

8. LL(k) Parsing

Canonical LL(k) parser

The most general deterministic parsing method in which the input string is parsed

- (1) in a single Left-to-right scan,
- (2) producing a Left parse, and
- (3) using lookahead of length k.

Canonical LL(k) parser

dual of the canonical LR(k) parser

generalization of strong LL(k) parser

Generalization of

shift-reduce parser

produce-shift parser

Stack symbols

valid stack string

valid stack string

viable prefix

viable suffix

LR(k) equivalence

LL(k) equivalence

8.1 Viable Suffixes

produce-shift parser

adding lookahead

adding lookback

SLL(k) parser

LL(k) parser

G_{abL} :

$$S \rightarrow aAab \mid bAb$$

$$A \rightarrow cAB \mid \varepsilon \mid a$$

$$B \rightarrow \varepsilon$$

$$L(G_{abL}) = a\{c\}^*\{\varepsilon, a\}ab \cup b\{c\}^*\{\varepsilon, a\}b.$$

G_{abL} is not SLL(k) for any k.

$$First_k(\varepsilon Follow_k(A)) = \{k:ab, k:b\}$$

$$First_k(a Follow_k(A)) = \{k:aab, k:ab\}$$

$$\begin{aligned} First_k(\varepsilon Follow_k(A)) \cap First_k(a Follow_k(A)) \\ = \{k:ab\} \end{aligned}$$

\therefore produce-produce conflict for $A \rightarrow a$ and $A \rightarrow \varepsilon$.

adding lookahead

$$A \mid ab \Rightarrow^{A \rightarrow a} a \mid ab \quad A \rightarrow a$$

$$\Rightarrow^{A \rightarrow \varepsilon} \mid ab \quad A \rightarrow \varepsilon.$$

G_{abL} is not SLL(k) for any k.

adding lookahead and lookback

$$bB^mA \mid ab \Rightarrow^{A \rightarrow a} bB^ma \mid ab \quad A \rightarrow a$$

$$baB^mA \mid ab \Rightarrow^{A \rightarrow \varepsilon} baB^m \mid ab \quad A \rightarrow \varepsilon.$$

G_{abL} is LL(2)

A string γ is a **viable stack string** of pda M , if

$$\$ \gamma_s \mid w \$ \Rightarrow^* \$ \gamma \mid y \$ \Rightarrow^* \$ \gamma_f \mid \$ \text{ in } M.$$

stack string in some accepting computation M .

$$G_{abL} : S \rightarrow aAab \mid bAb$$

$$A \rightarrow cAB \mid \varepsilon \mid a$$

$$B \rightarrow \varepsilon.$$

Viable stack string of G_{abL} (in regular exp)

$$\begin{aligned} & \{S\} \cup \{baAa\} \cup \{baA\} \cup \{ba\} \cup \{b\} \cup \{\varepsilon\} \\ & \cup \{bAb\} \cup \{bA\} \\ & \cup \{baB^nAc \mid n \geq 1\} \cup \{bB^nAc \mid n \geq 1\} \\ & \cup \{baB^nA \mid n \geq 1\} \cup \{bB^nA \mid n \geq 1\} \\ & \cup \{baB^na \mid n \geq 1\} \cup \{bB^na \mid n \geq 1\} \\ & \cup \{baB^n \mid n \geq 1\} \cup \{bB^n \mid n \geq 1\} \end{aligned}$$

An action r is **valid** for viable stack string γ of M if

$$\$ \gamma' \mid y' \$ \Rightarrow^r \$ \gamma \mid y \$ \text{ in } M \ .\exists. k:y\$ \in \text{First}_k(\gamma^R\$).$$

Note that the action r is valid for γ not for γ' .

Two viable stack string belongs to the same **equivalent class** if they have same set of **valid actions**.

At most $|\Sigma| + |\Sigma|^k |P| \leq |G|^{k+1}$ distinct actions

\therefore at most $2^{|G|^{k+1}}$ equivalent classes

<i>prev. class:</i>	<i>valid action</i>	<i>equi. class</i>
[S]	$S a \rightarrow baAa a$	[baAa]
[S]	$S b \rightarrow bAb b$	[bAb]
[baAa]	$a a \rightarrow $	[baA]
[bAb]	$b b \rightarrow $	[bA]
[baA]	$A c \rightarrow BAc c$	[baB ⁺ Ac]
[baA]	$A aa \rightarrow a aa$	[baa]
[baA]	$A ab \rightarrow ab$	[ba]
[baa]	$a a \rightarrow $	[ba]
[ba]	$a a \rightarrow $	[b]
[b]	$b b \rightarrow $	[ε]
[bA]	$A c \rightarrow BAc c$	[bB ⁺ Ac]
[bA]	$A ab \rightarrow a ab$	[bA]
[bA]	$A b\$ \rightarrow b\$$	[b]
[baB ⁺ Ac]	$c c \rightarrow$	[baB ⁺ A]
[bB ⁺ Ac]	$c c \rightarrow$	[bB ⁺ A]
[baB ⁺ A]	$A c \rightarrow BAc c$	[baB ⁺ Ac]
[baB ⁺ A]	$A aa \rightarrow a aa$	[baB ⁺ a]
[baB ⁺ A]	$A ab \rightarrow ab$	[baB ⁺]
[baB ⁺ a]	$a a \rightarrow $	[baB ⁺]
[baB ⁺]	$B ab \rightarrow ab$	[baB ⁺]
[baB ⁺]	$B ab \rightarrow ab$	[ba]
[bB ⁺ A]	$A c \rightarrow BAc c$	[bB ⁺ Ac]
[bB ⁺ A]	$A ab \rightarrow a ab$	[bB ⁺ Aa]
[bB ⁺ A]	$A b\$ \rightarrow b\$$	[bB ⁺]
[bB ⁺ Ac]	$c c \rightarrow$	[bB ⁺ A]
[bB ⁺ Aa]	$a a \rightarrow $	[bB ⁺ A]
[bB ⁺]	$B b\$$	[bB ⁺]
[bB ⁺]	$B b\$$	[b]
—	—	[S]

The equivalence should be

(1) right-invariant

(2) any two equivalent viable stack strings
should end with the same symbol

$$\begin{array}{ll} \{\varepsilon, bA\} & \{\varepsilon\}, \{bA\} \\ \{bB^n \mid n \geq 0\} & \{b\}, \{bB^n \mid n \geq 1\} \\ \{bB^n A \mid n \geq 1\} \cup \{baB^n A \mid n \geq 1\}, & \\ & \{bB^n A \mid n \geq 1\}, \{baB^n A \mid n \geq 1\} \end{array}$$

$\{baa\}$

$\{baB^n a \mid n \geq 1\}$ same valid actions

$\{A \mid aa \rightarrow a \mid aa\}$

$\{baB^n a \mid n \geq 0\}$

16 equivalent classes

$[\varepsilon], [S],$

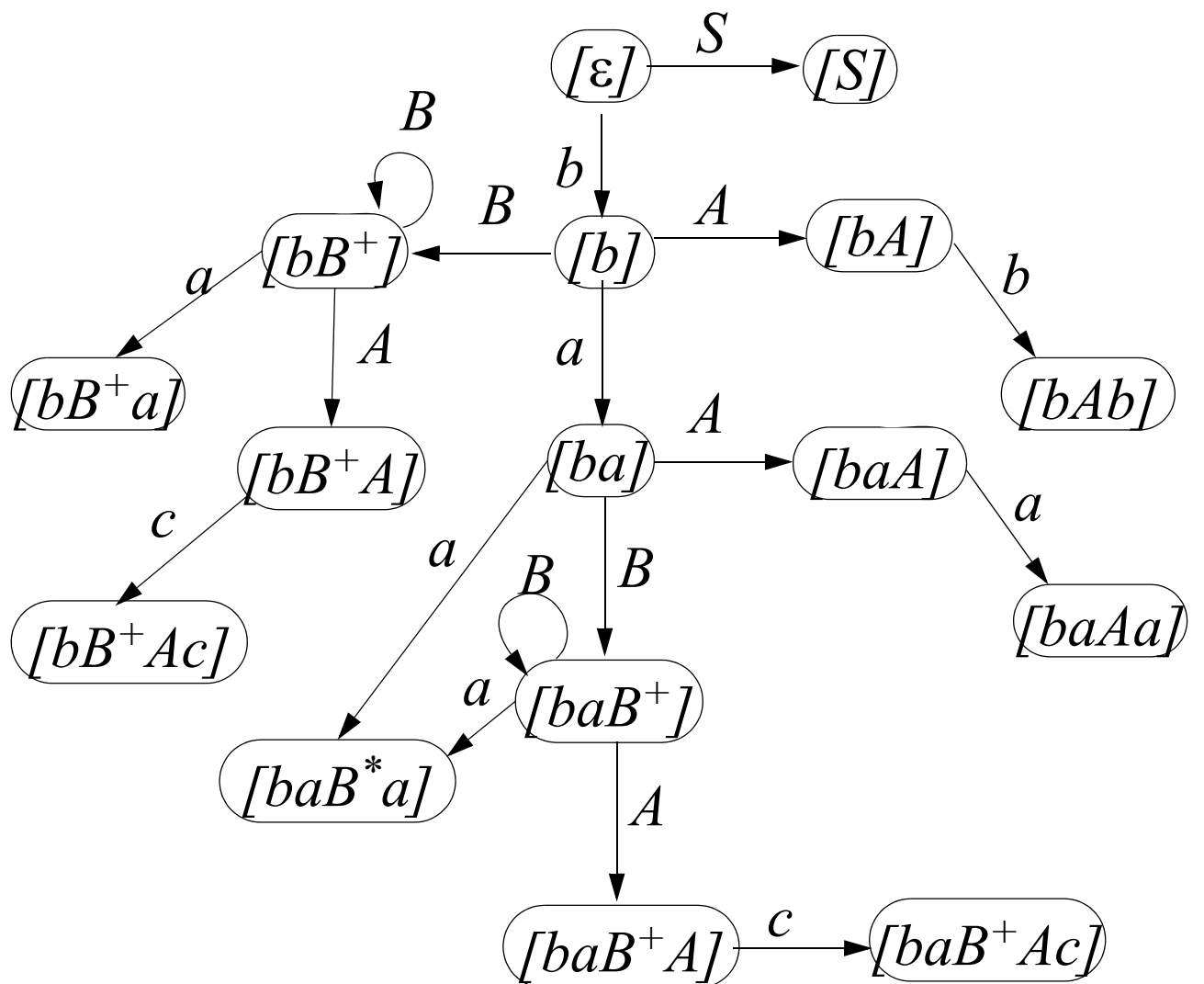
$[b], [ba[, [baA], [baAa], [bA], [bAb],$

$[baB^+], [baB^+A], [baB^+c], [baB^*a],$

$[bB^+], [bB^+A], [bB^+c], [bB^+a].$

$$S \rightarrow aAab \mid bAb$$

$$A \rightarrow cAB \mid a \mid \varepsilon$$

$$B \rightarrow \varepsilon$$


A simplified canonical LL(k) parser

the new produce action

$$[\delta A] \mid y \rightarrow Y_n \dots Y_1 \mid y,$$

where δA is viable stack string,

$$A \rightarrow X_1 \dots X_n \text{ is a rule,}$$

$$y \in \text{First}_k(X_1 \dots X_n \delta^R \$),$$

$$Y_i = [\delta X_n \dots X_i] \quad X_i \in N$$

$$X_i \quad X_i \in \Sigma.$$

same shift action

$$a \mid a \rightarrow \mid.$$

the shift action is same as that of SLL(k) parser.

Example: simplified canonical LL(2) parser of G_{abL}

$$(p. 202) [S]_2 \mid aa \rightarrow ba[baA]_2 a \mid aa, \quad S \rightarrow aAab.$$

in the SLL(2) parser

$$A \mid ab \rightarrow a \mid ab, \quad A \mid ab \rightarrow \mid ab.$$

in the simplified canonical LL(2) parser

$$[bA]_2 \mid ab \rightarrow a \mid ab, \quad [bB^+A]_2 \mid ab \rightarrow a \mid ab.$$

$$[baA]_2 \mid ab \rightarrow \mid ab, \quad [baB^+A]_2 \mid ab \rightarrow \mid ab.$$

G is not SLL(2) but simplified canonical LL(2).

String $\gamma \in V^*$ is a **viable suffix** of G , if

$$S \xRightarrow{lm}^* xA\delta \xRightarrow{lm} x\alpha\beta\delta (=x\alpha\gamma^R)$$

where $\delta \in V^*$, $x \in \Sigma^*$, and $A \rightarrow \alpha\beta \in P$.

γ is a **complete viable suffix**, if $\alpha = \varepsilon$.

Note that the viable suffix is **reversed**, since the **top** of the stack is the **rightmost** symbol.

Fact 8.1 Any (reversal of) viable suffix is a (suffix) prefix of some complete viable suffix.

Lemma 8.2 $n \geq 0$, $\phi_1, \phi_2 \in V^*$,

(a) $\phi_1 \xRightarrow{rm}^{r_1 \dots r_n} \phi_2$ in G iff

$$\phi_1^R \xRightarrow{lm}^{r_1^R \dots r_n^R} \phi_2^R \text{ in } G^R$$

(b) $\phi_1 \xRightarrow{lm}^{r_1 \dots r_n} \phi_2$ in G iff

$$\phi_1^R \xRightarrow{rm}^{r_1^R \dots r_n^R} \phi_2^R \text{ in } G^R$$

Proof induction on n .

$r^R: A \rightarrow \omega^R$ in G^R whenever $r: A \rightarrow \omega$ in G .

The set of viable suffixes of G coincides with the set of viable prefixes of G^R .

Lemma 8.3

(a) $\gamma \in V^*$ is a (complete) viable prefix of G iff γ is a (complete) viable suffix of G^R .

(b) $\gamma \in V^*$ is a (complete) viable suffix of G iff γ is a (complete) viable prefix of G^R .

Proof. (a) (only if)

If $S \xRightarrow{rm}^* \delta A y \xRightarrow{rm}^r \underline{\delta \alpha} \beta y (= \gamma \beta y)$ holds in G , then ,

$$\begin{aligned} S \xRightarrow{lm}^* (\delta A y)^R &= y^R A \delta^R \xRightarrow{lm}^* y^R (\alpha \beta)^R \delta^R \\ &= y^R \beta^R (\delta \alpha)^R = y^R \beta^R \gamma^R \text{ holds in } G^R. \text{ (L 8.2)} \end{aligned}$$

(if)

$$(G^R)^R = G.$$

“only if” part of (b) = ”if” part of (a)

“only if” part of (a) = “if” part of (b)

Theorem 8.4 *the set of all viable suffixes of G is a regular language over V*

Given a grammar $G = (N, \Sigma, P, S)$,

Let $G_{VS} = (N_{VS}, \Sigma_{VS}, P_{VS}, [S])$ where

$$N_{VS} = \{[A] \mid A \in N\},$$

$$\Sigma_{VS} = N \cup \Sigma, \text{ and}$$

$$P_{VS} = \{[A] \rightarrow \beta^R \mid A \rightarrow \alpha\beta \in P\} \cup$$

$$\{[A] \rightarrow \beta^R [B] \mid A \rightarrow \alpha B \beta \in P, B \in N\}.$$

$(G_{abL})_{VS}$:

$$[S] \rightarrow \varepsilon \mid b \mid ba \mid baA \mid baAa \mid bA \mid bAb \mid ba[A] \mid b[A]$$

$$[A] \rightarrow \varepsilon \mid B \mid BA \mid BA c \mid [B] \mid B[A] \mid a$$

