



Compiler Construction

Lecture 9: Syntax Analysis V ($LR(0)$ and $SLR(1)$ Parsing)

Summer Semester 2017

Thomas Noll

Software Modeling and Verification Group

RWTH Aachen University

<https://moves.rwth-aachen.de/teaching/ss-17/cc/>

Recap: $LR(0)$ Grammars

$LR(0)$ Grammars

The case $k = 0$ is relevant (in contrast to $LL(0)$): here the decision is just based on the contents of the pushdown, **without any lookahead**.

Corollary ($LR(0)$ grammar)

$G \in CFG_{\Sigma}$ has the **$LR(0)$ property** if for all rightmost derivations of the form

$$S \begin{cases} \Rightarrow_r^* \alpha A w \Rightarrow_r \alpha \beta w \\ \Rightarrow_r^* \gamma B x \Rightarrow_r \alpha \beta y \end{cases}$$

it follows that $\alpha = \gamma$, $A = B$, and $x = y$.

Goal: derive a **finite information** from the pushdown which suffices to resolve the nondeterminism (similar to abstraction of right context in LL parsing by fo-sets)

Recap: $LR(0)$ Grammars

$LR(0)$ Items and Sets

Definition ($LR(0)$ items and sets)

Let $G = \langle N, \Sigma, P, S \rangle \in CFG_{\Sigma}$ be start separated by $S' \rightarrow S$ and $S' \Rightarrow_r^* \alpha A w \Rightarrow_r \alpha \beta_1 \beta_2 w$ (i.e., $A \rightarrow \beta_1 \beta_2 \in P$).

- $[A \rightarrow \beta_1 \cdot \beta_2]$ is called an **$LR(0)$ item** for $\alpha \beta_1$.
- Given $\gamma \in X^*$, $LR(0)(\gamma)$ denotes the set of all **$LR(0)$ items** for γ , called the **$LR(0)$ set** (or: **$LR(0)$ information**) of γ .
- $LR(0)(G) := \{LR(0)(\gamma) \mid \gamma \in X^*\}$.

Corollary

1. For every $\gamma \in X^*$, $LR(0)(\gamma)$ is finite.
2. $LR(0)(G)$ is finite.
3. The item $[A \rightarrow \beta \cdot] \in LR(0)(\gamma)$ indicates the possible **reduction** $(w, \alpha \beta, z) \vdash (w, \alpha A, zi)$ where $\pi_i = A \rightarrow \beta$ and $\gamma = \alpha \beta$.
4. The item $[A \rightarrow \beta_1 \cdot Y \beta_2] \in LR(0)(\gamma)$ indicates an **incomplete handle** β_1 (to be completed by shift operations or ε -reductions).

Recap: $LR(0)$ Grammars

$LR(0)$ Conflicts

Definition ($LR(0)$ conflicts)

Let $G = \langle N, \Sigma, P, S \rangle \in CFG_{\Sigma}$ and $I \in LR(0)(G)$.

- I has a **shift/reduce conflict** if there exist $A \rightarrow \alpha_1 a \alpha_2, B \rightarrow \beta \in P$ such that

$$[A \rightarrow \alpha_1 \cdot a \alpha_2], [B \rightarrow \beta \cdot] \in I.$$

- I has a **reduce/reduce conflict** if there exist $A \rightarrow \alpha, B \rightarrow \beta \in P$ with $A \neq B$ or $\alpha \neq \beta$ such that

$$[A \rightarrow \alpha \cdot], [B \rightarrow \beta \cdot] \in I.$$

Lemma

$G \in LR(0)$ iff no $I \in LR(0)(G)$ contains conflicting items.

Proof.

omitted □

Recap: $LR(0)$ Grammars

Computing $LR(0)$ Sets I

Theorem (Computing $LR(0)$ sets)

Let $G = \langle N, \Sigma, P, S \rangle \in CFG_{\Sigma}$ be start separated by $S' \rightarrow S$ and reduced.

