



UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Escuela Técnica Superior de Ingeniería Informática



Procesadores de Lenguajes

Tema 1

Introducción

Javier Vélez Reyes
jvelez@lsi.uned.es

Javier Vélez Reyes jvelez@lsi.uned.es

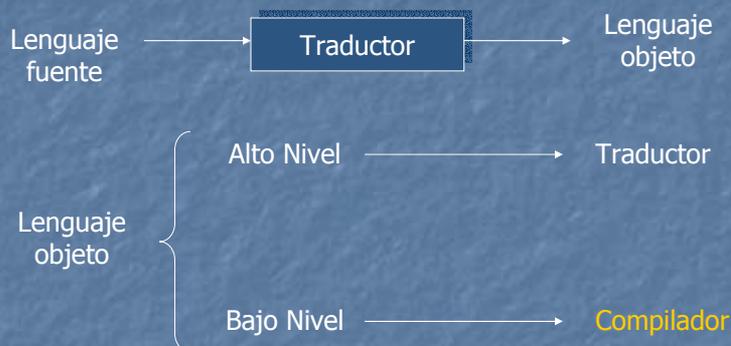
Objetivos del Tema

- Aprender qué es un compilador
- Conocer los tipos de compiladores que existen
- Conocer la diferencia entre compilador e interprete
- Familiarizarse con el contexto de un compilador
- Aprender la estructura y fases de un compilador

Índice General

- ¿Qué es un compilador?
- Compiladores e interpretes
- Contexto de un compilador
- Tipos de compiladores
- Estructura de un compilador

¿Qué es un compilador?



Un compilador es un programa que lee un programa escrito en lenguaje *fuentes*, y lo traduce a un lenguaje *objeto* de bajo nivel. Además generará una lista de los posibles errores que tenga el programa fuente

Índice General

- ¿Qué es un compilador?
- Compiladores e interpretes
- Contexto de un compilador
- Tipos de compiladores
- Estructura de un compilador

Compiladores e interpretes

- Compiladores
 - Una única compilación
 - Mayor velocidad ejecución
 - Mayor detalle de errores
 - Mayor consumo de memoria
- Interpretes
 - Interpretación en ejecución
 - Menor velocidad ejecución
 - Menor detalle de errores
 - Menor consumo de memoria

Índice General

- ¿Qué es un compilador?
- Compiladores e interpretes
- Contexto de un compilador
- Tipos de compiladores
- Estructura de un compilador

Contexto de un compilador

- Contexto
 - Precompilador
 - Compilador
 - Enlazador (montador)
 - Depurador
 - Ensamblador



Índice General

- ¿Qué es un compilador?
- Compiladores e interpretes
- Contexto de un compilador
- Tipos de compiladores
- Estructura de un compilador

Tipos de compiladores

- Tipos de compiladores
 - Ensamblador
 - Compilador cruzado
 - Compilador con montador
 - Autocompilador
 - Metacompilador
 - Descompilador

Índice General

- ¿Qué es un compilador?
- Compiladores e interpretes
- Contexto de un compilador
- Tipos de compiladores
- Estructura de un compilador

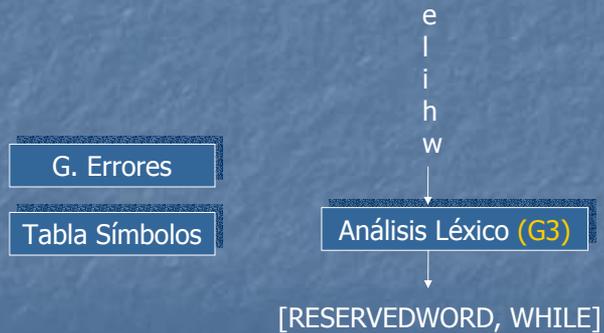
Estructura de un compilador



Análisis léxico I

- Tipos de tokens
 - Específicos
 - Palabras reservadas
 - Separadores
 - Operadores
 - No específicos
 - Identificadores
 - Constantes
 - Etiquetas
- Estructura
 - Tipo
 - Lexema

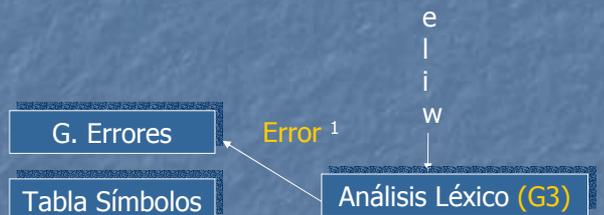
El analizador léxico o scanner, transforma el texto fuente en una secuencia a ordenada de elemento léxicamente válidos (tokens)



