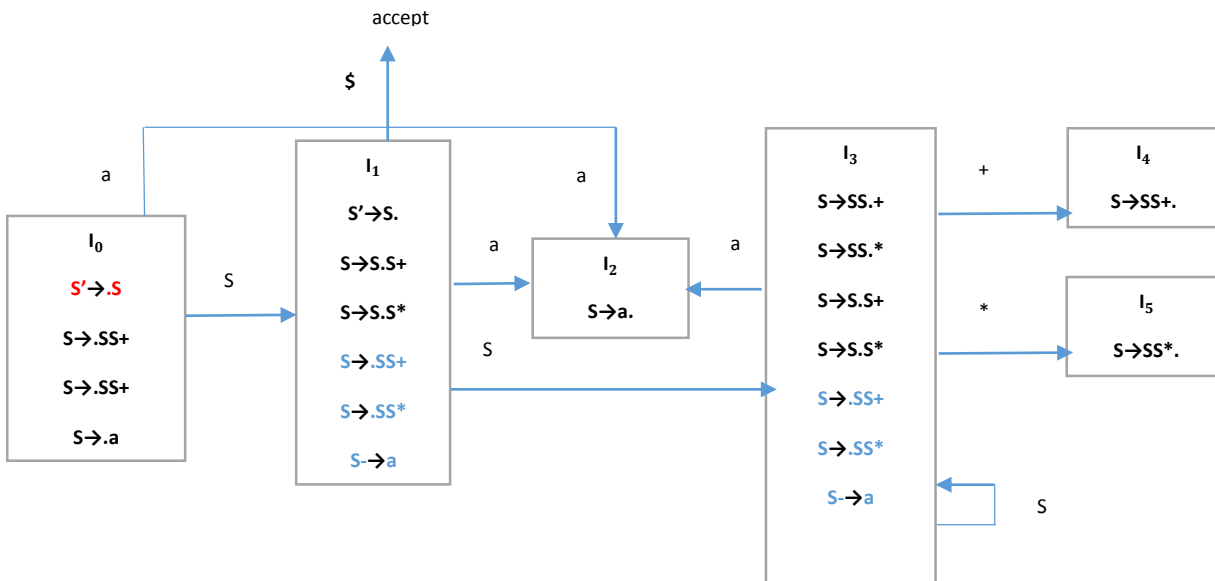


**CSC 340: Programming Language and Compilation**  
**Homework #3: LR Parsing**

---

**1- Exercise 4.6.2 [1]**

**Construct the SLR sets of items for the (augmented) grammar  $S \rightarrow SS+|SS^*|a$**   
**Show the parsing table for this grammar. Is the grammar SLR?**



Add a new symbol  $S'$  and a new production  $S' \rightarrow S$

**Numbering the productions:**

$$S \rightarrow SS+ \quad (1)$$

$$S \rightarrow SS^* \quad (2)$$

$$S \rightarrow a \quad (3)$$

**Finding follow of the grammar's symbols (you may need to find First to find the Follow):**

$$\text{FIRST}(S) = \{a\}$$

$$\text{FOLLOW}(S) = \{\$, +, *, a\}$$

**CSC 340: Programming Language and Compilation**  
**Homework #3: LR Parsing**

STATE	ACTION				GOTO
	+	*	a	\$	
0			S2		1
1			S2	acc	3
2	R3	R3	R3	R3	
3	S4	S5	S2		3
4	R1	R1	R1	R1	
5	R2	R2	R2	R2	

SLR Parsing Table T

\* in SLR place Reduce action in the follow of the lhs of S->a

\*in SLR, place Reduce action in the follow of the lhs of S->SS+

\* in SLR place Reduce action in the follow of the lhs of S->SS\*

No conflicts in SLR parsing table, therefore, the grammar is SLR.

