Análisis Sintáctico Descendente

CI4721 – Lenguajes de Programación II

Ernesto Hernández-Novich
<emhn@usb.ve>

Universidad "Simón Bolívar"

UNIVERSIDAD SIMÓN BOLÍVAR

Reconocedor descendente no determinístico

Sea $G = (N, \Sigma, P, S)$ una CFG cualquiera. Entonces el PDA extendido

$$Desc(G) = (\{q_0, q_1\}, \Sigma, N \cup \Sigma, \delta, q_0, \{q_1\})$$

con δ definida según

$$\delta(q_0, \lambda, \lambda) = \{(q_1, S)\} \tag{1}$$
$$\delta(q_1, \lambda, A) = \{A \to \alpha \in P : (q_1, \alpha)\}, \forall A \in N \tag{2}$$
$$\delta(q_1, \mathbf{a}, \mathbf{a}) = \{(q_1, \lambda)\}, \forall \mathbf{a} \in \Sigma \tag{3}$$

¿Cuál es la razón del no-determinismo?

UNIVERSIDAD SIMÓN BOLÍVAR

Eliminando la ambigüedad

"Husmear" antes de decidir

- El reconocedor pretende encontrar

$$S \underset{Left}{\overset{*}{\Rightarrow}} w$$

- En cualquier forma sentencial intermedia

$$S \underset{Left}{\overset{*}{\Rightarrow}} uA\alpha$$

  es natural que u sea un *prefijo* de w.
- Como $w = uax$ podemos "espiar" a y eso ayuda ...
  - ... a decidir cuál regla $A \to w_i$ expandir.
  - ... a notar que ninguna regla aplica – ¡Error de sintaxis!

Vamos a generalizar el "espiar".

UNIVERSIDAD SIMÓN BOLÍVAR

"Espiar" es el nombre técnico para "lookahead"

¿Qué podemos obtener de un no terminal?

Sea $G = (N, \Sigma, P, S)$ libre de contexto y $A \in N$.
Definimos el *lookahead* de A como

$$LA(A) = \{x | S \overset{*}{\Rightarrow} uA\alpha \overset{*}{\Rightarrow} ux \in \Sigma^*\}$$

UNIVERSIDAD SIMÓN BOLÍVAR

"Espiar" es el nombre técnico para "lookahead"

¿Qué podemos obtener de un no terminal?

Sea $G = (N, \Sigma, P, S)$ libre de contexto y $A \in N$.
Definimos el *lookahead* de A como

$$LA(A) = \{x | S \overset{*}{\Rightarrow} uA\alpha \overset{*}{\Rightarrow} ux \in \Sigma^*\}$$

$LA(A)$ está constituido por *todas* las cadenas de *terminales* derivables a partir de $A\alpha$ cuando $uA\alpha$ es una forma sentencial más izquierda.

UNIVERSIDAD SIMÓN BOLÍVAR

"Espiar" es el nombre técnico para "lookahead"

¿Qué podemos obtener de una regla particular?

Sea $G = (N, \Sigma, P, S)$ libre de contexto y $A \in N$.
Definimos el *lookahead* de la regla $A \to \alpha \in P$ como

$$LA(A \to \alpha) = \{x | \alpha\beta \overset{*}{\Rightarrow} x \in \Sigma^* \wedge S \overset{*}{\Rightarrow} uA\beta\}$$

UNIVERSIDAD SIMÓN BOLÍVAR

"Espiar" es el nombre técnico para "lookahead"

¿Qué podemos obtener de una regla particular?

Sea $G = (N, \Sigma, P, S)$ libre de contexto y $A \in N$.
Definimos el *lookahead* de la regla $A \to \alpha \in P$ como

$$LA(A \to \alpha) = \{x | \alpha\beta \overset{*}{\Rightarrow} x \in \Sigma^* \wedge S \overset{*}{\Rightarrow} uA\beta\}$$

$LA(A \to \alpha) \subseteq LA(A)$ en el cual las derivaciones $A\beta \overset{*}{\Rightarrow} x$ comienzan expandiendo la regla $A \to \alpha$.

