



**Università degli Studi di Udine**

---

DIPARTIMENTO DI SCIENZE MATEMATICHE, INFORMATICHE E FISICHE

Corso di Laurea Magistrale in Informatica

RELAZIONE DEL PROGETTO  
PER L'INSEGNAMENTO  
LINGUAGGI E COMPILATORI

**ProgettoLC parte1**  
**Filippo Callegari**

Studente:

**Filippo Callegari**

`callegari.filippo@spes.uniud.it`

Matricola 128602

Professore dell'insegnamento:

**Marco Comini**

## Esercizio 1

### Parte 1: grammatica LL(1)

Viene fornita la seguente grammatica equivalente:

$$\begin{aligned}
 E &\rightarrow T A \\
 A &\rightarrow \epsilon \mid \text{and } T A \\
 T &\rightarrow P O \\
 O &\rightarrow \epsilon \mid \text{or } P O \\
 P &\rightarrow \text{next } P \mid \text{always } P \mid F \\
 F &\rightarrow \text{true} \mid \text{atom} \mid ( E )
 \end{aligned}$$

Tale grammatica risulta **LL(1)**, poichè, secondo definizione, data  $A \rightarrow \alpha \mid \beta$ , per ogni coppia:

- $\text{FIRST}(\alpha) \cap \text{FIRST}(\beta) = \emptyset$
- se  $\alpha \xRightarrow{*} \epsilon$  allora  $\text{FOLLOW}(A) \cap \text{FIRST}(\beta) = \emptyset$

### Parte 2: parser top-down

Si calcolano *FIRST* per ogni simbolo non terminale:

- $\text{FIRST}(E) = \{ \text{next}, \text{always}, \text{true}, \text{atom}, ( \};$
- $\text{FIRST}(A) = \{ \epsilon, \text{and} \};$
- $\text{FIRST}(T) = \{ \text{next}, \text{always}, \text{true}, \text{atom}, ( \};$
- $\text{FIRST}(O) = \{ \epsilon, \text{or} \};$
- $\text{FIRST}(P) = \{ \text{next}, \text{always}, \text{true}, \text{atom}, ( \};$
- $\text{FIRST}(F) = \{ \text{true}, \text{atom}, ( \}.$

Si calcolano *FOLLOW* per ogni simbolo non terminale:

- $\text{FOLLOW}(E) = \{ \$, ) \};$
- $\text{FOLLOW}(A) = \{ \$, ) \};$
- $\text{FOLLOW}(T) = \{ \text{and}, \$, ) \};$
- $\text{FOLLOW}(O) = \{ \text{and}, \$, ) \};$
- $\text{FOLLOW}(P) = \{ \text{and}, \text{or}, \$, ) \};$
- $\text{FOLLOW}(F) = \{ \text{and}, \text{or}, \$, ) \}.$

La parsing table generata è quindi:

	always	and	atom	next	or	true	(	)	\$
A	e2	$A \rightarrow \text{and } TA$	e8	e1	e0	e8	e0	$A \rightarrow \epsilon$	$A \rightarrow \epsilon$
E	$E \rightarrow TA$	e3	$E \rightarrow TA$	$E \rightarrow TA$	e3	$E \rightarrow TA$	$E \rightarrow TA$	e6	e4
F	e2	e0	$F \rightarrow \text{atom}$	e1	e0	$F \rightarrow \text{true}$	$F \rightarrow (E)$	e0	e0
O	e2	$O \rightarrow \epsilon$	e7	e1	$O \rightarrow \text{or } PO$	e7	e5	$O \rightarrow \epsilon$	$O \rightarrow \epsilon$
P	$P \rightarrow \text{always } P$	e3	$P \rightarrow F$	$P \rightarrow \text{next } P$	e3	$P \rightarrow F$	$P \rightarrow F$	e6	e3
T	$T \rightarrow PO$	e0	$T \rightarrow PO$	$T \rightarrow PO$	e3	$T \rightarrow PO$	$T \rightarrow PO$	e9	e3

### Meccanismi di “Error Recovery”

Si cerca di definire un error recovery per ogni entry non appartenente al core del parser LL(1). Si scrive in notazione `SideEffect[Stampa]`, al fine di mostrare che cosa compare come messaggio e che azione viene intrapresa.