1. $LR(0)(\varepsilon)$ is the least set such that

- $[S' \rightarrow \cdot S] \in LR(0)(\varepsilon)$ and
- if $[A \rightarrow \cdot B\gamma] \in LR(0)(\varepsilon)$ and $B \rightarrow \beta \in P$,
then $[B \rightarrow \cdot \beta] \in LR(0)(\varepsilon)$.

2. $LR(0)(\alpha Y)$ ($\alpha \in X^*$, $Y \in X$) is the least set such that

- if $[A \rightarrow \gamma_1 \cdot Y\gamma_2] \in LR(0)(\alpha)$,
then $[A \rightarrow \gamma_1 Y \cdot \gamma_2] \in LR(0)(\alpha Y)$ and
- if $[A \rightarrow \gamma_1 \cdot B\gamma_2] \in LR(0)(\alpha Y)$ and $B \rightarrow \beta \in P$,
then $[B \rightarrow \cdot \beta] \in LR(0)(\alpha Y)$.

LR(0) Parsing Functions

The goto Function I

Observation: if $G \in LR(0)$, then $LR(0)(\gamma)$ yields **deterministic shift/reduce decision** for $NBA(G)$ in a configuration with pushdown γ

\implies **new pushdown alphabet:** $LR(0)(G)$ in place of X

Moreover $LR(0)(\gamma Y)$ is determined by $LR(0)(\gamma)$ and Y but **independent from γ** in the following sense:

$$LR(0)(\gamma) = LR(0)(\gamma') \implies LR(0)(\gamma Y) = LR(0)(\gamma' Y)$$

Definition 9.1 ($LR(0)$ goto function)

The function **goto** : $LR(0)(G) \times X \rightarrow LR(0)(G)$ is determined by

$$\text{goto}(I, Y) = I' \quad \text{iff} \quad \text{there exists } \gamma \in X^* \text{ such that} \\ I = LR(0)(\gamma) \text{ and } I' = LR(0)(\gamma Y).$$

LR(0) Parsing Functions

The goto Function II

Example 9.2 (cf. Example 8.16: $S' \rightarrow S, S \rightarrow B \mid C, B \rightarrow aB \mid b, C \rightarrow aC \mid c$)

$l_0 := LR(0)(\varepsilon) :$	$[S' \rightarrow \cdot S]$	$[S \rightarrow \cdot B]$	$[S \rightarrow \cdot C]$	goto	S	B	C	a	b	c
	$[B \rightarrow \cdot aB]$	$[B \rightarrow \cdot b]$		l_0	l_1	l_2	l_3	l_4	l_5	l_6
	$[C \rightarrow \cdot aC]$	$[C \rightarrow \cdot c]$		l_1						
$l_1 := LR(0)(S) :$	$[S' \rightarrow S \cdot]$			l_2						
$l_2 := LR(0)(B) :$	$[S \rightarrow B \cdot]$			l_3						
$l_3 := LR(0)(C) :$	$[S \rightarrow C \cdot]$			l_4		l_7	l_8	l_4	l_5	l_6
$l_4 := LR(0)(a) :$	$[B \rightarrow a \cdot B]$	$[C \rightarrow a \cdot C]$	$[B \rightarrow \cdot aB]$	l_5						
	$[B \rightarrow \cdot b]$	$[C \rightarrow \cdot aC]$	$[C \rightarrow \cdot c]$	l_6						
$l_5 := LR(0)(b) :$	$[B \rightarrow b \cdot]$			l_7						
$l_6 := LR(0)(c) :$	$[C \rightarrow c \cdot]$			l_8						
$l_7 := LR(0)(aB) :$	$[B \rightarrow aB \cdot]$			l_9						
$l_8 := LR(0)(aC) :$	$[C \rightarrow aC \cdot]$									
$l_9 := \emptyset$										