**2- Exercise 4.6.3 [1]**

**Show the actions of your parsing table from Exercise 4.6.2 on the input aa\*a+**

(Stack column explanation)

T[0,a] =s2

T[2,a] =r3, apply reduction then T[0,S] =1

T[1,a] =s2

T[2,\*] =r3, apply reduction then T[1,S] =3

T[3,\*] =s5

T[5,a] =r2, apply reduction then T[0,S] =1

T[1,a] =s2

T[2,+] =r3 , apply reduction then T[1,S] =3

T[3,+] =s4

T[4,\$] =r1, apply reduction then T[0,S] =1

T[1,\$] =accept

Line	Stack	Symbols	input	Action
1	0		aa*a+\$	Shift to 2
2	02	a	a*a+\$	Reduce by S->a
3	01	S	a*a+\$	Shift to 2
4	012	S a	*a+\$	Reduce by S->a
5	013	SS	*a+\$	Shift to 5
6	0135	SS*	a+\$	Reduce by S->SS*
7	01	S	a+\$	Shift to 2
8	012	Sa	+\$	Reduce by S->a
9	013	SS	+\$	Shift to 4
10	0134	SS+	\$	Reduce S->SS+
11	01	S	\$	accept

Parsing Table on the input aa\*a+

**CSC 340: Programming Language and Compilation**  
**Homework #3: LR Parsing**

---

**3- Example 4.45 [1]**

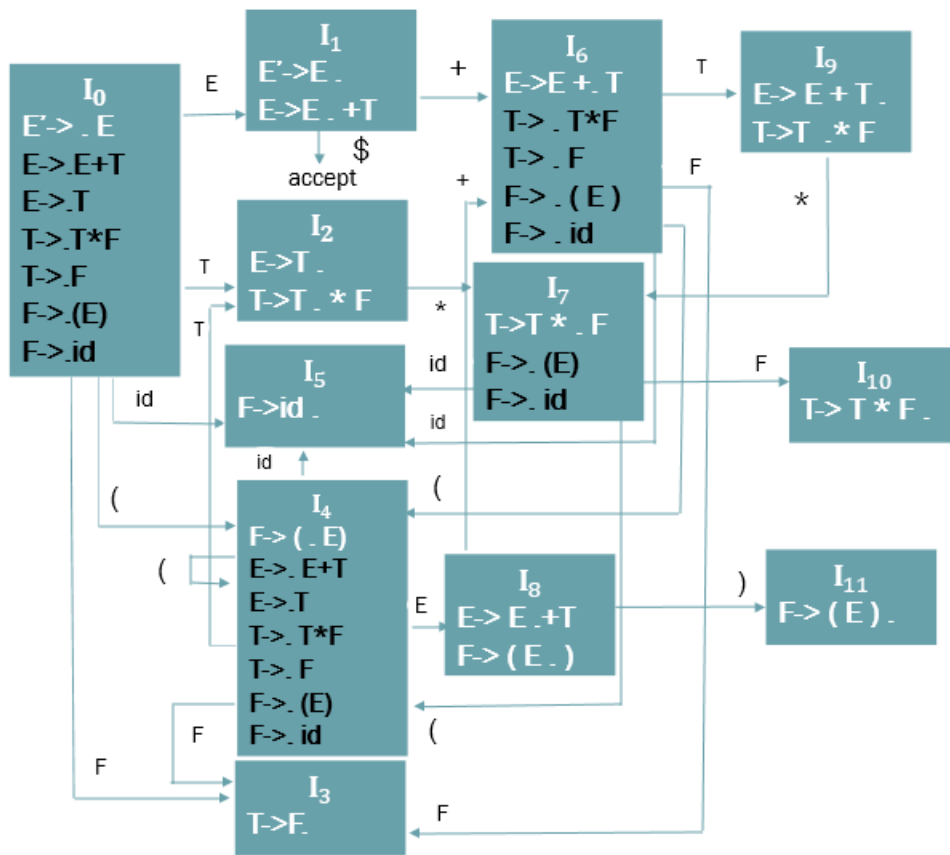
Consider the following expression grammar:

$$\begin{aligned} E &\rightarrow E+T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow (E) \mid id \end{aligned}$$

- i. Construct SLR(1) parsing table for the grammar.
- ii. Illustrate the action of the shift-reduce parser on the input  $id * id$  using SLR(1) parsing table.

