

# ProgettoLC parteB Riccardo Cavasin Relazione

Calcolo la collezione canonica di insiemi di item LR(0) sulla grammatica aumentata:

- 0.  $E' \rightarrow E$
- 1.  $E \rightarrow L \text{ = } R$
- 2.  $E \rightarrow L \text{ pp}$
- 3.  $E \rightarrow \text{pp} L$
- 4.  $L \rightarrow \text{id}$
- 5.  $L \rightarrow L \text{ [ } E \text{ ]}$
- 6.  $R \rightarrow \text{num}$
- 7.  $R \rightarrow L$
- 8.  $R \rightarrow R \text{ + } R$
- 9.  $R \rightarrow L \text{ pp}$
- 10.  $R \rightarrow \text{pp} L$

## Automa LR(0)

Per ogni stato sono distinti i kernel items (sopra), dagli altri items (sotto). Gli items di tipo  $A \rightarrow \alpha \cdot$  che generano un'azione reduce sono evidenziati in grassetto. È stato aggiunto un simbolo "\$" che rappresenta l'EOF, la fine dell'input.

- $I_0 = \text{closure}(\{E' \rightarrow \cdot E \$\})$ 
  - $E' \rightarrow \cdot E \$$

- $E \rightarrow \cdot L \text{ = } R$
  - $E \rightarrow \cdot L \text{ pp}$
  - $E \rightarrow \cdot \text{pp} L$
  - $L \rightarrow \cdot \text{id}$
  - $L \rightarrow \cdot L \text{ [ } E \text{ ]}$

- $I_1 = \text{goto}(I_0, E)$ 
  - $E' \rightarrow E \cdot \$$**
- $I_2 = \text{goto}(I_0, L) = \text{goto}(I_7, L)$

- $E \rightarrow L \cdot \text{ = } R$
  - $E \rightarrow L \cdot \text{pp}$
  - $L \rightarrow L \cdot \text{ [ } E \text{ ]}$
- $I_3 = \text{goto}(I_0, \text{pp}) = \text{goto}(I_7, \text{pp})$ 
  - $E \rightarrow \text{pp} \cdot L$

- $L \rightarrow \cdot \text{id}$
  - $L \rightarrow \cdot L \text{ [ } E \text{ ]}$

- $I_4 = \text{goto}(I_0, \text{id}) = \text{goto}(I_3, \text{id}) = \text{goto}(I_5, \text{id}) = \text{goto}(I_7, \text{id}) = \text{goto}(I_{12}, \text{id}) = \text{goto}(I_{14}, \text{id})$ 
  - $L \rightarrow \text{id} \cdot$**
- $I_5 = \text{goto}(I_2, \text{ = })$ 
  - $E \rightarrow L \text{ = } \cdot R$

- $R \rightarrow \cdot \text{num}$
  - $R \rightarrow \cdot L$
  - $R \rightarrow \cdot R \text{ + } R$
  - $R \rightarrow \cdot L \text{ pp}$
  - $R \rightarrow \cdot \text{pp} L$
  - $L \rightarrow \cdot \text{id}$
  - $L \rightarrow \cdot L \text{ [ } E \text{ ]}$

- $I_6 = \text{goto}(I_2, \boxed{\text{pp}})$ 
    - $\mathbf{E} \rightarrow \mathbf{L} \boxed{\text{pp}} \cdot$
  - $I_7 = \text{goto}(I_2, \boxed{[ ]}) = \text{goto}(I_8, \boxed{[ ]}) = \text{goto}(I_{11}, \boxed{[ ]}) = \text{goto}(I_{16}, \boxed{[ ]})$ 
    - $\mathbf{L} \rightarrow \mathbf{L} \boxed{[ ]} \cdot \mathbf{E} \boxed{[ ]}$
- 

