

3. Análisis Sintáctico

3.1. Conceptos fundamentales

- Especificación sintáctica de los Lenguajes de Programación
- Introducción al Análisis Sintáctico

3.2. Análisis Sintáctico Descendente

- Condición LL(1): Gramáticas LL(1)
- Construcción de Analizadores Sintácticos LL(1)

3.3. Análisis Sintáctico Ascendente

- Conceptos fundamentales: ASA por “Desplazamiento-Reducción”
- Gramáticas LR(0) y Gramáticas SLR(1)
- Relación entre gramáticas y resolución de conflictos
- Tratamiento y recuperación de errores

➤ Gramáticas incontextuales: $G = (N, T, P, S)$

$$S \in N; V = N \cup T; N \cap T = \emptyset; (A \rightarrow \alpha) \in P; A \in N; \alpha \in V^*$$

➤ Derivación directa

$$\delta A \gamma \Rightarrow \delta \alpha \gamma \text{ sii } \exists (A \rightarrow \alpha) \in P; \quad \delta, \gamma \in V^*$$

➤ Derivación

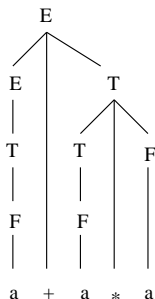
$$\alpha \xRightarrow{*} \beta \text{ sii } \exists \alpha_0, \dots, \alpha_m \in V^*: \alpha_0 = \alpha \Rightarrow \alpha_1 \Rightarrow \dots \alpha_{m-1} \Rightarrow \alpha_m = \beta$$

➤ Forma Sentencial: $\alpha \in V^* \text{ sii } \exists S \xRightarrow{*} \alpha$

➤ Lenguaje Generado: $L(G) = \{x \mid x \in T^* : S \xRightarrow{+} x\}$

Ejemplo-1

- 1) $E ::= E + T$
- 2) $E ::= T$
- 3) $T ::= T * F$
- 4) $T ::= F$
- 5) $F ::= (E)$
- 6) $F ::= a$



Secuencia de Derivación a Izquierdas

$$\begin{aligned} E &\Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \Rightarrow \\ &a + T \Rightarrow a + T * F \Rightarrow \\ &a + F * F \Rightarrow a + a * F \Rightarrow \\ &a + a * a \end{aligned}$$

Secuencia de Derivación a Derechas

$$\begin{aligned} E &\Rightarrow E + T \Rightarrow E + T * F \Rightarrow \\ &E + T * a \Rightarrow E + F * a \Rightarrow \\ &E + a * a \Rightarrow T + a * a \Rightarrow \\ &F + a * a \Rightarrow a + a * a \end{aligned}$$

➤ Árbol de Derivación. Dado $G = (N, T, P, S)$:

- la raíz está etiquetada con el símbolo inicial S ,
- cada hoja está etiquetada con un símbolo de $T \cup \{\epsilon\}$,
- cada nodo interior está etiquetado con un símbolo de N ,
- sea $A \in N$ la etiqueta de un nodo, y x_1, \dots, x_n las etiquetas de sus nodos hijos (de izquierda a derecha); entonces: $A \rightarrow x_1 \dots x_n \in P$.

➤ Teorema

Dada $G(S)$, $S \xRightarrow{*} \alpha$ (con $\alpha \in V^*$) sii existe un árbol de derivación que produce α .

➤ Gramática ambigua

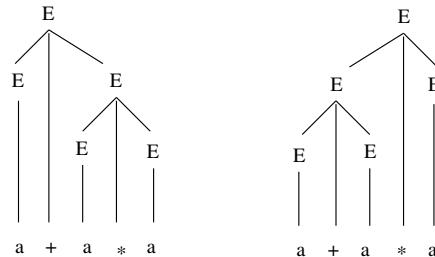
Una gramática es ambigua sii dado $x \in T^*$ existen más de un árbol de derivación que produce x .

➤ Proposición

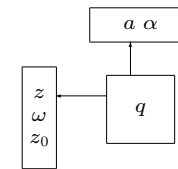
Determinar si una gramática es ambigua, es un problema indecidible.

Ejemplo-2

- 1) $E ::= E + E$
- 2) $E ::= E * E$
- 3) $E ::= (E)$
- 4) $E ::= a$



- **Autómata a pila.** $A_p = (Q, T, \Gamma, \delta, q_0, z_0, F)$ $q_0 \in Q; z_0 \in \Gamma; F \subseteq Q;$
 $\delta : Q \times (T \cup \{\epsilon\}) \times \Gamma \rightarrow \wp(Q \times \Gamma^*)$



$(q, a\alpha, z\omega z_0) \Leftarrow \text{configuración}$

- **Movimiento:** $(q, a\alpha, z\omega z_0) \vdash (q', \alpha, \beta\omega z_0)$ **si** $\exists (q', \beta) \in \delta(q, a, z)$.
con: $q, q' \in Q; a \in T \cup \{\epsilon\}; \alpha \in T^*; z \in \Gamma; \beta, \omega \in \Gamma^*$.

- **Lenguaje aceptado a estado final:**

$$L_F(A_p) = \{x \mid x \in T^* : (q_0, x, z_0) \vdash^* (q, \epsilon, \gamma); q \in F\}.$$

- **Lenguaje aceptado a pila vacía**

$$L_v(A_p) = \{x \mid x \in T^* : (q_0, x, z_0) \vdash^* (q, \epsilon, \epsilon)\}.$$

INTRODUCCIÓN AL AS

Aproximaciones al Análisis Sintáctico

➤ Análisis Sintáctico Descendente

Construye el árbol de análisis desde la raíz hasta las hojas

⇒ *Secuencia de Derivación a Izquierda*

➤ Análisis Sintáctico Ascendente

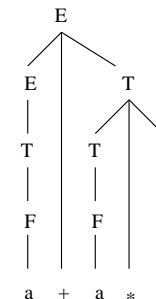
Construye el árbol de análisis desde las hojas hasta la raíz

⇒ *Inversa de la Secuencia de Derivación a Derecha*

INTRODUCCIÓN AL AS

Ejemplo-3

- 1) $E ::= E + T$
- 2) $E ::= T$
- 3) $T ::= T * F$
- 4) $T ::= F$
- 5) $F ::= (E)$
- 6) $F ::= a$



A.S.D.		A.S.A.	
forma sentencial	regla a derivar	forma sentencial	regla a reducir
<u>E</u>	r-1	<u>a</u> +a*a	r-6
<u>E</u> +T	r-2	<u>E</u> +a*a	r-4
<u>T</u> +T	r-4	<u>T</u> +a*a	r-2
<u>E</u> +T	r-6	<u>E</u> +a*a	r-6
<u>a</u> +T	r-3	<u>E</u> +F*a	r-4
<u>a</u> + <u>T</u> *F	r-4	<u>E</u> +T*a	r-6
<u>a</u> + <u>F</u> *F	r-6	<u>E</u> +T*F	r-3
<u>a</u> +a* <u>F</u>	r-6	<u>E</u> +T	r-1
<u>a</u> +a*a		<u>E</u>	

Tipos de Análisis Sintáctico: problema de indeterminismo

- Métodos con retroceso Complejidad exponencial
- Métodos tabulares: Complejidad cúbica
 - Algoritmo de *Cocke-Younger-Kasami*
 - Algoritmo de *Earley*
- Métodos deterministas: Complejidad lineal
 - ⇒ Caracterizar la clase de gramáticas deterministas
 - ⇒ Diseñar el algoritmo de AS determinista