

# Tema 6. Gramáticas de atributos

---

1. Gramáticas de atributos.
2. Orden de evaluación.
3. Definiciones S-atribuidas.
4. Definiciones L-atribuidas.



# **1. Introducción**

# Atributos y reglas semánticas

Especificación léxica --> Expresiones regulares

Especificación sintáctica --> Gramáticas independientes  
del contexto

**Especificación semántica --> Gramáticas de atributos**

- Asociamos atributos a los símbolos de la gramática.
- Los valores de los atributos se calculan mediante reglas semánticas asociadas a la reglas de producción.

# Definición y Ejemplo 1

Sea  $G=(N,T,P,S)$  una GIC. Se define:

Conjunto de **atributos** de  $X \in (N \cup \Sigma)$ :  $A(X) = \{ a_1, a_2, \dots, a_n \}$   
y se representan por  $X.a_1, X.a_2, \dots, X.a_n$

Conjunto de **acciones semánticas** de la regla  $r : X_0 \rightarrow X_1 X_2 \dots X_n \in P$ :

$$R(r) = \{ (X_i.a := f(X_r.b, \dots, X_s.c) \mid a \in A(X_i), b \in A(X_r), \dots, c \in A(X_s) \text{ con } 0 \leq i, r, \dots, s \leq n \}$$

|                         |   |
|-------------------------|---|
| $E' \rightarrow E \$$   | $\{ \text{printf}("%d", E.\text{valor}) \}$                 |
| $E \rightarrow E_1 + T$ | $\{ E.\text{valor} := E_1.\text{valor} + T.\text{valor} \}$ |
| $\quad \mid T$          | $\{ E.\text{valor} := T.\text{valor} \}$                    |
| $T \rightarrow T_1 * F$ | $\{ T.\text{valor} := T_1.\text{valor} * F.\text{valor} \}$ |
| $\quad \mid F$          | $\{ T.\text{valor} := F.\text{valor} \}$                    |
| $F \rightarrow ( E )$   | $\{ F.\text{valor} := E.\text{valor} \}$                    |
| $\quad \mid \text{num}$ | $\{ F.\text{valor} := \text{num.lexval} \}$                 |

# Definiciones

**Árbol sintáctico anotado:** Árbol sintáctico de una frase en el que en cada nodo aparecen los valores de los atributos de ese nodo.

## Tipos de Atributos

$X_i.a$  será un **atributo sintetizado** sii todas las acciones semánticas que lo definen son de la forma: **los atributos dependen de los hijos**

$r : X_0 \rightarrow X_1 X_2 \dots X_n \quad \{ (X_0.a := f(X_{i1}.bj_1, X_{i2}.bj_2, \dots, X_{ip}.bj_p)) \}$

$X_i.a$  será un **atributo heredado** sii todas las acciones semánticas que lo definen son de la forma:

$r : X_0 \rightarrow X_1 X_2 \dots X_n \quad \{ (X_{i1}.a := f(X_0.bj_0, X_{i1}.bj_1, \dots, X_{ip}.bj_p)) \}$

con  $i_1 \neq 0$

**los atributos dependen de los hermanos o del padre**

# Atributos sintetizados y heredados

|                                |   |
|--------------------------------|---|
| $D \rightarrow T L$            | $\{ L.tipo := T.tipo \}$  |
| $T \rightarrow \text{int}$     | $\{ T.tipo := \text{tentero} \}$                                      |
| $T \rightarrow \text{float}$   | $\{ T.tipo := \text{treal} \}$  |
| $L \rightarrow L_1, \text{id}$ | $\{ L_1.tipo := L.tipo ; \text{InsertaTDS}(\text{id.nom}, L.tipo) \}$ |
| $L \rightarrow \text{id}$      | $\{ \text{InsertaTDS}(\text{id.nom}, L.tipo) \}$                      |

Cadena  $w = \text{int id, id}$

## Ejemplo 2: Contar a's y b's

$S \rightarrow (A)$       $\{ \text{printf}(\text{"Num.a: \%d Num.b: \%d"}, A.as, A.bs) \}$

$A \rightarrow A_1, D$       $\{ A.as = A_1.as + D.as ; A.bs = A_1.bs + D.bs \}$

$A \rightarrow D$       $\{ A.as = D.as ; A.bs = D.bs \}$

$D \rightarrow a$       $\{ D.as = 1 ; D.bs = 0 \}$

$D \rightarrow b$       $\{ D.as = 0 ; D.bs = 1 \}$

$D \rightarrow (A)$       $\{ D.as = A.as ; D.bs = A.bs \}$

# Ejemplo 3: Convertir de binario a decimal

$S \rightarrow L \quad \{ \text{printf}(L.v) \}$   
 $L \rightarrow L_1 B \quad \{ L.v := L_1.v * 2 + B.v ; \}$   
 $L \rightarrow B \quad \{ L.v := B.v \}$   
 $B \rightarrow 0 \quad \{ B.v := 0 \}$   
 $B \rightarrow 1 \quad \{ B.v := 1 \}$



# Ejemplo 4: Convertir a base 10

$S \rightarrow h L$

$S \rightarrow b L$

$S \rightarrow o L$

$L \rightarrow L_1 B$

$L \rightarrow B$

$B \rightarrow 0$

$B \rightarrow 1$

$B \rightarrow 2$

.....