- $\mathbf{E} \rightarrow \cdot \mathbf{L} \boxed{=} \mathbf{R}$
  - $\mathbf{E} \rightarrow \cdot \mathbf{L} \boxed{\text{pp}}$
  - $\mathbf{E} \rightarrow \cdot \boxed{\text{pp}} \mathbf{L}$
  - $\mathbf{L} \rightarrow \cdot \boxed{\text{id}}$
  - $\mathbf{L} \rightarrow \cdot \mathbf{L} \boxed{[ ]} \mathbf{E} \boxed{[ ]}$
  - $I_8 = \text{goto}(I_3, \mathbf{L})$ 
    - $\mathbf{E} \rightarrow \boxed{\text{pp}} \mathbf{L} \cdot$
    - $\mathbf{L} \rightarrow \mathbf{L} \cdot \boxed{[ ]} \mathbf{E} \boxed{[ ]}$
  - $I_9 = \text{goto}(I_5, \mathbf{R})$ 
    - $\mathbf{E} \rightarrow \mathbf{L} \boxed{=} \mathbf{R} \cdot$
    - $\mathbf{R} \rightarrow \mathbf{R} \cdot \boxed{+} \mathbf{R}$
  - $I_{10} = \text{goto}(I_5, \boxed{\text{num}}) = \text{goto}(I_{14}, \boxed{\text{num}})$ 
    - $\mathbf{R} \rightarrow \boxed{\text{num}} \cdot$
  - $I_{11} = \text{goto}(I_5, \mathbf{L}) = \text{goto}(I_{14}, \mathbf{L})$ 
    - $\mathbf{R} \rightarrow \mathbf{L} \cdot$
    - $\mathbf{R} \rightarrow \mathbf{L} \cdot \boxed{\text{pp}}$
    - $\mathbf{L} \rightarrow \mathbf{L} \cdot \boxed{[ ]} \mathbf{E} \boxed{[ ]}$
  - $I_{12} = \text{goto}(I_5, \boxed{\text{pp}}) = \text{goto}(I_{14}, \boxed{\text{pp}})$ 
    - $\mathbf{R} \rightarrow \boxed{\text{pp}} \cdot \mathbf{L}$
- 

- $\mathbf{L} \rightarrow \cdot \boxed{\text{id}}$
  - $\mathbf{L} \rightarrow \cdot \mathbf{L} \boxed{[ ]} \mathbf{E} \boxed{[ ]}$
  - $I_{13} = \text{goto}(I_7, \mathbf{E})$ 
    - $\mathbf{L} \rightarrow \mathbf{L} \boxed{[ ]} \mathbf{E} \cdot \boxed{[ ]}$
  - $I_{14} = \text{goto}(I_9, \boxed{+}) = \text{goto}(I_{18}, \boxed{+})$ 
    - $\mathbf{R} \rightarrow \mathbf{R} \boxed{+} \cdot \mathbf{R}$
- 

- $\mathbf{R} \rightarrow \cdot \boxed{\text{num}}$
- $\mathbf{R} \rightarrow \cdot \mathbf{L}$
- $\mathbf{R} \rightarrow \cdot \mathbf{R} \boxed{+} \mathbf{R}$
- $\mathbf{R} \rightarrow \cdot \mathbf{L} \boxed{\text{pp}}$
- $\mathbf{R} \rightarrow \cdot \boxed{\text{pp}} \mathbf{L}$
- $\mathbf{L} \rightarrow \cdot \boxed{\text{id}}$
- $\mathbf{L} \rightarrow \cdot \mathbf{L} \boxed{[ ]} \mathbf{E} \boxed{[ ]}$
- $I_{15} = \text{goto}(I_{11}, \boxed{\text{pp}})$ 
  - $\mathbf{R} \rightarrow \mathbf{L} \boxed{\text{pp}} \cdot$
- $I_{16} = \text{goto}(I_{12}, \mathbf{L})$ 
  - $\mathbf{R} \rightarrow \boxed{\text{pp}} \mathbf{L} \cdot$
  - $\mathbf{L} \rightarrow \mathbf{L} \cdot \boxed{[ ]} \mathbf{E} \boxed{[ ]}$
- $I_{17} = \text{goto}(I_{13}, \boxed{[ ]})$ 
  - $\mathbf{L} \rightarrow \mathbf{L} \boxed{[ ]} \mathbf{E} \boxed{[ ]} \cdot$
- $I_{18} = \text{goto}(I_{14}, \mathbf{R})$ 
  - $\mathbf{R} \rightarrow \mathbf{R} \boxed{+} \mathbf{R} \cdot$
  - $\mathbf{R} \rightarrow \mathbf{R} \cdot \boxed{+} \mathbf{R}$

# Tabella di parsing SLR

Contrassegnato con "\*" nella tabella c'è l'unico conflitto shift/reduce (s14/r8) nella tabella di parsing. Il conflitto è dato dall'ambiguità dell'operatore +. Come specificato dalla consegna, si sceglie lo shift per ottenere l'associatività a destra. Eccetto per questo conflitto, la grammatica è SLR(1).