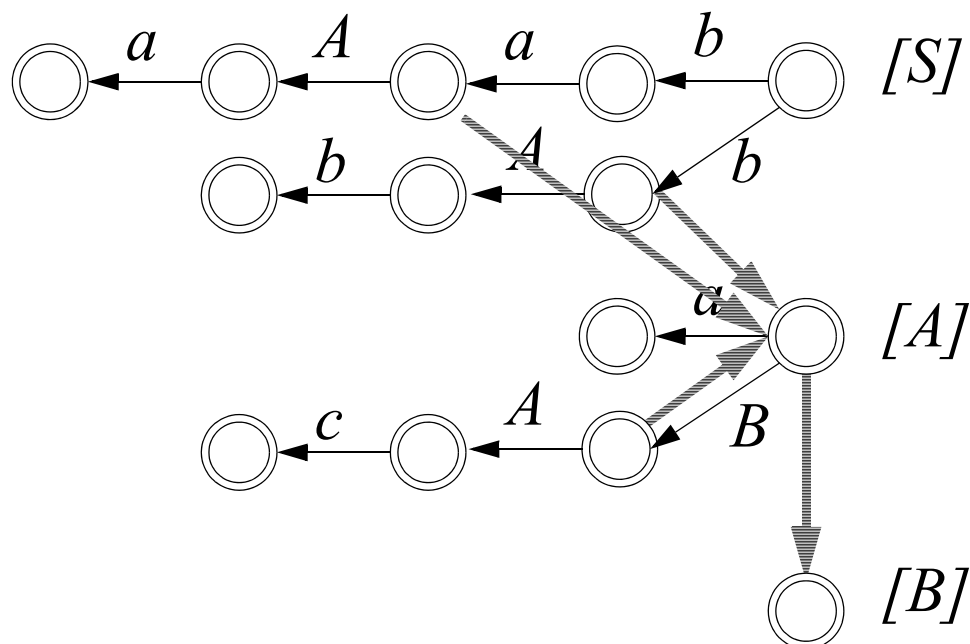
$$[B] \rightarrow \varepsilon$$

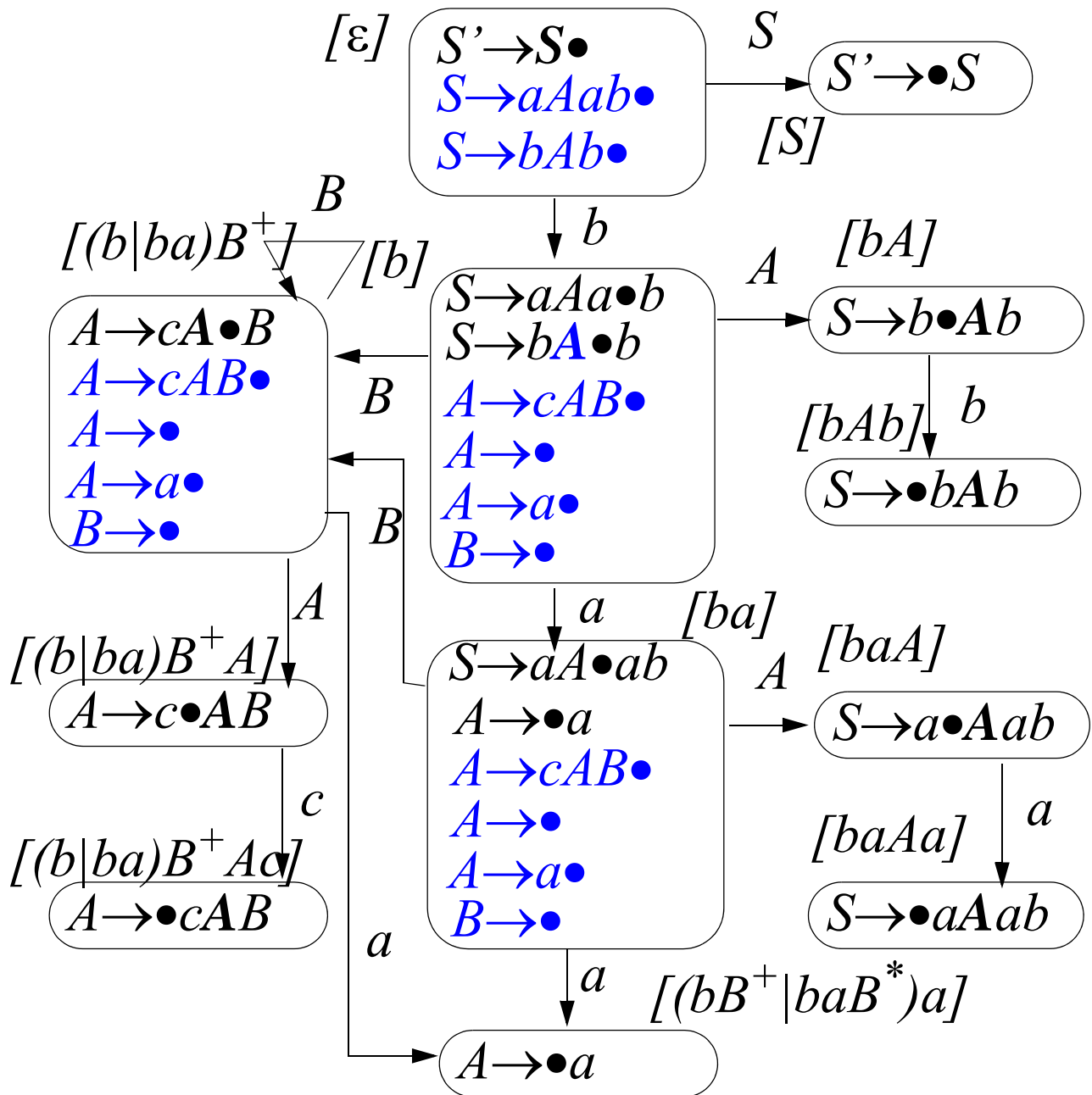
G_{abL} :

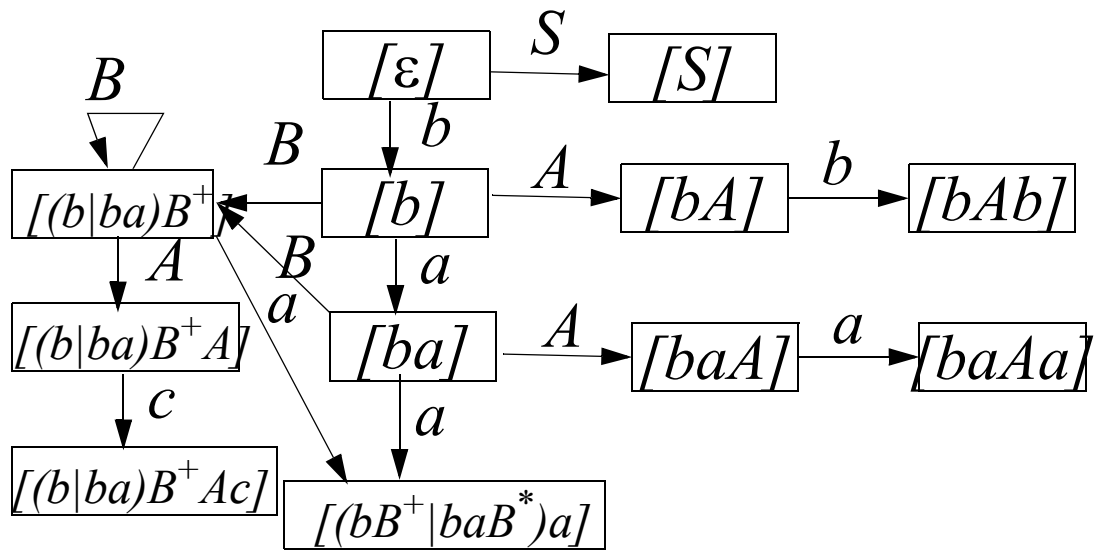
$$S \rightarrow aAab \mid bAb,$$

$$A \rightarrow cAB \mid \varepsilon \mid a,$$

$$B \rightarrow \varepsilon.$$







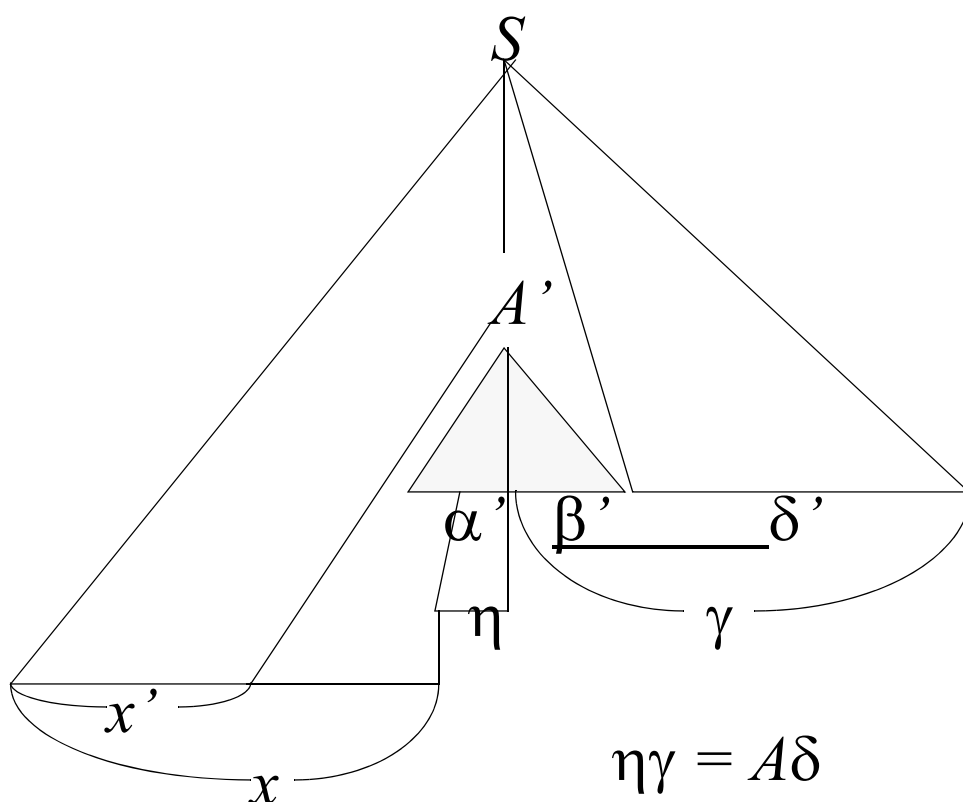
Lemma 8.5

If $S \xrightarrow[lm]{\pi} x\eta\gamma = xA\delta$ in G , and $\pi \neq \varepsilon$. Then

$$S \xrightarrow[rm]{\pi'} x'A'\delta' \xrightarrow[lm]{r} x'\alpha'\beta'\delta' = x'\alpha'\gamma, \quad x'\alpha' \xrightarrow[rm]{\pi''} x\eta$$

$$\pi'r\pi'' = \pi, \text{ and } 1:\beta' = 1:\gamma. \text{ (L6.2)}$$

If γ is a prefix of some nontrivially derived left sentential form not extending over the first nonterminal, then there is the derivation contains a segment that proves γ^R to be a viable prefix, even so that the right-hand side of the rule r cuts γ properly.



Lemma 8.6

$S \xRightarrow{lm}^+ xA\delta$. Then $\delta^R A$ is a **viable suffix**.

Proof. $\gamma = A\delta$, $\eta = \varepsilon$. (L 8.5)

Lemma 8.7 Any (suffix) prefix of a reversal of viable suffix is a (reversal of) **viable suffix**.

Proof. Let $\gamma_1\gamma_2$ is a viable suffix, then $\gamma_1\gamma_2$ is a viable prefix of G^R . If γ_1 is a viable prefix of G^R (L 6.4), then a viable suffix of $(G^R)^R = G$. (L 8.3(a))

Lemma 8.8 $A \rightarrow \alpha\beta \in P$, if δA is a viable suffix of G , then so is $\delta\beta^R$.

Lemma 8.9 Let M is its SLL(k) parser for G and

$\$ \gamma \eta \mid y \$ \Rightarrow^\pi \$ \mid \$$ in M . Then

$\$ \gamma \eta \mid y \$ \Rightarrow^{\pi'} \$ \gamma \mid z \$ \Rightarrow^{\pi''} \$ \mid \$$ in M , and $\pi = \pi' \pi''$.

Proof. induction on $|\pi|$. (L 6.6)

Lemma 8.10 Any prefix of a viable stack string of an SLL(k) parser is a viable stack string. (L 8.9)

Theorem 8.11 Any viable stack string of SLL(k) parser M is either S or viable suffix of G .

Conversely, any viable suffix of G is a viable stack string of M , provided that G is reduced. (T6.7)

Proof. (the first part of theorem)

$\gamma \neq S$, r the last produce action in π

$$\$S|w\$ \Rightarrow^\pi \$\gamma|z\$ \Rightarrow^* \$|\$ \text{ in } M$$

$$\begin{aligned} \$S|w\$ \Rightarrow^{\pi_1} \$\delta A|yz\$ \Rightarrow^r \$\delta\omega^R|yz\$ = \$\gamma y^R|yz\$ \\ \Rightarrow^{\pi_2} \$\gamma|z\$, \text{ where } \pi_1 r \pi_2 = \pi, w=xyz \end{aligned}$$

π_2 is a $|y\}$ -lengthed shift actions.

$$\therefore S \xRightarrow[lm]^* xA\delta^R \text{ in } G, A \rightarrow \omega \in P, \gamma y^R = \delta\omega^R. (L 5.27)$$

$$S \xRightarrow[lm]^* xA\delta^R \xRightarrow[lm] x\omega\delta^R = x(\gamma y^R)^R,$$

γy^R is viable suffix, so is γ . **Q.E.D.**

8.2 Valid LL(k) Items

Let $A \rightarrow \alpha\beta \in P$. Then $[A \rightarrow \alpha\cdot\beta, y]$ is a **k-item**.

$A \rightarrow \alpha\cdot\beta$ is a core item,

$y \in k:\Sigma^*$ is a lookahead string.

A k-item $[A \rightarrow \alpha\cdot\beta, y]$ is **LL(k)-valid**(or **valid**) for string $\gamma \in V^*$ if

$$S \xRightarrow{lm}^* xA\delta \xRightarrow{lm} x\alpha\beta\delta = x\alpha\gamma^R \text{ and } y \in \text{First}_k(\gamma^R).$$

The stack string is the prediction of remained input string.

Fact 8.12 If $[A \rightarrow \alpha\cdot\beta, y]$ is a LL(k) valid item for string γ , then γ is a **viable suffix**, $[A \rightarrow \alpha\cdot\beta, y]$ is a **k-item**, β^R is a suffix of $\gamma (= \delta^R\beta^R)$ and $y \in \text{First}_k(\gamma^R) = \text{First}_k(\beta\delta) \subseteq \text{First}_k(\beta\text{Follow}_k(A))$.

Conversely, if a string γ is a viable suffix, then some item is LL(k)-valid for γ . (**F6.12**)

Example : LL(k) valid item of G_{abL}

Let $\gamma \in V^*$. Then

$$\begin{aligned} \langle \gamma \rangle_{LL(k)}^G &= Valid_{LL(k)}^G(\gamma) \\ &= \{I \mid I \text{ is an } LL(k)\text{-valid item for } \gamma\} \\ &= \{[A \rightarrow \alpha.\beta, y] \mid S \xrightarrow{lm}^* xA\delta \xrightarrow{lm} x\alpha\beta\delta = x\alpha\gamma^R \\ &\quad y \in First_k(\gamma^R)\} \end{aligned}$$

$$\begin{aligned} Valid_{LL(k)}^G &\equiv Valid_{LL(k)} \equiv Valid_k \equiv Valid \\ \langle \rangle_{LL(k)}^G &\equiv \langle \rangle_{LL(k)} \equiv \langle \rangle_k \equiv \langle \rangle. \end{aligned}$$

String γ_1 is **LL(k)-equivalent(equivalent)** to string γ_2 , written $\gamma_1 \rho_{LL(k)} \gamma_2$ (or $\gamma_1 \rho_k \gamma_2$), if

$$Valid_k(\gamma_1) = Valid_k(\gamma_2).$$

The relation ρ_k is the **LL(k)-equivalence** for G .

$[\gamma]_{\rho_k}$ denotes an **equivalent class** of γ under ρ_k

$$\begin{aligned} [\gamma]_{\rho_k} &= \{\delta \mid \gamma \rho_k \delta\} \\ [\gamma]_{\rho_k} &\equiv [\gamma]_k \equiv [\gamma]. \end{aligned}$$

We denote $[\gamma]_{\rho_k}$ by $[\gamma]_k$ (or even $[\gamma]$).

Fact 8.13 *From the definition of a viable suffix implies that*

$$\begin{aligned} & [A_1 \rightarrow \alpha_1 \cdot \beta_1, y_1], [A_2 \rightarrow \alpha_2 \cdot \beta_2, y_2] \in \langle \gamma \rangle_k \\ & \{z \mid [A_1 \rightarrow \alpha_1 \cdot \beta_1, z] \in \langle \gamma \rangle_k\} \\ & = \{z \mid [A_2 \rightarrow \alpha_2 \cdot \beta_2, z] \in \langle \gamma \rangle_k\} \\ & = \text{First}_k(\gamma^R). \end{aligned}$$

Note that the viable suffix is reversed in LL parsing.

Theorem 8.14

The LL(k)-equivalence ρ_k for G is of finite index, and the index of ρ_k is at most $2^{|G| + (|\Sigma| + 1)^k}$. (T6.13)

Proof.

The number of distinct subsets of $k:\Sigma^$ is at most $2^{(k|\Sigma| + 1)}$, each of these subsets may occur in at most $2^{|G|}$ sets $\langle \gamma \rangle_0$. (F 8.13)*

Hence, $2^{|G|} \cdot 2^{(k|\Sigma| + 1)} = 2^{|G| + (|\Sigma| + 1)^k}$.