UNIVERSIDAD SIMÓN BOLÍVAR

El conjunto y sus partes

Sean $A \to \alpha_1, \ldots A \to \alpha_n$ las reglas para A, si se cumple

1. $LA(A) = \bigcup_{i=1}^{n} LA(A \to \alpha_i)$
2. $LA(A \to \alpha_i) \cap LA(A \to \alpha_j) = \emptyset$ cuando $1 \leq i < j \leq n$

decimos que los $LA(A \to \alpha_i)$ *particionan* $LA(A)$

UNIVERSIDAD SIMÓN BOLÍVAR

El conjunto y sus partes

Sean $A \to \alpha_1, \ldots A \to \alpha_n$ las reglas para A, si se cumple

1. $LA(A) = \bigcup_{i=1}^{n} LA(A \to \alpha_i)$
2. $LA(A \to \alpha_i) \cap LA(A \to \alpha_j) = \emptyset$ cuando $1 \leq i < j \leq n$

decimos que los $LA(A \to \alpha_i)$ *particionan* $LA(A)$

- La primera condición siempre se cumple para una gramática libre de contexto directamente por la definición de $LA(A)$.
- Si la segunda condición se cumple, y tenemos $S \overset{*}{\Rightarrow} uA\beta$ tratando de derivar $w = ux$, entonces:
  - x es elemento de *exactamente* uno de los $LA(A \to \alpha)$ – tenemos la regla a expandir.
  - x no pertenece a *ninguno* de los $LA(A \to \alpha)$ – es imposible derivar w.

UNIVERSIDAD SIMÓN BOLÍVAR

Un ejemplo trivial ...

¿Cómo expandir el no terminal S?

$S \to A\mathbf{abd}$

$S \to \mathbf{c}A\mathbf{bcd}$

$A \to \mathbf{a}$

$A \to \mathbf{b}$

$A \to \lambda$

Para el no-terminal S tenemos

$$LA(S) = \{\mathbf{aabd}, \mathbf{babd}, \mathbf{abd}, \mathbf{cabcd}, \mathbf{cbbcd}, \mathbf{cbcd}\}$$

partido por los conjuntos

$$LA(S \to A\mathbf{abd}) = \{\mathbf{aabd}, \mathbf{babd}, \mathbf{abd}\}$$
$$LA(S \to \mathbf{c}A\mathbf{bcd}) = \{\mathbf{cabcd}, \mathbf{cbbcd}, \mathbf{cbcd}\}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Un ejemplo trivial ...

¿Cómo expandir el no terminal S?

$S \to A\mathbf{abd}$

$S \to \mathbf{c}A\mathbf{bcd}$

$A \to \mathbf{a}$

$A \to \mathbf{b}$

$A \to \lambda$

Para el no-terminal S tenemos

$$LA(S) = \{\mathbf{aabd}, \mathbf{babd}, \mathbf{abd}, \mathbf{cabcd}, \mathbf{cbbcd}, \mathbf{cbcd}\}$$

partido por los conjuntos

$$LA(S \to A\mathbf{abd}) = \{\mathbf{aabd}, \mathbf{babd}, \mathbf{abd}\}$$
$$LA(S \to \mathbf{c}A\mathbf{bcd}) = \{\mathbf{cabcd}, \mathbf{cbbcd}, \mathbf{cbcd}\}$$

Basta *lookahead* de un símbolo para determinar cuál regla S expandir en cada caso.

UNIVERSIDAD SIMÓN BOLÍVAR

Un ejemplo trivial ...

¿Cómo expandir el no terminal A?

$S \to A\mathbf{abd}$

$S \to \mathbf{c}A\mathbf{bcd}$

$A \to \mathbf{a}$

$A \to \mathbf{b}$

$A \to \lambda$

Para el no-terminal A tenemos

$$LA(A) = \{\mathbf{aabd}, \mathbf{abcd}, \mathbf{babd}, \mathbf{bbcd}, \mathbf{abd}, \mathbf{bcd}\}$$

partido por los conjuntos

$$LA(A \to \mathbf{a}) = \{\mathbf{aabd}, \mathbf{abcd}\}$$
$$LA(A \to \mathbf{b}) = \{\mathbf{babd}, \mathbf{bbcd}\}$$
$$LA(A \to \lambda) = \{\mathbf{abd}, \mathbf{bcd}\}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Un ejemplo trivial ...