first	follow
$first(E') = \{ \boxed{+}, \boxed{id} \}$	$follow(E') = \{ \} \$$
$first(E) = \{ \boxed{+}, \boxed{id} \}$	$follow(E) = \{ \boxed{]}, \} \$$
$first(L) = \{ \boxed{id} \}$	$follow(L) = \{ \boxed{=}, \boxed{+}, \boxed{[}, \boxed{]}, \text{' + ' } \} \$$
$first(R) = \{ \boxed{num}, \boxed{+}, \boxed{id} \}$	$follow(R) = \{ \boxed{+}, \boxed{]}, \} \$$

stato \ action {	<div>+ </div>	<div>= </div>	<div>[ </div>	<div>] </div>	<div>id </div>	<div>num </div>	<div>pp </div>	<div>\$ </div>	<div>} goto {</div>	<div>E' </div>	<div>E </div>	<div>L </div>	<div>R </div>	<div>}</div>
$I_0$	e2	e3	e4	e2	s4	e6	s3	e1		e0	1	2	e0	
$I_1$	e0!	e0!	e0!	e0!	e0!	e0!	e0!	acc		e0	e0	e0	e0	
$I_2$	e14	s5	s7	e2	e18	e6	s6	e1		e0	e0	e0	e0	
$I_3$	e14	e2	e4	e2	s4	e5	e2	e1		e0	e0	8	e0	
$I_4$	r4	r4	r4	r4	e8^	e8^	r4	r4		e0	e0	e0	e0	
$I_5$	e15	e2	e2	e2	s4	s10	s12	e1		e0	e0	11	9	
$I_6$	e14	e16	e2	r2	e2	e2	e2	r2		e0	e0	e0	e0	
$I_7$	e14	e3	e4	e19	s4	e6	s3	e1		e0	13	2	e0	
$I_8$	e14	e16	s7	r3	e2	e2	e2	r3		e0	e0	e0	e0	
$I_9$	s14	e17	e19	r1	e7	e7	e2	r1		e0	e0	e0	e0	
$I_{10}$	r6	e9^	e9^	r6	e9^	e9^	e9^	r6		e0	e0	e0	e0	
$I_{11}$	r7	e17	s7	r7	e2	e10	s15	r7		e0	e0	e0	e0	
$I_{12}$	e2	e2	e4	e2	s4	e5	e2	e1		e0	e0	16	e0	
$I_{13}$	e0!	e0!	e2	s17	e0!	e0!	e2	e1		e0	e0	e0	e0	
$I_{14}$	e2	e2	e4	e11	s4	s10	s12	e1		e0	e0	11	18	
$I_{15}$	r9	e17	e12^	r9	e12^	e12^	e12^	r9		e0	e0	e0	e0	
$I_{16}$	r10	e17	s7	r10	e2	e7	e2	r10		e0	e0	e0	e0	
$I_{17}$	r5	r5	r5	r5	e13^	e13^	r5	r5		e0	e0	e0	e0	
$I_{18}$	s14*	e17	e19	r8	e7	e7	e2	r8		e0	e0	e0	e0	
$I_{19}$	s14*	e17	e19	r8	e7	e7	e2	r8		e0	e0	e0	e0	

# Errori

n°	azioni
e0	<i>print</i> : internal error exit
e1	<i>print</i> : program ended too early <i>exit</i>
e2	<i>print</i> : unexpected {lookahead} <i>skip</i>
e3	<i>print</i> : lvalue missing in assignment <i>push goto(I, L)</i>
e4	<i>print</i> : accessor without identifier <i>push goto(I, L)</i>
e5	<i>print</i> : <code>num</code> is not a valid lvalue <i>skip, push goto(I, L)</i>
e6	<i>print</i> : <code>num</code> is allowed only in a assignment rvalue <i>skip</i>
e7	<i>print</i> : missing <code>+</code> operator <i>push goto(I, <code>+</code>)</i>
e8^	//bubble up r4
e9^	//bubble up r6
e10	<i>print</i> : missing <code>+</code> operator <i>r7, push goto(I, <code>+</code>)</i>
e11	<i>print</i> : right <code>+</code> operand missing <i>pop</i>
e12^	//bubble up r9
e13^	//bubble up r5
e14	<i>print</i> : <code>+</code> is allowed only in a assignment rvalue <i>skip</i>
e15	<i>print</i> : left <code>+</code> operand missing <i>push goto(I, R)</i>
e16	<i>print</i> : {symbol[0] symbol[1]} is not a valid lvalue <i>pop 2, push goto(I, L)</i>
e17	<i>print</i> : only one assignment per expression <i>skip</i>
e18	<i>print</i> : unexpected {lookahead} (add "=") <i>push goto(I, <code>=</code>)</i>
e19	<i>print</i> : accessor cannot be empty <i>push goto(I, E)</i>

