



Tema 5

Traducción dirigida por la sintaxis

Javier Vélez Reyes
jvelez@lsi.uned.es

Javier Vélez Reyes jvelez@lsi.uned.es

Objetivos del Tema

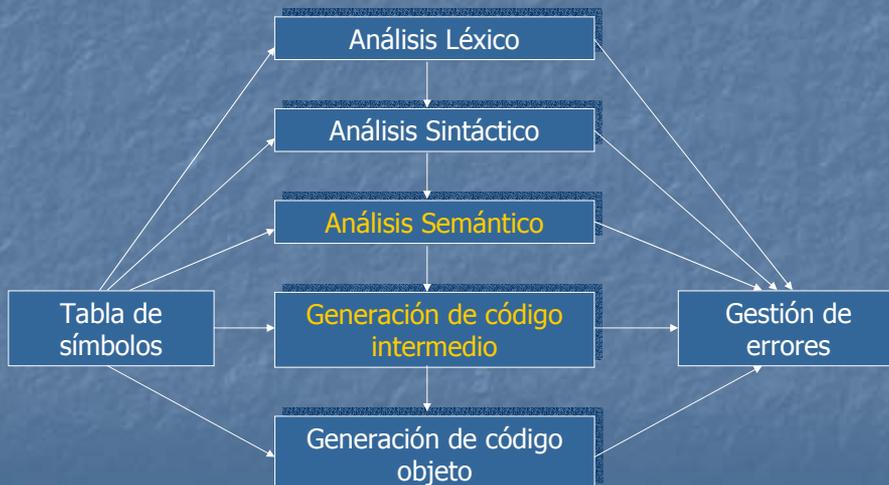
- Entender la traducción dirigida por la sintaxis
- Aprender a especificar gramáticas con atributos
- Entender el concepto de reglas semánticas
- Entender los problemas de dependencias
- Aprender a distinguir y reconocer
 - Compiladores de una sola pasada
 - Compiladores de doble pasada
- Entender y aprender a especificar
 - Definiciones dirigidas por la sintaxis
 - Esquemas de traducción

Índice General

- Introducción
 - Fases de análisis semántico y traducción
 - Traducción dirigida por la sintaxis
- Gramáticas con atributos
 - Atributos
 - Reglas semánticas
 - Tipos de atributos
 - Grafos de dependencias
- Especificación de un traductor
 - Definiciones dirigidas por la sintaxis
 - Esquemas de traducción

Introducción

- Estructura de un compilador



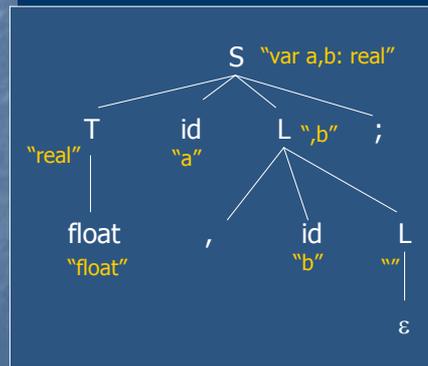
Introducción

- Principio de Traducción dirigida por la sintaxis

El significado de una frase está directamente relacionado con su estructura sintáctica según se representa en su árbol de análisis sintáctico.

- Se realizan en paralelo

- Análisis semántico
- Traducción

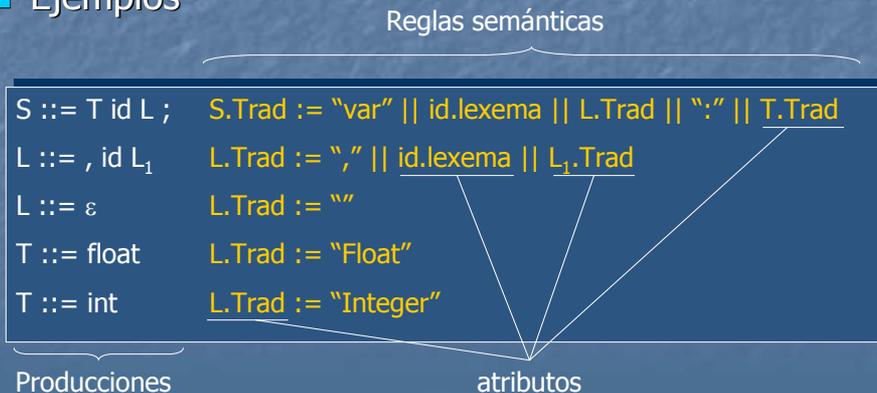


Gramáticas con Atributos

- Gramática con atributos

- Cada símbolo de la gramática tiene atributos
- Cada regla de producción tiene reglas semánticas