¿Cómo expandir el no terminal A?

$S \to A\mathbf{abd}$

$S \to \mathbf{c}A\mathbf{bcd}$

$A \to \mathbf{a}$

$A \to \mathbf{b}$

$A \to \lambda$

Para el no-terminal A tenemos

$$LA(A) = \{\mathbf{aabd}, \mathbf{abcd}, \mathbf{babd}, \mathbf{bbcd}, \mathbf{abd}, \mathbf{bcd}\}$$

partido por los conjuntos

$$LA(A \to \mathbf{a}) = \{\mathbf{aabd}, \mathbf{abcd}\}$$
$$LA(A \to \mathbf{b}) = \{\mathbf{babd}, \mathbf{bbcd}\}$$
$$LA(A \to \lambda) = \{\mathbf{abd}, \mathbf{bcd}\}$$

Hace falta *lookahead* de tres símbolos para determinar cuál regla A expandir en cada caso.

UNIVERSIDAD SIMÓN BOLÍVAR

Si la gramática no es trivial . . .

- Tanto $LA(A)$ como $LA(A \to \alpha)$ pueden contener cadenas de longitud arbitraria . . .
- . . . y una cantidad arbitraria de cadenas.
- El *lookahead* termina siendo un prefijo más corto que permita distinguir el $LA(A \to \alpha)$ de interés.

UNIVERSIDAD SIMÓN BOLÍVAR

Prefijos de longitud k

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$ definiremos la función

$$trunc_k :: \Sigma^* \to \Sigma^*$$

llamada **prefijo de longitud** k como

$$trunc_k(w) = \begin{cases} w & \text{if } |w| \leq k \\ u & \text{if } w = uv \wedge |u| = k \end{cases}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Prefijos de longitud k

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$ definiremos la función

$$trunc_k :: \Sigma^* \to \Sigma^*$$

llamada **prefijo de longitud** k como

$$trunc_k(w) = \begin{cases} w & \text{if } |w| \leq k \\ u & \text{if } w = uv \wedge |u| = k \end{cases}$$

que podemos extender a conjuntos

$$trunc_k :: \mathcal{P}(\Sigma^*) \to \mathcal{P}(\Sigma^*)$$

como

$$trunc_k(X) = \{trunc_k(w) | w \in X\}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Lookahead de longitud k

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$ definiremos

- *Lookahead* de A de longitud k

$$LA_k(A) = trunc_k(LA(A))$$

- *Lookahead* de la regla $A \to \alpha$ de longitud k

$$LA_k(A \to \alpha) = trunc_k(LA(A \to \alpha))$$

UNIVERSIDAD SIMÓN BOLÍVAR

Lookahead de longitud k

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$ definiremos

- *Lookahead* de A de longitud k

$$LA_k(A) = trunc_k(LA(A))$$

- *Lookahead* de la regla $A \to \alpha$ de longitud k

$$LA_k(A \to \alpha) = trunc_k(LA(A \to \alpha))$$

Aún necesitamos un método más eficiente para calcular los LA_k

UNIVERSIDAD SIMÓN BOLÍVAR

FIRST – Prefijos derivables de A

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$. Entonces, $\forall \alpha \in (N \cup \Sigma)^*$ definiremos

$$FIRST_k(\alpha) = trunc_k(\{w | \alpha \stackrel{*}{\Rightarrow} w\})$$

UNIVERSIDAD SIMÓN BOLÍVAR

FIRST – Prefijos derivables de A

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$. Entonces, $\forall \alpha \in (N \cup \Sigma)^*$ definiremos

$$FIRST_k(\alpha) = trunc_k(\{w | \alpha \stackrel{*}{\Rightarrow} w\})$$

Lemma

1. $FIRST_k(\lambda) = \{\lambda\}$

FIRST – Prefijos derivables de A

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$. Entonces, $\forall \alpha \in (N \cup \Sigma)^*$ definiremos