# Commenti sulla gestione degli errori

In alcuni casi, si è cercato di rimandare la gestione degli errori per poter dare messaggi più significativi. Questo approccio ha lo svantaggio di essere più complicato da realizzare correttamente e pronò ad errori. Le error routines che fanno "bubbling up" sono state marcate con "^". Inoltre, non vengono mai inseriti stati non validi nello stack, quindi la sezione goto della tabella ha solo errori interni.

Alcune routine d'errore fanno riferimento agli ultimi 0\dots n simboli nello stack partendo dalla cima.

Per come è fatto il parser SLR generato da questa grammatica, è impossibile gestire il caso d'errore `id [ id + id + ]` come "`id [ R ]`" senza aggiungere stati. Questo perché il parser ha già una regola di riduzione da R a "`L = R`" sul lookahead `]`.

## Esecuzione d'esempio

stack	symbols	input	action
$I_0$		<code>id [ id + id + ] = num id pp + num \$</code>	s4
$I_0 I_4$	<code>id</code>	<code>[ id + id + ] = num id pp + num \$</code>	r4
$I_0 I_2$	L	<code>[ id + id + ] = num id pp + num \$</code>	s7
$I_0 I_2 I_7$	L <code>[</code>	<code>id + id + ] = num id pp + num \$</code>	s4
$I_0 I_2 I_7 I_4$	L <code>[</code> <code>id</code>	<code>+ id + ] = num id pp + num \$</code>	r4
$I_0 I_2 I_7 I_2$	L <code>[</code> L	<code>+ id + ] = num id pp + num \$</code>	<code>+</code> is allowed only in a assignment rvalue <i>skip</i>
$I_0 I_2 I_7 I_2$	L <code>[</code> L	<code>id + ] = num id pp + num \$</code>	unexpected id (add "=") <i>push goto(I, =)</i>
$I_0 I_2 I_7 I_2 I_5$	L <code>[</code> L <code>(=)</code>	<code>id + ] = num id pp + num \$</code>	s4
$I_0 I_2 I_7 I_2 I_5 I_4$	L <code>[</code> L <code>(=)</code> <code>id</code>	<code>+ ] = num id pp + num \$</code>	r4
$I_0 I_2 I_7 I_2 I_5 I_{11}$	L <code>[</code> L <code>(=)</code> L	<code>+ ] = num id pp + num \$</code>	r7
$I_0 I_2 I_7 I_2 I_5 I_9$	L <code>[</code> L <code>(=)</code> R	<code>+ ] = num id pp + num \$</code>	s14
$I_0 I_2 I_7 I_2 I_5 I_9 I_{14}$	L <code>[</code> L <code>(=)</code> R <code>+</code>	<code>] = num id pp + num \$</code>	right <code>+</code> operand missing <i>pop</i>
$I_0 I_2 I_7 I_2 I_5 I_9$	L <code>[</code> L <code>(=)</code> R	<code>] = num id pp + num \$</code>	r1
$I_0 I_2 I_7 I_{13}$	L <code>[</code> E	<code>] = num id pp + num \$</code>	s17
$I_0 I_2 I_7 I_{13} I_{17}$	L <code>[</code> E <code>]</code>	<code>= num id pp + num \$</code>	r5
$I_0 I_2$	L	<code>= num id pp + num \$</code>	s5
$I_0 I_2 I_5$	L <code>=</code>	<code>num id pp + num \$</code>	s10
$I_0 I_2 I_5 I_{10}$	L <code>=</code> <code>num</code>	<code>id pp + num \$</code>	r9^
$I_0 I_2 I_5 I_9$	L <code>=</code> R	<code>id pp + num \$</code>	missing <code>+</code> operator <i>push goto(I, +)</i>
$I_0 I_2 I_5 I_9 I_{14}$	L <code>=</code> R <code>(+)</code>	<code>id pp + num \$</code>	s4