$B \rightarrow f$

# Ejemplo 4: Convertir a base 10

$S \rightarrow h L$

$S \rightarrow b L$

$S \rightarrow o L$

$L \rightarrow L_1 B$

$L \rightarrow B$

$B \rightarrow 0$

$B \rightarrow 1$

$B \rightarrow 2$

.....

$B \rightarrow f$

# Ejemplo 4: Convertir a base 10

*Ejemplo con comprobación de números erróneos*

$S \rightarrow h L$

$S \rightarrow b L$

$S \rightarrow o L$

$L \rightarrow L_1 B$

$L \rightarrow B$

$B \rightarrow o$

$B \rightarrow 1$

$B \rightarrow 2$

.....

$B \rightarrow f$



## **2. Orden de evaluación**

# Orden de evaluación

Si un atributo **c** de un nodo del árbol anotado depende de un atributo **b**, la regla semántica que define al nodo **c** debe ser evaluada después de la regla semántica que define a **b**.

Métodos para la evaluación semántica:

- Mediante grafo de dependencias
- Dirigidos por la sintaxis

# Orden de evaluación

**Grafo de dependencias:** Grafo dirigido acíclico con:

- *Un nodo para cada atributo*
- *Un arco  $b \rightarrow c$  si el atributo  $c$  depende del atributo  $b$*

- Cualquier orden topológico del grafo de dependencias proporciona un orden de evaluación de las reglas semánticas válido.

**Métodos dirigidos por la sintaxis:**

*Se restringe la clase de gramáticas atribuidas que puede ser utilizada. Gramáticas S y L atribuidas*

# Definiciones

Notaciones para asociar reglas semánticas con producciones:

- *Gramática de atributos:*

*Gramática más un conjunto de atributos y de reglas semánticas asociadas a cada regla.*

- *Esquema de traducción dirigida por la sintaxis (ETDS):*

*Gramática más un conjunto de atributos y de reglas semánticas asociadas a cada regla. Estas reglas pueden aparecer entre los símbolos de la parte derecha de las producciones.*

### 3. Definiciones S-atribuidas



# Gramática S-atribuida

*Definición atribuida que solo usa atributos sintetizados.*

Orden de evaluación:

*Se puede anotar el árbol sintáctico de forma ascendente: desde las hojas hasta la raíz.*

- Las definiciones S-atribuidas pueden transformarse en esquemas de traducción sin más que poner todas las acciones semánticas al final de las reglas de producción.

# Ejemplo: S-atribuida y análisis LR

Gramática de atributos que pase a **base 10** un número **binario**

- (1)  $S \rightarrow L$        $\{ \text{printf}(L.v) \}$
- (2)  $L \rightarrow L1 B$      $\{ L.v := L1.v * 2 + B.v \}$
- (3)  $L \rightarrow B$        $\{ L.v := B.v \}$
- (4)  $B \rightarrow 0$        $\{ B.v := 0 \}$
- (5)  $B \rightarrow 1$        $\{ B.v := 1 \}$

|   | 0   | 1   | \$   | S | L | B |
|---|-----|-----|------|---|---|---|
| 0 | d-4 | d-5 |      | 1 | 2 | 3 |
| 1 |     |     | Acep |   |   |   |
| 2 | d-4 | d-5 | r-1  |   |   | 6 |
| 3 | r-3 | r-3 | r-3  |   |   |   |
| 4 | r-4 | r-4 | r-4  |   |   |   |
| 5 | r-5 | r-5 | r-5  |   |   |   |
| 6 | r-2 | r-2 | r-2  |   |   |   |

# Ejemplo 5: Calculadora

$S \rightarrow (\text{print } E)$

$E \rightarrow (\text{Op } E) \mid (\text{Op } E E) \mid \text{num}$

$\text{Op} \rightarrow + \mid - \mid * \mid /$

ETDS que calcule el valor numérico de una expresiones y la imprima.