- Ejemplos



Atributos

- Atributos
 - Propiedades de los símbolos del lenguaje que almacenan contenidos relacionados con el significado de los símbolos a los que pertenecen.
- Se utiliza la notación **símbolo.atributo**
 - id.lexema
 - Var.valor
 - Var.dirección
 - Exp.código
- Se utilizan subíndices para diferenciar símbolos

```
L ::= , id L1    L.Trad := "," || id.lexema || L1.Trad
```

Reglas semánticas

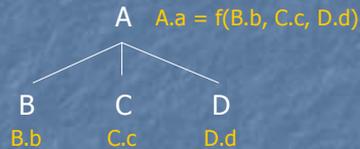
- Reglas o acciones semánticas
 - Define sólo el valor de los atributos de los símbolos que aparecen en la producción
- Notación
 - := operador de asignación
 - || operador de concatenación de cadenas
- Ejemplo

```
S ::= T id L ;    S.Trad := "var" || id.lexema || L.Trad || ":" || T.Trad
```

Tipos de Atributos

■ Atributos sintetizados

- Se calculan a partir de atributos de nodos hijos
- Se refieren a atributos del antecedente de la regla
- La información asciende por el árbol



- Los atributos sintetizados lo son durante toda la gramática
- Los atributos de terminales son sintetizados

Tipos de Atributos

■ Atributos heredados

- Calculados con atributos en nodos padre y hermanos
- Se refieren a atributos del consecuente de la regla
- Información descendente o de tránsito horizontal



- Los atributos heredados lo son durante toda la gramática
- Se utilizan para transferir información entre reglas

Tipos de Atributos

- Ejemplo 6.4 (página 104, texto base)

$E ::= E_1 + T$	$E.trad := "suma(" \parallel E_1.Trad \parallel ", " \parallel T.trad \parallel ")"$
$E ::= T$	$E.trad := T.trad$
$T ::= T_1 * F$	$T.trad := "por(" \parallel T_1.trad \parallel ", " \parallel F.trad \parallel ")"$
$T ::= F$	$T.trad := F.trad$
$F ::= num$	$F.trad := num.valor$

- Hacer el árbol de análisis con atributos de $2+3*5$

Tipos de Atributos

- Ejemplo 6.5 (página 105, texto base)

$D ::= T L ;$	$L.tipo := T.tipo$
$T ::= float$	$T.tipo = real$
$T ::= int$	$T.tipo = entero$
$L ::= L_1 , id$	$L_1.tipo := L.tipo; AsignaTipo(id.lexema, L.tipo)$
$L ::= id$	$AsignaTipo(id.lexema, L.tipo)$

- ¿Qué atributos son heredados?
- ¿Qué atributos son sintetizados?
- Hacer el árbol de análisis con atributos de float a,b;

Grafos de dependencias

- Motivaciones
 - ¿Cuándo debe ejecutarse una acción semántica?
 - Los atributos no pueden calcularse en cualquier orden
 - Todos los atributos de los que depende deben tener valor
- Grafo de dependencias
 - Determina el flujo de información
 - Determina un orden de asignación de valores
 - Induce cierto orden de ejecución de las reglas

Grafos de dependencias

■ Construcción

Para cada producción $A ::= X_1 X_2 \dots X_n$

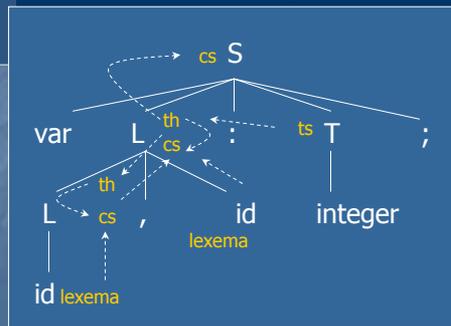
1. Se crea un nodo por cada atributo $X_i.a_j$ de cada símbolo
2. Para cada regla semántica de la forma $X_i.a_j = f(\dots, X_k.a_l, \dots)$
 - a. Para cada k y cada l
 - crear un arco desde el nodo $X_k.a_l$ hasta el nodo $X_i.a_j$