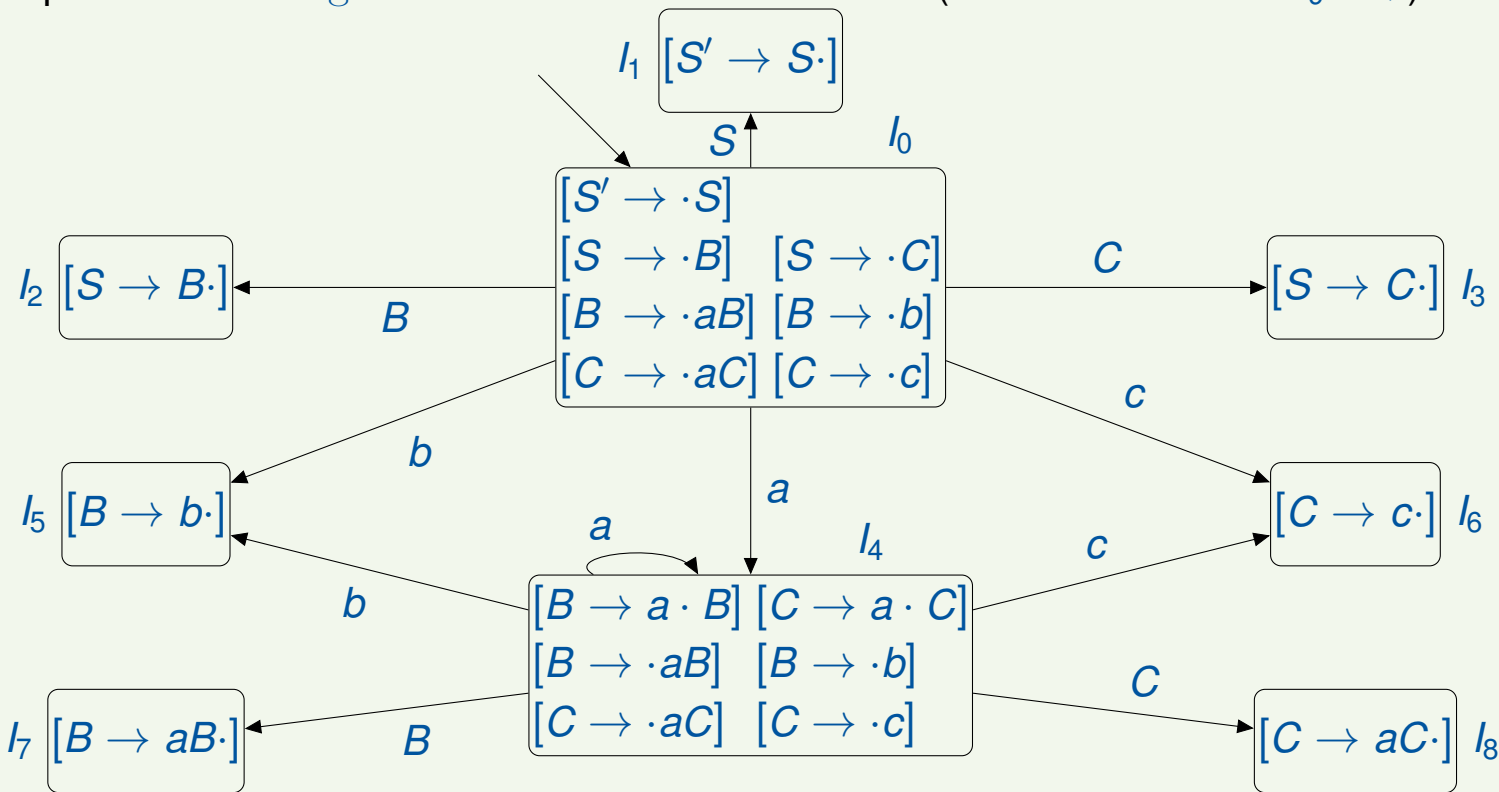
(empty = l_9)

LR(0) Parsing Functions

The goto Function III

Example 9.2 (continued)

Representation of goto function as finite automaton (omitted: sink state $l_9 = \emptyset$):



LR(0) Parsing Functions

The LR(0) Action Function

The parsing automaton will be defined using another table, the **action function**, which determines the shift/reduce decision (reminder: $\pi_0 = S' \rightarrow S$).