stack	symbols	input	action
$I_0 I_2 I_5 I_9 I_{14} I_4$	$L = R ( + ) id$	pp + num \$	r4
$I_0 I_2 I_5 I_9 I_{14} I_1 1$	$L = R ( + ) L$	pp + num \$	s15
$I_0 I_2 I_5 I_9 I_{14} I_1 1 I_{15}$	$L = R ( + ) L pp$	+ num \$	r9
$I_0 I_2 I_5 I_9 I_{14} I_1 8$	$L = R ( + ) R$	+ num \$	s14
$I_0 I_2 I_5 I_9 I_{14} I_1 8 I_1 4$	$L = R ( + ) R +$	num \$	s10
$I_0 I_2 I_5 I_9 I_{14} I_1 8 I_1 4 I_{10}$	$L = R ( + ) R +$ num	\$	r6
$I_0 I_2 I_5 I_9 I_{14} I_1 8 I_1 4 I_{18}$	$L = R ( + ) R + R$	\$	r8
$I_0 I_2 I_5 I_9 I_{14} I_1 8$	$L = R ( + ) R$	\$	r8
$I_0 I_2 I_5 I_9$	$L = R$	\$	r1
$I_0 I_1$	E	\$	acc

## Calcolo lookahead per parser LALR

Uso le varianti di closure( $I$ ) e goto( $I$ , X) con lookahead per calcolare i lookahead generati spontaneamente e i lookahead propagati. L'algorithmo per il calcolo dei lookahead utilizza un simbolo non terminale fittizio # per calcolare contemporaneamente entrambi.

- $I_0$ 
  - closure( $\{E' \rightarrow \cdot E, \# \}$ )
    - $E' \rightarrow \cdot E, \#$
    - $E \rightarrow \cdot L = R, \#$
    - $E \rightarrow \cdot L pp, \#$
    - $E \rightarrow \cdot pp L, \#$
    - $L \rightarrow \cdot id, = / pp$
    - $L \rightarrow \cdot L [ E ], = / pp$
- $I_1$ 
  - closure( $\{E' \rightarrow E \cdot, \# \}$ )
    - $E' \rightarrow E \cdot, \#$
- $I_2$ 
  - closure( $\{E \rightarrow L \cdot = R, \# \}$ )
    - $E \rightarrow L \cdot = R, \#$
  - closure( $\{E \rightarrow L \cdot pp, \# \}$ )
    - $E \rightarrow L \cdot pp, \#$
  - closure( $\{L \rightarrow L \cdot [ E ], \# \}$ )
    - $L \rightarrow L \cdot [ E ], \#$
- $I_3$ 
  - closure( $\{E \rightarrow pp \cdot L, \# \}$ )
    - $E \rightarrow pp \cdot L, \#$
    - $L \rightarrow \cdot id, \# / [$
    - $L \rightarrow \cdot L [ E ], \# / [$
- $I_4$ 
  - closure( $\{L \rightarrow id \cdot, \# \}$ )
    - $L \rightarrow id \cdot, \#$
- $I_5$ 
  - closure( $\{E \rightarrow L = \cdot R, \# \}$ )

- $E \rightarrow L = \cdot R, \#$
- $R \rightarrow \cdot \text{num}, \# / +$
- $R \rightarrow \cdot L, \# / +$
- $R \rightarrow \cdot R + R, \# / +$
- $R \rightarrow \cdot L \text{ pp}, \# / +$
- $R \rightarrow \cdot \text{pp} L, \# / +$
- $L \rightarrow \cdot \text{id}, \# / + / \text{pp}$
- $L \rightarrow \cdot L [ E ], \# / + / \text{pp}$

•  $I_6$

- $\text{closure}(\{E \rightarrow L \text{ pp} \cdot, \#\})$ 
  - $E \rightarrow L \text{ pp} \cdot, \#$

•  $I_7$

- $\text{closure}(\{L \rightarrow L [ \cdot E ], \#\})$ 
  - $L \rightarrow L [ \cdot E ], \#$
  - $E \rightarrow \cdot L = R, ]$
  - $E \rightarrow \cdot L \text{ pp}, ]$
  - $E \rightarrow \cdot \text{pp} L, ]$
  - $L \rightarrow \cdot \text{id}, = / \text{pp}$
  - $L \rightarrow \cdot L [ E ], = / \text{pp}$