Q.E.D.

Lemma 8.15 Let $k \leq l$. Then

$$\langle \gamma \rangle_k = \{ [A \rightarrow \alpha \cdot \beta, k:y] \mid [A \rightarrow \alpha \cdot \beta, y] \in \langle \gamma \rangle_l \}. \quad (\mathbf{L6.14})$$

Lemma 8.16 Let $k \leq l$. Then

LL(l)-equivalence is a **refinement** of LL(k)-equivalence. More specifically $[\gamma]_k = \cup [\delta]_l. (\mathbf{L6.15})$

canonical collection of set of LL(k)-valid items for $G =$ canonical LL(k) collection for $G: C_k$

**Simplified canonical LL(k) machine M
(or deterministic LL(k) machine)**

$$M = (C_k, V, \{ \langle \gamma \rangle_k \cdot X \rightarrow \langle \gamma X \rangle_k \}, \langle \varepsilon \rangle_k, \emptyset)$$

Example : the canonical LL(2) machine for G'_{abL}

Construction of the canonical LL(k) machine

$\partial_{LL(k)}$: relation on LL(k) items

$$[A \rightarrow \alpha B \cdot \beta, y] \partial_{LL(k)} [B \rightarrow \omega \cdot, y]$$

$$\mathbf{desc}_{LL(k)}, \partial_{LL(k)} \equiv \partial_k \equiv \partial.$$

I_2 is an **immediate LL(k)-descendant** of I_1 , if $I_1 \partial I_2$.

I_2 is an **LL(k)-descendant** of I_1 , if $I_1 \partial^* I_2$.

I_1 is an **(immediate) LL(k)-ancestor** of I_2 ,

if I_2 is an **(immediate) LL(k)-descendant** of I_1 .

$$\langle \gamma \rangle_k^n = \{ [A \rightarrow \alpha \cdot \beta, y] \mid S \xrightarrow{lm}^n x A \delta \xrightarrow{lm} x \alpha \beta \delta (= x \alpha \gamma^R), \\ y \in \text{First}_k(\gamma^R) \}$$

Fact 8.17 $k \geq 0$, $\langle \gamma \rangle_k = \cup_{n=0}^{\infty} \langle \gamma \rangle_k^n$. (F6.16)

Lemma 8.18 If $[A \rightarrow \alpha B \cdot \beta, y] \in \langle \gamma \rangle_k^n$ and

$$\alpha \Rightarrow^m v \in \Sigma^*. \text{ Then } \forall B \rightarrow \omega \in P,$$

$$[B \rightarrow \omega \cdot, y] \in \langle \gamma \rangle_k^{n+1}. \text{ (L6.17)}$$

Proof. $S \xrightarrow{lm}^n x A \delta \xrightarrow{lm} x \alpha B \beta \delta \xrightarrow{lm}^m x v B \beta \delta \xrightarrow{lm} x v \omega \beta \delta$.

Lemma 8.19 $\langle \gamma \rangle_k$ is closed under ∂ , i.e.,

$$\partial^*(\langle \gamma \rangle_k) = \langle \gamma \rangle_k. \text{ (L6.18)}$$

Lemma 8.20 If $[B \rightarrow \omega \cdot, y] \in \langle \gamma \rangle_k^n$ and $n > 0$. Then $[A \rightarrow \alpha B \cdot \beta, y] \in \langle \gamma \rangle_k^m$, $\alpha \Rightarrow^{n-m-1} \nu$, $m < n$. **(L6.19)**

Fact 8.21 $\langle \gamma \rangle_k^0 = \{[S \rightarrow \omega \cdot \gamma^R, \varepsilon] \mid S \rightarrow \omega \gamma^R \in P\}$.
(F6.20)

$[A \rightarrow \alpha \cdot \beta]$ is **LL-essential** (or **essential**), if $\beta \neq \varepsilon$ **inessential**, otherwise.

$\text{Ess}_{LL(k)}(q) = \{I \in q \mid I \text{ is LL-essential}\}$.

Lemma 8.22 Let $I \in \langle \gamma \rangle_k^n$, $k, n \geq 0$.

(1) $n = 0$, $\gamma = \varepsilon$, $I = [S \rightarrow \omega \cdot, \varepsilon]$.

(2) $\gamma \neq \varepsilon$ and I is **essential**.

(3) $n > 0$, and I has an **immediate ancestor** in some $\langle \gamma \rangle_k^m$, $m < n$. **(L6.21)**

Lemma 8.23

$\langle \varepsilon \rangle_k^n \subseteq \partial^*(\{[S \rightarrow \omega \cdot, \varepsilon] \mid S \rightarrow \omega \in P\})$,

$\langle \gamma \rangle_k^n \subseteq \partial^*(\text{Ess}(\langle \gamma \rangle_k^n))$, if $\gamma \neq \varepsilon$. **(L6.22)**

Lemma 8.24

$\langle \varepsilon \rangle_k^n = \partial^*(\{[S \rightarrow \omega \cdot, \varepsilon] \mid S \rightarrow \omega \in P\})$

$\langle \gamma \rangle_k^n = \partial^*(\text{Ess}(\langle \gamma \rangle_k^n))$, if $\gamma \neq \varepsilon$. **(L6.23)**

χ_k^X : relation on set of LL(k) items.

$[A \rightarrow \alpha X \cdot \beta, y] \chi_k^X [A \rightarrow \alpha \cdot X \beta, z], z \in \text{First}_k(Xy)$

pass-X, or χ^X for short

Basis $_{LL(k)}(q, X) = \{[A \rightarrow \alpha \cdot X \beta, z] \mid [A \rightarrow \alpha X \cdot \beta, y] \in q, \text{ and } z \in \text{First}_k(Xy)\} \equiv \chi_k^X(q).$

δ_k^X : relation on set of LL(k) items.

Goto $_{LL(k)}(q, X) = \partial_k^*(\text{Basis}_{LL}(q, X)) = \partial_k^*(\chi_k^X(q)) \equiv \delta_k^X(q).$

X-successor, δ_k^X for short

Lemma 8.25

If $[A \rightarrow \alpha \omega \cdot \beta, y] \in \langle \gamma \rangle_k^n$, then

$\gamma \omega^R$ is a viable suffix and

$[A \rightarrow \alpha \cdot \omega \beta, z] \in \langle \gamma \omega^R \rangle_k^n.$

If $[A \rightarrow \alpha \cdot \omega \beta, z] \in \langle \delta \rangle_k^n$, then

$\exists \gamma$ s.t. $[A \rightarrow \alpha \omega \cdot \beta, y] \in \langle \gamma \rangle_k^n$

$\delta = \omega \gamma^R, z \in \text{First}_k(\omega y).$ **(F6.24)**

Lemma 8.26 $\text{Ess}(\langle \gamma X \rangle_k) = \text{Basis}(\langle \gamma \rangle_k, X)$ **(L6.25)**

Lemma 8.27 $\langle \gamma X \rangle_k = \text{Goto}(\langle \gamma \rangle_k, X)$ **(L6.26)**

Algorithm for constructing the canonical LL(k) machine

begin

 compute the relation $\hat{\partial}$

for $X \in V$ **do**

 compute the χ_k^X relation

$q_s := \hat{\partial}_k^* (\{[S' \rightarrow S\cdot, \varepsilon]\})$;

$Q_M := \{q_s\}$;

$P_M := \emptyset$;

repeat

for $q \in Q_M$ **and** $X \in V$ **do**

begin

$p := \hat{\partial}_k^* (\chi_k^X(q))$;

$Q_M := Q_M \cup \{p\}$;

$P_M := P_M \cup \{q \cdot X \rightarrow p\}$

end

until nothing is added to Q_M and P_M .

end

Lemma 8.28 Let $M = (Q_M, V, P_M, q_s, F)$ be a canonical LL(k) machine for G . Then the followings hold

(a) M is deterministic.

(b) $q \in Q_M$, q has unique entry symbol.

(c) q is accessible upon reading γ , iff $q = \langle \gamma \rangle_k$.

(d) If $F = \{ \langle \gamma \rangle_k \mid \text{for a given } \gamma \}$, $L_M = [\gamma]_k$.

(e) If $F = \{ \langle \gamma \rangle_k \mid \langle \gamma \rangle_k \neq \emptyset \}$,

$L_M = \text{Set of viable suffixes of } X.$

(f) If $F = \{ \langle \gamma \rangle_k \mid \text{for all } \gamma \}$, $L_M = V^*$. **(L6.27)**

Theorem 8.29

(a) The LL(k) equivalence of G is the equivalence induced by the canonical LL(k) machine of G .

$[\gamma_1]_{\rho k} = [\gamma_2]_{\rho k}$ iff

$q_s \mid \gamma_1 \Rightarrow^* q_s \dots q \mid$ and $q_s \mid \gamma_2 \Rightarrow^* q_s \dots q \mid$.

(b) The LL(k) equivalence of G is **right invariance**.

If $[\gamma_1]_{\rho k} = [\gamma_2]_{\rho k}$, then $[\gamma_1 \cdot X]_{\rho k} = [\gamma_2 \cdot X]_{\rho k}$.

(c) The LL(k) equivalence of G is **ends with same symbols**.

If $[\gamma_1]_{\rho k} = [\gamma_2]_{\rho k}$, then $\gamma_1 : 1 = \gamma_2 : 1$. **(T6.28)**

8.3 Canonical LL(k) Parser

The **canonical LL(k) parser** for G is a pushdown transducer $M = ([G]_k, \Sigma, \Gamma, \tau, [\varepsilon]_k, [S]_k, \{[\varepsilon]_k\})$ where $[G]_k = \{[\gamma]_k \mid \gamma \in V^*\}$

$$\begin{aligned} \Gamma = & \{[\gamma]_k[\gamma A]_k \mid y \rightarrow [\gamma]_k[\gamma X_n]_k \dots [\gamma X_n \dots X_1]_k \mid y \\ & \mid [A \rightarrow \bullet X_1 \dots X_n, y] \in \langle \gamma X_n \dots X_1 \rangle_k\} \quad (pa) \\ \cup & \{[\gamma]_k[\gamma a]_k \mid ay \rightarrow [\gamma]_k \mid y \\ & \mid [A \rightarrow \alpha \bullet a \beta, k:ay] \in \langle \gamma a \rangle_k\} \quad (sa) \end{aligned}$$

$$\begin{aligned} \tau([\gamma]_k[\gamma A]_k \mid y \rightarrow [\gamma]_k[\gamma X_n]_k \dots [\gamma X_n \dots X_1]_k \mid y) \\ = A \rightarrow X_1 \dots X_n, \\ \tau([\gamma]_k[\gamma a]_k \mid ay \rightarrow [\gamma]_k \mid y) = \varepsilon. \end{aligned}$$

compare Γ in canonical **LR(k) parser**

$$\begin{aligned} \Gamma = & \{[\delta]_k[\delta X_1]_k \dots [\delta X_1 \dots X_n]_k \mid y \rightarrow [\delta]_k[\delta A]_k \mid y \\ & \mid [A \rightarrow X_1 \dots X_n, y] \in \langle \delta X_1 \dots X_n \rangle_k\} \quad (ra) \\ \cup & \{[\delta]_k \mid ay \rightarrow [\delta]_k[\delta a]_k \mid y \\ & \mid a \in \Sigma, [A \rightarrow \alpha \bullet a \beta, z] \in \langle \delta \rangle_k, \\ & y \in First_{\max\{k-1, 0\}}(\beta z)\} \quad (sa) \end{aligned}$$

$$\begin{aligned}
[B \rightarrow \alpha A \bullet \beta, x] \in \langle \gamma \rangle_k = q \quad [B \rightarrow \alpha \bullet A \beta, y] \in \langle \gamma A \rangle_k = q_A \\
[A \rightarrow X_1 \dots X_n \bullet, x] \in \langle \gamma \rangle_k = q (= q_0) \\
[A \rightarrow X_1 X_2 \dots \bullet X_n, x_n] \in \langle \gamma X_n \rangle_k = q_n \\
\dots \\
[A \rightarrow \bullet X_1 \dots X_n, x_1] \in \langle \gamma X_n \dots X_1 \rangle_k = q_1
\end{aligned}$$

Fig. 8.5 Algorithm for computing the parsing action of the canonical LL(k) parser.

```

for  $\forall q$  do
  for  $A \rightarrow X_1 \dots X_n \in P$  s.t.  $[A \rightarrow X_1 \dots X_n \bullet, x] \in q_0$  do
    where  $q_{i+1} = \text{GOTO}(q_i, X_{i+1})$  for  $1 \leq i \leq n$ 
     $q_A = \text{GOTO}(q, A)$ 
    for  $[A \rightarrow \bullet X_1 \dots X_n, y] \in q_1$  do where  $q = q_0$ 
      generate  $qq_A \mid y \rightarrow qq_n \dots q_1 \mid y$  od
    od
  for  $[A \rightarrow \alpha \bullet a \beta, ay] \in q$  do
    generate  $q \mid ay \rightarrow \mid y$  od
  od