Vengono definiti i seguenti errori nella parsing table:

0. ErroreGenerale, svuoto stack, termino[Errore generico di parsing.];
1. Rimuovo "next" da input[Token "next" inaspettato.];
2. Rimuovo "always" da input[Token "always" inaspettato.];
3. Inserisco "true" in input[Token "true" o "atom" mancante.];
4. Svuoto Stack, termino[Possibile stringa senza predicati presente!];
5. Cambio '(' con ')' [Token ')' in posizione errata!];
6. Cambio ')' con '(' [Token '(' in posizione errata!];
7. Inserisco 'or' in input[Token 'or' mancante, viene aggiunto per contesto];
8. Inserisco 'and' in input[Token 'and' mancante, viene aggiunto per contesto];
9. Rimuovo un carattere in input[carattere in posizione errata].

### Parte 3: esecuzione del parser

Viene somministrata la stringa

*next atom next atom and or and next next atom*

e se ne emula l'esecuzione con i passi di “Error Recovery” descritti.

Stack	Input	Regola
$E$	<i>next atom next atom and or and next next atom</i> \$	$E \rightarrow TA$
$TA$	<i>next atom next atom and or and next next atom</i> \$	$T \rightarrow PO$
$POA$	<i>next atom next atom and or and next next atom</i> \$	$P \rightarrow next P$
$nextPOA$	<i>next atom next atom and or and next next atom</i> \$	Match <i>next</i>
$POA$	<i>atom next atom and or and next next atom</i> \$	$P \rightarrow F$
$FOA$	<i>atom next atom and or and next next atom</i> \$	$F \rightarrow atom$
$atomOA$	<i>atom next atom and or and next next atom</i> \$	Match <i>atom</i>
$OA$	<i>next atom and or and next next atom</i> \$	$e1$
$OA$	<i>atom and or and next next atom</i> \$	$e7$
$OA$	<i>or atom and or and next next atom</i> \$	$O \rightarrow or PO$
$orPOA$	<i>or atom and or and next next atom</i> \$	Match <i>or</i>
$POA$	<i>atom and or and next next atom</i> \$	$P \rightarrow F$
$FOA$	<i>atom and or and next next atom</i> \$	$F \rightarrow atom$
$atomOA$	<i>atom and or and next next atom</i> \$	Match <i>atom</i>
$OA$	<i>and or and next next atom</i> \$	$O \rightarrow \epsilon$
$A$	<i>and or and next next atom</i> \$	$A \rightarrow and TA$
$andTA$	<i>and or and next next atom</i> \$	Match <i>and</i>
$TA$	<i>or and next next atom</i> \$	$e3$
$TA$	<i>true or and next next atom</i> \$	$T \rightarrow PO$
$POA$	<i>true or and next next atom</i> \$	$P \rightarrow F$
$FOA$	<i>true or and next next atom</i> \$	$F \rightarrow true$
$trueOA$	<i>true or and next next atom</i> \$	Match <i>true</i>
$OA$	<i>or and next next atom</i> \$	$O \rightarrow or PO$
$orPOA$	<i>or and next next atom</i> \$	Match <i>or</i>
$POA$	<i>and next next atom</i> \$	$e3$
$POA$	<i>true and next next atom</i> \$	$P \rightarrow F$
$FOA$	<i>true and next next atom</i> \$	$F \rightarrow true$
$trueOA$	<i>true and next next atom</i> \$	Match <i>true</i>
$OA$	<i>and next next atom</i> \$	$O \rightarrow \epsilon$
$A$	<i>and next next atom</i> \$	$A \rightarrow and TA$
$andTA$	<i>and next next atom</i> \$	Match <i>and</i>
$TA$	<i>next next atom</i> \$	$T \rightarrow PO$
$POA$	<i>next next atom</i> \$	$P \rightarrow next P$
$nextPOA$	<i>next next atom</i> \$	Match <i>next</i>
$POA$	<i>next atom</i> \$	$P \rightarrow next P$
$nextPOA$	<i>next atom</i> \$	Match <i>next</i>
$POA$	<i>atom</i> \$	$P \rightarrow F$
$FOA$	<i>atom</i> \$	$F \rightarrow atom$
$atomOA$	<i>atom</i> \$	Match <i>atom</i>
$OA$	\$	$O \rightarrow \epsilon$
$A$	\$	$A \rightarrow \epsilon$
	\$	Accettato