$$FIRST_k(\alpha) = trunc_k(\{w | \alpha \stackrel{*}{\Rightarrow} w\})$$

Lemma

1. $FIRST_k(\lambda) = \{\lambda\}$
2. $FIRST_k(\mathbf{a}) = \{\mathbf{a}\}$

FIRST – Prefijos derivables de A

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$. Entonces, $\forall \alpha \in (N \cup \Sigma)^*$ definiremos

$$FIRST_k(\alpha) = trunc_k(\{w | \alpha \stackrel{*}{\Rightarrow} w\})$$

Lemma

1. $FIRST_k(\lambda) = \{\lambda\}$
2. $FIRST_k(\mathbf{a}) = \{\mathbf{a}\}$
3. $FIRST_k(\mathbf{a}\alpha) = \{\mathbf{a}w | w \in FIRST_{k-1}(\alpha)\}$

UNIVERSIDAD SIMÓN BOLÍVAR

FIRST – Prefijos derivables de A

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$. Entonces, $\forall \alpha \in (N \cup \Sigma)^*$ definiremos

$$FIRST_k(\alpha) = trunc_k(\{w | \alpha \stackrel{*}{\Rightarrow} w\})$$

Lemma

1. $FIRST_k(\lambda) = \{\lambda\}$
2. $FIRST_k(\mathbf{a}) = \{\mathbf{a}\}$
3. $FIRST_k(\mathbf{a}\alpha) = \{\mathbf{a}w | w \in FIRST_{k-1}(\alpha)\}$
4. $FIRST_k(\alpha\beta) = trunc_k(FIRST_k(\alpha)FIRST_k(\beta))$

UNIVERSIDAD SIMÓN BOLÍVAR

FIRST – Prefijos derivables de A

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$. Entonces, $\forall \alpha \in (N \cup \Sigma)^*$ definiremos

$$FIRST_k(\alpha) = trunc_k(\{w | \alpha \stackrel{*}{\Rightarrow} w\})$$

Lemma

1. $FIRST_k(\lambda) = \{\lambda\}$
2. $FIRST_k(\mathbf{a}) = \{\mathbf{a}\}$
3. $FIRST_k(\mathbf{a}\alpha) = \{\mathbf{a}w | w \in FIRST_{k-1}(\alpha)\}$
4. $FIRST_k(\alpha\beta) = trunc_k(FIRST_k(\alpha)FIRST_k(\beta))$
5. *Si* $A \to \alpha \in P$ *entonces* $FIRST_k(\alpha) \subseteq FIRST_k(A)$

UNIVERSIDAD SIMÓN BOLÍVAR

Calculando $FIRST_k$

$S \to A\mathbf{abd}$

$S \to \mathbf{c}A\mathbf{bcd}$

$A \to \mathbf{a}$

$A \to \mathbf{b}$

$A \to \lambda$

$$FIRST_1(S) = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$$
$$FIRST_1(A) = \{\mathbf{a}, \mathbf{b}, \lambda\}$$

$$FIRST_2(S) = \{\mathbf{aa}, \mathbf{ba}, \mathbf{ab}, \mathbf{ca}, \mathbf{cb}\}$$
$$FIRST_2(A) = \{\mathbf{a}, \mathbf{b}, \lambda\}$$

$$FIRST_3(S) = \{\mathbf{aab}, \mathbf{bab}, \mathbf{abd}, \mathbf{cab}, \mathbf{cbb}, \mathbf{cbc}\}$$
$$FIRST_3(A) = \{\mathbf{a}, \mathbf{b}, \lambda\}$$

UNIVERSIDAD SIMÓN BOLÍVAR

FOLLOW – Prefijos que continúan derivaciones de A

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$. Entonces, $\forall A \in N$ definiremos

$$FOLLOW_k(A) = \{x | S \stackrel{*}{\Rightarrow} uA\alpha \wedge x \in FIRST_k(\alpha)\}$$