•  $I_8$

- $\text{closure}(\{E \rightarrow \text{pp} L \cdot, \#\})$ 
  - $E \rightarrow \text{pp} L \cdot, \#$
- $\text{closure}(\{L \rightarrow L \cdot [ E ], \#\})$ 
  - $L \rightarrow L \cdot [ E ], \#$

•  $I_9$

- $\text{closure}(\{E \rightarrow L = R \cdot, \#\})$ 
  - $E \rightarrow L = R \cdot, \#$
- $\text{closure}(\{R \rightarrow R \cdot + R, \#\})$ 
  - $R \rightarrow R \cdot + R, \#$

•  $I_{10}$

- $\text{closure}(\{R \rightarrow \text{num} \cdot, \#\})$ 
  - $R \rightarrow \text{num} \cdot, \#$

•  $I_{11}$

- $\text{closure}(\{R \rightarrow L \cdot, \#\})$ 
  - $R \rightarrow L \cdot, \#$
- $\text{closure}(\{R \rightarrow L \cdot \text{pp}, \#\})$ 
  - $R \rightarrow L \cdot \text{pp}, \#$
- $\text{closure}(\{L \rightarrow L \cdot [ E ], \#\})$ 
  - $L \rightarrow L \cdot [ E ], \#$

•  $I_{12}$

- $\text{closure}(\{R \rightarrow \text{pp} \cdot L, \#\})$ 
  - $R \rightarrow \text{pp} \cdot L, \#$
  - $L \rightarrow \cdot \text{id}, \# / [$
  - $L \rightarrow \cdot L [ E ], \# / [$

•  $I_{13}$

- $\text{closure}(\{L \rightarrow L [ E \cdot ], \#\})$ 
  - $L \rightarrow L [ E \cdot ], \#$

•  $I_{14}$

- $\text{closure}(\{R \rightarrow R + \cdot R, \#\})$ 
  - $R \rightarrow R + \cdot R, \#$
  - $R \rightarrow \cdot \text{num}, \# / +$
  - $R \rightarrow \cdot L, \# / +$
  - $R \rightarrow \cdot R + R, \# / +$
  - $R \rightarrow \cdot L \text{ pp}, \# / +$

- $R \rightarrow \cdot \text{pp } L, \# / +$
- $L \rightarrow \cdot \text{id}, \# / + / \text{pp} / [$
- $L \rightarrow \cdot L [ E ], \# / + / \text{pp} / [$

•  $I_{15}$

- $\text{closure}(\{R \rightarrow L \text{pp} \cdot, \# \})$ 
  - $R \rightarrow L \text{pp} \cdot, \#$

•  $I_{16}$

- $\text{closure}(\{R \rightarrow \text{pp } L \cdot, \# \})$ 
  - $R \rightarrow \text{pp } L \cdot, \#$
- $\text{closure}(\{L \rightarrow L \cdot [ E ], \# \})$ 
  - $L \rightarrow L \cdot [ E ], \#$

•  $I_{17}$

- $\text{closure}(\{L \rightarrow L [ E ] \cdot, \# \})$ 
  - $L \rightarrow L [ E ] \cdot, \#$

•  $I_{18}$

- $\text{closure}(\{R \rightarrow R + R \cdot, \# \})$ 
  - $R \rightarrow R + R \cdot, \#$
- $\text{closure}(\{R \rightarrow R \cdot + R, \# \})$ 
  - $R \rightarrow R \cdot + R, \#$

La variante di goto( $I, X$ ) ottiene risultati equivalenti a quelli già ottenuti nell'automa LR(0).

L'algoritmo per il calcolo dei lookahead converge al 5° passo. L'ultimo passo aggiunge solo *allo stato*  $I_{18}$  ed è stato collassato.