## Esercizio 2

Viene fornita di seguito la grammatica aumentata comune ai parser LALR e SLR. La regola si ritrova di fianco la sua enumerazione.

$$\begin{aligned}
 S &\rightarrow E \quad (0) \\
 E &\rightarrow L = R \quad (1) \mid R \quad (2) \\
 L &\rightarrow *R \quad (3) \mid id \quad (4) \mid L[E] \quad (5) \\
 R &\rightarrow num \quad (6) \mid L \quad (7) \mid R + R \quad (8) \mid Lpp \quad (9) \mid ppL \quad (10)
 \end{aligned}$$

Viene calcolato il *FIRST* per la grammatica:

- $FIRST(S) = \{*, id, num, pp\};$
- $FIRST(E) = \{*, id, num, pp\};$
- $FIRST(L) = \{*, id\};$
- $FIRST(R) = \{*, id, num, pp\}.$

Viene calcolato il *FOLLOW* per la grammatica:

- $FOLLOW(S) = \{\$ \};$
- $FOLLOW(E) = \{[, \$ \};$
- $FOLLOW(L) = \{=, [, \$, pp, +, ] \};$
- $FOLLOW(R) = \{=, [, \$, pp, +, ] \}.$

## Parser SLR

Di seguito viene presentata la collezione canonica per la grammatica:

- $I_0 = Closure(S) = \{S \rightarrow \cdot E, E \rightarrow \cdot L = R, E \rightarrow \cdot R, L \rightarrow \cdot * R, L \rightarrow \cdot id, L \rightarrow \cdot L[E], R \rightarrow \cdot num, R \rightarrow \cdot L, R \rightarrow \cdot R + R, R \rightarrow \cdot Lpp, R \rightarrow \cdot ppL\};$
- $I_1 = GOTO(I_0, E) = \{S \rightarrow E \cdot\};$
- $I_2 = GOTO(I_0, L) = \{E \rightarrow L \cdot = R, L \rightarrow L \cdot [E], R \rightarrow L \cdot, R \rightarrow L \cdot pp\};$
- $I_3 = GOTO(I_0, R) = \{E \rightarrow R \cdot, R \rightarrow R \cdot + R\};$
- $I_4 = GOTO(I_0, *) = \{L \rightarrow * \cdot R, R \rightarrow \cdot num, R \rightarrow \cdot L, R \rightarrow \cdot R + R, R \rightarrow \cdot Lpp, R \rightarrow \cdot ppL, L \rightarrow * \cdot R, L \rightarrow \cdot id, L \rightarrow \cdot L[E]\};$
- $I_5 = GOTO(I_0, id) = \{L \rightarrow id \cdot\};$
- $I_6 = GOTO(I_0, num) = \{R \rightarrow num \cdot\};$
- $I_7 = GOTO(I_0, pp) = \{R \rightarrow pp \cdot L, L \rightarrow \cdot * R, L \rightarrow \cdot id, L \rightarrow \cdot L[E]\};$
- $I_8 = GOTO(I_2, =) = \{E \rightarrow L = \cdot R, R \rightarrow \cdot num, R \rightarrow \cdot L, R \rightarrow \cdot R + R, R \rightarrow \cdot Lpp, R \rightarrow \cdot ppL, L \rightarrow * \cdot R, L \rightarrow \cdot id, L \rightarrow \cdot L[E]\};$
- $I_9 = GOTO(I_2, [) = \{L \rightarrow L [ \cdot E, E \rightarrow \cdot L = R, E \rightarrow \cdot R, L \rightarrow \cdot * R, L \rightarrow \cdot id, L \rightarrow \cdot L[E], R \rightarrow \cdot num, R \rightarrow \cdot L, R \rightarrow \cdot R + R, R \rightarrow \cdot Lpp, R \rightarrow \cdot ppL\};$