```

Fig 8.2 The Canonical LL(1) machine of G_{abL}

$First_1(S) = \{a, b\}$

$First_1(A) = \{\epsilon, a, c\}$ $First_1(B) = \{\epsilon\}$

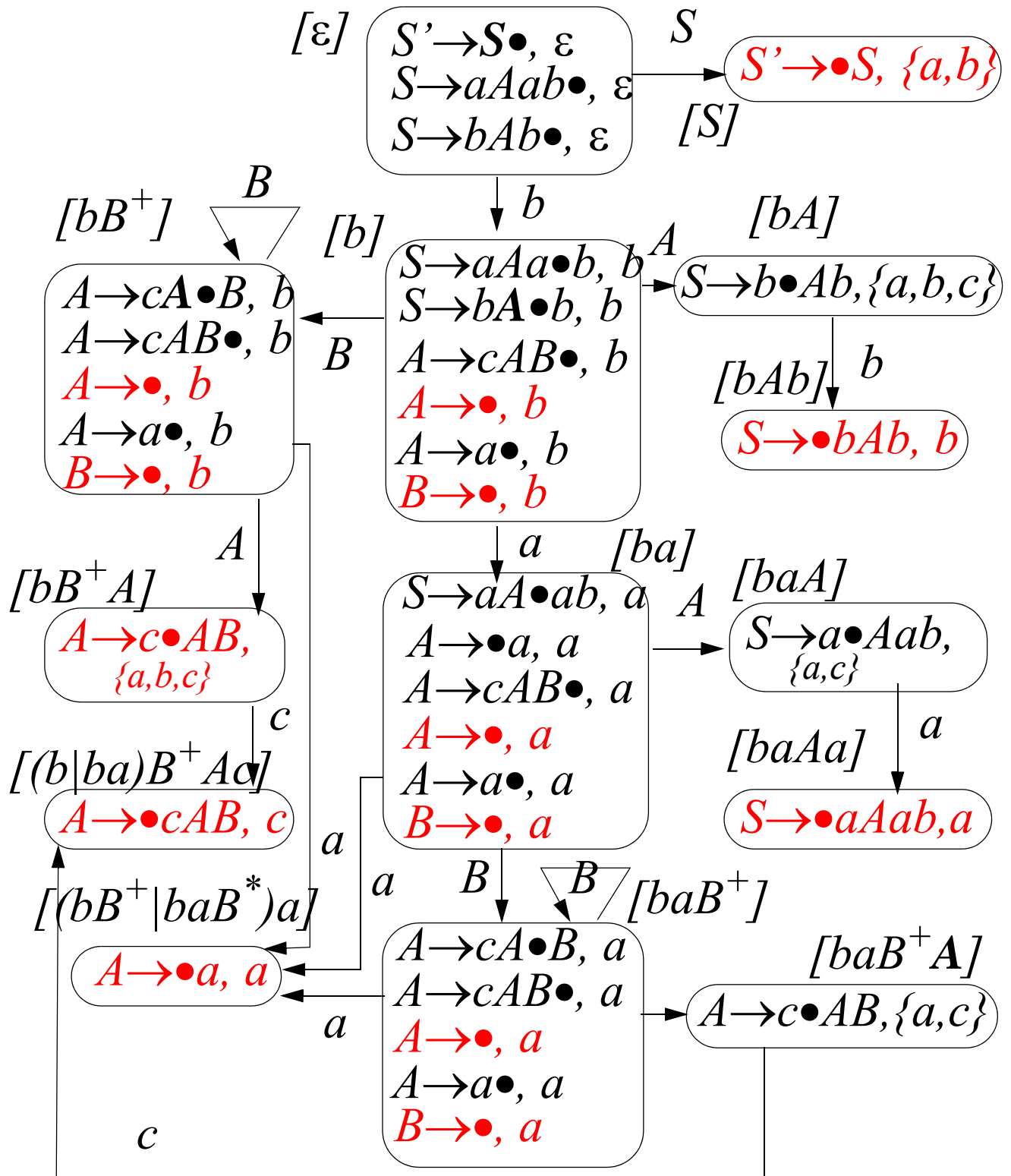
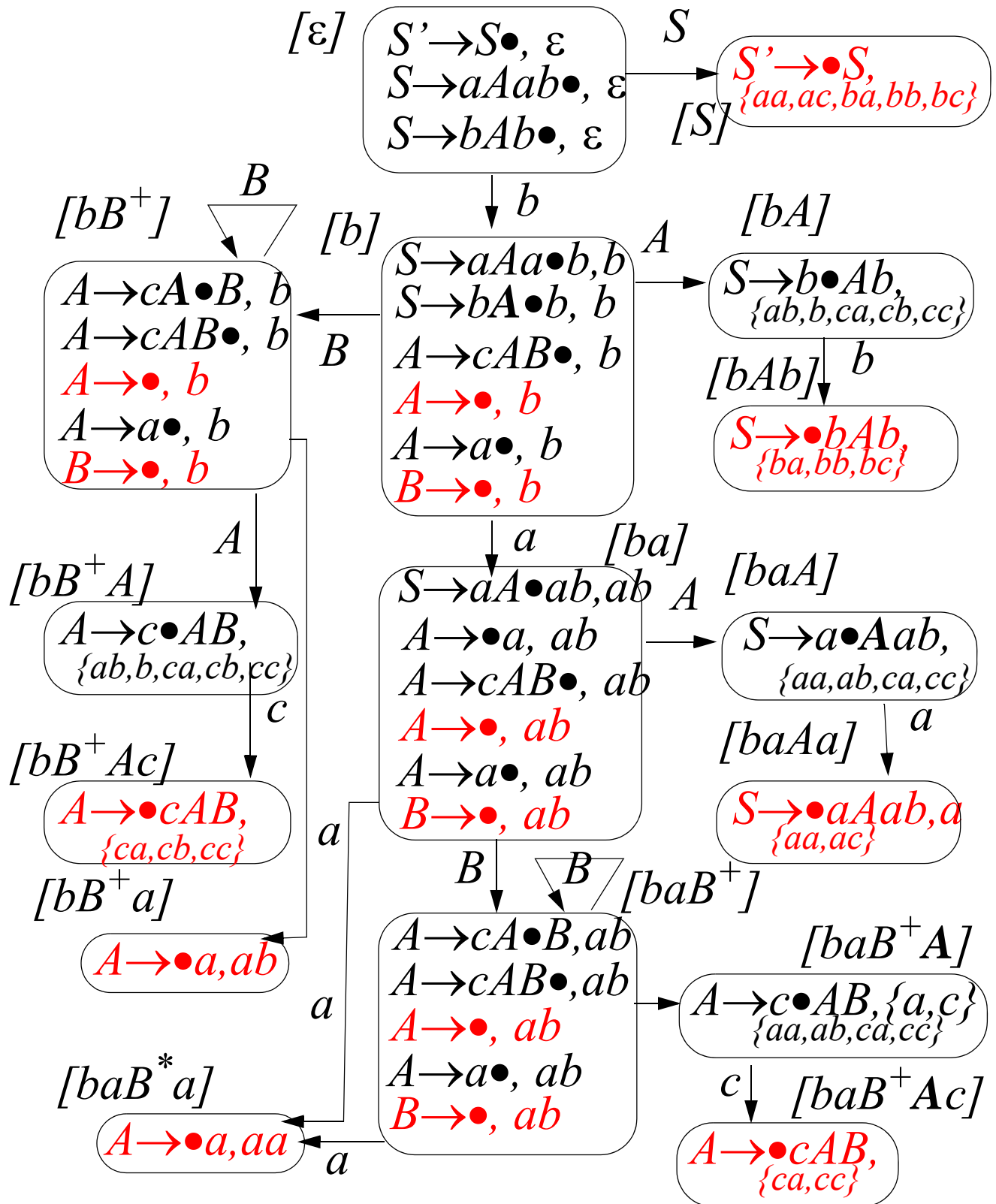


Fig 8.3 The Canonical LL(2) machine of G_{abL}

$First_2(S) = \{aa, ac, ba, bb, bc\}$

$First_2(A) = \{\epsilon, a, c, ca\}$ $First_2(B) = \{\epsilon\}$



$$\$[\epsilon][S] \mid accab\$ \Rightarrow^{S \rightarrow aAab}$$

$$\$[\epsilon][b][ba][baA][baAa] \mid accab\$ \Rightarrow^a$$

$$\$[\epsilon][b][ba][baA] \mid ccab\$ \Rightarrow^{A \rightarrow cAB}$$

$$\$[\epsilon][b][ba][baB][baBA][baBAc] \mid ccab\$ \Rightarrow^c$$

$$\$[\epsilon][b][ba][baB][baBA] \mid cab\$ \Rightarrow^{A \rightarrow cAB}$$

$$\$[\epsilon][b][ba][baB][baBB][baBBA][baBBAc] \mid cab\$$$

$$\Rightarrow^c \$[\epsilon][b][ba][baB][baBB][baBBA] \mid ab\$$$

$$\Rightarrow^{A \rightarrow \epsilon} \$[\epsilon][b][ba][baB][baBB] \mid ab\$$$

$$\Rightarrow^{B \rightarrow \epsilon} \$[\epsilon][b][ba][baB] \mid ab\$$$

$$\Rightarrow^{B \rightarrow \epsilon} \$[\epsilon][b][ba] \mid ab\$$$

$$\Rightarrow^a \$[\epsilon][b] \mid b\$ \qquad \Rightarrow^b \$[\epsilon] \mid \$$$

$$\$[\epsilon][S] \mid bccb\$$$

$$\Rightarrow^{S \rightarrow bAb} \$[\epsilon][b][bA][bAb] \mid bccb\$$$

$$\Rightarrow^b \$[\epsilon][b][bA] \mid ccb\$$$

$$\Rightarrow^{A \rightarrow cAB} \$[\epsilon][b][bB][bBA][bBAc] \mid ccb\$$$

$$\Rightarrow^c \$[\epsilon][b][bB][bBA] \mid cb\$$$

$$\Rightarrow^{A \rightarrow cAB} \$[\epsilon][b][bB][bBB][bBBA][bBBAc] \mid cb\$$$

$$\Rightarrow^c \$[\epsilon][b][bB][bBB][bBBA] \mid b\$$$

$$\Rightarrow^{A \rightarrow \epsilon} \$[\epsilon][b][bB][bBB] \mid b\$ \Rightarrow^{B \rightarrow \epsilon}$$

$$\$[\epsilon][b][bB] \mid b\$ \Rightarrow^{B \rightarrow \epsilon} \$[\epsilon][b] \mid b\$ \Rightarrow^b \$[\epsilon] \mid \$$$

Example: the difference between the simplified canonical $LL(k)$ parser and the true canonical $LL(k)$ parser. (p.222)

in simplified canonical $LL(k)$ parser,

$$[\delta A]_k \mid y \rightarrow \dots$$

in true canonical $LL(k)$ parser

$$[\delta]_k [\delta A]_k \mid y \rightarrow \dots$$

The canonical LL(k) parser yields a valid left parser.

LL(k) parser \Rightarrow left parser

L8.30, L8.31(L6.29, L6.30)

left parser \Rightarrow LL(k) parser

L8.32, L8.33, L8.34(L6.31, L6.32, L6.33)

LL(k) parser \Leftrightarrow left parser

T8.35(T6.34)

Lemma 8.30 *Let M be a LL(k) parser for G . If*

$\$[\$]_k[\$X_1]_k \dots [X_1 \dots X_m]_k \mid w\$ \Rightarrow^{\pi'} \Phi$ in M .

Then

$\Phi = \$[\$]_k[\$Y_1]_k \dots [Y_1 \dots Y_n]_k \mid y\$$, $w=xy$,

$X_m \dots X_1 \xrightarrow{tm} \tau(\pi') x Y_n \dots Y_1$ in G ,

and $|\pi'| = |\tau(\pi')| + |x|$.

Proof *Induction on $|\pi'|$. (L 5.12 or L 5.27)*

Lemma 8.31 *Let M be a canonical LL(k) parser for G . Then*

(1) $L(M) \subseteq L(G)$,

(2) $\forall \pi'$ is a left parse of w in M , $\tau(\pi')$ is a left parse of w in G ,

(3) $Time_G(w) \leq Time_M(w) - |w|$.

Lemma 8.32 Let M be a LL(k) parser for G . If

$\gamma a_n \dots a_1$ is a viable suffix of G , $k: y \$^k \in \text{First}_k(\gamma^R)$.

Then $\exists \pi'$ s.t. π' is a string of an n -length shift actions of M ,

$$\$ \Phi[\gamma]_k \dots [\gamma a_n \dots a_1]_k \mid a_1 \dots a_n y \$^k \Rightarrow^{\pi'} \$ \Phi[\gamma]_k \mid y \k$

Lemma 8.33 Let M be a LL(k) parser for G .

If $X_m \dots X_1 \xrightarrow{lm}^{\pi} x Y_n \dots Y_1$ in G ,

$[A \rightarrow \alpha.\beta] \in \langle \$X_1 \dots X_m \rangle_0$,

$k: y \$ \in \text{First}_k(Y_n \dots Y_1)$, and

either $Y_n \dots Y_1 = \varepsilon$ or Y_n is nonterminal,

Then

$\tau(\pi') = \pi$, $|\pi'| = |\pi| + |x|$, and

$$\$[\$]_k [\$X_1]_k \dots [\$X_1 \dots X_m]_k \mid xy \$ \Rightarrow^{\pi'}$$

$$\$[\$]_k [\$Y_1]_k \dots [\$Y_1 \dots Y_n]_k \mid y \$$$

Proof. Induction on $|\pi|$.

Lemma 8.34 *Let M be a canonical $LL(k)$ parser for G . Then*

- (1) $L(G) \subseteq L(M)$,
- (2) $\forall \pi$ is a left parse of w in G , $\tau(\pi') = \pi$, s.t.
 π' is a left parse of w in M ,
- (3) $Time_G(w) \leq Time_M(w) + |w|$.

Theorem 8.35 *Let M be a canonical $LL(k)$ parser for G . Then*

- (1) M is a left parser for G .
- (2) $\forall w \in L(G)$, M produces *all* left parses of w .
- (3) $Time_M(w) = Time_G(w) + |w|$.

8.3.1 Lewis-Sterns Canonical LL(k) Parser

The *canonical LL(k) parser* for G is a pushdown transducer $M = ([L]_k, \Sigma, [S, \{\$^k\}]_k, \{\varepsilon\}, \Gamma, \tau)$ where

$$[L]_k = \{[X, R]_k \mid X \in V, R \subseteq \text{Follow}_k(X)\}$$

$$\begin{aligned} \Gamma = & \{[A, R]_k \mid y \rightarrow [X_n, R_n]_k \dots [X_1, R_1]_k \mid y \\ & \mid A \rightarrow X_1 \dots X_n \in P, y \in \text{First}_k(X_1 \dots X_n R), \\ & \quad R_n = R, R_i = \text{First}_k(X_{i+1} \dots X_n R)\} \quad (pa) \\ \cup & \{[a, R]_k \mid ay \rightarrow \mid y \\ & \mid a \in \Sigma\} \quad (sa) \end{aligned}$$

$$\begin{aligned} \tau([A, R]_k \mid y \rightarrow [X_n, R_n]_k \dots [X_1, R_1]_k \mid y) \\ & = A \rightarrow X_1 \dots X_n, \\ \tau([a, R]_k \mid ay \rightarrow \mid y) & = \varepsilon. \end{aligned}$$

Algorithm Construction of $[L]_k$ and Γ .

$[L]_k := \{[S, \{\$^k\}]_k\}; \Gamma := \{\};$

repeat

for $[A, R]_k \in [L]_k$ **do**

for $A \rightarrow X_1 \dots X_n \in P$ **do**

$R_n := R;$

for $i := n$ **downto** 1 **do**

$[L]_k := [L]_k \cup [X_i, R_i]_k;$

$R_{i-1} := First_k(X_i \cdot R_i)$

if $X_i \in \Sigma$ **then**

$\Gamma := \Gamma \cup \{[X_i, R_i]_k \mid X_i y \rightarrow \mid y \mid y \in R_i\}$

fi

od;

$\Gamma := \Gamma \{[A, R]_k \mid y \rightarrow [X_n, R_n]_k \dots [X_1, R_1]_k \mid y$
 $\mid y \in R_0\}$

od

od

until $[L]_k$ **does not change**

8.4 LL(k) Grammars

A grammar G is an **LL(k) grammar**, if its canonical LL(k) parser is deterministic.

K -item $[A_1 \rightarrow \cdot \omega_1, y_1]$ and $[A_2 \rightarrow \cdot \omega_2, y_2]$ of G' exhibit a **produce-produce conflict** if $A_1 = A_2$, $y_1 = y_2$ but $\omega_1 \neq \omega_2$.

Lemma 8.36 The canonical LL(k) parser for G is **nondeterministic** iff there is a viable suffix δ , and k -item $[A \rightarrow \cdot \omega_1, y]$ $[A \rightarrow \cdot \omega_2, y]$ exhibiting a produce-produce conflict s.t.

$$(a) \begin{aligned} [A \rightarrow \cdot \omega_1, y] &\in \langle \delta \omega_1^R \rangle_k \\ [A \rightarrow \cdot \omega_2, y] &\in \langle \delta \omega_2^R \rangle_k. \end{aligned} \quad (\mathbf{L6.36})$$

Proof .

(\Leftarrow) Assume (a) holds, then

$$\begin{aligned} [\delta]_k [\delta A]_k \mid y &\rightarrow [\delta]_k [\delta X_m]_k \dots [\delta X_m \dots X_1]_k \mid y, \\ &X_1 \dots X_m = \omega_1 \\ [\delta]_k [\delta A]_k \mid y &\rightarrow [\delta]_k [\delta Y_n]_k \dots [\delta Y_n \dots Y_1]_k \mid y, \\ &Y_1 \dots Y_n = \omega_2 \end{aligned}$$

$\omega_1 \neq \omega_2$, the parser is nondeterministic.

(\Rightarrow) Assume the parser is nondeterministic

Let Φ be configuration to which r_1, r_2 are applicable.
then

$$(1) r_1 = [\delta]_k [\delta A]_k \mid y \rightarrow [\delta]_k [\delta X_m]_k \dots [\delta X_m \dots X_1] \mid y,$$

$$r_2 = [\gamma]_k [\gamma B]_k \mid z \rightarrow [\gamma]_k [\gamma Y_n]_k \dots [\gamma Y_n \dots Y_1] \mid z,$$

$$(2) r_1 = [\delta]_k [\delta A]_k \mid y \rightarrow [\delta]_k [\delta X_m]_k \dots [\delta X_m \dots X_1] \mid y,$$

$$r_2 = [\gamma a]_k \mid az \rightarrow \mid z,$$

$$(3) r_1 = [\delta a]_k \mid ay \rightarrow \mid y,$$

$$r_2 = [\delta b]_k \mid bz \rightarrow \mid z.$$

r_1, r_2 are both in same configuration, (2),(3) impossible. only case (1) remains.

$$[A \rightarrow \cdot X_1 \dots X_m, y] \in \langle \delta X_m \dots X_1 \rangle_k$$

$$[B \rightarrow \cdot Y_1 \dots Y_n, z] \in \langle \gamma Y_n \dots Y_1 \rangle_k$$

$$= \langle \delta Y_n \dots Y_1 \rangle_k$$

$A=B, y=z$, but $X_1 \dots X_m \neq Y_1 \dots Y_n$, otherwise $r_1=r_2$.

Q.E.D.

Nonterminal $A \in N$ has **the LL(k) property**

if $First_k(\omega_1\delta) \cap First_k(\omega_2\delta) = \emptyset$.

$xA\delta$ is left sentential forms, $A \rightarrow \omega_1, A \rightarrow \omega_2 \in P$

(L6.37, L6.38; SLL(k) property)

Theorem 8.37 The following statements are logically equivalent. (L6.39; T5.34)

(a) G is an LL(k) grammar

(b) $[A \rightarrow \cdot \omega_1, y] \in \langle \delta \omega_1^R \rangle_k$ and

$[A \rightarrow \cdot \omega_2, y] \in \langle \delta \omega_2^R \rangle_k$

always imply that $\omega_1 = \omega_2$.

(c) $\forall A \in N$ have the LL(k) property.

