

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/cc08/`

Summer semester 2008

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

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 , $NBA(G)$, 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 $\pi(i) = A \rightarrow \beta$
- **Initial configuration** for $w \in \Sigma^*$: $(w, \varepsilon, \varepsilon)$
- **Final configurations:** $\{\varepsilon\} \times \{S\} \times [p]^*$

Nondeterminism in NTA(G)

Remark: NTA(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:

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

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 $w' = v$.

Corollary ($LR(0)$ grammar)

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

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

it follows that $\alpha' = \alpha$, $A' = A$, and $w' = v$.

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 .
- $LR(0)(G) := \{LR(0)(\gamma) \mid \gamma \in X^*\}$.

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

Computing $LR(0)$ Sets I

Lemma

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

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 $[B \rightarrow \cdot \beta] \in LR(0)(\alpha Y)$.

Example

$G :$ $S' \rightarrow S$
 $S \rightarrow B \mid C$
 $B \rightarrow aB \mid b$
 $C \rightarrow aC \mid c$

$I_0 := LR(0)(\varepsilon) :$ $[S' \rightarrow \cdot S]$
 $[S \rightarrow \cdot B]$ $[S \rightarrow \cdot C]$
 $[B \rightarrow \cdot aB]$ $[B \rightarrow \cdot b]$
 $[C \rightarrow \cdot aC]$ $[C \rightarrow \cdot c]$

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

$I_2 := LR(0)(B) :$ $[S \rightarrow B \cdot]$

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

$I_4 := LR(0)(a) :$ $[B \rightarrow a \cdot B]$ $[C \rightarrow a \cdot C]$
 $[B \rightarrow \cdot aB]$ $[B \rightarrow \cdot b]$
 $[C \rightarrow \cdot aC]$ $[C \rightarrow \cdot c]$

$I_5 := LR(0)(b) :$ $[B \rightarrow b \cdot]$

$I_6 := LR(0)(c) :$ $[C \rightarrow c \cdot]$

$I_7 := LR(0)(aB) :$ $[B \rightarrow aB \cdot]$

$I_8 := LR(0)(aC) :$ $[C \rightarrow aC \cdot]$

$I_9 := \emptyset$

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

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

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:

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

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 10.1 ($LR(0)$ goto function)

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

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

The goto Function II

Example 10.2 (cf. Example 9.15)

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

The goto Function II

Example 10.2 (cf. Example 9.15)

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

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**

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

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

Example 10.3 (cf. Example 10.2)

$\mathfrak{A}(G)$ for $G : \begin{array}{l} S' \rightarrow S \\ S \rightarrow B \mid C \\ B \rightarrow aB \mid b \\ C \rightarrow aC \mid c \end{array}$ (on the board)

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)
- $\hat{q}_0 := \varepsilon([S' \rightarrow \cdot S])$
- $\hat{\delta} : \hat{Q} \times X \rightarrow \hat{Q} : (T, Y) \mapsto \varepsilon\left(\bigcup_{q \in T} \delta(q, Y)\right)$

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:

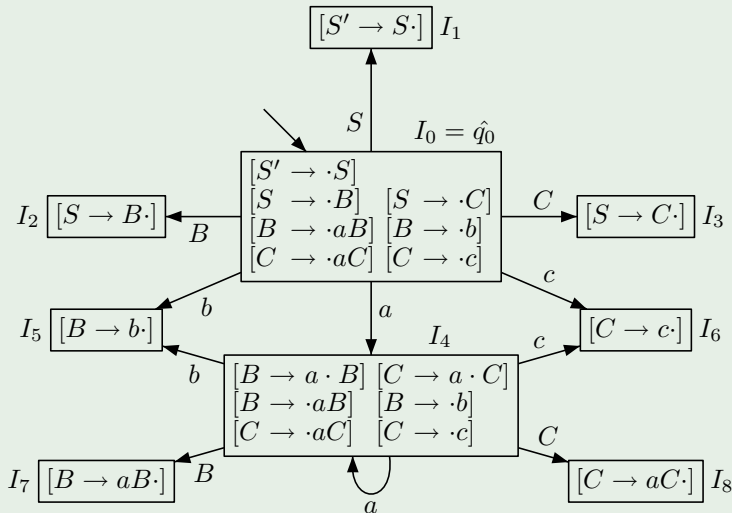
- $\hat{Q} := 2^Q$ (only reachable subsets)
- $\hat{q}_0 := \varepsilon([S' \rightarrow \cdot S])$
- $\hat{\delta} : \hat{Q} \times X \rightarrow \hat{Q} : (T, Y) \mapsto \varepsilon\left(\bigcup_{q \in T} \delta(q, Y)\right)$

Lemma 10.4

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

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

Example 10.5 (cf. Example 10.3)



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

- 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: $\pi(0) = S' \rightarrow S$)

The $LR(0)$ Action Function

The automaton will be defined using another table, the **action function**, which determines the shift/reduce decision.