**Solution [1]:**

A- LR(0) :



**CSC 340: Programming Language and Compilation**  
**Homework #3: LR Parsing**

---

B- Find Follow sets:

$$\text{First}(E) = \{ '(', \text{id} \}$$

$$\text{Follow}(E) = \{ \$, +, ' )' \}$$

$$\text{First}(T) = \{ '(', \text{id} \}$$

$$\text{Follow}(T) = \{ \$, +, ' )', * \}$$

$$\text{First}(F) = \{ '(', \text{id} \}$$

$$\text{Follow}(F) = \{ \$, +, ' )', * \}$$

C- Numbering the productions:

- (1)  $E \rightarrow E+T$
- (2)  $E \rightarrow T$
- (3)  $T \rightarrow T*F$
- (4)  $T \rightarrow F$
- (5)  $F \rightarrow (E)$
- (6)  $F \rightarrow \text{id}$

i. SLR parsing table:

**Reduce Move placement Explanation**

\* in SLR place Reduce action in the follow of the lhs of  $E \rightarrow T$

\*in SLR, place Reduce action in the follow of the lhs of  $T \rightarrow F$

\* in SLR place Reduce action in the follow of the lhs of  $F \rightarrow \text{id}$

\*in SLR, place Reduce action in the follow of the lhs of  $E \rightarrow E+T$

\* in SLR place Reduce action in the follow of the lhs of  $T \rightarrow T*F$

\* in SLR place Reduce action in the follow of the lhs of  $F \rightarrow (E)$

STATE	ACTION						GOTO		
	id	+	*	(	)	\$	E	T	F
0	S5			S4			1	2	3
1		S6				acc			
2		R2	S7		R2	R2			
3		R4	R4		R4	R4			
4	S5			S4			8	2	3
5		R6	R6		R6	R6			
6	S5			S4				9	3
7	S5			S4					10
8		S6			S11				
9		R1	S7		R1	R1			
10		R3	R3		R3	R3			
11		R5	R5		R5	R5			

SLR Parsing Table T

**CSC 340: Programming Language and Compilation**  
**Homework #3: LR Parsing**

---

ii. Parsing the input **id\*id**  
 - Use a stack to hold states

Line	Stack	Symbol	Input	Action
(1)	0	\$	id*id\$	Shift to 5
(2)	0 5	\$id	*id\$	Reduce by $F \rightarrow id$
(3)	0 3	\$F	*id\$	Reduce by $T \rightarrow F$
(4)	0 2	\$T	*id\$	Shift to 7
(5)	0 2 7	\$T*	id\$	Shift to 5
(6)	0 2 7 5	\$T*id	\$	reduce by $F \rightarrow id$
(7)	0 2 7 10	\$T*F	\$	Reduce $T \rightarrow T*F$
(8)	0 2	\$T	\$	Reduce by $E \rightarrow T$
(9)	0 1	\$E	\$	accept
The parse of id*id				

