Tema 3 | Análisis Sintáctico

Contenidos

Bibliografía básica

24/09/13

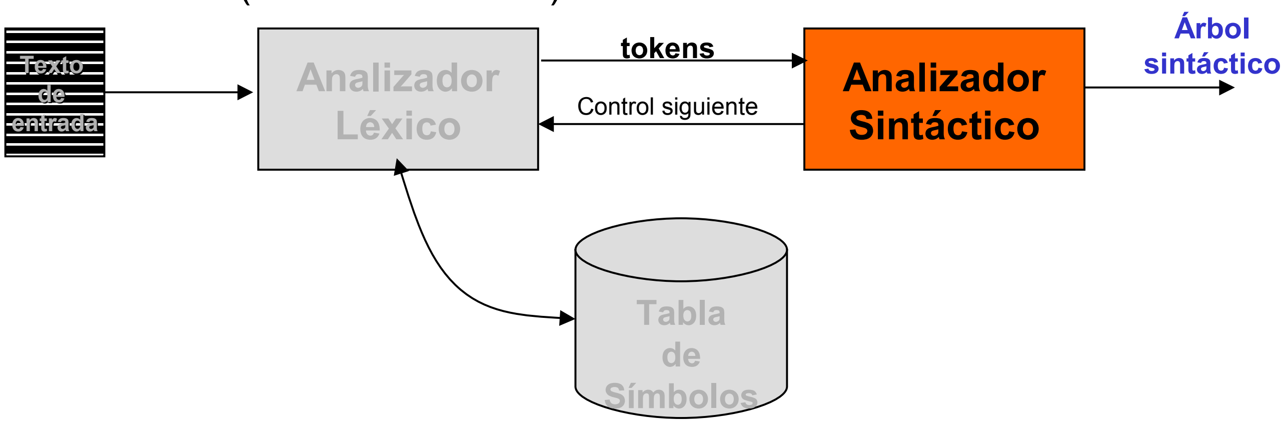
[Aho90]	Alfred V. Aho, Ravi Sethi, Jffrey D. Ullman *Compiladores. Principios, técnicas y herramientas.* Addison-Wesley Iberoamericana 1990.
[Broo93]	J.G. Brookshear *Teoría de la Computación. Lenguajes formales, autómatas y complejidad.* Addison-Wesley Iberoamericana 1993.
[Hopc79]	J.E. Hopcroft, J.D. Ullman *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley 1979.

3.1 Descripción Funcional

Objetivo: Analizar las secuencias de tokens y comprobar que son correctas sintácticamente.

A partir de una secuencia de tokens, el analizador sintáctico nos devuelve:

1. Si la secuencia es correcta o no sintácticamente (existe un conjunto de reglas gramaticales aplicables para poder estructurar la secuencia de tokens).
2. El orden en el que hay que aplicar las producciones de la gramática para obtener la secuencia de entrada (**árbol sintáctico**).



Si no se encuentra un árbol sintáctico para una secuencia de entrada, entonces la secuencia de entrada es incorrecta sintácticamente (tiene errores sintácticos).

3.2 Fundamentos. Gramáticas libres de contexto

Una gramática definida como $G = (V_N, V_T, P, S)$, donde:

- V_N es el conjunto de símbolos no terminales.
- V_T es el conjunto de símbolos terminales.
- P es el conjunto de producciones.
- S es el símbolo inicial.

se dice que es una *gramática libre de contexto* cuando el conjunto de producciones P obedece al formato:

$$P = \{A \rightarrow \alpha \;/\; A \in V,\ \alpha \subset (V_N \cup V_T)^*\}$$

es decir, sólo admiten tener un símbolo no terminal en su parte izquierda. La denominación *libre de contexto* se debe a que se puede cambiar A por α, independientemente del contexto en el que aparezca A.

3.2 Fundamentos. Árbol Sintáctico

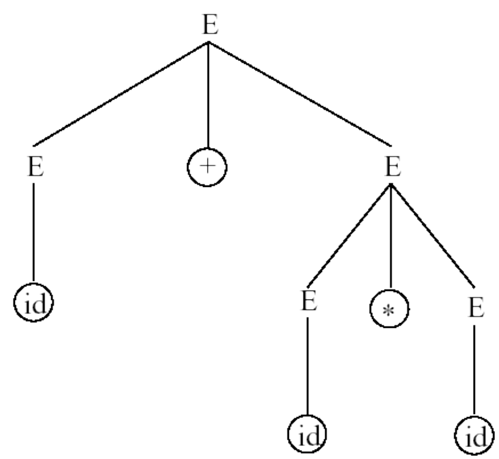
Es una **representación gráfica** donde aparecen las producciones de la gramática aplicadas y en el orden que son aplicadas para obtener una secuencia de símbolos de un lenguaje.

En las hojas aparecen símbolos terminales y en los nodos interiores aparecen símbolos no terminales.