Definition 9.3 (LR(0) action function)

The **LR(0) action function** $\text{act} : LR(0)(G) \rightarrow \{\text{red } i \mid i \in [p]\} \cup \{\text{shift, accept, error}\}$ is defined by

$$\text{act}(I) := \begin{cases} \text{red } i & \text{if } i \neq 0, \pi_i = A \rightarrow \alpha \text{ and } [A \rightarrow \alpha \cdot] \in I \\ \text{shift} & \text{if } [A \rightarrow \alpha_1 \cdot a\alpha_2] \in I \\ \text{accept} & \text{if } [S' \rightarrow S \cdot] \in I \\ \text{error} & \text{if } I = \emptyset \end{cases}$$

Corollary 9.4

For every $G \in CFG_{\Sigma}$, $G \in LR(0)$ iff act is well defined.

Together, act and goto form the **LR(0) parsing table** of G .

The LR(0) Parsing Automaton

The LR(0) Parsing Table

Example 9.5 (cf. Example 9.2)

$G: S' \rightarrow S \quad (0)$
 $S \rightarrow B \mid C \quad (1, 2)$
 $B \rightarrow aB \mid b \quad (3, 4)$
 $C \rightarrow aC \mid c \quad (5, 6)$

$LR(0)(G)$	act	goto					
		S	B	C	a	b	c
l_0	shift	l_1	l_2	l_3	l_4	l_5	l_6
l_1	accept						
l_2	red 1						
l_3	red 2						
l_4	shift		l_7	l_8	l_4	l_5	l_6
l_5	red 4						
l_6	red 6						
l_7	red 3						
l_8	red 5						
l_9	error						

(empty = l_9)

$l_0 := LR(0)(\epsilon) : [S' \rightarrow \cdot S]$
 $[S \rightarrow \cdot B] \quad [S \rightarrow \cdot C]$
 $[B \rightarrow \cdot aB] \quad [B \rightarrow \cdot b]$
 $[C \rightarrow \cdot aC] \quad [C \rightarrow \cdot c]$

$l_1 := LR(0)(S) : [S' \rightarrow S \cdot]$

$l_2 := LR(0)(B) : [S \rightarrow B \cdot]$

$l_3 := LR(0)(C) : [S \rightarrow C \cdot]$

$l_4 := LR(0)(a) : [B \rightarrow a \cdot B] \quad [C \rightarrow a \cdot C]$
 $[B \rightarrow \cdot aB] \quad [B \rightarrow \cdot b]$
 $[C \rightarrow \cdot aC] \quad [C \rightarrow \cdot c]$

$l_5 := LR(0)(b) : [B \rightarrow b \cdot]$

$l_6 := LR(0)(c) : [C \rightarrow c \cdot]$

$l_7 := LR(0)(aB) : [B \rightarrow aB \cdot]$

$l_8 := LR(0)(aC) : [C \rightarrow aC \cdot]$

$l_9 := \emptyset$

The $LR(0)$ Parsing Automaton

The $LR(0)$ Parsing Automaton I

Definition 9.6 ($LR(0)$ parsing automaton)

Let $G = \langle N, \Sigma, P, S \rangle \in LR(0)$. The (deterministic) $LR(0)$ parsing automaton of G is defined by the following components:

- Input alphabet Σ
- Pushdown alphabet $\Gamma := LR(0)(G)$
- Output alphabet $\Delta := [\rho] \cup \{0, \text{error}\}$
- Configurations $\Sigma^* \times \Gamma^* \times \Delta^*$
- Initial configuration (w, l_0, ε) where $l_0 := LR(0)(\varepsilon)$
- Final configurations $\{\varepsilon\} \times \{\varepsilon\} \times \Delta^*$
- Transitions:

shift: $(aw, \alpha l, z) \vdash (w, \alpha l', z)$ if $\text{act}(l) = \text{shift}$ and $\text{goto}(l, a) = l'$

reduce: $(w, \alpha l_1 \dots l_n, z) \vdash (w, \alpha l', zi)$ if $\text{act}(l_n) = \text{red } i$, $\pi_i = A \rightarrow Y_1 \dots Y_n$, $\text{goto}(l, A) = l'$

accept: $(\varepsilon, l_0 l, z) \vdash (\varepsilon, \varepsilon, z 0)$ if $\text{act}(l) = \text{accept}$

error: $(w, \alpha l, z) \vdash (\varepsilon, \varepsilon, z \text{error})$ if $\text{act}(l) = \text{error}$

The LR(0) Parsing Automaton

The LR(0) Parsing Automaton II

Example 9.7 (cf. Example 9.5)

$G: S' \rightarrow S \quad (0)$
 $S \rightarrow B \mid C \quad (1, 2)$
 $B \rightarrow aB \mid b \quad (3, 4)$
 $C \rightarrow aC \mid c \quad (5, 6)$

LR(0) parsing of *aac*:

LR(0)(G)	act	goto					
		S	B	C	a	b	c
l_0	shift	l_1	l_2	l_3	l_4	l_5	l_6
l_1	accept						
l_2	red 1						
l_3	red 2						
l_4	shift		l_7	l_8	l_4	l_5	l_6
l_5	red 4						
l_6	red 6						
l_7	red 3						
l_8	red 5						
l_9	error						

(empty = l_9)

(aac, l_0, ε)
 $\vdash (ac, l_0 l_4, \varepsilon)$
 $\vdash (c, l_0 l_4 l_4, \varepsilon)$
 $\vdash (\varepsilon, l_0 l_4 l_4 l_6, \varepsilon)$
 $\vdash (\varepsilon, l_0 l_4 l_4 l_8, 6)$
 (*)
 $\vdash (\varepsilon, l_0 l_4 l_8, 65)$
 $\vdash (\varepsilon, l_0 l_3, 655)$
 $\vdash (\varepsilon, l_0 l_1, 6552)$
 $\vdash (\varepsilon, \varepsilon, 65520)$

Check by rightmost derivation (on the board)

Remark: in the corresponding computation of $NBA(G)$, (*) is nondeterministic (handle C vs. aC)

The $LR(0)$ Parsing Automaton

The $LR(0)$ Parsing Automaton III

Theorem 9.8 (Correctness of $LR(0)$ Parsing Automaton)

If $G \in LR(0)$, then the $LR(0)$ parsing automaton of G is deterministic, and for every $w \in \Sigma^*$ and $z \in \{0, \dots, p\}^*$:

$$(w, l_0, \varepsilon) \vdash^* (\varepsilon, \varepsilon, z) \quad \text{iff} \quad \overleftarrow{z} \text{ is a rightmost analysis of } w$$

Proof.

omitted □

SLR(1) Parsing

Removing Conflicts in LR(0) Parsing

In practice: often $G \notin LR(0)$

Example 9.9

$$G_{AE} : E' \rightarrow E \qquad E \rightarrow E+T \mid T$$
$$T \rightarrow T*F \mid F \qquad F \rightarrow (E) \mid a \mid b$$