UNIVERSIDAD SIMÓN BOLÍVAR

FOLLOW – Prefijos que continúan derivaciones de A

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$. Entonces, $\forall A \in N$ definiremos

$$FOLLOW_k(A) = \{x | S \stackrel{*}{\Rightarrow} uA\alpha \wedge x \in FIRST_k(\alpha)\}$$

Lemma

1. $\lambda \in FOLLOW_k(S)$

FOLLOW – Prefijos que continúan derivaciones de A

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$. Entonces, $\forall A \in N$ definiremos

$$FOLLOW_k(A) = \{x | S \stackrel{*}{\Rightarrow} uA\alpha \wedge x \in FIRST_k(\alpha)\}$$

Lemma

1. $\lambda \in FOLLOW_k(S)$
2. *Si* $A \to \alpha B \in P$ *entonces* $FOLLOW_k(A) \subseteq FOLLOW_k(B)$

UNIVERSIDAD SIMÓN BOLÍVAR

FOLLOW – Prefijos que continúan derivaciones de A

Sea $G = (N, \Sigma, P, S)$ y $k > 0 \in \mathbb{N}$. Entonces, $\forall A \in N$ definiremos

$$FOLLOW_k(A) = \{x | S \stackrel{*}{\Rightarrow} uA\alpha \wedge x \in FIRST_k(\alpha)\}$$

Lemma

1. $\lambda \in FOLLOW_k(S)$
2. *Si* $A \to \alpha B \in P$ *entonces* $FOLLOW_k(A) \subseteq FOLLOW_k(B)$
3. *Si* $A \to \alpha B \beta \in P$ *entonces* $trunc_k(FIRST_k(\beta) FOLLOW_k(A)) \subseteq FOLLOW_k(B)$

UNIVERSIDAD SIMÓN BOLÍVAR

Calculando $FOLLOW_k$

$S \rightarrow A\mathbf{abd}$

$S \rightarrow \mathbf{c}A\mathbf{bcd}$

$A \rightarrow \mathbf{a}$

$A \rightarrow \mathbf{b}$

$A \rightarrow \lambda$

$$FOLLOW_1(S) = \{\lambda\}$$
$$FOLLOW_1(A) = \{\mathbf{a}, \mathbf{b}\}$$

$$FOLLOW_2(S) = \{\lambda\}$$
$$FOLLOW_2(A) = \{\mathbf{ab}, \mathbf{bc}\}$$

$$FOLLOW_3(S) = \{\lambda\}$$
$$FOLLOW_3(A) = \{\mathbf{abd}, \mathbf{bcd}\}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Podemos calcular los *lookaheads*

Theorem

Sea $G = (N, \Sigma, P, S)$.
Para todo $k > 0 \in \mathbb{N}, A \in N$ *y* $A \to \alpha \in P$ *podemos calcular*

1. $LA_k(A) = trunc_k(FIRST_k(A)FOLLOW_k(A))$
2. $LA_k(A \to \alpha) = trunc_k(FIRST_k(\alpha)FOLLOW_k(A))$

UNIVERSIDAD SIMÓN BOLÍVAR

Podemos calcular los *lookaheads*

Theorem

Sea $G = (N, \Sigma, P, S)$.
Para todo $k > 0 \in \mathbb{N}, A \in N$ *y* $A \to \alpha \in P$ *podemos calcular*

1. $LA_k(A) = trunc_k(FIRST_k(A)FOLLOW_k(A))$
2. $LA_k(A \to \alpha) = trunc_k(FIRST_k(\alpha)FOLLOW_k(A))$

Como $\alpha = r_1 r_2 \dots r_n$ siendo $r_i \in (N \cup \Sigma)$, entonces

$$\begin{aligned} LA_k(A \to \alpha) &= trunc_k(FIRST_k(\alpha)FOLLOW_k(A)) \\ &= trunc_k(FIRST_k(r_1) \dots FIRST_k(r_n)FOLLOW_k(A)) \end{aligned}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Calculando LA_k usando $FIRST_k$ y $FOLLOW_k$