- $I_{10} = GOTO(I_2, pp) = \{R \rightarrow Lpp\};$
- $I_{11} = GOTO(I_3, +) = \{R \rightarrow R + \cdot R, R \rightarrow \cdot num, R \rightarrow \cdot L, R \rightarrow \cdot R + R, R \rightarrow \cdot Lpp, R \rightarrow \cdot ppL, L \rightarrow \cdot * R, L \rightarrow \cdot id, L \rightarrow \cdot L[E]\};$
- $I_{12} = GOTO(I_4, R) = \{L \rightarrow *R\cdot, R \rightarrow R\cdot + R\};$
- $I_{13} = GOTO(I_4, L) = \{R \rightarrow L\cdot, R \rightarrow L\cdot pp, L \rightarrow L\cdot [E]\};$
- $I_{14} = GOTO(I_7, L) = \{R \rightarrow ppL\cdot, L \rightarrow L\cdot [E]\};$
- $I_{15} = GOTO(I_8, R) = \{E \rightarrow L = R\cdot, R \rightarrow R\cdot + R\};$
- $I_{16} = GOTO(I_9, E) = \{L \rightarrow L[E\cdot]\};$
- $I_{17} = GOTO(I_{11}, R) = \{R \rightarrow R + R\cdot, R \rightarrow R\cdot + R\};$
- $I_{18} = GOTO(I_{16}, \cdot) = \{L \rightarrow L[E]\cdot\}.$

Si presenta la rimanenza per la creazione delle *ACTION* e *GOTO*:

- $GOTO(I_4, num) \equiv I_6;$
- $GOTO(I_4, pp) \equiv I_7;$
- $GOTO(I_4, *) \equiv I_4;$
- $GOTO(I_4, id) \equiv I_5;$
- $GOTO(I_7, *) \equiv I_4;$
- $GOTO(I_7, id) \equiv I_5;$
- $GOTO(I_8, num) \equiv I_6;$
- $GOTO(I_8, L) \equiv I_{13};$
- $GOTO(I_8, pp) \equiv I_7;$
- $GOTO(I_8, *) \equiv I_4;$
- $GOTO(I_8, id) \equiv I_5;$
- $GOTO(I_9, L) \equiv I_2;$
- $GOTO(I_9, R) \equiv I_3;$
- $GOTO(I_9, *) \equiv I_4;$
- $GOTO(I_9, id) \equiv I_5;$
- $GOTO(I_9, num) \equiv I_6;$
- $GOTO(I_9, pp) \equiv I_7;$
- $GOTO(I_{11}, num) \equiv I_6;$
- $GOTO(I_{11}, L) \equiv I_{13};$
- $GOTO(I_{11}, pp) \equiv I_7;$
- $GOTO(I_{11}, *) \equiv I_4;$
- $GOTO(I_{11}, id) \equiv I_5;$
- $GOTO(I_{12}, +) \equiv I_{11};$

- $GOTO(I_{13}, pp) \equiv I_{10}$ ;
- $GOTO(I_{13}, \mid) \equiv I_9$ ;
- $GOTO(I_{14}, \mid) \equiv I_9$ ;
- $GOTO(I_{15}, +) \equiv I_{11}$ ;
- $GOTO(I_{17}, +) \equiv I_{11}$ ;

### Tabella di parsing SLR

Di seguito la tabella di parsing SLR.

	id	num	pp	*	+	=	[	]	\$	E	L	R	S
0	s5	s6	s7	s4	e5	e6	e7	e8	e11	1	2	3	
1	e0	e0	e0	e0	e0	e0	e0	e0	acc				
2	e14	e15	s10 / r7	e4	r7	s8 / r7	s9 / r7	r7	r7				
3	e14	e15	e17	e4	s11	e6	e7	r2	r2				
4	s5	s6	s7	s4	e5	e6	e7	e8	e17		13	12	
5	e14	e15	r4	e13	r4	r4	r4	r4	r4				
6	e14	e15	r6	e13	r6	r6	r6	r6	r6				
7	s5	e15	e3	s4	e17	e17	e17	e8	e17		14		
8	s5	s6	s7	s4	e17	e6	e7	e8	e18		13	15	
9	s5	s6	s7	s4	e5	e17	e7	e17	e18	16	2	3	
10	e14	e15	r9	e4	r9	r9	r9	r9	r9				
11	s5	s6	s7	s4	e5	e6	e12	e12	e17		13	17	
12	e14	e15	r3	e4	s11 / r3	r3	r3	r3	r3				
13	e14	e15	s10 / r7	e16	r7	r7	s9 / r7	r7	r7				
14	e14	e15	r10	e4	r10	r10	s9 / r10	r10	r10				
15	e14	e15	e3	e4	s11	e6	e7	r1	r1				
16	e14	e15	e3	e4	e5	e6	e7	s18					
17	e14	e15	r8	e4	s11 / r8	r8	r8	r8	r8				
18	e14	e15	r5	e4	r5	r5	r5	r5	r5				

