Compiler Construction

Lecture 10: Syntactic Analysis VI ($LR(0)$ Parsing)

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)

RWTH Aachen University

noll@cs.rwth-aachen.de

http://www-i2.informatik.rwth-aachen.de/i2/cc09/

Winter semester 2009/10

Outline

Nondeterministic Bottom-Up Automaton

Definition (Nondeterministic bottom-up parsing automaton)

Let $G = \langle N, \Sigma, P, S \rangle \in CFG_\Sigma$. The nondeterministic bottom-up parsing automaton of G, $\mathrm{NBA}(G)$, is defined by the following components.

- Input alphabet: Σ
- Pushdown alphabet: X
- Output alphabet: $[p]$
- Configurations: $\Sigma^* \times X^* \times [p]^*$ (top of pushdown to the right)
- Transitions for $w \in \Sigma^*$, $\alpha \in X^*$, and $z \in [p]^*$:
  shifting steps: $(aw, \alpha, z) \vdash (w, \alpha a, z)$ if $a \in \Sigma$
  reduction steps: $(w, \alpha\beta, z) \vdash (w, \alpha A, zi)$ if $\pi(i) = A \to \beta$
- Initial configuration for $w \in \Sigma^*$: $(w, \varepsilon, \varepsilon)$
- Final configurations: $\{\varepsilon\} \times \{S\} \times [p]^*$

Nondeterminisn in NBA(G)

Remark: NBA(G) is generally nondeterministic

- Shift or reduce? Example:

$$(bw, \alpha a, z) \vdash \begin{cases} (w, \alpha ab, z) \\ (bw, \alpha A, zi) \end{cases} \text{ if } \pi(i) = A \rightarrow a$$

- If reduce: which "handle" β? Example:

$$(w, \alpha ab, z) \vdash \begin{cases} (w, \alpha A, zi) \\ (w, \alpha aB, zj) \end{cases} \text{ if } \pi(i) = A \rightarrow ab \text{ and } \pi(j) = B \rightarrow b$$

- If reduce β: which left-hand side A? Example:

$$(w, \alpha a, z) \vdash \begin{cases} (w, \alpha A, zi) \\ (w, \alpha B, zj) \end{cases} \text{ if } \pi(i) = A \rightarrow a \text{ and } \pi(j) = B \rightarrow a$$

- When to terminate parsing? Example:

$$\underbrace{(\varepsilon, S, z)}_{\text{final}} \vdash (\varepsilon, A, zi) \text{ if } \pi(i) = A \rightarrow S$$

$LR(k)$ Grammars

Goal: resolve remaining nondeterminism of NBA(G) by supporting lookahead of $k \in \mathbb{N}$ symbols on the input

$\implies$ $LR(k)$: reading of input from left to right with k-lookahead, computing a rightmost analysis

Definition ($LR(k)$ grammar)

Let $G = \langle N, \Sigma, P, S\rangle \in CFG_\Sigma$ be start separated and $k \in \mathbb{N}$. Then G has the $LR(k)$ property (notation: $G \in LR(k)$) if for all rightmost derivations of the form

$$S \begin{cases} \Rightarrow_r^* & \alpha A w & \Rightarrow_r \alpha\beta w \\ \Rightarrow_r^* & \alpha' A' w' & \Rightarrow_r \alpha\beta v \end{cases}$$

such that $\text{first}_k(w) = \text{first}_k(v)$, it follows that $\alpha' = \alpha$, $A' = A$, and $w' = v$.

Remarks:

- If $G \in LR(k)$, then the reduction of $\alpha\beta w$ to $\alpha A w$ is already determined by $\alpha\beta\,\text{first}_k(w)$.
- Therefore NBA(G) in configuration $(w, \alpha\beta, z)$ can decide whether to shift or to reduce and, in the second case, how to reduce.

$LR(0)$ Items and Sets

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

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

- $[A \to \beta_1 \cdot \beta_2]$ 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.
- $LR(0)(G) := \{LR(0)(\gamma) \mid \gamma \in X^*\}$.

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 $A \to \alpha_1 a \alpha_2, B \to \beta \in P$ such that
$$[A \to \alpha_1 \cdot a\alpha_2], [B \to \beta \cdot] \in I.$$
- I has a reduce/reduce conflict if there exist $A \to \alpha, B \to \beta \in P$ with $A \neq B$ or $\alpha \neq \beta$ such that
$$[A \to \alpha \cdot], [B \to \beta \cdot] \in I.$$

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 $S' \to S$ and reduced.

