3. Análisis Sintáctico

3.1. Conceptos fundamentales

- Especifiación 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: $G = (\Sigma,\ N,\ S,\ P)$

  $S \in N; \quad V = N \cup \Sigma; \quad N \cap \Sigma = \emptyset; \quad (A \to \alpha) \in P; \quad A \in N; \quad \alpha \in V^*$

- Derivación directa

  $\delta\ A\ \gamma\ \Rightarrow\ \delta\ \alpha\ \gamma$ **sii** $\exists (A \to \alpha) \in P; \qquad \delta, \gamma \in V^*$

- Derivación

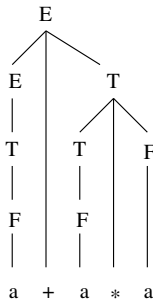
  $\alpha \overset{*}{\Rightarrow} \beta$ **sii** $\exists\ \alpha_0, \ldots \alpha_m \in V^* : \quad \alpha = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \alpha_{m-1} \Rightarrow \alpha_m = \beta$

- Forma Sentencial: $\alpha \in V^*$ **sii** $\exists\ S \overset{*}{\Rightarrow} \alpha$

- Lenguaje Generado: $L(G) = \{x \mid x \in \Sigma^* :\ S \overset{+}{\Rightarrow} x\}$

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

- Teorema

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

- Gramática ambigua

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

- Proposición

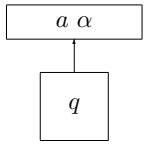
  Determinar si una gramática es ambigua, es un problema indecidible.

Ejemplo-2

E → E + E
E → E * E
E → (E)
E → a

Máquinas aceptoras: Autómatas

➢ Autómata de Estados Finitos



$$AEF = (\Sigma,\ Q,\ q_0,\ F,\ \delta)$$

$$q_0 \in Q; \quad F \subseteq Q; \quad \delta : Q \times \Sigma \rightarrow \wp(Q)$$

Configuración $\Rightarrow$ $(q,\ a\alpha)$

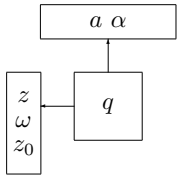
➢ Movimiento: $(q,\ a\alpha) \vdash (q',\ \alpha)$ **sii** $\exists\ q' \in \delta(q,\ a)$

con: $q, q' \in Q; \quad a \in \Sigma; \quad \alpha \in \Sigma^*$

➢ Lenguaje aceptado por un AEF

$$L(AEF) = \{x \mid x \in \Sigma^* : \quad (q_0,\ x) \overset{*}{\vdash} (q,\ \epsilon) \quad q \in F\}.$$

➢ Autómata a Pila



$$A_p = (\Sigma,\ \Gamma,\ Q,\ q_0,\ F,\ z_0,\ \delta)$$

$$q_0 \in Q; \quad z_0 \in \Gamma; \quad F \subseteq Q;$$
$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \ \rightarrow \ \wp(Q \times \Gamma^*)$$

Configuración $\Rightarrow$ $(q,\ a\alpha,\ z\omega z_0)$

➢ Movimiento: $(q,\ a\alpha,\ z\omega z_0) \vdash (q',\ \alpha,\ \beta\omega z_0)$ **sii** $\exists (q',\ \beta) \in \delta(q,\ a,\ z)$

con: $q, q' \in Q; \quad a \in \Sigma \cup \{\epsilon\}; \quad \alpha \in \Sigma^*; \quad z \in \Gamma; \quad \beta, \omega \in \Gamma^*$

➢ Lenguaje aceptado a estado final:

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

➢ Lenguaje aceptado a pila vacía

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

Aproximaciones al Análisis Sintáctico

➢ **Análisis Sintáctico Descendente**

Construye el árbol de análisis desde la raíz hasta las hojas

⇨ *Secuencia de Derivación a Izquierda*

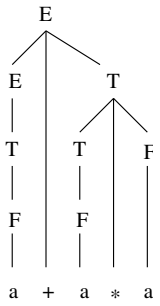
➢ **Análisis Sintáctico Ascendente**

Construye el árbol de análisis desde las hojas hasta la raíz

⇨ *Inversa de la Secuencia de Derivación a Derecha*

Ejemplo-3 (ASD)

1) E –> E + T
2) E –> T
3) T –> T * F
4) T –> F
5) F –> (E)
6) F –> a



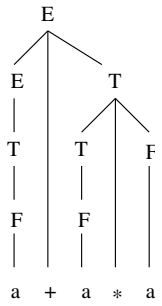
Secuencia de Derivación a Izquierdas

$$E \Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \Rightarrow a + T \Rightarrow a + T * F \Rightarrow a + F * F \Rightarrow a + a * F \Rightarrow a + a * a$$

A.S.D.	
Forma sentencial	Regla a derivar
E	r-1
E+T	r-2
T+T	r-4
F+T	r-6
a+T	r-3
a+T*F	r-4
a+F*F	r-6
a+a*F	r-6
a+a*a	

Ejemplo-3 (ASA)

1) E –> E + T
2) E –> T
3) T –> T * F
4) T –> F
5) F –> (E)
6) F –> a



Secuencia de Derivación a Derechas

$$E \Rightarrow E + T \Rightarrow E + T * F \Rightarrow E + T * a \Rightarrow E + F * a \Rightarrow E + a * a \Rightarrow T + a * a \Rightarrow F + a * a \Rightarrow a + a * a$$

A.S.A.	
Forma sentencial	Regla a reducir
a+a*a	r-6
F+a*a	r-4
T+a*a	r-2
E+a*a	r-6
E+F*a	r-4
E+T*a	r-6
E+T*F	r-3
E+T	r-1
E	

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