stato	item lookahead {	init	pass 1	pass 2	pass 3	pass 4 (+1)
$I_0$	$E' \rightarrow \cdot E$	\$	\$	\$	\$	\$
$I_1$	$E' \rightarrow E \cdot$		\$	\$	\$	\$
$I_2$	$E \rightarrow L \cdot$ $= R$ $E \rightarrow L \cdot$ $\text{pp}$ $L \rightarrow L \cdot [$ $E ]$	$] ]$ $= / \text{pp} / [$	$] / \$$ $] / \$$ $= / \text{pp} / [$	$] / \$$ $] / \$$ $= / \text{pp} / [$	$] / \$$ $] / \$$ $= / \text{pp} / [$	$] / \$$ $] / \$$ $= / \text{pp} / [$
$I_3$	$E \rightarrow \text{pp} \cdot$ $L$	$] ]$	$] / \$$	$] / \$$	$] / \$$	$] / \$$
$I_4$	$L \rightarrow \text{id} \cdot$	$= / \text{pp} / [ / +$	$= / \text{pp} / [ / + / ]$	$= / \text{pp} / [ / + / ] / \$$	$= / \text{pp} / [ / + / ] / \$$	$= / \text{pp} / [ / + / ] / \$$
$I_5$	$E \rightarrow L =$ $\cdot R$		$] ]$	$] / \$$	$] / \$$	$] / \$$
$I_6$	$E \rightarrow L \text{pp}$ $\cdot$		$] ]$	$] / \$$	$] / \$$	$] / \$$
$I_7$	$L \rightarrow L [ \cdot$ $E ]$		$= / \text{pp} / [ / +$	$= / \text{pp} / [ / + / \$ / ]$	$= / \text{pp} / [ / + / \$ / ]$	$= / \text{pp} / [ / + / \$ / ]$
$I_8$	$E \rightarrow \text{pp } L$ $\cdot$ $L \rightarrow L \cdot [$ $E ]$	$[ ]$	$] ]$ $[ / ] ]$	$] / \$$ $[ / ] / \$$	$] / \$$ $[ / ] / \$$	$] / \$$ $[ / ] / \$$
$I_9$	$E \rightarrow L =$ $R \cdot$	$+ +$	$+ +$	$] ]$ $+ / ] ]$	$] / \$$ $+ / ] / \$$	$] / \$$ $+ / ] / \$$



stato	item lookahead {	init	pass 1	pass 2	pass 3	pass 4 (+1)
	$R \rightarrow R \cdot$ + R					
$I_{10}$	$R \rightarrow \text{num} \cdot$	+	+	+ / ]	+ / ] / \$	+ / ] / \$
$I_{11}$	$R \rightarrow L \cdot$ $R \rightarrow L \cdot$ pp $L \rightarrow L \cdot$ [ ] E ]	<div>+</div> <div>+</div> <div>+ / pp / [ ]</div>	<div>+</div> <div>+</div> <div>+ / pp / [ ]</div>	<div>+ / ]</div> <div>+ / ]</div> <div>+ / pp / [ / ]</div>	<div>+ / ] / \$</div> <div>+ / ] / \$</div> <div>+ / pp / [ / ] / \$</div>	<div>+ / ] / \$</div> <div>+ / ] / \$</div> <div>+ / pp / [ / ] / \$</div>
$I_{12}$	$R \rightarrow \text{pp} \cdot$ L	+	+	+ / ]	+ / ] / \$	+ / ] / \$
$I_{13}$	$L \rightarrow L$ [ ] E · ]			= / pp / [ / +	= / pp / [ / + / \$ / ]	= / pp / [ / + / \$ / ]
$I_{14}$	$R \rightarrow R$ + · R		+	+	+ / ]	+ / ] / \$
$I_{15}$	$R \rightarrow L$ pp ·		+	+	+ / ]	+ / ] / \$
$I_{16}$	$R \rightarrow \text{pp} L$ · $L \rightarrow L \cdot$ [ ] E ]	[	<div>+</div> <div>[ / +</div>	<div>+</div> <div>[ / +</div>	<div>+ / ]</div> <div>[ / + / ]</div>	<div>+ / ] / \$</div> <div>[ / + / ] / \$</div>
$I_{17}$	$L \rightarrow L$ [ ] E ] ·				= / pp / [ / +	= / pp / [ / + / \$ / ]
$I_{18}$	$R \rightarrow R$ + $R \cdot$ $R \rightarrow R \cdot$ + R	+	+	<div>+</div> <div>+</div>	<div>+</div> <div>+</div>	<div>+ / ] / (/ \$)</div> <div>+ / ] (/ \$)</div>

## Tabella di parsing LALR

Le tabelle di parsing sono uguali. La grammatica (disambiguata con l'associatività a destra dell'operatore 

+

) era SLR(1), quindi è anche LALR(1).