1. $LR(0)(\varepsilon)$ *is the least set such that*
   - $[S' \to \cdot S] \in LR(0)(\varepsilon)$ *and*
   - *if* $[A \to \cdot B\gamma] \in LR(0)(\varepsilon)$ *and* $B \to \beta \in P$, *then* $[B \to \cdot\beta] \in LR(0)(\varepsilon)$.
2. $LR(0)(\alpha Y)$ *(*$\alpha \in X^*, Y \in X$*) is the least set such that*
   - *if* $[A \to \gamma_1 \cdot Y\gamma_2] \in LR(0)(\alpha)$, *then* $[A \to \gamma_1 Y \cdot \gamma_2] \in LR(0)(\alpha Y)$ *and*
   - *if* $[A \to \gamma_1 \cdot B\gamma_2] \in LR(0)(\alpha Y)$ *and* $B \to \beta \in P$, *then* $[B \to \cdot\beta] \in LR(0)(\alpha Y)$.

Computing $LR(0)$ Sets II

Example

$$
\begin{aligned}
G:\quad S' &\to S\\
S &\to B \mid C\\
B &\to aB \mid b\\
C &\to aC \mid c
\end{aligned}
$$

$$
\begin{array}{lll}
I_0 := LR(0)(\varepsilon): & [S' \to \cdot S] & \\
 & [S \to \cdot B] & [S \to \cdot C]\\
 & [B \to \cdot aB] & [B \to \cdot b]\\
 & [C \to \cdot aC] & [C \to \cdot c]\\
I_1 := LR(0)(S): & [S' \to S\cdot] & \\
I_2 := LR(0)(B): & [S \to B\cdot] & \\
I_3 := LR(0)(C): & [S \to C\cdot] & \\
I_4 := LR(0)(a): & [B \to a \cdot B] & [C \to a \cdot C]\\
 & [B \to \cdot aB] & [B \to \cdot b]\\
 & [C \to \cdot aC] & [C \to \cdot c]\\
I_5 := LR(0)(b): & [B \to b\cdot] & \\
I_6 := LR(0)(c): & [C \to c\cdot] & \\
I_7 := LR(0)(aB): & [B \to aB\cdot] & \\
I_8 := LR(0)(aC): & [C \to aC\cdot] & \\
I_9 := \emptyset & &
\end{array}
$$

no conflicts $\Longrightarrow$ $G \in LR(0)$

Outline

1. Repetition: Bottom-Up Parsing
2. $LR(0)$ Parsing
3. The $LR(0)$ Parsing Automaton

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
$\Longrightarrow$ 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 10.1 ($LR(0)$ goto function)

The function $\text{goto} : LR(0)(G) \times X \to LR(0)(G)$ is determined by

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

Example 10.2 (cf. Example 9.14)

$$
\begin{array}{lll}
I_0 := LR(0)(\varepsilon): & [S' \to \cdot S] & \\
 & [S \to \cdot B] & [S \to \cdot C] \\
 & [B \to \cdot aB] & [B \to \cdot b] \\
 & [C \to \cdot aC] & [C \to \cdot c] \\
I_1 := LR(0)(S): & [S' \to S\cdot] & \\
I_2 := LR(0)(B): & [S \to B\cdot] & \\
I_3 := LR(0)(C): & [S \to C\cdot] & \\
I_4 := LR(0)(a): & [B \to a \cdot B] & [C \to a \cdot C] \\
 & [B \to \cdot aB] & [B \to \cdot b] \\
 & [C \to \cdot aC] & [C \to \cdot c] \\
I_5 := LR(0)(b): & [B \to b\cdot] & \\
I_6 := LR(0)(c): & [C \to c\cdot] & \\
I_7 := LR(0)(aB): & [B \to aB\cdot] & \\
I_8 := LR(0)(aC): & [C \to aC\cdot] & \\
I_9 := \emptyset & &
\end{array}
$$

goto	S	B	C	a	b	c
I_0	I_1	I_2	I_3	I_4	I_5	I_6
I_1						
I_2						
I_3						
I_4		I_7	I_8	I_4	I_5	I_6
I_5						
I_6						
I_7						
I_8						
I_9						

(empty $= I_9$)

Computing $LR(0)(G)$ and goto I

Goal: obtaining $LR(0)(G)$ and goto by powerset construction

Given $G = \langle N, \Sigma, P, S\rangle \in CFG_\Sigma$ (start separated by $S' \to S$ and reduced), the nondeterministic finite automaton $\mathfrak{A}(G) := \langle Q, X, \delta, q_0, F\rangle \in NFA_X$ is defined as follows.