*Ejemplo:*  $(\text{print } (/ 20 (* 2 (- 5)))) \rightarrow \text{Imprimirá } -2$

## 4. Definiciones L-atribuidas

# Definición L-atribuida

Definición atribuida en la que cada atributo heredado de  $X_j$  ( $1 \leq j \leq n$ ), en la producción  $A \rightarrow X_1 X_2 \dots X_n$  depende solo de:

- a) *Los atributos de los símbolos  $X_1, X_2, \dots, X_{j-1}$  a la izquierda de  $X_j$*
- b) *Los atributos heredados de  $A$*

|                                |   |
|--------------------------------|---|
| $D \rightarrow T L$            | $\{ L.tipo := T.tipo \}$  |
| $T \rightarrow \text{int}$     | $\{ T.tipo := \text{tentero} \}$                                      |
| $T \rightarrow \text{float}$   | $\{ T.tipo := \text{treal} \}$  |
| $L \rightarrow L_1, \text{id}$ | $\{ L_1.tipo := L.tipo ; \text{InsertaTDS}(\text{id.nom}, L.tipo) \}$ |
| $L \rightarrow \text{id}$      | $\{ \text{InsertaTDS}(\text{id.nom}, L.tipo) \}$                      |

# Orden de evaluación ETDS

```
Procedure Primero_en_Profundidad (N: nodo):  
  begin  $\{N \rightarrow X_1, X_2, \dots, X_p\}$   
    For i := 1 to p do  
      begin  
        evaluar atributos heredados de  $X_i$ ;  
        Primero_en_Profundidad ( $X_i$ ) ;  
      end ;  
      evaluar atributos sintetizados de N;  
    end
```

# Esquema de Traducción Dirigido por Sintaxis (ETDS)

- ⇒ En un ETDS las acciones semánticas no pueden usar atributos de símbolos que aparezcan a su derecha:
- Se debe usar el algoritmo **Primero\_en\_Profundidad** para determinar el orden en el que deben aparecer las reglas entre los símbolos de la producción.

*Ejemplo:*

```
D → T    { L.tipo := T.tipo } L
T → int  { T.tipo := entero }
T → float { T.tipo := real }
L →      { L1.tipo := L.tipo } L1, id {InsertaTDS (id.nom, L.tipo) }
L → id   { InsertaTDS(id.nom, L.tipo) }
```

# Propiedades de los ETDS

(derivadas del algoritmo Primero\_en\_Profundidad)

- Un atributo heredado para un símbolo del lado derecho de una producción debe calcularse en una acción antes de dicho símbolo.
- Una acción no puede usar un atributo sintetizado de un símbolo a la derecha de dicha acción.
- Una acción para calcular un atributo sintetizado del no-terminal del lado izquierdo solo puede ejecutarse después de las reglas que calculan todos los atributos que usa.



# Ejemplo 6: Calculadora detecta errores

$S \rightarrow (\text{ print } E )$

$E \rightarrow ( \text{ Op } E ) \mid ( \text{ Op } E E ) \mid \text{ num}$

$\text{Op} \rightarrow + \mid - \mid * \mid /$

*Detectar divisiones por cero e informar del **número del operador** en el que se ha producido.*

Ej.:  $(\text{print } (+ (* 2 10) (/ 5 (- 3 3)))) \rightarrow \text{Error: Div por cero en op 3}$

# Ejemplo 7

$S \rightarrow A$

$A \rightarrow (\text{num } A A) \mid (\text{num})$

- ETDS que obtenga el mínimo nivel de profundidad del número de mayor valor.
- *Nivel de profundidad* de un nodo es el número mínimo de ramas de la raíz al nodo.

Ej. (2 (5) (8)) Nivel de prof. del mayor número = 2

## Ejemplo 8

La siguiente gramática reconoce líneas formadas por palabras. Las palabras pueden ser correctas o incorrectas. Escribe un ETDS que cada vez que se reconozca una palabra **incorrecta** muestre el **número de línea** en la que se encuentra.

$D \rightarrow L \text{ newline } RD$

$RD \rightarrow L \text{ newline } RD$

$\mid \varepsilon$

$L \rightarrow P \text{ RL}$

$RL \rightarrow P \text{ RL}$

$\mid \varepsilon$

$P \rightarrow \text{correcta}$

$\mid \text{incorrecta}$

# Ejemplo 9

**S -> find2 ( LE )**

**LE -> LE , E | E**

**E -> cte | ctr | op**

Construye un ETDS que detecte la primera aparición de dos "cte" seguidas en la lista y devuelva en un atributo de S la posición que ocupa. Si no hay dos apariciones seguidas de "cte" devolverá o.

Ej.

find2 ( cte op cte cte op cte cte )

Devolverá 3