$S \to A\mathbf{abd}$

$S \to \mathbf{c}A\mathbf{bcd}$

$A \to \mathbf{a}$

$A \to \mathbf{b}$

$A \to \lambda$

$$\begin{aligned} LA_3(S \to A\mathbf{abd}) &= trunc_3(FIRST_3(A)FIRST_3(\mathbf{a})FIRST_3(\mathbf{b})FIRST_3(\mathbf{d})FOLLOW_3(S)) \\ &= trunc_3(\{\mathbf{a},\mathbf{b},\lambda\}\{\mathbf{a}\}\{\mathbf{b}\}\{\mathbf{d}\}\{\lambda\}) \\ &= trunc_3(\{\mathbf{aabd},\mathbf{babd},\mathbf{abd}\}) \\ &= \{\mathbf{aab},\mathbf{bab},\mathbf{abd}\} \end{aligned}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Gramática Fuertemente $LL(k)$

Sea $G = (N, \Sigma, P, S)$ con **marcador de final (*endmarker*)** $\$^k$.
Diremos que G es **fuertemente** $LL(k)$ si siempre que existan dos derivaciones más izquierdas

$$S \overset{*}{\Rightarrow} u_1 A \alpha_1 \Rightarrow u_1 \beta \alpha_1 \overset{*}{\Rightarrow} u_1 z v_1$$
$$S \overset{*}{\Rightarrow} u_2 A \alpha_2 \Rightarrow u_2 \gamma \alpha_2 \overset{*}{\Rightarrow} u_2 z v_2$$

con $u_i, v_i, z \in \Sigma^*$ y $|z| = k$, entonces $\beta = \gamma$.

UNIVERSIDAD SIMÓN BOLÍVAR

Gramática Fuertemente $LL(k)$

Sea $G = (N, \Sigma, P, S)$ con **marcador de final (*endmarker*)** $\$^k$.
Diremos que G es **fuertemente** $LL(k)$ si siempre que existan dos derivaciones más izquierdas

$$S \overset{*}{\Rightarrow} u_1 A \alpha_1 \Rightarrow u_1 \beta \alpha_1 \overset{*}{\Rightarrow} u_1 z v_1$$
$$S \overset{*}{\Rightarrow} u_2 A \alpha_2 \Rightarrow u_2 \gamma \alpha_2 \overset{*}{\Rightarrow} u_2 z v_2$$

con $u_i, v_i, z \in \Sigma^*$ y $|z| = k$, entonces $\beta = \gamma$.

- El marcador $\$^k$ se agrega a todas las entradas para garantizar que siempre habrá k símbolos en el *lookahead*.
- Si S no es recursivo, se agrega $\$^k$ al final de cada regla de S.
- Si S es recursivo, se agrega la regla $S' \to S\$^k$

UNIVERSIDAD SIMÓN BOLÍVAR

Relación con los *lookaheads*

Theorem

G es fuertemente $LL(k) \Leftrightarrow \forall A \in N, LA_k(A \to \alpha_i)$ *particionan* $LA_k(A)$

Relación con los *lookaheads*

Theorem

G es fuertemente $LL(k) \Leftrightarrow \forall A \in N, LA_k(A \to \alpha_i)$ *particionan* $LA_k(A)$

Demostración ($\Leftarrow$).

1. Supongamos que $LA_k(A \to \alpha_i)$ particionan $LA_k(A)$.

Relación con los *lookaheads*

Theorem

G es fuertemente $LL(k) \Leftrightarrow \forall A \in N, LA_k(A \to \alpha_i)$ *particionan* $LA_k(A)$

Demostración ($\Leftarrow$).

1. Supongamos que $LA_k(A \to \alpha_i)$ particionan $LA_k(A)$.
2. Sea $z \in \Sigma^*$ con $|z| = k$ que aparece en dos derivaciones

$$S \overset{*}{\Rightarrow} u_1 A \alpha_1 \Rightarrow u_1 \beta \alpha_1 \overset{*}{\Rightarrow} u_1 z v_1$$
$$S \overset{*}{\Rightarrow} u_2 A \alpha_2 \Rightarrow u_2 \gamma \alpha_2 \overset{*}{\Rightarrow} u_2 z v_2$$

Así, $z \in LA_k(A \to \beta)$ y $z \in LA_k(A \to \gamma)$

Relación con los *lookaheads*

Theorem

G es fuertemente $LL(k) \Leftrightarrow \forall A \in N, LA_k(A \to \alpha_i)$ *particionan* $LA_k(A)$

Demostración ($\Leftarrow$).

