

Análisis sintáctico predictivo para procesar cadenas leyéndolas de izquierda a derecha (Left-to-right) y que construye una derivación por la izquierda (Leftmost).

Un analizador predictivo se basa en la idea de elegir la producción $A \rightarrow \alpha$ de la gramática, si el siguiente token a procesar es a y pertenece a $\text{FIRST}(\alpha)$. Estas elecciones se reúnen en una tabla auxiliar M llamada de **tabla de análisis predictivo** cuyas entradas están determinadas por X un símbolo no-terminal y a un símbolo terminal. Las entradas de la tabla $[X, a]$ indican qué producción debe usarse si se quiere derivar una cadena $a\omega$ a partir de X .

Para construir la tabla de parsing LL(1):

Para cada producción $X \rightarrow \alpha$ hacer:

1. Para cada terminal a en $\text{FIRST}(\alpha)$ agregar la entrada $M[X, a] = X \rightarrow \alpha$
2. Si $\varepsilon \in \text{FIRST}(\alpha)$ entonces:

para cada $b \in \text{FOLLOW}(X)$ agregar $M[X, b] = X \rightarrow \alpha$

Si $\varepsilon \in \text{FIRST}(\alpha)$ y $\# \in \text{FOLLOW}(X)$ entonces:

agregar $M[X, \#] = X \rightarrow \alpha$

Una entrada de la tabla para este tipo de parsers tiene una sola producción. Las entradas vacías de la tabla, indican que no existe una producción para derivar la entrada. Alternativamente se puede completar las entradas vacías con alguna función auxiliar para el manejo de errores.

Algoritmo para LL(1)

Para reconocer una cadena perteneciente al lenguaje de una gramática **LL(1)** usamos un autómata de pila cuyo estado inicial contiene un símbolo inicial nuevo S y el fondo de la pila es el símbolo de fin de cadena $\#$.

El algoritmo construye una derivación izquierda y/o un árbol de sintaxis como objeto final. En cada paso se puede observar la derivación o la parte del árbol con w la (sub)cadena que se ha reconocido hasta el momento, la pila contiene una secuencia de símbolos α tal que $S \rightarrow^* w\alpha$. En cada iteración se considera el tope de la pila y el siguiente token, si se tiene un símbolo no-terminal entonces se consulta la tabla para decidir qué producción utilizar, en otro caso se comparan los símbolos terminales y deben coincidir.

- Se inicia con a el primer símbolo de ω y $S\#$ en la pila.
- Mientras la pila tenga elementos:
sea X el tope de la pila, a el token actual y M la tabla de análisis.
 - Si X es un símbolo terminal y $X = a$ entonces se saca a X del tope de la pila y se lee el siguiente token
 - Si X es un símbolo terminal y $X \neq a$ entonces hay un error
 - Si X es un símbolo no-terminal entonces:
 - $M[X, a]$ contiene una producción $X \rightarrow Y_1..Y_k$
se saca a X del tope de la pila y se agrega a $Y_k...Y_1$ donde Y_1 es el nuevo tope;
se construye el nodo en el árbol de derivación o se registra la producción como parte de la derivación
 - $M[X, a]$ no contiene una producción entonces hay un error

¹Resumen tomado del libro "Compilers, Principles, Techniques and Tools", Aho A. et al., Capítulo 4.