

Indica si la secuencia de componentes léxicos, encontrados en el análisis léxico, están en el orden correspondiente a las reglas gramaticales del lenguaje

Análisis Sintáctico

Una gramática G es LL(1) si, y sólo si cada vez que $A \rightarrow \alpha \mid \beta$, son dos producciones distintas de G, se aplican las siguientes condiciones:

1. Para el no terminal α , tanto α como β derivan cadenas que empiecen con α .
2. A lo mucho, sólo α o β puede derivar la cadena vacía.
3. Si $\beta \Rightarrow^* \epsilon$, entonces β no deriva a ninguna cadena que empiece con una terminal en Siguiente(α). De igual forma, si $\alpha \Rightarrow^* \epsilon$, entonces α no deriva a ninguna cadena que comience con un terminal en Siguiente(α)

La primera "L" se refiere al hecho de que se procesa la entrada de izquierda a derecha. (del inglés "Left-right"). La segunda "L" hace referencia al hecho de que rastrea una derivación por la izquierda para la cadena de entrada. El número 1 entre paréntesis significa que se utiliza sólo un símbolo de entrada para predecir la dirección del análisis sintáctico.

Gramáticas LL(1)

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	$F \rightarrow id$			$F \rightarrow (E)$	

Es un árbol etiquetado en el cual los nodos interiores están etiquetados por no terminales

Árboles de análisis sintáctico

ALAN REYES GUTIERREZ

Capítulo 3. Análisis Sintáctico

Conjuntos Primero y Siguienente

Si X es un símbolo de la gramática (un terminal o no terminal) o ϵ , entonces el conjunto Primero(X), compuesto de terminales, y posiblemente de ϵ , se calcula aplicando para todos los símbolos gramaticales de X las siguientes reglas hasta que no puedan agregarse más terminales o ϵ a ningún conjunto Primero.

Una gramática libre de contexto es una especificación para la estructura sintáctica de un lenguaje de programación

Gramáticas libres de contexto

Una derivación para la expresión: $x + y * 60$

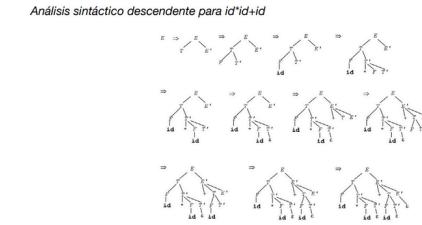
- 1) $Exp \Rightarrow Exp Op Exp$
- 2) $\Rightarrow Exp Op Exp Op Exp$
- 3) $\Rightarrow Exp Op Exp Op num$
- 4) $\Rightarrow Exp Op Exp * num$
- 5) $\Rightarrow Exp Op id * num$
- 6) $\Rightarrow Exp + id * num$
- 7) $\Rightarrow id + id * num$

Una gramática libre de contexto consta de cuatro componentes:

1. Un conjunto de símbolos terminales.
2. Un conjunto de símbolos no terminales.
3. Un conjunto de producciones
4. Uno de los símbolos no terminales designado como símbolo inicial.

Análisis sintáctico descendente para la expresión: $x + y * 60$

Se denomina descendente porque para generar el árbol de análisis sintáctico se realiza de la raíz a las hojas.



Análisis sintáctico descendente

Cada procedimiento sigue la secuencia del lado derecho de la producción, si se evalúa un no terminal, se llama al procedimiento correspondiente; y cuando se evalúa un terminal, se verifica que sea el terminal buscado en la producción, en caso de que sí lo sea, se continúa con la secuencia de la producción, de lo contrario se llama a una rutina de error

Bosquejo de un procedimiento para un no terminal en un analizador sintáctico descendente

```
void A() {
    Elegir una producción A,  $A \rightarrow X_1 X_2 \dots X_k$ ;
    for( i = 1 a k ){
        if(  $X_i$  es un no terminal )
            llamar al procedimiento  $X_i()$ ;
        else if(  $X_i$  es igual al símbolo de entrada actual a )
            avanzar la entrada hasta el siguiente símbolo;
        else /* rutina de error y retroceso() */;
    }
}
```