

**Algoritmo 3.2:** Análisis sintáctico predictivo, controlado por una tabla. (Aho, Lam, Sethi, & Ullman, 2008, pág. 226)

**Entrada:** Una cadena  $w$  y una tabla de análisis sintáctico  $M$  para la gramática  $G$ .

**Salida:** Si  $w$  está en el lenguaje de la gramática  $L(G)$ , una derivación por la izquierda de  $w$ ; de lo contrario, una indicación de error.

**Método:** Al principio, el analizador sintáctico se encuentra en una configuración con  $w$  en el búfer de entrada, y el símbolo inicial  $S$  de  $G$  en la parte superior de la pila, por encima de  $\$$ .

establecer  $ip$  para que apunte al primer símbolo de  $w$ ;  
establecer  $X$  con el símbolo de la parte superior de la pila;  
while ( $X \neq \$$ ) { /\* mientras la pila no está vacía \*/  
  if ( $X$  es  $a$ ) extraer de la pila y avanzar ( $ip$ ; /\*  $a$  símbolo al que apunta  $ip$  \*/  
  else if ( $X$  es un terminal) error();  
  else if ( $M[X, a]$  es una entrada de error) error();  
  else if ( $M[X, a] = X \rightarrow Y_1 Y_2 \dots Y_k$ ) {  
    enviar de salida la producción  $X \rightarrow Y_1 Y_2 \dots Y_k$ ;  
    extraer de la pila;  
    meter  $Y_k, Y_{k-1}, \dots, Y_1$  en la pila, con  $Y_1$  en la parte superior;  
  }  
  establecer  $X$  con el símbolo de la cima de la pila;  
}

Tabla 3.1

Tabla de análisis sintáctico para la gramática 3.3

No terminal	Símbolo de entrada					
	id	+	*	(	)	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Contiene opciones de producción a emplear en el paso apropiado del análisis sintáctico

Arreglo bidimensional indizado por no terminales y terminales

Tabla de análisis sintáctico

La segunda L rastrea una derivación por la izquierda para la cadena de entrada

Una gramática  $G$  es LL(1) si, y sólo si cada vez que  $A \rightarrow \alpha \mid \beta$ ,

La primera L procesa la entrada de derecha a izquierda

$1(X) = \{X\}$ .

Si X es un terminal entonces primero

$X \rightarrow Y_1 Y_2 \dots Y_k$

Si X es un no terminal

Si  $X \rightarrow e$  es una producción, entonces se agrega

$\epsilon$  a  $\text{Primero}(X)$ .

En donde  $\text{Primero}(\beta)$  contiene a e, entonces todo lo que hay en  $\text{Siguiente}(A)$  está en  $\text{Siguiente}(B)$ .

Entonces todo lo que hay en  $\text{Primero}(\beta)$  excepto e está en  $\text{Siguiente}(B)$ .

Si hay una producción  $A \rightarrow \alpha\beta$ , o una producción  $A \rightarrow \alpha\beta\beta$ .

Si hay una producción  $A \rightarrow \alpha\beta\beta$ .

Asociadas a una gramática  $G$

$G = (\{0, 1\}, \{\bar{N}, C\}, N, P)$

Elegir producción que se aplicará con base al siguiente símbolo de entrada

Si X es un símbolo de la gramática (un terminal o no terminal) o  $\epsilon$ ,

Análisis sintáctico por descenso recursivo

Considere la gramática

$S \rightarrow cAd$

$A \rightarrow ab \mid a$

Entrada  $w = cad$

Revisar las dos posibilidades de A

Colocar \$ en  $\text{Siguiente}(S)$ .

El terminal buscado en la producción

Se construye un árbol con tres nodos T'. En el primer nodo T' (en preorder) se elige la producción  $T' \rightarrow *FT'$ ; y en el segundo y tercer nodo se elige la producción  $T' \rightarrow \epsilon$

Procedimiento correspondiente

Lado derecho de la producción

Procedimiento para cada no terminal de la gramática

El terminal buscado en la producción

Se construye un árbol con tres nodos T'. En el primer nodo T' (en preorder) se elige la producción  $T' \rightarrow *FT'$ ; y en el segundo y tercer nodo se elige la producción  $T' \rightarrow \epsilon$