(d) $S \xrightarrow{lm}^* xA\delta \xrightarrow{lm} x\omega_1\delta \xrightarrow{lm}^* xv_1$,

$S \xrightarrow{lm}^* xA\delta \xrightarrow{lm} x\omega_2\delta \xrightarrow{lm}^* xv_2$, and

$k:y_1 = k:y_2$ always imply that $\omega_1 = \omega_2$.

Proof.

(a) = (b) (L 8.36) (b) = (b') by LL(k)-validity (p. 207),

(b') $S' \xrightarrow{lm}^* x_1 A \delta^R \xrightarrow{lm} x_1 \omega_1 \delta^R, y \in First_k(\omega_1 \delta^R)$,

$S' \xrightarrow{lm}^* x_2 A \delta^R \xrightarrow{lm} x_2 \omega_2 \delta^R, y \in First_k(\omega_2 \delta^R)$,

always imply that $\omega_1 = \omega_2$.

$G' \rightarrow G, S' \rightarrow S, x_2 \rightarrow x_1 (S' \xrightarrow{lm}^* x_1 A \delta^R, A \rightarrow \omega_2 \in P)$

$(b'') S \xrightarrow{lm}^* xA\delta^R \xrightarrow{lm} x\omega_1\delta^R, y \in \text{First}_k(\omega_1\delta^R),$

$S \xrightarrow{lm}^* xA\delta^R \xrightarrow{lm} x\omega_2\delta^R, y \in \text{First}_k(\omega_2\delta^R),$

always imply that $\omega_1 = \omega_2$. $(b'') = (c) = (d)$.

Theorem 8.38 for $1 \leq l \leq k$,

$LA(k+1)LL(l)$ grammars $\subset LA(k+1)LL(l+1)$ grammars.

for $0 \leq l \leq k$,

$LA(k)LL(l)$ grammars $\subset LA(k+1)LL(l)$ grammars.

Proof. examples (p. 231).

Equivalence of LL(1) and SLL(1)

Lemma 8.39 Every LL(1) grammar is an SLL(1) grammar.

Proof. Let G be an LL(1) grammar and consider,

$$(1) S \xrightarrow{lm}^* x_1 A \delta_1 \xrightarrow{lm} x_1 \omega_1 \delta_1 \xrightarrow{lm}^* x_1 v_1,$$

$$S \xrightarrow{lm}^* x_2 A \delta_2 \xrightarrow{lm} x_2 \omega_2 \delta_2 \xrightarrow{lm}^* x_2 v_2.$$

Assume

$$\omega_1 \xrightarrow{lm}^* y_1, \delta_1 \xrightarrow{lm}^* z_1, y_1 z_1 = v_1$$

$$\omega_2 \xrightarrow{lm}^* y_2, \delta_2 \xrightarrow{lm}^* z_2, y_2 z_2 = v_2$$

$$(2) S \xrightarrow{lm}^* x_1 A \delta_1 \xrightarrow{lm} x_1 \omega_1 \delta_1 \xrightarrow{lm}^* x_1 y_1 z_1$$

$$S \xrightarrow{lm}^* x_1 A \delta_1 \xrightarrow{lm} x_1 \omega_2 \delta_1 \xrightarrow{lm}^* x_1 y_2 z_1, \text{ and}$$

$$(3) S \xrightarrow{lm}^* x_2 A \delta_2 \xrightarrow{lm} x_2 \omega_1 \delta_2 \xrightarrow{lm}^* x_2 y_1 z_2$$

$$S \xrightarrow{lm}^* x_2 A \delta_2 \xrightarrow{lm} x_2 \omega_2 \delta_2 \xrightarrow{lm}^* x_2 y_2 z_2$$

if $y_1 \neq \varepsilon$, then $1:v_1 = 1:v_2 \Rightarrow 1:y_1 = 1:y_2 z_2$. (3)

if $y_2 \neq \varepsilon$, then $1:v_1 = 1:v_2 \Rightarrow 1:y_1 z_1 = 1:y_2$. (2)

if $y_1 = \varepsilon, y_2 = \varepsilon$, then $1:v_1 = 1:v_2 \Rightarrow 1:z_1 = 1:z_2$.

\therefore If $G = LL(1)$,

(1) and $1:v_1 = 1:v_2$ implies $\omega_1 = \omega_2$.

$\therefore G$ is SLL(1).

(T 5.34) **Q.E.D.**

Lemma 8.40 $k \geq 0$, any SLL(k) grammar is an LALL(k) grammar and any LALL(k) grammar is an LL(k) grammar.

Theorem 8.41 The classes of SLL(1), LALL(1), and LL(1) grammars coincide.

Theorem 8.42 $k > 1$, the class of SLL(k) grammar is properly contained in the class of LALL(k) grammar.
Proof. counter example grammar which is LALL(k) but not SLL(k), for $k > 1$. (p.233)

Properly containment of theorem 8.42 are properties of grammar only, but not properties of language. i.e. any LL(k) grammar can be transformed into an equivalent SLL(k) grammar.

Transformation of LL(k) grammar into SLL(k)

$$T_k(G) = (\{[A, R] \mid A \in N, R \subseteq \text{Follow}_k(A)\}, \Sigma,$$

$$P_{T(k)}, [S, \{\$^k\}]), \text{ where}$$

$$P_{T(k)} = [A, R] \rightarrow X_1' \dots X_m', \text{ and}$$

$$X_i' = [X_i, \text{First}_k(X_{i+1} \dots X_m R)], X_i \in N,$$

$$X_i, X_i \in \Sigma.$$

Nonterminal A is split into several nonterminals $[A, R_1] \dots [A, R_n]$, each representing an occurrence of the original nonterminals in a particular context.

Let h_k homomorphism s.t.

$$h_k([A, R] \rightarrow X_1' \dots X_m') = A \rightarrow X_1 \dots X_m.$$

$$G_{abL}: LL(2)$$

$$S \rightarrow aAab \mid bAb$$

$$A \rightarrow cAB \mid \varepsilon \mid a$$

$$B \rightarrow \varepsilon.$$

$$T_k(G_{abL}): SLL(2)$$

$$[S, \{\$^2\}] \rightarrow a [A, \{ab\}] a b \mid b [A, \{b\$\}] b$$

$$[A, \{ab\}] \rightarrow c [A, \{ab\}] [B, \{ab\}] \mid \varepsilon \mid a$$

$$[A, \{b\$\}] \rightarrow c [A, \{b\$\}] [B, \{b\$\}] \mid \varepsilon \mid a$$

$$[B, \{ab\}] \rightarrow \varepsilon$$

$$[B, \{b\$\}] \rightarrow \varepsilon$$

Covering LL(k) grammars by SLL(k) grammars

(1) $T_k(G)$ left-to-left covers G .

T8.45(L8.43, L8.44)

(2) $T_k(G)$ is SLL(k), iff G is LL(k).

T8.47

T8.48, T8.49

Lemma 8.43

Let $[A, R] \Rightarrow_{lm}^{\pi} xX_1' \dots X_n'$ in $T_k(G)$. Then

$$A \xRightarrow{lm}^{h_k(\pi')} xX_1 \dots X_n \text{ in } G.$$

where $X_1' = [X_i, \text{First}_k(X_{i+1} \dots X_n R)]$, if $X_i \in N$,
 X_i , if $X_i \in \Sigma$.

Proof. induction on $|\pi'|$.

Lemma 8.44 Let

$A \Rightarrow_{lm}^{\pi} xX_1 \dots X_n$ in G , an either $X_1 \dots X_n = \varepsilon$ or $X_1 \in N$.

Then $\exists \pi' \in \hat{P}^* . \exists . h_k(\pi') = \pi$,

$$[A, R] \Rightarrow_{lm}^{\pi'} xX_1' \dots X_n' \text{ in } T_k(G).$$

Proof. By induction on $|\pi|$.

Theorem 8.45 $T_k(G)$ left-to-left covers G w.r.t. h_k .

Corollary 8.46 If (M, τ) is a left parser of $T_k(G)$, then $(M, \tau h_k)$ is a left parser of G .

Theorem 8.47 $T_k(G)$ is SLL(k) whenever G is LL(k).

G is LL(k) whenever $T_k(G)$ is so.

Proof. (proof of the first part of theorem)

Let $[A, R] \rightarrow X'_1 \dots X'_m$, $[A, R] \rightarrow Y'_1 \dots Y'_n \in P_{Tk}$

$0 \leq m \leq p$, $0 \leq n \leq q$, s.t.

$$(1) [S, \{\$\}] \xrightarrow{lm}^* x_1[A, R] X'_{m+1} \dots X'_p \\ \xrightarrow{lm} x_1 X'_1 \dots X'_m X'_{m+1} \dots X'_p \xrightarrow{lm}^* x_1 y_1,$$

$$[S, \{\$\}] \xrightarrow{lm}^* x_2[A, R] Y'_{n+1} \dots Y'_q \\ \xrightarrow{lm} x_2 Y'_1 \dots Y'_n Y'_{n+1} \dots Y'_q \xrightarrow{lm}^* x_2 y_2,$$

$k: y_1 = k: y_2$ implies $X'_1 \dots X'_m \stackrel{?}{=} Y'_1 \dots Y'_n$.

$$(4) X'_1 \dots X'_m \xrightarrow{lm}^* u_1, X'_{m+1} \dots X'_p \xrightarrow{lm}^* v_1, u_1 v_1 = y_1.$$

$$Y'_1 \dots Y'_n \xrightarrow{lm}^* u_2, Y'_{n+1} \dots Y'_q \xrightarrow{lm}^* v_2, u_2 v_2 = y_2.$$

$$(5) X_1 \dots X_m \xrightarrow{lm}^* u_1, X_{m+1} \dots X_p \xrightarrow{lm}^* v_1.$$

$$Y_1 \dots Y_n \xrightarrow{lm}^* u_2, Y_{n+1} \dots Y_q \xrightarrow{lm}^* v_2.$$

$$(6) \text{First}_k(X_{m+1} \dots X_p \$) = R = \text{First}_k(Y_{n+1} \dots Y_q \$)$$

$$(7) S \xrightarrow{lm}^* x_1 A X_{m+1} \dots X_p. (L 8.43)$$

(8) $X_{m+1}\dots X_p \xRightarrow{\overline{lm}}^* v_2'$ s.t. $k:v_2 = k:v_2'$, so

$k: u_1v_1 = k: y_1 = k: y_2 = k: u_2v_2 = k: u_2v_2'$ then

$S \xRightarrow{\overline{lm}}^* x_1AX_{m+1}\dots X_p \xRightarrow{\overline{lm}} x_1X_1\dots X_mX_{m+1}\dots X_p \xRightarrow{\overline{lm}}^* x_1y_1$,

$S \xRightarrow{\overline{lm}}^* x_1AX_{m+1}\dots X_p \xRightarrow{\overline{lm}} x_1Y_1\dots Y_nX_{m+1}\dots X_p \xRightarrow{\overline{lm}}^* x_1y_2'$,

$k: y_1 = k: y_2'$ where $y_2' = u_2v_2'$

Thus if G is LL(k) (i.e. $X_1\dots X_m = Y_1\dots Y_n$) then

$T_k(G)$ is SLL(k) (i.e. $X'_1\dots X'_m = Y'_1\dots Y'_n$).

(proof of the LL(k)-ness of $T_k(G)$ implies the LL(k)-ness of G .) Assume that (8) is true, then

(9) $[S, \{\$\}] \xRightarrow{\overline{lm}}^* x_1[A, R]X'_{m+1}\dots X'_p$.

(10) $[S, \{\$\}] \xRightarrow{\overline{lm}}^* x_1[A, R]X'_{m+1}\dots X'_p$

$\xRightarrow{\overline{lm}}^* x_1X'_1\dots X'_mX'_{m+1}\dots X'_p \xRightarrow{\overline{lm}}^* x_1y_1$,

$[S, \{\$\}] \xRightarrow{\overline{lm}}^* x_1[A, R]X'_{m+1}\dots X'_p$

$\xRightarrow{\overline{lm}}^* x_1Y'_1\dots Y'_nX'_{m+1}\dots X'_p \xRightarrow{\overline{lm}}^* x_1y_2'$,

$k: y_1 = k: y_2'$.

If $T_k(G)$ is LL(k) (i.e. $X'_1\dots X'_m = Y'_1\dots Y'_n$),

then G is LL(k) (i.e. $X_1\dots X_m = Y_1\dots Y_n$).

Q.E.D.

Theorem 8.47 $T_k(G)$ is SLL(k) whenever G is LL(k).
 G is LL(k) whenever $T_k(G)$ is so.

Proof.

Consider a derivation in G

$$S \Rightarrow_{lm}^* xAX_{m+1}\dots X_p. \text{ Then}$$

$$[S, \$^k] \Rightarrow_{lm}^* x[A, R]X'_{m+1}\dots X'_p$$

$$\begin{aligned} \therefore X'_i &= [X_i, \text{First}_k(X_{i+1}\dots X_n R)], \text{ if } X_i \in N, \\ &= X_i, \text{ if } X_i \in \Sigma. \end{aligned}$$

$$\therefore \text{First}_k(X'_{m+1}\dots X'_p) = \text{First}_k(X_{m+1}\dots X_p)$$

Assume G is LL(k).

$$\forall A \in N, \text{First}_k(\omega_1 \delta) \cap \text{First}_k(\omega_2 \delta) = \emptyset.$$

$$A \rightarrow \omega_1 \neq A \rightarrow \omega_2, \exists S \Rightarrow_{lm}^* xA\delta.$$

$$\therefore \exists [S, \$^k] \Rightarrow_{lm}^* x[A, R]\Delta \text{ where } \tau_k(\Delta) = \delta.$$

where $\tau_k([A, \text{First}_k($

Theorem 8.48 $k \geq 1$, any grammar can be transformed into an equivalent grammar that left to left covers the original grammar, and is SLL(k) iff the original is LL(k). In fact, the transformed grammar is structurally equivalent to the original grammar.

Theorem 8.49 For any k , the families of LL(k) language, LALL(k) languages, and SLL(k) languages coincide.

Theorem 8.50 Any LL(k) grammar is unambiguous.

Proof. Let $S \xrightarrow[lm]{\pi} w$ and $S \xrightarrow[lm]{\pi} w$.

Assume the contrary $\pi_1 = \pi r_1 \pi'_1$, $\pi_2 = \pi r_2 \pi'_2$, $r_1 \neq r_2$

s.t. $S \xrightarrow[lm]{\pi} xA\delta \xrightarrow[lm]{r_1 \pi'_1} xy$,

$S \xrightarrow[lm]{\pi} xA\delta \xrightarrow[lm]{r_2 \pi'_2} xy$, where $xy = w$.

As G is LL(k), $r_1 = r_2$. (T 8.37)

Thus $\pi_1 = \pi_2$ and G is unambiguous. **Q.E.D.**

Theorem 8.51 A reduced left recursive grammar is not LL(k), $k \geq 0$.

Proof. (T5.40, C5.41).

Comparison on the classes of LL(k) and LR(k) grammars.

$$S \Rightarrow^* xAz \Rightarrow x\alpha z \Rightarrow^* xyz$$

LR(k) parser

reduce $A \rightarrow \alpha$ after scanning xy and seeing $k:z$

LL(k) parser

produce $A \rightarrow \alpha$ after scanning x and seeing $k:yz$

LL(k) grammars \subset LR(k) grammar

*rightmost derivation \Rightarrow leftmost derivation
(viable suffix)*

*leftmost derivation \Rightarrow rightmost derivation
(viable prefix)*

Lemma 8.52 Let G is LL(k),

$$S \xrightarrow{\overline{lm}}^{\pi_1} x_1 A \delta_1 \xrightarrow{\overline{lm}}^{r_1} x_1 \omega_1 \delta_1, \omega_1 \xrightarrow{\overline{lm}}^{\pi'_1} v_1, \delta_1 \xrightarrow{\overline{lm}}^* y_1,$$

$$S \xrightarrow{\overline{lm}}^{\pi_2} x_2 A \delta_2 \xrightarrow{\overline{lm}}^{r_2} x_2 \omega_2 \delta_2, \omega_2 \xrightarrow{\overline{lm}}^{\pi'_2} v_2, \delta_2 \xrightarrow{\overline{lm}}^* y_2,$$

$$x_2 v_2 = x_1 v_1 v \text{ and } k:y_1 = k:vy_2.$$

Then either $\pi_1 r_1 \pi'_1$ is a prefix of $\pi_2 r_2 \pi'_2$ or vice versa.

Proof. For the sake of contradiction $\pi_1 r_1 \pi'_1$ and $\pi_2 r_2 \pi'_2$ are not prefix of each other.

(case1) $\pi_1 = \pi_2$ and $r_1 \neq r_2$.