1. Supongamos que $LA_k(A \to \alpha_i)$ particionan $LA_k(A)$.
2. Sea $z \in \Sigma^*$ con $|z| = k$ que aparece en dos derivaciones

$$S \overset{*}{\Rightarrow} u_1 A \alpha_1 \Rightarrow u_1 \beta \alpha_1 \overset{*}{\Rightarrow} u_1 z v_1$$
$$S \overset{*}{\Rightarrow} u_2 A \alpha_2 \Rightarrow u_2 \gamma \alpha_2 \overset{*}{\Rightarrow} u_2 z v_2$$

   Así, $z \in LA_k(A \to \beta)$ y $z \in LA_k(A \to \gamma)$
3. Como $LA_k(A \to \alpha_i)$ particionan $LA_k(A)$, entonces $\beta = \gamma$
   Por tanto, G es fuertemente $LL(k)$.

Relación con los *lookaheads*

Theorem

G es fuertemente $LL(k) \Leftrightarrow \forall A \in N, LA_k(A \to \alpha_i)$ *particionan* $LA_k(A)$

UNIVERSIDAD SIMÓN BOLÍVAR

Relación con los *lookaheads*

Theorem

G es fuertemente $LL(k) \Leftrightarrow \forall A \in N, LA_k(A \to \alpha_i)$ *particionan* $LA_k(A)$

Demostración ($\Rightarrow$).

1. Supongamos que G es fuertemente $LL(k)$ y sea $z \in LA_k(A)$.

UNIVERSIDAD SIMÓN BOLÍVAR

Relación con los *lookaheads*

Theorem

G es fuertemente $LL(k) \Leftrightarrow \forall A \in N, LA_k(A \to \alpha_i)$ *particionan* $LA_k(A)$

Demostración ($\Rightarrow$).

1. Supongamos que G es fuertemente $LL(k)$ y sea $z \in LA_k(A)$.
2. Entonces solamente existe una regla A que puede usarse a partir de formas sentenciales $uA\alpha$ para derivar cadenas $uzw \in \Sigma^*$

UNIVERSIDAD SIMÓN BOLÍVAR

Relación con los *lookaheads*

Theorem

G es fuertemente $LL(k) \Leftrightarrow \forall A \in N, LA_k(A \to \alpha_i)$ *particionan* $LA_k(A)$

Demostración ($\Rightarrow$).

1. Supongamos que G es fuertemente $LL(k)$ y sea $z \in LA_k(A)$.
2. Entonces solamente existe una regla A que puede usarse a partir de formas sentenciales $uA\alpha$ para derivar cadenas $uzw \in \Sigma^*$
3. Necesariamente z estará en exactamente un $LA_k(A \to \alpha_i)$.

UNIVERSIDAD SIMÓN BOLÍVAR

Relación con los *lookaheads*

Theorem

G es fuertemente $LL(k) \Leftrightarrow \forall A \in N, LA_k(A \to \alpha_i)$ *particionan* $LA_k(A)$

Demostración ($\Rightarrow$).

1. Supongamos que G es fuertemente $LL(k)$ y sea $z \in LA_k(A)$.
2. Entonces solamente existe una regla A que puede usarse a partir de formas sentenciales $uA\alpha$ para derivar cadenas $uzw \in \Sigma^*$
3. Necesariamente z estará en exactamente un $LA_k(A \to \alpha_i)$.
4. En consecuencia los $LA_k(A \to \alpha_i)$ particionan $LA_k(A)$.

□

UNIVERSIDAD SIMÓN BOLÍVAR

Ambigüedad gramatical

Theorem

Si G es fuertemente $LL(k)$ para algún k entonces G no es ambigua.

- El reconocedor para la gramática es determinístico –
  Hay exactamente una regla aplicable para cada paso de derivación.
- La demostración queda como ejercicio.