LR(0)(G_{AE}) with conflicts:

$$I_0 : [E' \rightarrow \cdot E] \quad [E \rightarrow \cdot E+T] \quad [E \rightarrow \cdot T]$$
$$[T \rightarrow \cdot T*F] \quad [T \rightarrow \cdot F] \quad [F \rightarrow \cdot (E)]$$
$$[F \rightarrow \cdot a] \quad [F \rightarrow \cdot b]$$
$$I_4 : [F \rightarrow (\cdot E)] \quad [E \rightarrow \cdot E+T] \quad [E \rightarrow \cdot T]$$
$$[T \rightarrow \cdot T*F] \quad [T \rightarrow \cdot F] \quad [F \rightarrow \cdot (E)]$$
$$[F \rightarrow \cdot a] \quad [F \rightarrow \cdot b]$$
$$I_8 : [T \rightarrow T* \cdot F] \quad [F \rightarrow \cdot (E)]$$
$$[F \rightarrow \cdot a] \quad [F \rightarrow \cdot b]$$
$$I_{11} : [T \rightarrow T*F \cdot]$$
$$I_1 : [E' \rightarrow E \cdot] \quad [E \rightarrow E \cdot +T]$$
$$I_2 : [E \rightarrow T \cdot] \quad [T \rightarrow T \cdot *F]$$
$$I_3 : [T \rightarrow F \cdot]$$
$$I_5 : [F \rightarrow a \cdot]$$
$$I_6 : [F \rightarrow b \cdot]$$
$$I_7 : [E \rightarrow E+ \cdot T] \quad [T \rightarrow \cdot T*F] \quad [T \rightarrow \cdot F]$$
$$[F \rightarrow \cdot (E)] \quad [F \rightarrow \cdot a] \quad [F \rightarrow \cdot b]$$
$$I_9 : [F \rightarrow (E \cdot)] \quad [E \rightarrow E \cdot +T]$$
$$I_{10} : [E \rightarrow E+T \cdot] \quad [T \rightarrow T \cdot *F]$$
$$I_{12} : [F \rightarrow (E) \cdot]$$

SLR(1) Parsing

Adding Lookahead I

Goal: resolving conflicts by considering next input symbol

Observations:

- $[A \rightarrow \beta_1 \cdot a\beta_2] \in LR(0)(\alpha\beta_1)$
 $\implies \exists \alpha \in X^*, w \in \Sigma^* :$

$$S' \Rightarrow_r^* \alpha A w \Rightarrow_r$$

$\alpha\beta_1 : a\beta_2 w$

↑ ↓
pushdown next input symbol

Thus: shift only on lookahead a

- $[A \rightarrow \beta \cdot] \in LR(0)(\alpha\beta)$
 $\implies \exists \alpha \in X^*, x \in \Sigma_\epsilon, w \in \Sigma^* (x = \epsilon \text{ only if } w = \epsilon):$

$$S' \Rightarrow_r^* \alpha A x w \Rightarrow_r$$

$\alpha\beta : xw$

↑ ↓
pushdown input

$$\implies x \in \text{fo}(A) \subseteq \Sigma_\epsilon$$

Thus: reduce with $A \rightarrow \beta$ only if lookahead $x \in \text{fo}(A)$

SLR(1) Parsing

Adding Lookahead II

Example 9.10 (cf. Example 9.9)

$$\begin{aligned}G_{AE} : E' &\rightarrow E && (0) \\ E &\rightarrow E+T \mid T && (1, 2) \\ T &\rightarrow T*F \mid F && (3, 4) \\ F &\rightarrow (E) \mid a \mid b && (5, 6, 7)\end{aligned}$$

$A \in N$	$\text{fo}(A)$
E'	$\{\varepsilon\}$
E	$\{+,), \varepsilon\}$

- $I_1 = \{[E' \rightarrow E\cdot], [E \rightarrow E\cdot +T]\}$:
 - accept on lookahead ε
 - shift on lookahead $+$
- $I_2 = \{[E \rightarrow T\cdot], [T \rightarrow T\cdot *F]\}$:
 - red 2 on lookahead $+/)/\varepsilon$
 - shift on lookahead $*$
- $I_{10} = \{[E \rightarrow E+T\cdot], [T \rightarrow T\cdot *F]\}$:
 - red 1 on lookahead $+/)/\varepsilon$
 - shift on lookahead $*$

\implies **SLR(1) parsing** (Simple LR(1))