Gli errori mantengono la notazione espressa nell'esercizio precedente, ovvero `SideEffect[StampaErrore]`. Gli errori definiti quindi sono:

0. `ErroreGenerale`, svuoto stack, `termino[Errore generico di parsing.]`;
1. Rimuovo token *id* dall'input[Token *id* in posizione inaspettata];
2. Rimuovo token *num* dall'input[Token *num* in posizione inaspettata];
3. Rimuovo token *pp* dall'input[Token *pp* in posizione inaspettata];
4. Rimuovo token *\** dall'input[Token *\** in posizione inaspettata];
5. Rimuovo token *+* dall'input[Token *+* in posizione inaspettata];
6. Rimuovo token *=* dall'input[Token *=* in posizione inaspettata];
7. Rimuovo token *[* dall'input[Token *[* in posizione inaspettata];

8. Rimuovo token ] dall'input[Token ] in posizione inaspettata];
9. Rimuovo token \$ dall'input[Token \$ in posizione inaspettata];
10. Termino l'esecuzione[Stringa vuota non accettata!];
11. Pop stack stato attuale e simbolo precedente[Token *id* o *num* non trovato, ripristino a stato precedente];
12. Aggiunta in input token + per contesto su token \*[Token + non trovato, aggiunto per contesto];
13. Aggiunta in input token + per contesto su token *id*[Token + non trovato, aggiunto per contesto];
14. Aggiunta in input token + per contesto su token *num*[Token + non trovato, aggiunto per contesto];
15. Aggiunta in input token \* per contesto[Token *id* o *num* non trovato, aggiunto per contesto];
16. Aggiunta in input token *id* per contesto[Token *id* non trovato, aggiunto per contesto];
17. Aggiunta in input token *num* per contesto[Token *num* non trovato, aggiunto per contesto];

## Parser LALR

Viene ora calcolato, seguendo quanto spiegato a lezione, i kernel per gli item del parser SLR al fine di calcolare la parsing table del parser LALR. Sappiamo per certo che gli stati del parser SLR ed LALR coincidono e differiscono solo nel calcolo dei *lookahead*, passo affrontato successivamente.

## Calcolo dei Kernel

Definiamo di seguito i kernel per ogni item del parser SLR.

- $K_0 = \{S \rightarrow \cdot E\};$
- $K_1 = \{S \rightarrow E \cdot\};$
- $K_2 = \{E \rightarrow L \cdot = R, L \rightarrow L \cdot [E], R \rightarrow L \cdot, R \rightarrow L \cdot pp\};$
- $K_3 = \{E \rightarrow R \cdot, R \rightarrow R \cdot + R\};$
- $K_4 = \{L \rightarrow * \cdot R\};$
- $K_5 = \{L \rightarrow id \cdot\};$
- $K_6 = \{R \rightarrow num \cdot\};$
- $K_7 = \{R \rightarrow pp \cdot L\};$
- $K_8 = \{E \rightarrow L = \cdot R\};$
- $K_9 = \{L \rightarrow L [\cdot E]\};$



- $K_{10} = \{R \rightarrow Lpp\cdot\};$
- $K_{11} = \{R \rightarrow R + \cdot R\};$
- $K_{12} = \{L \rightarrow *R\cdot, R \rightarrow R \cdot + R\};$
- $K_{13} = \{R \rightarrow L\cdot, R \rightarrow L \cdot pp, L \rightarrow L \cdot [E]\};$
- $K_{14} = \{R \rightarrow ppL\cdot, L \rightarrow L \cdot [E]\};$
- $K_{15} = \{E \rightarrow L = R\cdot, R \rightarrow R \cdot + R\};$
- $K_{16} = \{L \rightarrow L[E\cdot]\};$
- $K_{17} = \{R \rightarrow R + R\cdot, R \rightarrow R \cdot + R\};$
- $K_{18} = \{L \rightarrow L[E]\cdot\}.$

Si definisce la tabella di propagazione dei lookahead:

$I_0$	$S \rightarrow \cdot E$	$I_1$
		$I_2$
		$I_3$
		$I_4$
		$I_5$
		$I_6$
		$I_7$
$I_2$	$E \rightarrow L\cdot = R$	$I_8$
	$L \rightarrow L \cdot [E]$	$I_9$
	$R \rightarrow L\cdot$	
	$R \rightarrow L \cdot pp$	$I_{10}$
$I_3$	$E \rightarrow R\cdot$	
	$R \rightarrow R \cdot + R$	$I_{11}$
$I_4$	$L \rightarrow * \cdot R$	$I_4$
		$I_5$
		$I_6$
		$I_7$
		$I_{12}$
		$I_{13}$
$I_7$	$R \rightarrow pp \cdot L$	$I_4$
		$I_5$
		$I_{14}$
$I_8$	$E \rightarrow L = \cdot R$	$I_4$
		$I_5$
		$I_6$
		$I_7$
		$I_{13}$
		$I_{15}$
$I_9$	$L \rightarrow L[\cdot E]$	$I_{16}$

$I_{11}$	$R \rightarrow R + \cdot R$	$I_4$
		$I_5$
		$I_6$
		$I_7$
		$I_{13}$
		$I_{17}$
$I_{12}$	$\frac{L \rightarrow *R \cdot}{R \rightarrow R \cdot + R}$	$I_{11}$
$I_{13}$	$\frac{R \rightarrow L \cdot}{R \rightarrow L \cdot pp}$	$I_{10}$
	$\frac{L \rightarrow L \cdot [E]}{L \rightarrow L \cdot [E]}$	$I_9$
$I_{14}$	$\frac{R \rightarrow ppL \cdot}{L \rightarrow L \cdot [E]}$	$I_9$
$I_{15}$	$\frac{E \rightarrow L = R \cdot}{R \rightarrow R \cdot + R}$	$I_{11}$
$I_{16}$	$L \rightarrow L[E \cdot]$	$I_{18}$
$I_{17}$	$\frac{R \rightarrow R + R \cdot}{R \rightarrow R \cdot + R}$	$I_{11}$

Si passa quindi al calcolo dei lookahead attraverso la propagazione:

	spontanee	propagate
$I_0$	$S \rightarrow \cdot E$	\$
$I_1$	$S \rightarrow E \cdot$	\$
$I_2$	$\frac{E \rightarrow L \cdot = R}{L \rightarrow L \cdot [E]}$	\$
	$\frac{R \rightarrow L \cdot}{R \rightarrow L \cdot pp}$	\$
$I_3$	$\frac{E \rightarrow R \cdot}{R \rightarrow R \cdot + R}$	\$
$I_4$	$L \rightarrow * \cdot R$	\$
$I_5$	$L \rightarrow id \cdot$	\$
$I_6$	$R \rightarrow num \cdot$	\$ = [ pp
$I_7$	$R \rightarrow pp \cdot L$	\$ = [ pp
$I_8$	$E \rightarrow L = \cdot R$	] \$
$I_9$	$L \rightarrow L[E \cdot]$	= [ pp + ] \$
$I_{10}$	$R \rightarrow Lpp \cdot$	+ ] \$ = [ pp
$I_{11}$	$R \rightarrow R + \cdot R$	+ ] \$ = [ pp
$I_{12}$	$\frac{L \rightarrow *R \cdot}{R \rightarrow R \cdot + R}$	= [ pp + ] \$
$I_{13}$	$\frac{R \rightarrow L \cdot}{R \rightarrow L \cdot pp}$	= [ pp ] \$
	$\frac{L \rightarrow L \cdot [E]}{L \rightarrow L \cdot [E]}$	\$ ] = [ pp
$I_{14}$	$\frac{R \rightarrow ppL \cdot}{L \rightarrow L \cdot [E]}$	+ ] \$ = [ pp
$I_{15}$	$\frac{E \rightarrow L = R \cdot}{R \rightarrow R \cdot + R}$	] \$
$I_{16}$	$L \rightarrow L[E \cdot]$	= [ pp + ] \$
$I_{17}$	$\frac{R \rightarrow R + R \cdot}{R \rightarrow R \cdot + R}$	+ ] \$ = [ pp
$I_{18}$	$L \rightarrow L[E \cdot]$	] \$ = [ pp