(Reminder: $\pi(0) = S' \rightarrow S$)

Definition 10.6 ($LR(0)$ 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

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

The $LR(0)$ Action Function

The automaton will be defined using another table, the **action function**, which determines the shift/reduce decision.

(Reminder: $\pi(0) = S' \rightarrow S$)

Definition 10.6 ($LR(0)$ 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

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

Corollary 10.7

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

The $LR(0)$ Action Function

The automaton will be defined using another table, the **action function**, which determines the shift/reduce decision.

(Reminder: $\pi(0) = S' \rightarrow S$)

Definition 10.6 ($LR(0)$ 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

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

Corollary 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 .

The $LR(0)$ Parsing Table

Example 10.8 (cf. Example 10.2)

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

$I_0 := LR(0)(\varepsilon) : \quad \begin{array}{l} [S' \rightarrow \cdot S] \\ [S \rightarrow \cdot B] \quad [S \rightarrow \cdot C] \\ [B \rightarrow \cdot aB] \quad [B \rightarrow \cdot b] \\ [C \rightarrow \cdot aC] \quad [C \rightarrow \cdot c] \end{array}$

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

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

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

$I_4 := LR(0)(a) : \quad \begin{array}{l} [B \rightarrow a \cdot B] \quad [C \rightarrow a \cdot C] \\ [B \rightarrow \cdot aB] \quad [B \rightarrow \cdot b] \\ [C \rightarrow \cdot aC] \quad [C \rightarrow \cdot c] \end{array}$

$I_5 := LR(0)(b) : \quad [B \rightarrow b \cdot]$

$I_6 := LR(0)(c) : \quad [C \rightarrow c \cdot]$

$I_7 := LR(0)(aB) : \quad [B \rightarrow aB \cdot]$

$I_8 := LR(0)(aC) : \quad [C \rightarrow aC \cdot]$

$I_9 := \emptyset$

The $LR(0)$ Parsing Table

Example 10.8 (cf. Example 10.2)

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

$LR(0)(G)$	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)

$I_0 := LR(0)(\varepsilon) : \quad [S' \rightarrow \cdot S]$
 $\quad [S \rightarrow \cdot B] \quad [S \rightarrow \cdot C]$
 $\quad [B \rightarrow \cdot aB] \quad [B \rightarrow \cdot b]$
 $\quad [C \rightarrow \cdot aC] \quad [C \rightarrow \cdot c]$

$I_1 := LR(0)(S) : \quad [S' \rightarrow S \cdot]$
 $I_2 := LR(0)(B) : \quad [S \rightarrow B \cdot]$
 $I_3 := LR(0)(C) : \quad [S \rightarrow C \cdot]$
 $I_4 := LR(0)(a) : \quad [B \rightarrow a \cdot B] \quad [C \rightarrow a \cdot C]$
 $\quad [B \rightarrow \cdot aB] \quad [B \rightarrow \cdot b]$
 $\quad [C \rightarrow \cdot aC] \quad [C \rightarrow \cdot c]$

$I_5 := LR(0)(b) : \quad [B \rightarrow b \cdot]$
 $I_6 := LR(0)(c) : \quad [C \rightarrow c \cdot]$
 $I_7 := LR(0)(aB) : \quad [B \rightarrow aB \cdot]$
 $I_8 := LR(0)(aC) : \quad [C \rightarrow aC \cdot]$
 $I_9 := \emptyset$

The $LR(0)$ Parsing Automaton I

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 $\Delta := [p] \cup \{0, \text{error}\}$
- 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: $(aw, \alpha I, z) \vdash (w, \alpha II', z)$ if $\text{act}(I) = \text{shift}$ and $\text{goto}(I, a) = I'$

reduce: $(w, \alpha II_1 \dots I_n, z) \vdash (w, \alpha II', zi)$ if $\text{act}(I_n) = \text{red } i$,
 $\pi(i) = A \rightarrow Y_1 \dots Y_n$, and $\text{goto}(I, A) = I'$

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

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

The $LR(0)$ Parsing Automaton II

Example 10.10 (cf. Example 10.8)

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

$LR(0)(G)$	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)

The $LR(0)$ Parsing Automaton II

Example 10.10 (cf. Example 10.8)

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

$LR(0)$ parsing of aac :

(aac, I_0, ε)

$LR(0)(G)$	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)

The $LR(0)$ Parsing Automaton II

Example 10.10 (cf. Example 10.8)

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

$LR(0)$ parsing of aac :

$$\begin{aligned}
 &(aac, I_0, \varepsilon) \\
 \vdash & (ac, I_0 I_4, \varepsilon)
 \end{aligned}$$

$LR(0)(G)$	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)

The $LR(0)$ Parsing Automaton II

Example 10.10 (cf. Example 10.8)

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

$LR(0)$ parsing of aac :

(aac, I_0, ε)
 $\vdash (ac, I_0 I_4, \varepsilon)$
 $\vdash (c, I_0 I_4 I_6, \varepsilon)$

$LR(0)(G)$	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)