SLR(1) Parsing

The SLR(1) Action Function

Definition 9.11 (SLR(1) action function)

The **SLR(1) action function**

$$\text{act} : LR(0)(G) \times \Sigma_\varepsilon \rightarrow \{\text{red } i \mid i \in [p]\} \cup \{\text{shift, accept, error}\}$$

is defined by

$$\text{act}(I, \mathbf{x}) := \begin{cases} \text{red } i & \text{if } i \neq 0, \pi_i = A \rightarrow \alpha, [A \rightarrow \alpha \cdot] \in I, \\ & \text{and } \mathbf{x} \in \text{fo}(A) \\ \text{shift} & \text{if } [A \rightarrow \alpha_1 \cdot \mathbf{x} \alpha_2] \in I \text{ and } \mathbf{x} \in \Sigma \\ \text{accept} & \text{if } [S' \rightarrow S \cdot] \in I \text{ and } \mathbf{x} = \varepsilon \\ \text{error} & \text{otherwise} \end{cases}$$

Definition 9.12 (SLR(1) grammar)

A grammar $G \in CFG_\Sigma$ has the **SLR(1) property** (notation: $G \in SLR(1)$) if its **SLR(1) action function** is well defined.

act and the **LR(0) goto** function (Definition 9.1) form the **SLR(1) parsing table** of G .

SLR(1) Parsing

The SLR(1) Parsing Table

Example 9.13 (cf. Example 9.9)

l_0 :	$[E' \rightarrow \cdot E]$	$[E \rightarrow \cdot E+T]$	$[E \rightarrow \cdot T]$	l_1 :	$[E' \rightarrow E \cdot]$	$[E \rightarrow E \cdot +T]$
	$[T \rightarrow \cdot T*F]$	$[T \rightarrow \cdot F]$	$[F \rightarrow \cdot (E)]$	l_2 :	$[E \rightarrow T \cdot]$	$[T \rightarrow T \cdot *F]$
	$[F \rightarrow \cdot a]$	$[F \rightarrow \cdot b]$		l_3 :	$[T \rightarrow F \cdot]$	
l_4 :	$[F \rightarrow (\cdot E)]$	$[E \rightarrow \cdot E+T]$	$[E \rightarrow \cdot T]$	l_5 :	$[F \rightarrow a \cdot]$	
	$[T \rightarrow \cdot T*F]$	$[T \rightarrow \cdot F]$	$[F \rightarrow \cdot (E)]$	l_6 :	$[F \rightarrow b \cdot]$	
	$[F \rightarrow \cdot a]$	$[F \rightarrow \cdot b]$		l_7 :	$[E \rightarrow E+ \cdot T]$	$[T \rightarrow \cdot T*F]$ $[T \rightarrow \cdot F]$
l_8 :	$[T \rightarrow T* \cdot F]$	$[F \rightarrow \cdot (E)]$		l_9 :	$[F \rightarrow (E \cdot)]$	$[E \rightarrow E \cdot +T]$
	$[F \rightarrow \cdot a]$	$[F \rightarrow \cdot b]$		l_{10} :	$[E \rightarrow E+T \cdot]$	$[T \rightarrow T \cdot *F]$
l_{11} :	$[T \rightarrow T*F \cdot]$			l_{12} :	$[F \rightarrow (E) \cdot]$	

$A \in N$	$fo(A)$
E'	$\{\epsilon\}$
E	$\{+,), \epsilon\}$
T	$\{+, *,), \epsilon\}$
F	$\{+, *,), \epsilon\}$