Análisis léxico II

- Tipos de tokens
 - Específicos
 - Palabras reservadas
 - Separadores
 - Operadores
 - No específicos
 - Identificadores
 - Constantes
 - Etiquetas
- Estructura
 - Tipo
 - Lexema

El analizador léxico o scanner, transforma el texto fuente en una secuencia a ordenada de elemento léxicamente válidos (tokens)



¹ Los errores léxicos son difíciles de detectar y suelen delegarse en el análisis sintáctico

Análisis léxico III

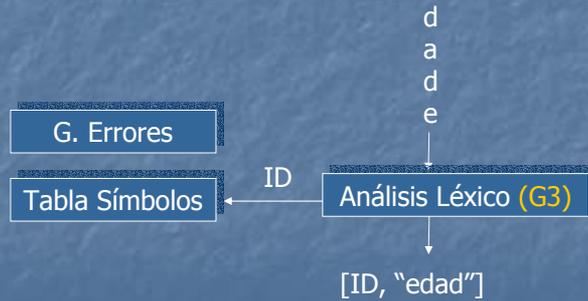
■ Tipos de tokens

- Específicos
 - Palabras reservadas
 - Separadores
 - Operadores
- No específicos
 - Identificadores
 - Constantes
 - Etiquetas

■ Estructura

- Tipo
- Lexema

El analizador léxico o scanner, transforma el texto fuente en una secuencia a ordenada de elemento léxicamente válidos (tokens)



Análisis sintáctico I

■ Definición

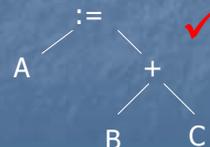
El analizador sintáctico o parser recibe los tokens y comprueba su ordenación correcta. Genera un árbol sintáctico

■ Funciones

- Guiar la traducción
- Gestión de errores
- prelación de operadores
 - $A/B * C = A/(B * C)$
 - $A/B * C = (A/B) * C$

A := B + C

Análisis Sintáctico (G2)



Análisis sintáctico II

Definición

El analizador sintáctico o parser recibe los tokens y comprueba su ordenación correcta. Genera un árbol sintáctico

Funciones

- Guiar la traducción
- Gestión de errores
- prelación de operadores
 - $A/B * C = A/(B * C)$
 - $A/B * C = (A/B) * C$



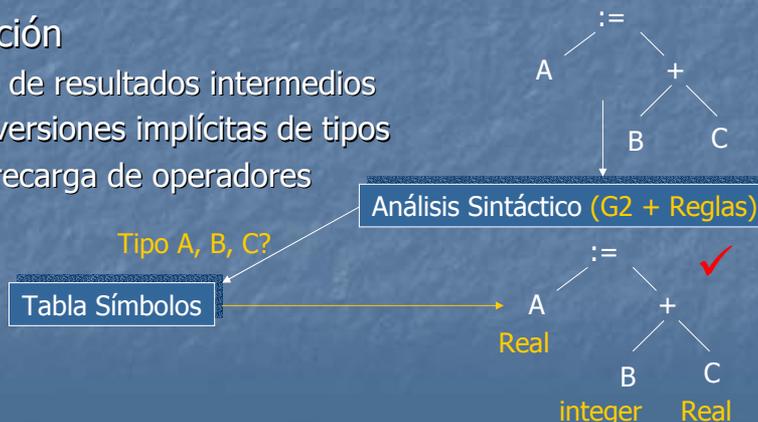
Análisis semántico I

Definición

El analizador semántico comprueba que el árbol sintáctico es semánticamente válido. Genera un árbol semántico o etiquetado

Validación

- Tipo de resultados intermedios
- Conversiones implícitas de tipos
- Sobrecarga de operadores



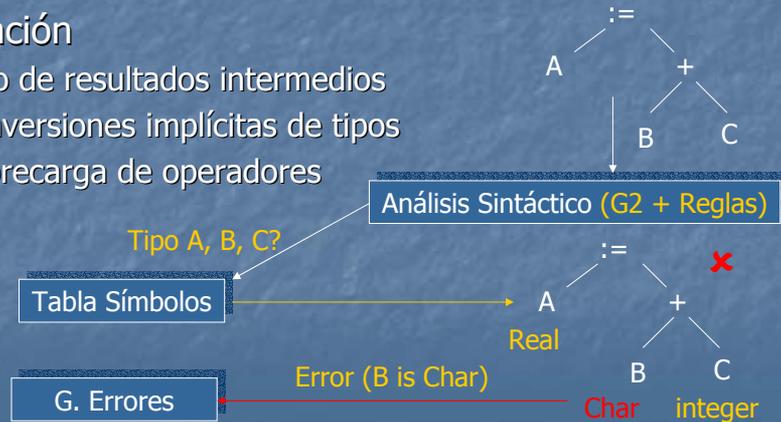
Análisis semántico I

Definición

El analizador semántico comprueba que el árbol sintáctico es semánticamente válido. Genera un árbol semántico o etiquetado