The $LR(0)$ Parsing Automaton II

Example 10.10 (cf. Example 10.8)

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

$LR(0)$ parsing of aac :

(aac, I_0, ε)
 $\vdash (ac, I_0I_4, \varepsilon)$
 $\vdash (c, I_0I_4I_4, \varepsilon)$
 $\vdash (\varepsilon, I_0I_4I_4I_6, \varepsilon)$

$LR(0)(G)$	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)

The $LR(0)$ Parsing Automaton II

Example 10.10 (cf. Example 10.8)

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

$LR(0)$ parsing of aac :

(aac, I_0, ε)
 $\vdash (ac, I_0 I_4, \varepsilon)$
 $\vdash (c, I_0 I_4 I_4, \varepsilon)$
 $\vdash (\varepsilon, I_0 I_4 I_4 I_6, \varepsilon)$
 $\vdash (\varepsilon, I_0 I_4 I_4 I_8, 6)$

$LR(0)(G)$	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)

The $LR(0)$ Parsing Automaton II

Example 10.10 (cf. Example 10.8)

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

$LR(0)$ parsing of aac :

$LR(0)(G)$	act	goto					
		S	B	C	a	b	c
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						

(aac, I_0, ε)
 $\vdash (ac, I_0I_4, \varepsilon)$
 $\vdash (c, I_0I_4I_4, \varepsilon)$
 $\vdash (\varepsilon, I_0I_4I_4I_6, \varepsilon)$
 $\vdash (\varepsilon, I_0I_4I_4I_8, 6)$
 $\vdash (\varepsilon, I_0I_4I_8, 65)$

(empty = I_9)

The $LR(0)$ Parsing Automaton II

Example 10.10 (cf. Example 10.8)

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

$LR(0)$ parsing of aac :

$LR(0)(G)$	act	goto					
		S	B	C	a	b	c
I_0	shift	$\textcolor{red}{I}_1$	I_2	I_3	I_4	I_5	I_6
I_1	accept						
I_2	red 1						
I_3	$\textcolor{red}{red 2}$						
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						

(aac, I_0, ε)
 $\vdash (ac, I_0 I_4, \varepsilon)$
 $\vdash (c, I_0 I_4 I_4, \varepsilon)$
 $\vdash (\varepsilon, I_0 I_4 I_4 I_6, \varepsilon)$
 $\vdash (\varepsilon, I_0 I_4 I_4 I_8, 6)$
 $\vdash (\varepsilon, I_0 I_4 I_8, 65)$
 $\vdash (\varepsilon, I_0 \textcolor{red}{I}_3, 655)$

(empty = I_9)

The $LR(0)$ Parsing Automaton II

Example 10.10 (cf. Example 10.8)

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

$LR(0)(G)$	act	goto					
		S	B	C	a	b	c
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 :

(aac, I_0, ε)
 $\vdash (ac, I_0I_4, \varepsilon)$
 $\vdash (c, I_0I_4I_4, \varepsilon)$
 $\vdash (\varepsilon, I_0I_4I_4I_6, \varepsilon)$
 $\vdash (\varepsilon, I_0I_4I_4I_8, 6)$
 $\vdash (\varepsilon, I_0I_4I_8, 65)$
 $\vdash (\varepsilon, I_0I_3, 655)$
 $\vdash (\varepsilon, I_0I_1, 6552)$

The $LR(0)$ Parsing Automaton II

Example 10.10 (cf. Example 10.8)

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

$LR(0)(G)$	act	goto					
		S	B	C	a	b	c
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 :

(aac, I_0, ε)
 $\vdash (ac, I_0I_4, \varepsilon)$
 $\vdash (c, I_0I_4I_4, \varepsilon)$
 $\vdash (\varepsilon, I_0I_4I_4I_6, \varepsilon)$
 $\vdash (\varepsilon, I_0I_4I_4I_8, 6)$
 $\vdash (\varepsilon, I_0I_4I_8, 65)$
 $\vdash (\varepsilon, I_0I_3, 655)$
 $\vdash (\varepsilon, I_0I_1, 6552)$
 $\vdash (\varepsilon, \varepsilon, 65520)$

The $LR(0)$ Parsing Automaton II

Example 10.10 (cf. Example 10.8)

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

$LR(0)(G)$	act	goto					
		S	B	C	a	b	c
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 :

(aac, I_0, ε)
 $\vdash (ac, I_0I_4, \varepsilon)$
 $\vdash (c, I_0I_4I_4, \varepsilon)$
 $\vdash (\varepsilon, I_0I_4I_4I_6, \varepsilon)$
 $\vdash (\varepsilon, I_0I_4I_4I_8, 6)$
 $\vdash (\varepsilon, I_0I_4I_8, 65)$
 $\vdash (\varepsilon, I_0I_3, 655)$
 $\vdash (\varepsilon, I_0I_1, 6552)$
 $\vdash (\varepsilon, \varepsilon, 65520)$

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 $z \in \{0, \dots, p\}^*$:*

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

Proof.

omitted □