(case2) $\pi_1 = \pi r'_1 \pi''_1, \pi_2 = \pi r'_2 \pi''_2$, where $r'_1 \neq r'_2$

(case3) $\pi_1 r_1 = \pi_2 r_2, \pi'_1 = \pi r'_1 \pi''_1$, and $\pi'_2 = \pi r'_2 \pi''_2$,
where $r'_1 \neq r'_2$

Each case is in contradiction to the assumption that G is LL(k).

Lemma 8.53 Let LL(k) grammar G has

$$S \xrightarrow{\overline{lm}}^* x_1 A_1 \delta_1 \xrightarrow{\overline{lm}} x_1 \omega_1 \delta_1, \omega_1 \xrightarrow{\overline{lm}}^* v_1, \delta_1 \xrightarrow{\overline{lm}}^* y_1,$$

$$S \xrightarrow{\overline{lm}}^* x_2 A_2 \delta_2 \xrightarrow{\overline{lm}} x_2 \omega_2 \delta_2, \omega_2 \xrightarrow{\overline{lm}}^* v_2, \delta_2 \xrightarrow{\overline{lm}}^* y_2,$$

$$x_2 v_2 = x_1 v_1 v \text{ and } k:y_1 = k:vy_2.$$

Then either $\delta_1 \Rightarrow^* v \delta_2$ or $v \delta_2 \Rightarrow^* \delta_1$.

Proof. (L 8.52).

$S \Rightarrow_{rm}^+ \gamma Ay$ vs. $S \Rightarrow_{lm}^+ xA\delta^R$. ($\gamma \Rightarrow^* x$ and $\delta^R \Rightarrow^* y$)
viable prefix γ induced by $S \Rightarrow_{lm}^+ xA\delta^R$
viable suffix δ^R induced by $S \Rightarrow_{rm}^+ \gamma Ay$

Let γ be a viable prefix of G , $\pi \in P^*$. Then γ is a viable prefix **induced** by π , if

(p1) $\gamma = \varepsilon$ and $\pi = \varepsilon$ or

(p2) $\gamma = \eta\alpha$, $\pi = \pi_1 r \pi_2$, . \exists . η is a viable prefix induced by π_1 and

$$S \xRightarrow_{lm}^{\pi_1} xA\phi^R \xRightarrow_{lm}^r x\alpha B\beta\phi^R, \alpha \xRightarrow_{lm}^{\pi_2} u.$$

Let δ be a viable suffix of G , $\hat{\pi} \in P^*$ then δ is a viable suffix **induced** by $\hat{\pi}$ if

(s1) $\delta = \varepsilon$ and $\hat{\pi} = \varepsilon$ or

(s2) $\delta = \phi\beta^R$, $\hat{\pi} = \hat{\pi}_1 r \hat{\pi}_2$ s.t. ϕ is a viable suffix induced by $\hat{\pi}_1$ and

$$S \xRightarrow_{rm}^{\hat{\pi}_1} \eta Ay \xRightarrow_{rm}^r \eta\alpha B\beta y, \beta \xRightarrow_{rm}^{\hat{\pi}_2} v.$$

(refer to figures in p.243-244)

Lemma 8.54 *No rule string in P^* induces more than one viable prefix (resp. viable suffix).*

Proof. *Assume π induces two distinct viable prefixes γ_1 and γ_2 .*

$$S \xrightarrow{lm}^{\pi_1} x_1 A_1 \phi_1^R \xrightarrow{lm}^{r_1} x_1 \alpha_1 B_1 \beta_1 \phi_1^R, \quad \alpha_1 \xrightarrow{lm}^{\pi_1'} u_1,$$

$$S \xrightarrow{lm}^{\pi_2} x_2 A_2 \phi_2^R \xrightarrow{lm}^{r_2} x_2 \alpha_2 B_2 \beta_2 \phi_2^R, \quad \alpha_2 \xrightarrow{lm}^{\pi_2'} u_2$$

$$x_1 u_1 B_1 \beta_1 \phi_1^R = x_2 u_2 B_2 \beta_2 \phi_2^R \quad (\pi_1 r_1 \pi_1' = \pi_2 r_2 \pi_2')$$

therefore $x_1 u_1 = x_2 u_2$, $B_1 = B_2$, $\beta_1 \phi_1^R = \beta_2 \phi_2^R$.

If $\pi_1 \neq \pi_2$, then $\pi_1 r_1 \pi_1' = \pi_2 r_2 \pi_2'$ imply that

$\pi_1 r_1$ is a prefix of π_2 or $\pi_2 r_2$ is a prefix of π_1 .

Assume $\pi_1 r_1$ is a prefix of π_2 , $\pi_2 = \pi_1 r_1 \pi'$,

$\pi' r_2 \pi_2' = \pi_1'$, then

$$S \xrightarrow{lm}^{\pi_1 r_1} x_1 \alpha_1 B_1 \beta_1 \phi_1^R \xrightarrow{lm}^{\pi'} x_2 A_2 \phi_2^R$$

$$\xrightarrow{lm}^{r_2} x_2 \alpha_2 B_2 \beta_2 \phi_2^R \xrightarrow{lm}^{\pi_2'} x_2 u_2 B_2 \beta_2 \phi_2^R$$

$B_1 \beta_1 \phi_1^R$ can not be involved in the derivation

that uses $\pi' r_2 \pi_2'$.

$B_1 \beta_1 \phi_1^R$ can not be involved in the derivation

$$\underline{x_1} \underline{\alpha_1} \underline{B_1} \underline{\beta_1} \underline{\phi_1^R} \xrightarrow{lm}^{\pi'} \underline{x_2} \underline{A_2} \underline{\phi_2^R} \xrightarrow{lm}^{r_2} \underline{x_2} \underline{\alpha_2} \underline{B_2} \underline{\beta_2} \underline{\phi_2^R}.$$

$B_1\beta_1\phi_1^R$ is a suffix of ϕ_2^R , then contradiction
because $\beta_1\phi_1^R = \beta_2\phi_2^R$. **Q.E.D.**

Lemma 8.55 *Let G has at least one rule with left handside S , and*

$$(a) S \xrightarrow[lm]{\pi} xA\delta^R \text{ and } \delta^R \Rightarrow^* y.$$

Then π induces some viable prefix γ , and δ is a viable suffix induced by $\hat{\pi} \in P^$, s.t.*

$$(b) S \xrightarrow[rm]{\hat{\pi}} \gamma Ay \text{ and } \gamma \Rightarrow^* x$$

Proof. *Base case, if $\pi = \varepsilon$, immediate by definition. If $\pi \neq \varepsilon$ and $\pi = \pi_1 r \pi_2$, $r = A_1 \rightarrow \alpha A \beta$ then*

$$(1) S \xrightarrow[lm]{\pi_1} x_1 A_1 \phi^R \xrightarrow[lm]{r} x_1 \alpha A \beta \phi^R, x_1 \alpha \xrightarrow[lm]{\pi_2} x$$

$$(2) S \xrightarrow[lm]{\pi_1} x_1 A_1 \phi^R, \phi^R \Rightarrow^* y_1 \text{ implies (by I.H.)}$$

π_1 induces some viable prefix η ,

ϕ is a viable suffix induced by $\hat{\pi}_1 \in P^$, and*

$$(3) S \xrightarrow[rm]{\hat{\pi}_1} \eta A_1 y_1 \text{ and } \eta \Rightarrow^* x_1.$$

$$(4) S \xrightarrow[rm]{\hat{\pi}_1} \eta A_1 y_1 \xrightarrow[rm]{r} \eta \alpha A \beta y_1 \text{ and } \beta \xrightarrow[rm]{\hat{\pi}_2} v$$

$$(5) S \xrightarrow[rm]{\hat{\pi}} \eta \alpha A v y_1 = \gamma Ay, \gamma = \eta \alpha \Rightarrow^* x_1 u = x$$

$$(6) S \xrightarrow[lm]{\pi_1} x_1 A_1 \phi^R \xrightarrow[lm]{r} x_1 \alpha A \beta \phi^R, \text{ and } \alpha \xrightarrow[lm]{\pi_2} u$$

in (4) $\delta = \phi \beta^R$, ϕ is a viable suffix induced by $\hat{\pi}_1$

δ is a viable suffix induced by $\hat{\pi} = \hat{\pi}_1 r \hat{\pi}_2$

in (6) $\gamma = \eta \alpha$, η is a viable suffix induced by π_1

γ is a viable suffix induced by $\pi = \pi_1 r \pi_2$

Q.E.D.

Lemma 8.56 *Let G has at least one rule with left handside S , and*

$$(a) S \xrightarrow[\text{rm}]{\hat{\pi}} \gamma Ay \text{ and } \gamma \Rightarrow^* x.$$

Then $\hat{\pi}$ reduces some viable suffix δ , and

γ is a viable prefix induced by $\pi \in P^$, and*

$$(b) S \xrightarrow[\text{lm}]{\pi} xA\delta^R \text{ and } \delta^R \Rightarrow^* y.$$

Theorem 8.57 $k \geq 0$, any reduced LL(k) grammar is an LR(k) grammar.

Proof. $S \Rightarrow^+ S$ impossible. Let

$$(1) S \xRightarrow{rm}^{\hat{\pi}_1} \gamma_1 A_1 y_1 \xRightarrow{rm}^{r_1} \gamma_1 \omega_1 y_1 = \gamma y_1$$

$$S \xRightarrow{rm}^{\hat{\pi}_2} \gamma_2 A_2 y_2 \xRightarrow{rm}^{r_2} \gamma_2 \omega_2 y_2 = \gamma v y_2$$

$$\text{and } k:y_1 = k:vy_2$$

$$(2) S \xRightarrow{rm}^{\hat{\pi}_1} \gamma_1 A_1 y_1, \gamma_1 \Rightarrow^* x_1 \text{ (G is reduced)}$$

$$S \xRightarrow{rm}^{\hat{\pi}_2} \gamma_2 A_2 y_2, \gamma_2 \Rightarrow^* x_2$$

$$(3) S \xRightarrow{lm}^{\pi_1} x_1 A_1 \delta_1^R, \delta_1^R \Rightarrow^* y_1 \text{ (L 8.56)}$$

$$S \xRightarrow{lm}^{\pi_2} x_2 A_2 \delta_2^R, \delta_2^R \Rightarrow^* y_2$$

where δ_1, δ_2 are viable suffix induced by $\hat{\pi}_1$ and $\hat{\pi}_2$ respectively, γ_1, γ_2 are viable prefix induced by π_1 and π_2

$$(4) S \xRightarrow{lm}^{\pi_1} x_1 A_1 \delta_1^R \xRightarrow{lm}^{r_1} x_1 \omega_1 \delta_1^R, \omega_1 \Rightarrow^* v_1, \delta_1^R \Rightarrow^* y_1$$

$$S \xRightarrow{lm}^{\pi_2} x_2 A_2 \delta_2^R \xRightarrow{lm}^{r_2} x_2 \omega_2 \delta_2^R, \omega_2 \Rightarrow^* v_2, \delta_2^R \Rightarrow^* y_2$$

$$x_2 v_2 = x_1 v_1 v \text{ and } k:y_1 = k:vy_2 \text{ (from 3, 1)}$$

If G is LL(k) grammar. Then either δ_1^R derives $v\delta_2^R$ or $v\delta_2^R$ derives δ_1^R (L 8.53).

(case 1) δ_1^R derives $v\delta_2^R$

$$(5) S \xrightarrow{lm} \pi_1 x_1 A_1 \delta_1^R \text{ and } \delta_1^R \xRightarrow{*} v\delta_2^R \xRightarrow{*} vy_2$$

$$(6) S \xrightarrow{rm}^* \gamma_1 A_1 v y_2 \xrightarrow{rm} \gamma_1 \omega_1 v y_2 = \gamma v y_2$$

$$S \xrightarrow{rm}^* \gamma_2 A_2 y_2 \xrightarrow{rm} \gamma_2 \omega_2 y_2 = \gamma v y_2$$

G is LL(k) and hence unambiguous, then

$\gamma_1 A_1 v y_2 = \gamma_2 A_2 y_2$, hence $\gamma_1 = \gamma_2$, $A_1 = A_2$, $v = \epsilon$,

$\gamma_1 \omega_1 = \gamma = \gamma v = \gamma_2 \omega_2 = \gamma_1 \omega_2$ implies $\omega_1 = \omega_2$.

(1) implies $\gamma_1 = \gamma_2$, $A_1 = A_2$, and $\omega_1 = \omega_2$, so G is LR(k).

(case 2) $v\delta_2^R$ derives δ_1^R

the proof is analogous to that of case 1.

Q.E.D.

Theorem 8.58 $k \geq 0$, the class of reduced LL(k) grammars is properly contained in the class of reduced LR(k) grammars

Proof. By Theorem 8.57 and LR(0) grammar may be left-recursive.

Proposition 8.59 $k \geq 0$, the family of LL(k) languages is properly contained in the family of LR(1) languages

Theorem 8.60 $k \geq 1$, the class of SLL(k) grammars is incompatible with the class of SLR(k) grammars, and the class of LALL(k) grammars is incompatible with the class of LALR(k) grammars

Proof. example grammars (p. 248).

8.5 Construction of LL(k) Parsers

Construction of canonical LL(0) machine.

Let $G = (V, \Sigma, P, S)$. $M_0 = (I_0 \cup \{q_s\}, V, q_s, I_f, \delta_0)$ is the **nondeterministic LL(0) machine** for G , where

I_0 : set of 0-items, V : input alphabet,

$q_s \notin I_0$: initial state, $I_f = I_0$: set of final states,

$\delta_0: (I_0 \cup \{q_s\}) \times (V \cup \{\epsilon\}) \rightarrow 2^{I_0}$ of the form;

$$q_s \rightarrow [S \rightarrow \omega \cdot],$$

$$[A \rightarrow \alpha X \cdot \beta] X \rightarrow [A \rightarrow \alpha \cdot X \beta], X \in V, \text{ and}$$

$$[A \rightarrow \alpha B \cdot \beta] \epsilon \rightarrow [B \rightarrow \omega \cdot]$$

Lemma 8.61 *The set of viable suffixes of any G is the language accepted by M_0 of G , and for any viable suffix γ ,*

$$\langle \gamma \rangle_0 = \{q \mid \exists q \in I_0: q \in \delta'_0(q_s, \gamma)\}$$

where $\delta'_0(q, \alpha X) = \{p \mid \exists \alpha \in V^*, X \in V, r \in \delta'_0(q, \alpha): p \in \delta_0(r, X)\}$.

Theorem 8.62 *Deterministic LL(0) machine for any grammar G can be computed in time $O(2^{|G|+2\log|G|})$. (T 3.30)*

Let $G = (V, \Sigma, P, S)$. $M_1 = (I_1 \cup \{q_s\}, V, q_s, I_f, \delta_1)$ is the **nondeterministic LL(1) machine** for G where

I_1 : set of 1-items, V : input alphabet,

$q_s \notin I_1$: initial state, $I_f = I_1$: set of final states,

$\delta_1: (I_1 \cup \{q_s\}) \times (V \cup \{\epsilon\}) \rightarrow 2^{I_1}$ of the form;

$$q_s \rightarrow [S \rightarrow \omega; \epsilon],$$

$$[A \rightarrow \alpha X \cdot \beta, y] X \rightarrow [A \rightarrow \alpha \cdot X \beta, z], \quad X \in V, \\ z \in \text{First}_1(Xy).$$

$$[A \rightarrow \alpha B \cdot \beta, y] \rightarrow [B \rightarrow \omega; y]$$

Lemma 8.63 *The set of viable suffixes of any grammar G is the language accepted by M_1 of G , and for any viable suffix γ ,*

$$\langle \gamma \rangle_1 = \{q \mid \exists q \in I_1: q \in \delta'_1(q_s, \gamma)\}$$

where $\delta'_1(q, \alpha X) = \{p \mid \exists \alpha \in V^*, X \in V, r \in \delta'_1(q, \alpha): \\ p \in \delta_1(r, X)\}$.

Theorem 8.64 *The canonical LL(1) machine for any grammar G can be constructed in time*

$$O(2^{2|G|+3\log|G|}).$$

Theorem 8.65 *The canonical LL(1) parser of any grammar G can be constructed in time*

$$O(2^{2|G|+3\log|G|}).$$

Construction of LALL(1) Parsers

To construct LALL(1) parser of a grammar G
 construct canonical LL(0) machine of its
 $\$$ -augmented grammar G' of G
 and lookahead symbols of states.

For state q in the canonical LL(0) machine of G'
 $Lall(q) = \{b \in \Sigma \cup \{\$\} \mid [A \rightarrow \alpha \cdot \beta, b] \in \langle [\gamma]_0 \rangle_1\}$
 where $Lall(q)$ is **Lall(1) lookahead set of q** .

