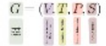
Capítulo 3: Análisis sintáctico

Funciones del análisis sintáctico

- Verificar que la secuencia de tokens del analizador léxico cumple la gramática del lenguaje 
- Construcción de la estructura jerárquica (árbol de análisis o AST) que representa el programa. 
- Detección y recuperación de errores sintácticos. 
- Interfaz con análisis semántico y generación de código intermedio (en compiladores dirigidos por sintaxis). 

Gramáticas y especificación de la sintaxis

- Definición de gramática libre de contexto: conjuntos de terminales, no terminales, reglas de producción, axioma.ntes temas 
- Ambigüedad, recursividad, problemas en gramáticas. 
- Transformaciones para hacer gramáticas analizadas por métodos eficientes (por ejemplo para análisis lineal). 

Métodos de análisis sintáctico

- Análisis descendente (top-down): partiendo del símbolo inicial, aplicando reglas hacia las hojas. 
- Análisis ascendente (bottom-up): desde los tokens (hojas) hacia la raíz. 
- Clasificación de gramáticas: LL(k), LR(k), SLR, LALR.

Árboles de análisis sintáctico / ASTs

- Árbol de análisis sintáctico (concreto) vs árbol de sintaxis abstracta (AST).
- Representación y uso del árbol para fases posteriores (análisis semántico, generación de código).
- Ejemplos de árbol de expresiones o estructuras de programación. 

Manejo de errores sintácticos

- Importancia de seguir analizando tras el primer error para reportar varios de una vez.

- Técnicas de recuperación: modo pánico, inserción/eliminación de tokens, sincronización.
- Qué ocurre si el programa fuente no cumple la sintaxis.

Implementación del parser

- Herramientas automáticas: generadores de parsers (por ejemplo ANTLR, YACC/Bison)

- Tablas de análisis, estados, elementos de control para parsers LR.