UNIVERSIDAD SIMÓN BOLÍVAR

Algoritmo de cálculo de $FIRST_k$

Inicialización

input: $G = (N, \Sigma, P, S)$ CFG

{ Aplicación del Lema 2 de *FIRST* }

for each $\mathbf{a} \in \Sigma$ **do**

$F'(\mathbf{a}) \leftarrow \{\mathbf{a}\}$

end for

{ El *FIRST* de un no terminal comienza vacío o con λ si hay una λ-producción – Aplicación del Lema 1 de *FIRST* }

for each $A \in N$ **do**

$$F(A) \leftarrow \begin{cases} \{\lambda\} & \text{if } A \to \lambda \in P \\ \emptyset & \text{otherwhise} \end{cases}$$

end for

UNIVERSIDAD SIMÓN BOLÍVAR

Algoritmo de cálculo de $FIRST_k$

Cómputo

repeat

{Conservar conjuntos de la iteración anterior}

for each $A \in N$ **do**

$F'(A) \leftarrow F(A)$

end for

{Actualizar $FIRST$ para todo A aplicando el Lema 4 para $FIRST$}

for each $A \rightarrow u_1 u_2 \ldots u_n$ with $n > 0$ **do**

$F(A) \leftarrow F(A) \cup trunc_k(F'(u_1)F'(u_2)\ldots F'(u_k))$

end for

{Detenerse cuando deje de haber actualizaciones}

until $F(A) = F'(A)$ for all $A \in N$

output: $FIRST_k(A) = F(A)$

UNIVERSIDAD SIMÓN BOLÍVAR

Algoritmo de cálculo de $FOLLOW_k$

Inicialización

inputs: $G = (N, \Sigma, P, S)$ y $FIRST_k(A)$ for all $A \in N$

{ Aplicación del Lema 1 para $FOLLOW$ }
$FL(S) \leftarrow \{\lambda\}$
{ El $FOLLOW$ del resto de los no terminales comienza vacío }
for each $A \in N - \{S\}$ **do**
  $FL(A) \leftarrow \emptyset$
end for

Algoritmo de cálculo de $FOLLOW_k$

Cómputo

repeat
  {Conservar conjuntos de la iteración anterior}
  for each $A \in N$ **do**
    $FL'(A) \leftarrow FL(A)$
  end for
  for each $A \to w = u_1 u_2 \dots u_n$ **do**
    { Aplicación del Lema 2 para $FOLLOW$ }
    $L \leftarrow FL'(A)$
    if $u_n \in N$ **then**
      $FL(u_n) \leftarrow FL(u_n) \cup L$
    end if
    { Aplicación del Lema 3 para $FOLLOW$ }
    for $i = n - 1$ **to** 1 **do**
      $L \leftarrow trunc_k(FIRST_k(u_{i+1})L)$
      if $u_i \in N$ **then**
        $FL(u_i) \leftarrow FL(u_i) \cup L$
      end if
    end for
  end for
until $FL(A) = FL'(A)$ for all $A \in N$

output: $FOLLOW_k(A) = FL(A)$

UNIVERSIDAD SIMÓN BOLÍVAR

Consideraciones

- Una gramática ambigüa no puede ser fuertemente $LL(k)$.
- Demuestre que una gramática que contiene producciones de la forma

$$A \to \mathbf{a}\alpha$$
$$A \to \mathbf{a}\beta$$

  con $\alpha \neq \beta$, no puede ser fuertemente $LL(1)$.
  ¿Puede encontrar alguna condición para generalizar a $LL(k)$?
- Demuestre que una gramática que incluye algún no terminal A recursivo por *izquierda* no puede ser fuertemente $LL(k)$ para **ningún** $k > 0$.

UNIVERSIDAD SIMÓN BOLÍVAR

Bibliografía

- [*Sudkamp*]
  - Secciones 19.1 a 19.6
  - Ejercicios 19.2 a 19.6
- Practique las técnicas de transformación de gramáticas que permiten factorizar producciones por izquierda y eliminar la recursión izquierda generalizada.

UNIVERSIDAD SIMÓN BOLÍVAR