Fact 8.66 The LALL(1) parser has a produce action
 of the form

$$[\delta]_0[\delta A]_0 \mid b \rightarrow [\delta]_0[\delta X_n]_0 \dots [\delta X_n \dots X_1]_0 \mid b$$

iff $[A \rightarrow \cdot X_1 \dots X_n] \in \langle \delta X_n \dots X_1 \rangle_0$ and
 $b \in Lall(\langle \delta X_n \dots X_1 \rangle_0)$

Lemma 8.67 For any state q in the canonical LL(0)
 machine,

$$Lall(q) = \{b \in First_1(\gamma^R) \mid \gamma \in \$V^*, \langle \gamma \rangle_0 = q\}$$

In other words, $Lall(q)$ is the union of all sets
 $First_1(\gamma^R)$.

Proof. (F 8.13)

Computation of Lall(1) lookahead set

q contains-item $A \rightarrow \alpha \cdot \beta$, if $A \rightarrow \alpha \cdot \beta \in q$.

(q, A) goes-to $Goto(q, A)$, if $Goto(q, A) \neq \emptyset$

q has-null-transition (q, B) , if $Goto(q, B) \neq \emptyset$ and B is nullable.

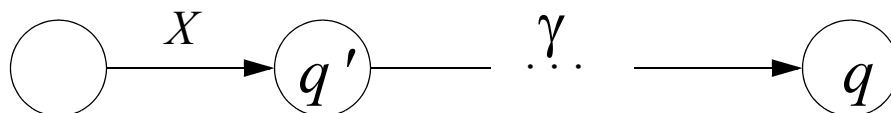
has-Lall-lookahead =

$(\text{goes-to}^{-1} \text{ has-null-transition}^{-1})^* \text{ contains-item}$
 $\text{points first-of}^{-1}$

where

$A \rightarrow \alpha \cdot X \beta$ points X

a **first-of** X if and only if $a \in \text{First}_1(X)$



Portion of a canonical LL(0) machine.
 If γ is nullable and $a \in \text{First}_1(X)$, then

q $(\text{goes-to}^{-1} \text{ has-null-transition}^{-1})^* q'$
 contains-item $A \rightarrow \alpha \cdot X \beta$ points X **first-of** a

that is, q has-Lall-lookahead a

Theorem 8.68 Let G be a reduced grammar. $a \in \Sigma$,
 $a \in \text{Lall}(q)$ iff q has-Lall-lookahead a .

Proof. (\Leftarrow)

q has-Lall-lookahead $a =$

q (goes-to⁻¹ has-null-transition⁻¹)* q'

contains-item points first-of⁻¹ a for some q'

If $q' = \langle \gamma \rangle_0$ for a viable suffix γ , then

$q = \langle \gamma \delta \rangle_0$, $\delta \Rightarrow^* \epsilon$.

q' contains-item points first-of⁻¹ a

implies that $a \in \text{First}_1(\gamma^R)$.

$\therefore a \in \text{Lall}(q)$, since $a \in \text{First}_1(\delta^R \gamma^R)$, δ is nullable.

(\Rightarrow) Assume that $a \in \text{Lall}(q)$,

$\exists \gamma$ of G' such that $\langle \gamma \rangle_0 = q$ and

$a \in \text{First}_1(\gamma^R)$.

$a \in \text{First}_1(\gamma^R)$ implies that $\gamma^R = \alpha^R X \beta^R$, where

$\alpha^R \Rightarrow^* \epsilon$, $a \in \text{First}_1(X)$.

q (goes-to⁻¹ has-null-transition⁻¹)* q'

contains-item points first-of⁻¹ a

where $q' = \langle \beta X \rangle_0$.

$\therefore q$ has-Lall-lookahead a . **Q.E.D.**

8.6 Non-Left-Recursive Grammatical Covers

A reduced left-recursive grammar cannot be an LL(k) grammar. (T 8.51)

*Nonterminal A is directly left-recursive
if the grammar has a rule $A \rightarrow A\alpha$.*

Transformations of left-recursion to right-recursion

$$G: A \rightarrow A\alpha$$

$$| \beta$$

Consider

$$A \Rightarrow_{lm}^* A\alpha^n \Rightarrow_{lm} \beta\alpha^n \Rightarrow_{lm}^* y\alpha^n \Rightarrow_{lm}^* yx_i^n = w$$

$$\pi_L(w) = (A \rightarrow A\alpha)^n \cdot (A \rightarrow \beta) \cdot \pi_L(y) \cdot \pi_L(x_i)^n.$$

$$A \Rightarrow_{rm} A\alpha \Rightarrow_{rm}^* Ax_n \Rightarrow_{rm}^* A(x_i)^n \alpha^n \Rightarrow_{rm} \beta(x_i)^n \Rightarrow_{rm}^* yx_i^n.$$

$$\pi_R(w) = \pi_R(y) (A \rightarrow \beta) (\pi_R(x_i) (A \rightarrow A\alpha))^n.$$

$$\hat{G}: A \rightarrow \beta A'$$

$$A \rightarrow \alpha A' | \varepsilon$$

$$A \Rightarrow_{lm} \beta A' \Rightarrow_{lm}^* yA' \Rightarrow_{lm} y\alpha A' \Rightarrow_{lm}^* y\alpha^n \Rightarrow_{lm}^* yx_i^n = w$$

$$\hat{\pi}_L(w) = (A \rightarrow \beta A') \cdot \pi_L(y) \cdot ((A' \rightarrow \alpha A') \cdot \pi_L(x_i))^n \cdot (A' \rightarrow \varepsilon).$$

$$A \Rightarrow_{rm} \beta A' \Rightarrow_{rm} \beta\alpha A' \Rightarrow_{lm}^* \beta\alpha^n A' \Rightarrow_{rm} \beta\alpha^n \Rightarrow_{rm}^* yx_i^n.$$

$$\hat{\pi}_R(w) = \pi_R(y) \cdot (\pi_R(x_i))^n \cdot (A' \rightarrow \varepsilon) \cdot (A' \rightarrow \alpha A')^n \cdot (A \rightarrow \beta A').$$

No way to make a homomorphism that produce a cover.

$$G': \quad A \rightarrow A\alpha R_1 \mid \beta R_2 \\ R_1 \rightarrow \varepsilon, R_2 \rightarrow \varepsilon$$

$$\hat{G}': \quad A \rightarrow \beta R_2 A' \\ A' \rightarrow \alpha R_1 A' \mid \varepsilon \\ R_1 \rightarrow \varepsilon, R_2 \rightarrow \varepsilon$$

$$h(R_1 \rightarrow \varepsilon) = A \rightarrow A\alpha \quad h(R_2 \rightarrow \varepsilon) = A \rightarrow \beta$$

$$h(A \rightarrow \beta R_2 A') = h(A' \rightarrow \alpha R_1 A' \mid \varepsilon) = \varepsilon$$

$$\hat{\pi}_L'(w) = (A \rightarrow \beta R_2 A') \pi_L(y) (R_2 \rightarrow \varepsilon) ((A' \rightarrow \alpha R_1 A') \pi_L(x_i) \\ (R_1 \rightarrow \varepsilon))^n (A' \rightarrow \varepsilon).$$

$$h(\hat{\pi}_L'(w)) = \pi_R(y) (A \rightarrow \beta) (\pi_R(x_i) (A \rightarrow A\alpha))^n = \pi_R(w)$$

$$G'': \quad A \rightarrow R_1 A\alpha \mid R_2 \beta \\ R_1 \rightarrow \varepsilon, R_2 \rightarrow \varepsilon$$

Removing of left recursion from a grammar such that a cover is obtained

*A reduced grammar, which is ε -free and does not contain any nonterminal A such that $A \Rightarrow^+ A$, called a **proper grammar***

Lemma 8.71 *Any proper grammar G can be transformed in time $O(|G|)$ into a grammar $\hat{G} = (\hat{V}, T, \hat{P}, S)$ such that the following statements hold*

- (1) \hat{G} is not directly left-recursive
- (2) $V \subseteq \hat{V}$
- (3) $L_{\hat{G}}(X) = L_G(X)$ for all $X \in V$

Proof .

$$A \rightarrow A\alpha_1 \mid \dots \mid A\alpha_n \mid \beta_1 \mid \dots \mid \beta_m, \quad n \geq 1, \quad m \geq 1$$

\Rightarrow

$$A \rightarrow \beta_1 A' \mid \dots \mid \beta_m A',$$

$$A' \rightarrow \alpha_1 A' \mid \dots \mid \alpha_n A' \mid \varepsilon$$

Q.E.D.

Theorem 8.72 Any proper grammar G can be transformed in time $O(|G|^2)$ into a grammar $\hat{G} = (\hat{V}, \Sigma, \hat{P}, S)$ such that the following statements hold

- (1) \hat{G} is not left-recursive
- (2) $V \subseteq \hat{V}$
- (3) $L_{\hat{G}}(X) = L_G(X)$ for all $X \in V$

Proof.

Algorithm for removing left recursion from a proper grammar $G = (\{A_1, \dots, A_n\} \cup \Sigma, \Sigma, P, S)$.

$\hat{P} := P;$

for $i := 2$ **to** n **do begin**

for $j := 1$ **to** $i-1$ **do**

if $A_i \rightarrow A_j \alpha$ for some α is in \hat{P} **then**

 replace $A_i \rightarrow A_j \alpha$ by the set of rules

$\{A_i \rightarrow \beta_j \alpha \mid A_j \rightarrow \beta_j \in \hat{P}\};$

 eliminate the direct left recursion

 possibly caused by nonterminal A_i (use the method described in the proof of L 8.71).

end

A non left-recursive covering grammar, G_T , for a given any proper grammar G

$G_T = (V \cup P, \Sigma \cup P, P_T, S)$, where

$$P_T = \{A \rightarrow \omega(A, \omega) \mid (A, \omega) \in P\}.$$

Lemma 8.73 Let a string $w \in (\Sigma \cup P)^*$, $n > 0$, such that

(a) $X \Rightarrow^n w$ in G_T

Then,

(b) $w = y_1 r_1 y_2 r_2 \dots y_n r_n$, for some $y_i \in \Sigma^*$ and

$$X \xRightarrow{r_n \dots r_1}_{rm} y_1 y_2 \dots y_n \text{ in } G.$$

Proof.

Base $n=1$, immediate.

Case $n > 0$, follows from construction of G_T and I.H.

(a) implies that (L 4.1)

$$\exists X \rightarrow X_1 \dots X_l (X, X_1 \dots X_l) \in P_T . \exists.$$

$$X \Rightarrow X_1 \dots X_l (X, X_1 \dots X_l),$$

$$X_i \Rightarrow^{n_i} w_i, \quad \forall i=1, \dots, l,$$

$$w_1 \dots w_l (X, X_1 \dots X_l) = w, \text{ and}$$

$$n_1 + \dots + n_l = n-1$$

$r_n \dots r_1 = r_n (X \rightarrow X_1 \dots X_l)$ and $r_{n-1} \dots r_1$ (from I.H.)

Q.E.D.

Lemma 8.74 *Let $n \geq 0$, such that*

$$(a) X \xRightarrow{r_n \dots r_1}_{r_m} y \text{ in } G$$

Then,

$$(b) y = y_1 y_2 \dots y_n, \text{ and}$$

$$X \Rightarrow^n y_1 r_1 y_2 r_2 \dots y_n r_n \text{ in } G_T.$$

Let $\hat{G}_T = (\hat{V}, \Sigma \cup P, \hat{P}_T, S)$ be a grammar obtained by removing left recursion from G_T by means of theorem 8.72.

Let $\hat{G}_N = (\hat{V}, \Sigma, \hat{P}_N, S)$ where

$$\hat{P}_N = \hat{P}_T \cup \{ r \rightarrow \varepsilon \mid r \in P \}$$

Then \hat{G}_N left-to-right covers G w.r.t. h .

where, $h : \hat{P}_N^ \rightarrow P^*$ by:*

$$(1) h(r \rightarrow \varepsilon) = r, \quad \text{where } r \in P$$

$$(2) h(A \rightarrow \omega) = \varepsilon, \text{ where } A \rightarrow \omega \in \hat{P}_T$$

\hat{G}_N interprets each rule in P as a nonterminal.

Lemma 8.75 Let y be a sentence in $L(\hat{G}_T)$ and $\hat{\pi}$ a left parse of y in \hat{G}_T . Then $h(\hat{\pi})$ is right parse of y in G .

Proof. As $\hat{\pi}$ is left parse, then

$$\exists \hat{\pi} = \hat{\pi}_1(r_1 \rightarrow \varepsilon)\hat{\pi}_2(r_2 \rightarrow \varepsilon)\dots\hat{\pi}_n(r_n \rightarrow \varepsilon)\hat{\pi}_{n+1}$$

$$S \xrightarrow[\overline{lm}]{\hat{\pi}_1} y_1 r_1 \beta_1 \xrightarrow[\overline{lm}]{} y_1 \beta_1$$

$$\xrightarrow[\overline{lm}]{\hat{\pi}_2} y_1 y_2 r_2 \beta_2 \xrightarrow[\overline{lm}]{} y_1 y_2 \beta_2$$

...

$$\xrightarrow[\overline{lm}]{\hat{\pi}_n} y_1 y_2 \dots y_n r_n \beta_n \xrightarrow[\overline{lm}]{} y_1 y_2 \dots y_n \beta_n$$

$$\xrightarrow[\overline{lm}]{\hat{\pi}_{n+1}} y_1 y_2 \dots y_n y_{n+1} = y, \text{ where}$$

$$\hat{\pi}_{n+1} = \varepsilon, y_{n+1} = \varepsilon = \beta_n \text{ (rule in } \hat{P}_T \text{ do not derive } \Sigma^*)$$

$$S \xrightarrow[\overline{lm}]{\hat{\pi}_1 \dots \hat{\pi}_n} y_1 r_1 \dots y_n r_n \text{ in } \hat{G}_T$$

$$S \xrightarrow[\overline{lm}]{}^+ y_1 r_1 \dots y_n r_n \text{ in } G_T. \text{ (L 8.72)}$$

$$S \xrightarrow[\overline{rm}]{r_n \dots r_1} y_1 \dots y_n \text{ in } G. \text{ (L 8.73)}$$

$$h(\hat{\pi}) = r_1 \dots r_n \text{ Q.E.D.}$$

Lemma 8.76 Let y be a sentence in $L(G)$ and π a right parse of y in G . Then y has in \hat{G}_N a left parse $\hat{\pi}$ such that $h(\hat{\pi}) = \pi$.

Proof. $\pi = r_1 \dots r_n$, π is right parse,

$$S \xRightarrow{rm}^{r_n \dots r_1} y \text{ in } G,$$

$$S \Rightarrow^n y_1 r_1 \dots y_n r_n \text{ in } G_T, y_1 \dots y_n = y, \text{ (L 8.74)}$$

$$S \Rightarrow^+ y_1 r_1 \dots y_n r_n \text{ in } \hat{G}_T, \text{ (L 8.72)}$$

$$S \xRightarrow{lm}^{\pi'} y_1 r_1 \dots y_n r_n \text{ in } G_T, \text{ (L 4.2) where } \pi' \in \hat{P}_T^*.$$

Get $\hat{\pi}$ a left parse of \hat{G}_N from π' by inserting $r_i \rightarrow \varepsilon$ at that place in π' where r_i become the first nonterminal in the derivation.

$$\hat{\pi} = \hat{\pi}_1(r_1 \rightarrow \varepsilon) \dots \hat{\pi}_n(r_n \rightarrow \varepsilon), \text{ where } \hat{\pi}_1 \dots \hat{\pi}_n = \pi' \\ \text{then}$$

$$S \xRightarrow{lm}^{\hat{\pi}} y_1 \dots y_n = y \text{ in } \hat{G}_N \text{ with } h(\hat{\pi}) = \pi \text{ Q.E.D.}$$

Theorem 8.77 For any proper grammar G , there is a non-left-recursive grammar \hat{G} and a homomorphism h such that (\hat{G}, h) is a left-to-right cover to G .

8.7 Predictive LR(k) Grammars

Converting a proper grammar into LL(k) form.