- State set $Q := \{[A \to \beta_1 \cdot \beta_2] \mid A \to \beta_1\beta_2 \in P\}$ ($LR(0)$ items of G)
- Input alphabet $X = N \cup \Sigma$
- Transition function $\delta : Q \times X_\varepsilon \to 2^Q$ where
  - $\delta([A \to \beta_1 \cdot Y\beta_2], Y) \ni [A \to \beta_1 Y \cdot \beta_2]$
  - $\delta([A \to \beta_1 \cdot B\beta_2], \varepsilon) \ni [B \to \cdot\beta]$ if $B \to \beta \in P$
- Initial state $q_0 := [S' \to \cdot S] \in Q$
- Final states $F := \emptyset$ (irrelevant)

Example 10.3 (cf. Example 10.2)

$\mathfrak{A}(G)$ for G :

$$\begin{aligned} S' &\to S \\ S &\to B \mid C \\ B &\to aB \mid b \\ C &\to aC \mid c \end{aligned}$$

(on the board)

Computing $LR(0)(G)$ and goto II

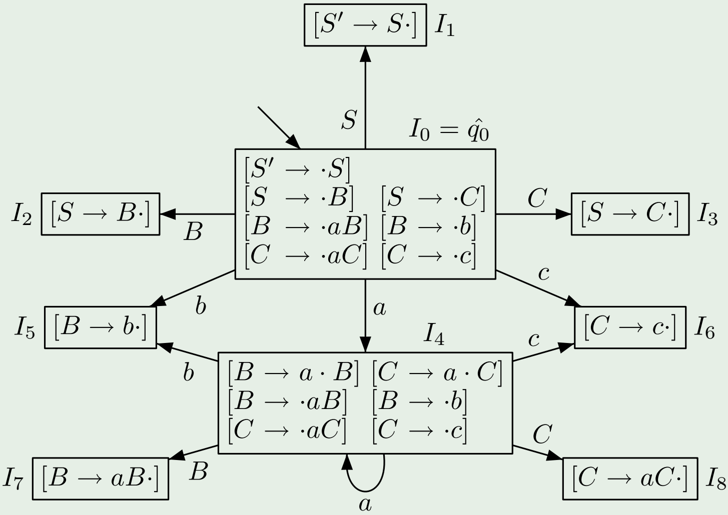
Transformation of $\mathfrak{A}(G)$ into $\widehat{\mathfrak{A}(G)} := \langle \hat{Q}, X, \hat{\delta}, \hat{q_0}, \emptyset \rangle \in DFA_X$ by powerset construction:

- $\hat{Q} := 2^Q$ (only reachable subsets)
- $\hat{q_0} := \varepsilon([S' \to \cdot S])$
- $\hat{\delta} : \hat{Q} \times X \to \hat{Q} : (T, Y) \mapsto \varepsilon\left(\bigcup_{q \in T} \delta(q, Y)\right)$

Lemma 10.4

If $\widehat{\mathfrak{A}(G)}$ is constructed as above, then $\hat{Q} = LR(0)(G)$ and $\hat{\delta} = \mathrm{goto}$.

Example 10.5 (cf. Example 10.3)



(omitted: sink state $I_9 = \emptyset$)

Outline

1. Repetition: Bottom-Up Parsing
2. $LR(0)$ Parsing
3. The $LR(0)$ Parsing Automaton

The $LR(0)$ Action Function

The automaton will be defined using another table, the action function, which determines the shift/reduce decision.
(Reminder: $\pi(0) = S' \to S$)

Definition 10.6 ($LR(0)$ action function)

The $LR(0)$ action function

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

is defined by

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

Corollary 10.7

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.