$LR(0)(G_{AE})$	act							goto								
	+	*	()	a	b	ϵ	E	T	F	+	*	()	a	b
l_0			shift		shift	shift		l_1	l_2	l_3			l_4		l_5	l_6
l_1	shift						accept									
l_2	red 2	shift		red 2			red 2									
l_3	red 4	red 4		red 4			red 4									
l_4			shift		shift	shift		l_9	l_2	l_3			l_4		l_5	l_6
l_5	red 6	red 6		red 6			red 6									
l_6	red 7	red 7		red 7			red 7									
l_7			shift		shift	shift			l_{10}	l_3			l_4		l_5	l_6
l_8			shift		shift	shift				l_{11}			l_4		l_5	l_6
l_9	shift			shift											l_{12}	
l_{10}	red 1	shift		red 1			red 1									
l_{11}	red 3	red 3		red 3			red 3									
l_{12}	red 5	red 5		red 5			red 5									

The SLR(1) Parsing Automaton

Definition 9.14 (SLR(1) parsing automaton)

The **SLR(1) parsing automaton** is defined as in the **LR(0)** case (see Definition 9.6), except for the **transition relation**:

shift: $(aw, \alpha l, z) \vdash (w, \alpha l', z)$ if $\text{act}(l, a) = \text{shift}$ and $\text{goto}(l, a) = l'$

reduce_a: $(aw, \alpha ll_1 \dots l_n, z) \vdash (aw, \alpha l', zi)$ if $\text{act}(l_n, a) = \text{red } i$, $\pi_i = A \rightarrow Y_1 \dots Y_n$, and $\text{goto}(l, A) = l'$

reduce_ε: $(\varepsilon, \alpha ll_1 \dots l_n, z) \vdash (\varepsilon, \alpha l', zi)$ if $\text{act}(l_n, \varepsilon) = \text{red } i$, $\pi_i = A \rightarrow Y_1 \dots Y_n$, and $\text{goto}(l, A) = l'$

accept: $(\varepsilon, l_0 l, z) \vdash (\varepsilon, \varepsilon, z 0)$ if $\text{act}(l, \varepsilon) = \text{accept}$

error_a: $(aw, \alpha l, z) \vdash (\varepsilon, \varepsilon, z \text{error})$ if $\text{act}(l, a) = \text{error}$

error_ε: $(\varepsilon, \alpha l, z) \vdash (\varepsilon, \varepsilon, z \text{error})$ if $\text{act}(l, \varepsilon) = \text{error}$

Examples of $SLR(1)$ Conflicts

$SLR(1)$ Conflicts

Problem: not all conflicts can be resolved using fo sets

Example 9.15

$G_{LR} : S' \rightarrow S \quad S \rightarrow L=R \mid R \quad L \rightarrow *R \mid a \quad R \rightarrow L$

$LR(0)(G_{LR}) : I_0 := LR(0)(\varepsilon) : \quad [S' \rightarrow \cdot S] \quad [S \rightarrow \cdot L=R] \quad [S \rightarrow \cdot R]$
 $\quad [L \rightarrow \cdot *R] \quad [L \rightarrow \cdot a] \quad [R \rightarrow \cdot L]$

$I_1 := LR(0)(S) : \quad [S' \rightarrow S \cdot]$

$I_2 := LR(0)(L) : \quad [S \rightarrow L \cdot =R] \quad [R \rightarrow L \cdot]$

$I_3 := LR(0)(R) : \quad [S \rightarrow R \cdot]$

$I_4 := LR(0)(*) : \quad [L \rightarrow * \cdot R] \quad [R \rightarrow \cdot L] \quad [L \rightarrow \cdot *R] \quad [L \rightarrow \cdot a]$

$I_5 := LR(0)(a) : \quad [L \rightarrow a \cdot]$

$I_6 := LR(0)(L=) : \quad [S \rightarrow L= \cdot R] \quad [R \rightarrow \cdot L] \quad [L \rightarrow \cdot *R] \quad [L \rightarrow \cdot a]$

$I_7 := LR(0)(*R) : \quad [L \rightarrow *R \cdot]$

$I_8 := LR(0)(*L) : \quad [R \rightarrow L \cdot]$

$I_9 := LR(0)(L=R) : \quad [S \rightarrow L=R \cdot]$

But: conflict in I_2 not $SLR(1)$ -solvable since $= \in fo(R)$