$A \in V$ if $A \rightarrow \omega \in P$, $X \in V \cup \{\epsilon\}$, $l:\omega = X$

\perp -augmented grammar G_{\perp}

$$G_{\perp} = (N \cup \{S'\}, \Sigma \cup \{\perp\}, P \cup \{S' \rightarrow \perp S\}, S')$$

The LL-transformed grammar $T_{LL}(G)$ for G

$$T_{LL}(G) = (V_t, \Sigma, P_t, [S', \perp])$$

If $A \rightarrow Z\alpha X\beta \in P$, $\alpha, \beta \in V^*$, $X, Z \in V$. Then

$$(1) [A, Z\alpha] \rightarrow a[A, Z\alpha a],$$

where $X \xrightarrow{L^*} a$, $a \in \Sigma$.

$$(2) [A, Z\alpha Y] \rightarrow [B, Y][A, Z\alpha B]$$

where $X \xrightarrow{L^*} B$, $B \xrightarrow{L} Y$, $Y \in V \cup \{\perp\}$.

$$(3) [A, Z\alpha X\beta] \rightarrow \epsilon.$$

Assume

$$X = B_0 L B_1 L B_2 L \dots L B_n L a$$

$$(1) [A, Z\alpha] \rightarrow a[A, Z\alpha a]$$

$$(2) [A, Z\alpha a] \rightarrow [B_n, a] [A, Z\alpha B_n]$$

$$[A, Z\alpha B_n] \rightarrow [B_{n-1}, B_n] [A, Z\alpha B_{n-1}]$$

...

$$[A, Z\alpha B_2] \rightarrow [B_1, B_2] [A, Z\alpha B_1]$$

$$[A, Z\alpha B_1] \rightarrow [B_0, B_1] [A, Z\alpha B_0]$$

$$\begin{aligned} \therefore A &\Rightarrow Z\alpha B_0\beta \Rightarrow^* xB_0\beta \Rightarrow xB_1\gamma_1\beta \Rightarrow xB_2\gamma_2\gamma_1\beta \\ &\Rightarrow^* xB_n\gamma_n\dots\gamma_1\beta \Rightarrow x\alpha\gamma_{n+1}\gamma_n\dots\gamma_1\beta \\ &\Rightarrow^* xay_{n+1}\gamma_n\dots\gamma_1\beta \Rightarrow^* xay_{n+1}y_n\dots y_1\beta \\ &\Rightarrow^* xay_{n+1}y_n\dots y_1z \end{aligned}$$

$$\begin{aligned} [A, Z\alpha] &\Rightarrow a[A, Z\alpha a] \Rightarrow a[B_n, a] [A, Z\alpha B_n] \\ &\Rightarrow^* ay_{n+1}[A, Z\alpha B_n] \Rightarrow ay_{n+1}[B_{n-1}, B_n] [A, Z\alpha B_{n-1}] \\ &\Rightarrow^* ay_{n+1}y_n[A, Z\alpha B_{n-1}] \\ &\Rightarrow^* ay_{n+1}y_n\dots y_2[B_0, B_1] [A, Z\alpha B_0] \\ &\Rightarrow^* ay_{n+1}y_n\dots y_2y_1 [A, Z\alpha B_0] \\ &\Rightarrow^* ay_{n+1}y_n\dots y_2y_1z \end{aligned}$$

Lemma 8.78 $T_{LL}(G)$ is of size $O(|G|^2)$, can be constructed from G in time $O(|G|^2)$.

G_{assign} :

$$S' \rightarrow \perp S$$

$$S \rightarrow i \leftarrow A \mid i \leftarrow B \quad S L i$$

$$A \rightarrow A * P \mid P \quad A L A, A L P$$

$$B \rightarrow A = A \quad B L A$$

$$P \rightarrow i \mid (A) \quad P L i, P L ($$

$T_{LL}(G_{assign})$ is:

$$[S', \perp] \rightarrow i[S', \perp i]$$

$$[S', \perp i] \rightarrow [S, i][S', \perp S]$$

$$[S', \perp S] \rightarrow \varepsilon$$

$$[S, i] \rightarrow \leftarrow [S, i \leftarrow]$$

$$[S, i \leftarrow] \rightarrow ([S, i \leftarrow (] \mid i[S, i \leftarrow i]$$

$$[S, i \leftarrow (] \rightarrow [P, (][S, i \leftarrow P]$$

$$[S, i \leftarrow i] \rightarrow [P, i][S, i \leftarrow P]$$

$$[S, i \leftarrow P] \rightarrow [A, P][S, i \leftarrow A]$$

$$[S, i \leftarrow A] \rightarrow [A, A][S, i \leftarrow A] \mid [B, A][S, i \leftarrow B] \mid \varepsilon$$

$$[S, i \leftarrow B] \rightarrow \varepsilon$$

...

$$E' \rightarrow \perp E \quad E \rightarrow E + T \mid T \quad T \rightarrow T * F \mid F \quad F \rightarrow a \mid (E)$$

$$\begin{aligned} [E', \perp] &\rightarrow a [E', \perp a] \mid ([E', \perp (] && a \mid (\\ [E', \perp a] &\rightarrow [F, a] [E', \perp F] \\ [E', \perp (] &\rightarrow [F, (] [E', \perp F] \\ [E', \perp F] &\rightarrow [T, F] [E', \perp T] \\ [E', \perp T] &\rightarrow [T, T] [E', \perp T] \mid [E, T] [E', \perp E] && * \mid +, \$ \\ [E', \perp E] &\rightarrow [E, E] [E', \perp E] \mid \varepsilon && + \mid \$ \end{aligned}$$

$$\begin{aligned} [E, E] &\rightarrow + [E, E+] \\ [E, E+] &\rightarrow a [E, E+a] \mid ([E, E+(] && a \mid (\\ [E, E+a] &\rightarrow [F, a] [E, E+F] \\ [E, E+(] &\rightarrow [F, (] [E, E+F] \\ [E, E+F] &\rightarrow [T, F] [E, E+T] \\ [E, E+T] &\rightarrow [T, T] [E, E+T] \mid \varepsilon && * \mid +, \$ \\ [E, T] &\rightarrow \varepsilon \\ [T, T] &\rightarrow * [T, T*] \\ [T, T*] &\rightarrow a [T, T*a] \mid ([T, T*(] && a \mid (\\ [T, T*a] &\rightarrow [F, a] [T, T*F] \\ [T, T*(] &\rightarrow [F, (] [T, T*F] \\ [T, T*F] &\rightarrow \varepsilon \\ [T, F] &\rightarrow \varepsilon \\ [F, (] &\rightarrow a [F, (a] \mid ([F, ((] && a \mid (\\ [F, (a] &\rightarrow [F, a] [F, (F] \\ [F, ((] &\rightarrow [F, (] [F, (F] \\ [F, (F] &\rightarrow [T, F] [F, (T] \\ [F, (T] &\rightarrow [T, T] [F, (T] \mid [E, T] [F, (E)] && * +, \$ \\ [F, (E)] &\rightarrow [E, E] [E, (E)] \mid) [F, (E)] && + \mid) \\ [F, (E)] &\rightarrow \varepsilon \\ [F, a] &\rightarrow \varepsilon \end{aligned}$$

		$[E', \perp]$			
\Rightarrow	a	$[E', \perp a]$			
\Rightarrow	a	$[F, a]$	$[E', \perp F]$		
\Rightarrow	a		$[E', \perp F]$		$(F \rightarrow a)$
\Rightarrow	a	$[T, F]$	$[E', \perp T]$		
\Rightarrow	a		$[E', \perp T]$		$(T \rightarrow F)$
\Rightarrow	a	$[E, T]$		$[E', \perp E]$	
\Rightarrow	a			$[E', \perp E]$	$(E \rightarrow T)$
\Rightarrow	a	$[E, E]$		$[E', \perp E]$	
\Rightarrow	a	$+$	$[E, E+]$	$[E', \perp E]$	
\Rightarrow	$a+$	a	$[E, E+a]$	$[E', \perp E]$	
\Rightarrow	$a+a$	$[F, a]$	$[E, E+F]$	$[E', \perp E]$	
\Rightarrow	$a+a$		$[E, E+F]$	$[E', \perp E]$	$(F \rightarrow a)$
\Rightarrow	$a+a$	$[T, F]$	$[E, E+T]$	$[E', \perp E]$	
\Rightarrow	$a+a$		$[E, E+T]$	$[E', \perp E]$	$(T \rightarrow F)$
\Rightarrow	$a+a$	$[T, T]$	$[E, E+T]$	$[E', \perp E]$	
\Rightarrow	$a+a$	$*$	$[T, T^*]$	$[E, E+T]$	$[E', \perp E]$
\Rightarrow	$a+a^*$	a	$[T, T^*a]$	$[E, E+T]$	$[E', \perp E]$
\Rightarrow	$a+a^*a$	$[F, a]$	$[T, T^*F]$	$[E, E+T]$	$[E', \perp E]$
\Rightarrow	$a+a^*a$	$[T, T^*F]$	$[E, E+T]$	$[E', \perp E]$	$(F \rightarrow a)$
\Rightarrow	$a+a^*a$		$[E, E+T]$	$[E', \perp E]$	$(T \rightarrow T^*F)$
\Rightarrow	$a+a^*a$			$[E', \perp E]$	$(E \rightarrow E+T)$
\Rightarrow	$a+a^*a$				$(E' \rightarrow \perp E)$

Let homomorphism, h_T

from the rule strings of $T_{LL}(G)$

to the rule strings of G

$$h_T([A, \alpha] \rightarrow \eta) = A \rightarrow \alpha, \eta = \varepsilon \text{ and } A \rightarrow \alpha \in P;$$

$$= \varepsilon, \quad \text{otherwise.}$$

Then $T_{LL}(G)$ left-to-right covers G w.r.t. homomorphism h_T .

Lemma 8.79 Let G be ε -free, π_T a rule string of $T_{LL}(G)$, r_T a rule of $T_{LL}(G)$ and

$$(a) [A, \alpha Y] \xRightarrow{lm}^{\pi_T r_T} x \text{ in } T_{LL}(G), Y \in V \cup \{\perp\}$$

Then

$$(b) [A, \alpha Y] \xRightarrow{lm}^{\pi_T} x [A, \alpha X \beta] \xRightarrow{lm}^{r_T} x \text{ in } T_{LL}(G),$$

$$X \beta \xRightarrow{rm}^{h_T(\pi_T)^R} Yx \text{ in } G_{\perp}$$

Proof. induction on $|\pi_T|$

Base case, if $\pi_T = \varepsilon$, $r_T = [A, \alpha Y] \rightarrow x$,

$x \in \Sigma^*$, so $x = \varepsilon$ and $A \rightarrow \alpha Y$ in G_{\perp} ,

$X = Y$, $\beta = \varepsilon$, so $X\beta \Rightarrow Y$.

immediately (a) implies (b).

(case 1) the first rule in π_T is of type 1

$$\text{let } r'_T = [A, \alpha Y] \rightarrow a[A, \alpha Ya]$$

$$\pi_T = r'_T \pi'_T, \quad x = ax' \text{ where}$$

$$[A, \alpha Ya] \xRightarrow{lm}^{\pi'_T r'_T} x' \text{ in } T_{LL}(G)$$

by I.H.

$$[A, \alpha Ya] \xRightarrow{lm}^{\pi'_T} x' [A, \alpha YX'\beta'] \xRightarrow{lm}^{r'_T} x' \text{ in } T_{LL}(G)$$

$$X'\beta' \xRightarrow{rm}^{h_T(\pi'_T)^R} ax' \text{ in } G_\perp$$

then

$$[A, \alpha Y] \xRightarrow{lm}^{\pi_T} ax' [A, \alpha YX'\beta'] \xRightarrow{lm}^{r'_T} ax' \text{ in } T_{LL}(G)$$

$$YX'\beta' \xRightarrow{rm}^{h_T(\pi_T)^R} Yax' = Yx \text{ in } G_\perp$$

observe that

$$h_T(\pi_T)^R = h_T(r'_T \pi'_T)^R = h_T(\pi'_T)^R$$

$$\text{because } h_T(r'_T) = \varepsilon \text{ for type 1}$$

(b) holds when we choose $X=Y$ and $\beta=X'\beta'$

case 2: the first rule in π_T is of type 2

let $r'_T = [A, \alpha Y] \rightarrow [B, Y][A, \alpha B]$

$\pi_T r_T = r'_T \pi'_T r''_T \pi''_T r_T$, $x = x' x''$ where

$[B, Y] \xrightarrow{lm} \pi'_T r'_T x'$ and $[A, \alpha B] \xrightarrow{lm} \pi''_T r_T x''$ in $T_{LL}(G)$

by I.H.

$[B, Y] \xrightarrow{lm} \pi'_T x' [B, X'\beta'] \xrightarrow{lm} r'_T x'$ in $T_{LL}(G)$

$X'\beta' \xrightarrow{rm} h_T(\pi'_T)^R Yx'$ in G_\perp

$[A, \alpha B] \xrightarrow{lm} \pi''_T x'' [A, \alpha X\beta] \xrightarrow{lm} r_T x''$ in $T_{LL}(G)$

$X\beta \xrightarrow{rm} h_T(\pi''_T)^R Bx''$ in G_\perp

then

$[A, \alpha Y] \xrightarrow{lm} \pi_T x' x'' [A, \alpha X\beta] \xrightarrow{lm} r_T x' x'' = x$ in $T_{LL}(G)$

$X\beta \xrightarrow{rm} h_T(\pi''_T)^R Bx'' \xrightarrow{rm} X'\beta' x'' \xrightarrow{rm} h_T(\pi'_T)^R Yx' x'' = Yx$

in G_\perp

observe that $r'' = B \rightarrow X'\beta'$, $h_T(r''_T) = r''$

$h_T(\pi''_T)^R r'' h_T(r'_T)^R =$

$(h_T(r'_T) h_T(\pi'_T) h_T(r''_T) h_T(\pi''_T))^R = h_T(\pi_T)^R$

where $h_T(r'_T) = \varepsilon$ (r'_T is of type 2).

Q.E.D.

Lemma 8.80 $L(T_{LL}(G)) \subseteq L(G)$

moreover left parse π' of sentence w in $T_{LL}(G)$,
 $h_T(\pi')$ is a right parse of w in G .

Proof. in Lemma 8.79, let $A=S'$, $\alpha=\varepsilon$, $Y=\perp$, $x=w$,
 $\pi_T r_T = \pi'$ s.t.

$[S', \perp] \xRightarrow{lm}^{\pi_T w} [S', X\beta] \xRightarrow{lm}^{r_T} w$ in $T_{LL}(G)$ and

$X\beta \xRightarrow{rm}^{h_T(\pi_T)^R} \perp w$ in G_{\perp} .

Then

$S \xRightarrow{rm}^{h_T(\pi_T)^R} w$ in G

$h_T(\pi_T)^R = h_T(\pi_T r_T)^R = h_T(\pi')^R$,

because $h_T(r_T) = \varepsilon$, $r_T = [S', \perp S] \rightarrow \varepsilon$ **Q.E.D.**

Lemma 8.81 Let G be an ε -free grammar

$A \xRightarrow{rm} \alpha X\beta$ and $X\beta \xRightarrow{rm}^{\pi} Yx$ in G_{\perp} . Then,

$[A, \alpha Y] \xRightarrow{lm}^{\pi_T x} [A, \alpha X\beta]$ in $T_{LL}(G)$ and $h_T(\pi_T) = \pi^R$

Proof. induction on $|\pi|$.

Lemma 8.82 $L(G) \subseteq L(T_{LL}(G))$, and right parse π of
sentence w in G , left parse π' of w in $T_{LL}(G)$ such that
 $h_T(\pi') = \pi$.

Theorem 8.83 $(T_{LL}(G), h_T)$ is a left-to-right cover of G .

A grammar G is a **Predictive LR(k) grammar (PLR(k))**, $k \geq 1$, if the transformed grammar $T_{LL}(G)$ is $LL(k)$

Proposition 8.84 $k \geq 1$,

$LL(k)$ grammar \subset $PLR(k)$ grammar \subset $LR(k)$ grammar

Intuitive characterization PLR(k) grammar

Let $A \rightarrow X\alpha$,

$S \Rightarrow^* wAz$, $X \Rightarrow^* x$, $\alpha \Rightarrow^* y$ ($S \Rightarrow^* wxyz$)

$LL(k)$ parser emits the rule $A \rightarrow X\alpha$

after scanning w , seeing $k:xyz$

$PLR(k)$ parser recognize the rule $A \rightarrow X\alpha$

after scanning wx , seeing $k:yz$

$LR(k)$ parser recognize the rule $A \rightarrow X\alpha$

after scanning wxy , seeing $k:z$

Proposition 8.85 *If G is proper, then $T_{LL}(G)$ is non-left-recursive.*

A grammar is left-factored if it has no two distinct rules $A \rightarrow \alpha\beta_1$ and $A \rightarrow \alpha\beta_2$. $\alpha \neq \varepsilon$

Fact 8.86 *For any G , $T_{LL}(G)$ is left-factored.*

Theorem 8.87 *Left-factoring cannot produce a $PLR(k)$ grammar from a non- $PLR(k)$ grammar, and a $PLR(k)$ grammar cannot be converted into a non- $PLR(k)$ grammar by left-factoring.*

Proof. *Process of constructing $T_{LL}(G)$ also involves the process of left-factoring.*