Análisis Sintáctico

Propósito

- Verifica que los componentes léxicos se encuentran en un orden prescrito.
- Recibe una secuencia de tokens o lexema del analizador léxico y decide si la secuencia está bien escrita.
- Hace uso de reglas de derivación de la gramática del lenguaje

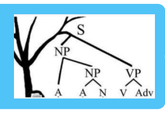
Relación con el Análisis Léxico

- Depende directamente de la salida del análisis léxico.
- El análisis sintáctico agrupa estos tokens en estructuras más complejas (expresiones, sentencias, bloques de código).
- Ejemplo:
  - Código fuente
    x = a + b * 5;
  - Análisis léxico
    ID(x) OP(=) ID(a) PO(+) ID(b) PO(*) NUM(5) ;
  - Análisis sintáctico: los organiza en una estructura jerárquica según las reglas del lenguaje

Gramáticas libres de contexto

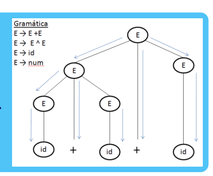
- Es una especificación para la estructura sintáctica de un lenguaje de programación
- Consta de cuatro componentes:
  - Terminales { id, num, +, -, *, / }
  - No terminales: { Exp, Op }.
  - Producciones: { Exp → Exp Op Exp; Exp → id; Exp → num; Op → + | - | * | / }
  - Símbolo inicial: { Exp }

Generación del Árbol Sintáctico

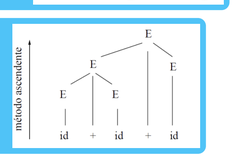


- Cada nodo interno representa una regla gramatical aplicada.

- Cada nodo hoja representa un token del lenguaje.
- La raíz del árbol representa una construcción completa, como una expresión o una sentencia.
- Dos métodos para analizadores sintácticos
  - Descendentes
    Se parte de la raíz y se hacen las derivaciones hasta llegar a las hojas.

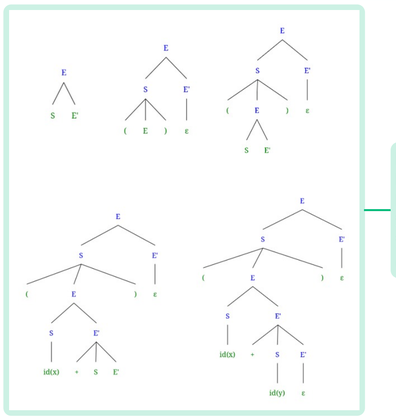


  - Ascendentes
    Se parte de las hojas (terminales) y se van haciendo reducciones hasta llegar al nodo raíz.



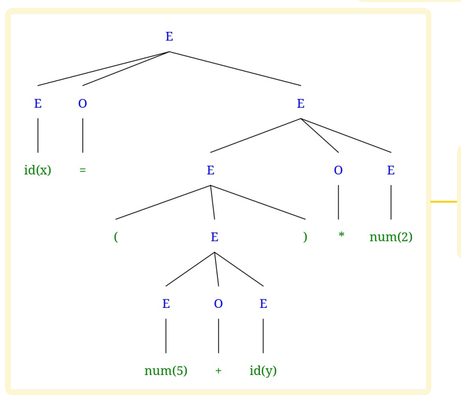
Análisis sintáctico descendente

- Analiza una cadena de componentes léxicos de entradamediante la búsqueda de los pasos para una derivación por la izquierda
- Ejemplo:
  E -> SE'
  E' -> +SE' | -SE' | ε
  S -> (E) | id
  Símbolo inicial: E
  (x + y)



Arboles de análisis sintáctico

- Muestra de forma gráfica la manera en que el símbolo inicial de una gramática deriva a una secuencia de componentes.
- Ejemplo:
  E -> E O E | (E) | num | id
  O -> + | - | * | /
  Símbolo inicial: E
  x = (5 + y) * 2



Análisis Sintáctico Descendente Recursivo

- Definir un procedimiento para cada no terminal de la gramática
- Ejemplo:
  El pseudo-código para análisis sintáctico descendente recursivo de la siguiente gramática
  S -> (E) | id | num

```
function parse(input):
    index = 0

    if S() and index == length(input):
        print("La cadena es válida")
    else:
        print("La cadena no es válida")


function S():
    //lookhead obtiene el siguiente token de la cadena
    if lookahead() == '(':
        //Consume se encarga de consumir el token y se lee el siguiente
        consume('(')
        if not E():
            return false
        if lookahead() == ')':
            consume(')')
            return true
        else:
            return false
    elif lookahead() == 'id':
        consume('id')
        return true
    elif lookahead() == 'num':
        consume('num')
        return true
    else:
        return false
```


Universidad Autónoma del Estado de Hidalgo
Licenciatura en Computación
Asignatura: Autómatas y Compiladores
Estudiante: ***Norberto Hernández Cárdenas***
Semestre: Sexto, Grupo 3
Fecha: 23 de marzo 2025

Referencias:
1) Sahagún, D. U. C., & Águila, O. A. Z. (2024). Análisis Sintáctico. En Compiladores: fases de análisis (pp. 45-68). Editorial Transdigital.