Ejemplo 4.1: Sean las producciones de la gramática G:

A la siguiente secuencia de entrada: `id+id*id` le corresponde el árbol sintáctico siguiente:

$$
\begin{array}{lcl}
E & \rightarrow & E+E \\
 & | & E*E \\
 & | & (E) \\
 & | & id
\end{array}
$$

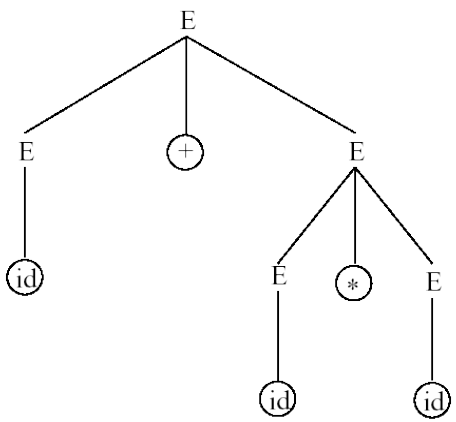


$$
\begin{array}{lcl}
E & \rightarrow & E+E \\
 & & id+E \\
 & & id+E*E \\
 & & id+id*id
\end{array}
$$

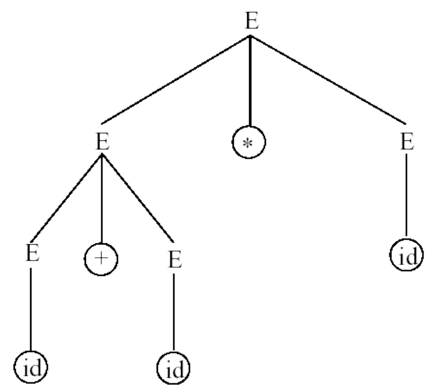
3.2 Fundamentos. Gramática Ambigua

Una gramática es ambigua cuando admite más de un árbol sintáctico para una misma secuencia de símbolos de entrada.

Ejemplo 3.2: Dadas las producciones de la gramática del ejemplo 4.1 y dada la misma secuencia de entrada **`id+id*id`**, se puede apreciar que le pueden corresponder dos árboles sintácticos.



$$\begin{aligned} E &\rightarrow E + E \\ &\ id + E \\ &\ id + E * E \\ &\ id + id * id \end{aligned}$$



$$\begin{aligned} E &\rightarrow E * E \\ &\ E * id \\ &\ E + E * id \\ &\ id + id * id \end{aligned}$$

3.2 Fundamentos. Formas Normales

GRAMÁTICA EN FORMA NORMAL DE CHOMSKY (CNF): Toda gramática libre de contexto cuyas producciones son de la forma:

$$A \rightarrow Bc$$
$$A \rightarrow a$$

Donde A y B son símbolos no terminales y a y c son terminales.

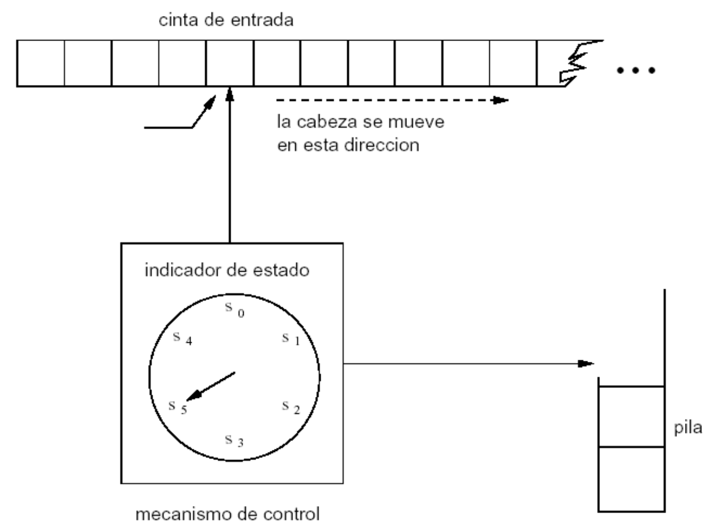
GRAMÁTICA EN FORMA NORMAL DE GREIBACH (GNF): Toda gramática libre de contexto cuyas producciones son de la forma:

$$A \rightarrow a\alpha$$

Donde A es un símbolo no terminal, a es un símbolo terminal y α es una cadena de símbolos perteneciente al conjunto de símbolos no terminales y la cadena vacía.