Validación

- Tipo de resultados intermedios
- Conversiones implícitas de tipos
- Sobrecarga de operadores



Generación de código intermedio

Definición

El generador de código intermedio transforma un árbol de semántico en una representación en un lenguaje intermedio cercano al código objeto

Lenguajes sencillos

- Tercetos
- Cuartetos

```
WHILE (A>B) AND (A<2*B-5) DO
  A:=A+B
```

Generación de código intermedio

```
L1:  IF A>B GOTO L2
      GOTO L3
L2:  T1 := 2*B
      T2 := T1 - 5
      IF A< T2 GOTO L4
      GOTO L3
L4:  A := A + B
      GOTO L1
L3:  ...
```

Optimización de código I

- Definición

El optimizador de código realiza modificaciones sobre el código intermedio para mejorar la eficiencia en velocidad y tamaño.

- Fases

- Independiente de la máquina
- Dependiente de la máquina

- Eliminación de saltos consecutivos

```
L1: IF A>B GOTO L2
      GOTO L3
L2: T1 := 2*B
      T2 := T1 - 5
      IF A< T2 GOTO L4
      GOTO L3
L4: A := A + B
      GOTO L1
L3: ...
```

Optimizador

```
L1: IF A<=B GOTO L2
      T1 := 2*B
      T2 := T1 - 5
      IF A>= T2 GOTO L2
      A := A + B
      GOTO L1
L2: ...
```

Optimización de código II

- Definición

El optimizador de código realiza modificaciones sobre el código intermedio para mejorar la eficiencia en velocidad y tamaño.

- Fases

- Independiente de la máquina
- Dependiente de la máquina

- Factorización de expresiones comunes

```
A := B + C + D
E := B + C + F
```

Optimizador

```
T1 := B + C
A := T1 + D
E := T1 + F
```

Optimización de código III

- Definición

El optimizador de código realiza modificaciones sobre el código intermedio para mejorar la eficiencia en velocidad y tamaño.

- Fases

- Independiente de la máquina
- Dependiente de la máquina

- Extracción de invariantes



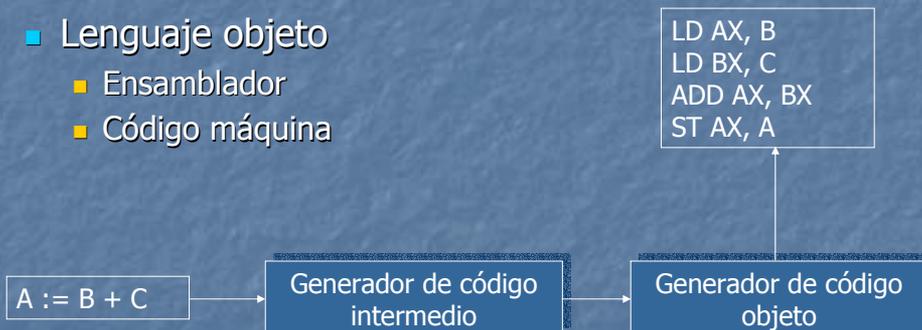
Generación de código objeto

- Definición

El generador de código objeto transforma el código intermedio optimizado en código objeto de bajo nivel

- Lenguaje objeto

- Ensamblador
- Código máquina



La tabla de símbolos

- Almacena estructuras de datos
 - Variables
 - Constantes
 - Etiquetas
 - Tipos
 - Valores
 - Signatura de funciones
- Operaciones
 - Insertar símbolo
 - Consultar símbolo
 - Borrar símbolo

Gestión de errores

- Detección de errores
 - Léxicos (se delegan al sintáctico)
 - Sintácticos
 - Semánticos
- Recuperación de errores
 - Parar al primer error
 - Recuperar volviendo a un contexto fiable

Bibliografía

- [AJO] AHO, SETHI, ULLMAN: *Compiladores: Principios, técnicas y herramientas*; Addison-Wesley Iberoamericana, 1990



- [GARRIDO] A. Garrido, J. Iñesta, F. Moreno y J. Pérez. 2002. *Diseño de compiladores*. Universidad de Alicante.