Unir todos los grafos

Grafos de dependencias

■ Ejemplo 6.6 (Pagina 107, texto base)

$S ::= \text{var } L : T ;$	$L.th := T.ts$
	$S.cs := L.cs$
$L ::= L_1 , id$	$L.cs := L_1.cs \parallel L.th \parallel id.lexema \parallel ", "$
	$L_1.th := L.th$
$L ::= id$	$L.cs := L.th \parallel id.lexema \parallel ", "$
$T ::= \text{real}$	$T.ts := \text{float}$
$T ::= \text{integer}$	$T.ts := \text{int}$



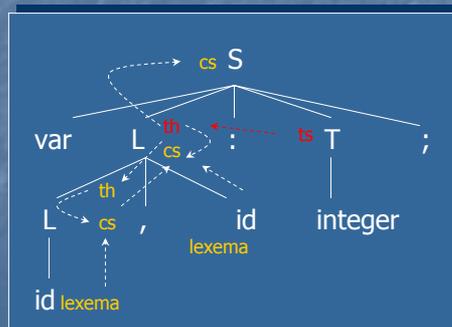
Grafos de dependencias

■ Traductores

- 1 pasada, traducen a la vez que analizan ☺
- Varias pasadas, requieren varias pasadas ☹

■ Para construir traductores de una sola pasada...

- Los atributos no pueden depender de otros calculados a su derecha



Especificación de un traductor

- 2 notaciones
 - Definiciones dirigidas por la sintaxis (DDS)
 - Formalismo de alto nivel para describir traducciones
 - Se ocultan los detalles de implementación
 - No se impone orden de ejecución de las **reglas**
 - Calcular los atributos en la producción en que aparecen
 - Esquemas de traducción (ETDS)
 - Notación de bajo nivel para especificar un traductor
 - El traductor es de una sola pasada
 - Explícita el orden de la ejecución de las **acciones**

Definiciones dirigidas por la sintaxis

- Gramática con atributos
 - Toda gramática con atributos es una DDS
 - Las reglas semánticas sólo manipulan atributos
 - No existen otras acciones semánticas
 - No existen efectos laterales
- DDS
 - Son una generalización de las gramáticas con atributos
 - Las reglas semánticas manipulan atributos
 - Existen otras acciones semánticas
 - Escritura de valores
 - Condicionales
 - Modificación de variables globales
 - Existen efectos laterales

Definiciones dirigidas por la sintaxis

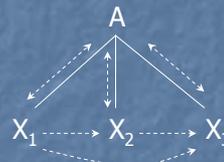
■ Ejemplo DDS

```
E ::= E1 OP T      if (OP.lexema == '+') {  
                    E.valor := E1.valor + T.valor  
                    }  
                    else {  
                    E.valor := E1.valor + T.valor  
                    }  
                    write(E.valor)
```

Definiciones dirigidas por la sintaxis

■ Gramáticas con atributos por la izquierda (GAI)

- Subconjunto de gramáticas con atributos
- Generan traductores de una sola pasada
- La información manipulada está disponible en la ejecución
- La información siempre fluye
 - De arriba abajo
 - De abajo a arriba
 - De izquierda a derecha
 - Nunca de derecha a izquierda



Una DDS es una GAI se es una si es una gramática de atributos en la que para cada atributo heredado $X_i.h$ de una producción $A ::= X_1 \dots X_n$ $X_i.h$ depende sólo de:

1. Los atributos de $X_1 \dots X_{i-1}$
2. Los atributos heredados de A

Definiciones dirigidas por la sintaxis

■ Ejemplo DDS

```
S ::= var L : T ;      L.th := T.ts
                       S.cs := L.cs
L ::= L1 , id        L.cs := L1.cs || L.th || id.lexema || ";"
                       L1.th := L.th
L ::= id              L.cs := L.th || id.lexema || ";"
T ::= real           T.ts := float
T ::= integer       T.ts := int
```