3.2 Fundamentos. Autómatas de Pila (1)

Un Autómata de Pila se define formalmente como una séptupla: $AP = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$



- Q es el conjunto finito de estados.
- Σ es el alfabeto de entrada (es finito).
- Γ es el alfabeto de la pila.
- δ es la función de transición y es una aplicación de la forma:

$$\delta : Q \times \{\Sigma \cup \{\lambda\}\} \times \Gamma \rightarrow Q \times \Gamma^*$$

de tal forma que:

$$\delta(q,a,z) = \{(p_1,\gamma_1),(p_2,\gamma_2),\ldots,(p_m,\gamma_m)\}$$

donde:

$$q \in Q, \;\; a \in (\Sigma \cup \{\varepsilon\}), \;\; z \in \Gamma, \;\; p_i \in Q \;\; i = 1,\ldots,m \;\; \gamma_i \subseteq \Gamma^*$$

- q_0 es el estado inicial y cumple que $q_0 \in Q$.
- Z_0 es el símbolo inicial que contiene la pila antes de comenzar. Evidentemente, $Z_0 \in \Gamma$.
- F es el conjunto de estados finales. Evidentemente, $F \subset Q$.

3.2 Fundamentos. Autómatas de Pila (2)

- **Descripción instantánea de un Autómata de Pila**
  Expresa el estado actual del autómata en base a la cadena de símbolos de entrada y al estado en el que se encuentra la pila.

$$DI = (q, \omega, \gamma), \text{ donde } q \in Q,\ \omega \in \Sigma^*,\ \gamma \in \Gamma^*$$

  se dice que $(q, a\omega, Z\alpha) \Rightarrow_m (p, \omega, \beta\alpha)$ si $(p, \beta) \subseteq \delta(q, a, z)$. También se puede definir el símbolo $\Rightarrow_m^*$ como la clausura de varias transiciones.

- **Lenguaje aceptado por un Autómata de Pila**
  Sea el autómata de pila $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, se define $L(M)$ como el lenguaje aceptado por M:

$$L(M) = \{\omega \ / \ (q_0, \omega, z_0) \Rightarrow_m^* (p, \varepsilon, \gamma) \text{ para algún } p \in F,\ \gamma \in \Gamma^*\}$$

3.2 Fundamentos. Autómatas de Pila (3)

- **Lenguaje aceptado por un Autómata de Pila Vacía**
  Se denota por $N(M)$ y se define para una pila M:

$$N(M) \;=\; \{\omega \;/\; (q_0, \omega, z_0) \Rightarrow_m^* (p, \varepsilon, \varepsilon) \text{ para algún } p \in Q$$

- **Equivalencia entre un Autómata de Pila y un Lenguaje Libre de Contexto**

Teorema 4.1 *Si L es un lenguaje libre de contexto, entonces existe un autómata de pila $N(M)$ tal que $L = N(M)$*

(Ver demostración en [Broo93])

Teorema 4.2 *Si $L = N(M)$ para un autómata de pila, entonces L es libre de contexto.*

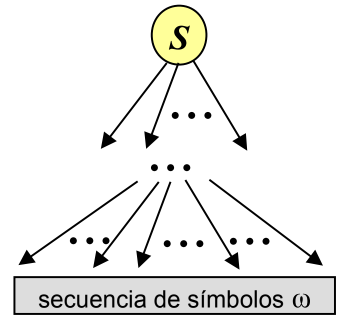
(Ver demostración en [Broo93])

3.3 Estrategias de Análisis Sintáctico

Dada una secuencia de símbolos ω, existen dos estrategias para verificar si se trata de una secuencia de símbolos de un lenguaje $L(G)$.

(a) *Análisis descendente (Top-Down):* Partir del símbolo inicial de la gramática y generar los árboles sintácticos hasta que se alcance la secuencia de símbolos ω.

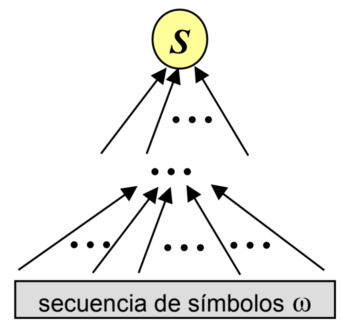
Sea S el símbolo inicial de la gramática. Se dice que ω es correcta sintácticamente si $S \Rightarrow_G^* \omega$.



(b) *Análisis ascendente (Bottom-Up):* Partir de la propia secuencia ω y buscar las subcadenas que coincidan con las partes derechas de las producciones y reescribirlas por la parte izquierda (Reducción), hasta alcanzar el símbolo inicial de la gramática.

Sea $A \rightarrow \alpha$ una producción de la gramática G. Una reducción es $\alpha \rightarrow A$. Se denota las reducciones sucesivas como $\rightarrow_G^*$.

Se dice que ω es correcta sintácticamente si se cumple que $\omega \rightarrow_G^* S$, donde S es el símbolo inicial de la gramática G. Es decir, aplicando reducciones sucesivas a ω se alcanza S.



Ejemplo 3.2: Sea la gramática definida por las producciones siguientes:

$$S \rightarrow aABe$$
$$A \rightarrow cB \mid b$$
$$B \rightarrow d \mid b$$

Y dada la cadena de entrada ***acdbe***, comprobar si es correcta sintácticamente de acuerdo con la gramática G