**CSC 340: Programming Language and Compilation**  
**Homework #3: LR Parsing**

---

**4- Example 4.54 [1]**

**Construct the parsing table using LR(1) for the following grammar:**

$S \rightarrow CC$   
 $C \rightarrow cC \mid d$

**Solution [1][2]:**

$\text{First}(S) = \{c, d\}$

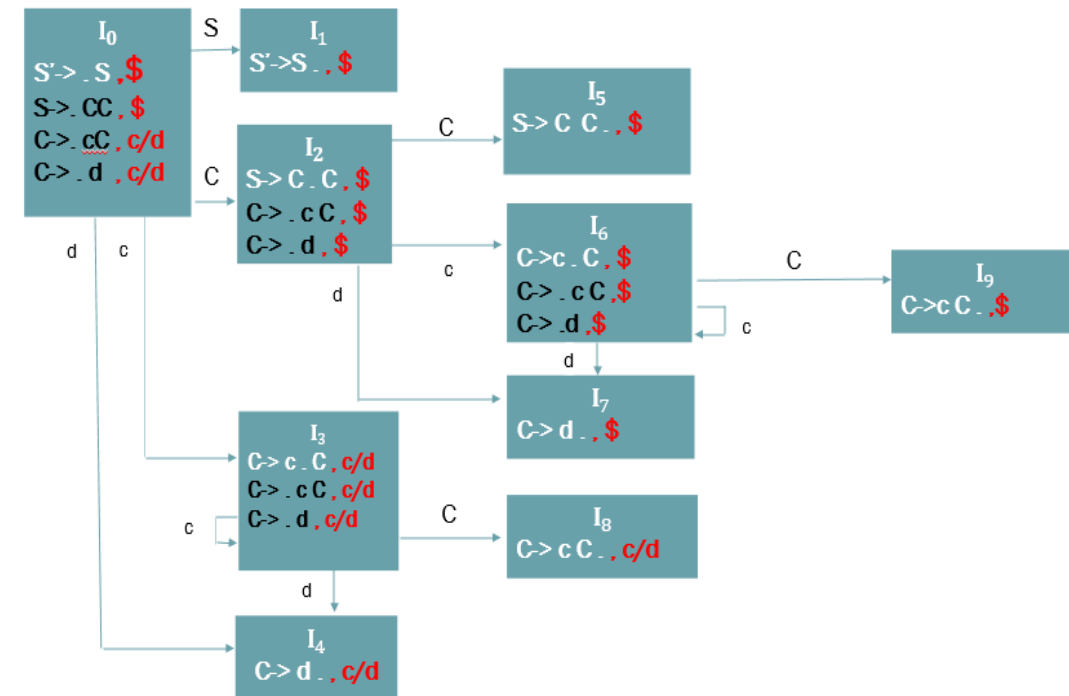
$\text{Follow}(S) = \{\$ \}$

$\text{First}(C) = \{c, d\}$

$\text{Follow}(C) = \{\$ \}$

- For Augmented Production (initial production) put \$ as a look Ahead.
- In case we want to do the closure, to find the lookAhead: we find the first of whatever remaining.
- When we change the DOT, when we transfer, we do Not change the LookAhead.
- While applying the closure, LookAhead might change.
- First of any terminal = the terminal itself
  - $\text{First}(\$) = \{\$ \}$
  - $\text{First}(C\$) = \text{First}(C) = \{c, d\}$
  - We use the notation  $[C \rightarrow \cdot cC, c/d]$  as a shorthand for the two items  $[C \rightarrow \cdot cC, c]$  and  $[C \rightarrow \cdot cC, d]$
  - $\text{First}(c) = \{c\}$
  - $\text{First}(d) = \{d\}$
- In the table, the only difference in LR(1) is where to put the reduce move. Place the reduce move only in the LookAhead symbols.

**CSC 340: Programming Language and Compilation**  
**Homework #3: LR Parsing**



The GOTO Graph for the given grammar

Numbering the grammar:

- (1)  $S \rightarrow CC$
- (2)  $C \rightarrow cC$
- (3)  $C \rightarrow d$

**Reduce Move placement Explanation**

\* in LR(1), place Reduce action in the LookAhead symbols : 'c' and 'd'

\*in LR(1), place Reduce action in the LookAhead symbols: \$

\*in LR(1), place Reduce action in the LookAhead symbols: \$

\* in LR(1), place Reduce action in the LookAhead symbols : 'c' and 'd'

\*in LR(1), place Reduce action in the LookAhead symbols: \$

STATE	ACTION			GOTO	
	c	d	\$	S	C
0	S3	S4		1	2
1			acc		
2	S6	S7			5
3	S3	S4			8
4	R3	R3			
5			R1		
6	S6	S7			9
7			R3		
8	R2	R2			
9			R2		

LR(1) Parsing Table

**CSC 340: Programming Language and Compilation**  
**Homework #3: LR Parsing**

---

**References:**

[1] “Compilers Principles, Techniques, & Tools” Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman

[2] [Compiler Design Lecture 14 -- CLR\(1\) and LALR\(1\) Parsers:](#)