### Tabella di parsing LALR

Di seguito la tabella di parsing LALR.

	id	num	pp	*	+	=	[	]	\$	E	L	R	S
0	s5	s6	s7	s4	e5	e6	e7	e8	e11	1	2	3	
1	e0	e0	e0	e0	e0	e0	e0	e0	acc				
2	e14	e15	s10	e4	r7	s8	s9	r7	r7				
3	e14	e15	e17	e4	s11	e6	e7	r2	r2				
4	s5	s6	s7	s4	e5	e6	e7	e8	e17		13	12	
5	e14	e15	r4	e13	r4	r4	r4	r4	r4				
6	e14	e15	r6	e13	r6	r6	r6	r6	r6				
7	s5	e15	e3	s4	e17	e17	e17	e8	e17		14		
8	s5	s6	s7	s4	e17	e6	e7	e8	e18		13	15	
9	s5	s6	s7	s4	e5	e17	e7	e17	e18	16	2	3	
10	e14	e15	r9	e4	r9	r9	r9	r9	r9				
11	s5	s6	s7	s4	e5	e6	e12	e12	e17		13	17	
12	e14	e15	r3	e4	s11 / r3	r3	r3	r3	r3				
13	e14	e15	s10 / r7	e16	r7	r7	s9 / r7	r7	r7				
14	e14	e15	r10	e4	r10	r10	s9 / r10	r10	r10				
15	e14	e15	e3	e4	s11	e6	e7	r1	r1				
16	e14	e15	e3	e4	e5	e6	e7	s18					
17	e14	e15	r8	e4	s11 / r8	r8	r8	r8	r8				
18	e14	e15	r5	e4	r5	r5	r5	r5	r5				

Gli errori definiti sono i medesimi del parser SLR.

## Parsing della stringa

Di seguito viene mostrata il parsing della stringa  $id \ [ \ id + id + \ ] = num \ id \ pp + num$  fornita. Si mostra un parsing unico in quanto le azioni intraprese dal parser SLR e LALR sono le medesime.