Example 10.8 (cf. Example 10.2)

$$
\begin{array}{llll}
G: & S' & \to S & (0)\\
 & S & \to B \mid C & (1,2)\\
 & B & \to aB \mid b & (3,4)\\
 & C & \to aC \mid c & (5,6)
\end{array}
$$

$LR(0)(G)$	act	goto					
		S	B	C	a	b	c
I_0	shift	I_1	I_2	I_3	I_4	I_5	I_6
I_1	accept						
I_2	red 1						
I_3	red 2						
I_4	shift		I_7	I_8	I_4	I_5	I_6
I_5	red 4						
I_6	red 6						
I_7	red 3						
I_8	red 5						
I_9	error						

(empty $= I_9$)

$$
\begin{array}{lll}
I_0 := LR(0)(\varepsilon): & [S' \to \cdot S] & \\
 & [S \to \cdot B] & [S \to \cdot C]\\
 & [B \to \cdot aB] & [B \to \cdot b]\\
 & [C \to \cdot aC] & [C \to \cdot c]\\
I_1 := LR(0)(S): & [S' \to S \cdot] & \\
I_2 := LR(0)(B): & [S \to B \cdot] & \\
I_3 := LR(0)(C): & [S \to C \cdot] & \\
I_4 := LR(0)(a): & [B \to a \cdot B] & [C \to a \cdot C]\\
 & [B \to \cdot aB] & [B \to \cdot b]\\
 & [C \to \cdot aC] & [C \to \cdot c]\\
I_5 := LR(0)(b): & [B \to b \cdot] & \\
I_6 := LR(0)(c): & [C \to c \cdot] & \\
I_7 := LR(0)(aB): & [B \to aB \cdot] & \\
I_8 := LR(0)(aC): & [C \to aC \cdot] & \\
I_9 := \emptyset & &
\end{array}
$$

Definition 10.9 ($LR(0)$ 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 $\Delta := [p] \cup \{0, \mathsf{error}\}$
- Configurations $\Sigma^* \times \Gamma^* \times \Delta^*$
- Initial configuration (w, I_0, ε) where $I_0 := LR(0)(\varepsilon)$
- Final configurations $\{\varepsilon\} \times \{\varepsilon\} \times \Delta^*$
- Transitions:

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

reduce: $(w, \alpha I I_1 \ldots I_n, z) \vdash (w, \alpha I I', zi)$ if $\mathrm{act}(I_n) = \mathsf{red}\, i$, $\pi(i) = A \to Y_1 \ldots Y_n$, and $\mathrm{goto}(I, A) = I'$

accept: $(\varepsilon, I_0 I, z) \vdash (\varepsilon, \varepsilon, z\,0)$ if $\mathrm{act}(I) = \mathsf{accept}$

error: $(w, \alpha I, z) \vdash (\varepsilon, \varepsilon, z\,\mathsf{error})$ if $\mathrm{act}(I) = \mathsf{error}$

Example 10.10 (cf. Example 10.8)

$$
\begin{array}{llll}
G: & S' & \to S & (0) \\
 & S & \to B \mid C & (1,2) \\
 & B & \to aB \mid b & (3,4) \\
 & C & \to aC \mid c & (5,6)
\end{array}
$$

$LR(0)(G)$	act	goto S	B	C	a	b	c
I_0	shift	I_1	I_2	I_3	I_4	I_5	I_6
I_1	accept						
I_2	red 1						
I_3	red 2						
I_4	shift		I_7	I_8	I_4	I_5	I_6
I_5	red 4						
I_6	red 6						
I_7	red 3						
I_8	red 5						
I_9	error						

(empty $= I_9$)

$LR(0)$ parsing of aac:

$$
\begin{array}{lll}
 & (aac, I_0 &, \varepsilon) \\
\vdash & (ac, I_0 I_4 &, \varepsilon) \\
\vdash & (c, I_0 I_4 I_4 &, \varepsilon) \\
\vdash & (\varepsilon, I_0 I_4 I_4 I_6 &, \varepsilon) \\
\vdash & (\varepsilon, I_0 I_4 I_4 I_8 &, 6) \\
\vdash & (\varepsilon, I_0 I_4 I_8 &, 65) \\
\vdash & (\varepsilon, I_0 I_3 &, 655) \\
\vdash & (\varepsilon, I_0 I_1 &, 6552) \\
\vdash & (\varepsilon, \varepsilon &, 65520)
\end{array}
$$

Check by rightmost derivation (on the board)

The $LR(0)$ Parsing Automaton III

Theorem 10.11 (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 $z \in \{0, \ldots, p\}^*$:

$$(w, I_0, \varepsilon) \vdash^* (\varepsilon, \varepsilon, z) \quad \textit{iff} \quad \overleftarrow{z} \textit{ is a rightmost analysis of } w$$

Proof.

omitted □