■ Ejemplo GAI

```
S ::= var D ;      S.cs := D.cs
D ::= id , D1    D.cs := D.ts || id.lexema || ";"
                       D.ts := D1.ts
D ::= id : T      D.cs := T.ts || id.lexema || ";"
                       D.ts := T.ts
T ::= integer     T.ts := int
T ::= real       T.ts := float
```

Esquemas de traducción

■ Esquemas de traducción

- Traductores de una sola pasada
- Acciones semánticas entre los símbolos de la derecha
 - Se encierran entre llaves {acción semántica}
 - Son fragmentos de código en un lenguaje de programación
 - Se explicita el orden de ejecución de las acciones

■ Ejemplo 6.9 (Página 111, texto base)

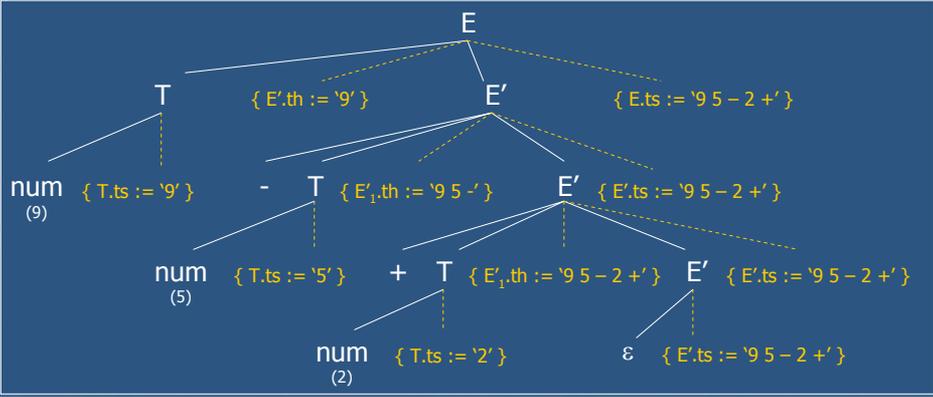
```
E ::= T {E'.th := T.ts} E' {E.ts := E'.ts}
E' ::= op T {E'.th := E'.th || T.ts || op.lexema} E'1 {E'.ts := E'1.ts}
E' ::= ε {E'.ts := E'.th}
T ::= num {T.ts := num.lexema}
```

Esquemas de traducción

■ Ejemplo 6.9 (Página 111, texto base)

$E ::= T \{E'.th := T.ts\} E' \{E.ts := E'.ts\}$
 $E' ::= op T \{E'.th := E'.th \mid T.ts \mid op.lexema\} E'_1 \{E'.ts := E'_1.ts\}$
 $E' ::= \varepsilon \{E'.ts := E'.th\}$
 $T ::= num \{T.ts := num.lexema\}$

9-5+2



Esquemas de traducción

■ Restricciones de diseño

- Si existen atributos sintetizados colocar la acción semántica después de todos los símbolos implicados o al final de la producción

$A := B C \{ A.a := f(B.a, C.a) \} D$

$A := B C D \{ A.a := f(B.a, C.a) \}$

- Si existen atributos heredados deben calcularse antes de que aparezca el símbolo en la parte derecha

$A := B C \{ D.h := f(B.a, C.a) \} D$

Esquemas de traducción

■ Restricciones de diseño

- Un atributo sintetizado no debe usarse antes de que aparezca el símbolo en la parte derecha

Error ! $A := B \{ A.s := f(B.s, C.s) \} C D$

Correcto ! $A := B C \{ A.s := f(B.s, C.s) \} D$

- Si existen acciones con efectos laterales deben situarse en el punto exacto de la parte derecha de la regla en la que deberían evaluarse. Hay que verificar que no utilizan atributos de símbolos situados a la derecha de dicho punto.

Esquemas de traducción

■ Pasos de diseño de un ETDS

- Decidir qué atributos son necesarios
- Añadir a la gramática las acciones semánticas necesarias
- Estudiar de qué tipo es cada atributo
 - Sintetizado
 - Heredado
- Comprobar que se cumplen las restricciones de diseño
 - Estudiar grafo de dependencias
 - Modificar la gramática
 - Reubicar las acciones semánticas

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.