Stack	Input	Azione
0	$id \ [ \ id + id + \ ] = num \ id \ pp + num \ \$$	$ACTION(0, id) = s5$
$<0, id> \ 5$	$[ \ id + id + \ ] = num \ id \ pp + num \ \$$	$ACTION( \ 5, [ \ ] ) = r4$
$<0, L>$	$[ \ id + id + \ ] = num \ id \ pp + num \ \$$	GOTO 2
$<0, L> \ 2$	$[ \ id + id + \ ] = num \ id \ pp + num \ \$$	$ACTION( \ 2, [ \ ] ) = s9$
$<0, L> \ <2, [ \ ] > \ 9$	$id + id + \ ] = num \ id \ pp + num \ \$$	$ACTION( \ 9, id ) = s5$
$<0, L> \ <2, [ \ ] > \ <9, id> \ 5$	$+ id + \ ] = num \ id \ pp + num \ \$$	$ACTION( \ 5, + ) = r4$
$<0, L> \ <2, [ \ ] > \ <9, L>$	$+ id + \ ] = num \ id \ pp + num \ \$$	GOTO 2
$<0, L> \ <2, [ \ ] > \ <9, L> \ 2$	$+ id + \ ] = num \ id \ pp + num \ \$$	$ACTION( \ 2, + ) = r7$
$<0, L> \ <2, [ \ ] > \ <9, R>$	$+ id + \ ] = num \ id \ pp + num \ \$$	GOTO 3
$<0, L> \ <2, [ \ ] > \ <9, R> \ 3$	$+ id + \ ] = num \ id \ pp + num \ \$$	$ACTION( \ 3, + ) = s11$
$<0, L> \ <2, [ \ ] > \ <9, R> \ <3, +> \ 11$	$id + \ ] = num \ id \ pp + num \ \$$	$ACTION(11, id) = s5$
$<0, L> \ <2, [ \ ] > \ <9, R> \ <3, +> \ <11, id> \ 5$	$+ \ ] = num \ id \ pp + num \ \$$	$ACTION( \ 5, + ) = r4$
$<0, L> \ <2, [ \ ] > \ <9, R> \ <3, +> \ <11, L>$	$+ \ ] = num \ id \ pp + num \ \$$	GOTO 13
$<0, L> \ <2, [ \ ] > \ <9, R> \ <3, +> \ <11, L> \ 13$	$+ \ ] = num \ id \ pp + num \ \$$	$ACTION(13, +) = r7$
$<0, L> \ <2, [ \ ] > \ <9, R> \ <3, +> \ <11, R>$	$+ \ ] = num \ id \ pp + num \ \$$	GOTO 17
$<0, L> \ <2, [ \ ] > \ <9, R> \ <3, +> \ <11, R> \ 17$	$] = num \ id \ pp + num \ \$$	$ACTION(17, ) = s11$
$<0, L> \ <2, [ \ ] > \ <9, R> \ <3, +> \ <11, R> \ <17, +> \ 11$	$] = num \ id \ pp + num \ \$$	E12
$<0, L> \ <2, [ \ ] > \ <9, R> \ <3, +> \ <11, R> \ 17$	$] = num \ id \ pp + num \ \$$	$ACTION(17, [ \ ] ) = r8$
$<0, L> \ <2, [ \ ] > \ <9, R>$	$] = num \ id \ pp + num \ \$$	GOTO 3
$<0, L> \ <2, [ \ ] > \ <9, R> \ 3$	$] = num \ id \ pp + num \ \$$	$ACTION( \ 3, [ \ ] ) = r2$
$<0, L> \ <2, [ \ ] > \ <9, E>$	$] = num \ id \ pp + num \ \$$	GOTO 16
$<0, L> \ <2, [ \ ] > \ <9, E> \ 16$	$] = num \ id \ pp + num \ \$$	$ACTION(16, [ \ ] ) = s18$
$<0, L> \ <2, [ \ ] > \ <9, E> \ <16, [ \ ] > \ 18$	$= num \ id \ pp + num \ \$$	$ACTION(18, =) = r5$
$<0, L>$	$= num \ id \ pp + num \ \$$	GOTO 2
$<0, L> \ 2$	$= num \ id \ pp + num \ \$$	$ACTION( \ 2, = ) = s8$
$<0, L> \ <2, => \ 8$	$num \ id \ pp + num \ \$$	$ACTION( \ 8, num ) = s6$
$<0, L> \ <2, => \ <8, num> \ 6$	$id \ pp + num \ \$$	E14
$<0, L> \ <2, => \ <8, num> \ 6$	$+ id \ pp + num \ \$$	$ACTION( \ 6, + ) = r6$
$<0, L> \ <2, => \ <8, R>$	$+ id \ pp + num \ \$$	GOTO 15
$<0, L> \ <2, => \ <8, R> \ 15$	$+ id \ pp + num \ \$$	$ACTION(15, +) = s11$
$<0, L> \ <2, => \ <8, R> \ <15, +> \ 11$	$id \ pp + num \ \$$	$ACTION(11, id) = s5$
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, id> \ 5$	$pp + num \ \$$	$ACTION( \ 5, pp ) = r4$
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, L>$	$pp + num \ \$$	GOTO 13
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, L> \ 13$	$pp + num \ \$$	$ACTION(13, pp) = s10$
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, L> \ <13, pp> \ 10$	$+ num \ \$$	$ACTION(10, +) = r9$
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, R>$	$+ num \ \$$	GOTO 17
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, R> \ 17$	$+ num \ \$$	$ACTION(17, +) = s11$
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, R> \ <17, +> \ 11$	$num \ \$$	$ACTION(11, num) = s6$
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, R> \ <17, +> \ <11, num> \ 6$	$\$$	$ACTION( \ 6, \$ ) = r6$
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, R> \ <17, +> \ <11, R>$	$\$$	GOTO 17
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, R> \ <17, +> \ <11, R> \ 17$	$\$$	$ACTION(17, \$) = r8$
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, R>$	$\$$	GOTO 17
$<0, L> \ <2, => \ <8, R> \ <15, +> \ <11, R> \ 17$	$\$$	$ACTION(17, \$) = r8$
$<0, L> \ <2, => \ <8, R>$	$\$$	GOTO 15
$<0, L> \ <2, => \ <8, R> \ 15$	$\$$	$ACTION(15, \$) = r1$
$<0, E>$	$\$$	GOTO 1
$<0, E> \ 1$	$\$$	$ACTION( \ 1, \$ ) = acc$