

Atributo: Propiedad que se asocia a los símbolos no terminales.

- Notación: `simbolo . nombre_atributo`
- Tipados

Gramática de Atributos

Gramática Libre de Contexto **extendida**, con

- Atributos en los no terminales.
- Reglas semánticas asociadas a las producciones.

Funciones:

- `IF(expr, val1, val2)`
Evalúa `expr`, devuelve `val1` si es verdadero, y `val2` si es falso.
- `CONDITION(expr)`
Rechaza la cadena si `expr` es falso.

Las gramáticas de atributos son **más expresivas** que las GLC.

- **Sintetizados:** Su valor se calcula en base al valor de los atributos en los nodos hijos.

Ejemplo: $A \rightarrow BC$ $\{A.a = f_1(B.b, C.c)\}$

- **Heredados:** Su valor se calcula en base al valor de los atributos en los nodos hermanos, o padre.

Ejemplo: $A \rightarrow BC$ $\{B.val = f_2(A.a, C.c)\}$

Tipos de Atributos

- Todos los atributos son, o sintetizados, o heredados.
- Ejercicios: siempre aclarar qué tipo de atributo es.
- **Ningún atributo puede ser sintetizado en una producción, y heredado en otra.**

Grafo de Dependencias

- Grafo dirigido, que muestra las dependencias entre los atributos para un determinado árbol de derivación.
- Para que los atributos puedan calcularse el grafo no debe tener ciclos.
- Implementación de una Gramática de Atributos:
 - 1 Parsear la cadena y generar el árbol de derivación.
 - 2 Calcular el grafo de dependencias.
 - 3 Elegir un **ordenamiento** de ese grafo.
 - 4 Recorrer el árbol en orden y calcular los valores.

Gramáticas de Atributos eficientes:

- **S-atribuidas:** Todos sus atributos son sintetizados.
Se implementa con parser LR
- **L-atribuidas:** Atributos sintetizados y heredados, pero si son heredados, deben depender solamente de:
 - Los atributos que están a su izquierda, en la producción, ó
 - Los atributos heredados del padre.

Se implementa con parser recursivo descendente

Ejercicio 1

Sea la siguiente gramática:

$$S \rightarrow aS \mid bS \mid \lambda$$

Computar atributo que cuente el número mayor de a -es consecutivas.

Ejercicio 1 - Solución

Atributos:

S.cantA	int	sintetizado
S.mayor	int	sintetizado

Gramática:

$$\begin{aligned} S &\rightarrow \lambda && \{S.cantA = 0, S.mayor = 0\} \\ S_1 &\rightarrow aS_2 && \{IF(S_2.cantA + 1 > S_2.mayor, S_1.mayor = S_2.cantA + 1, \\ &&& S_1.mayor = S_2.mayor), \\ &&& S_1.cantA = S_2.cantA + 1\} \\ S_1 &\rightarrow bS_2 && \{S_1.cantA = 0, S_1.mayor = S_2.mayor\} \end{aligned}$$

Ejercicio 2

Sea la siguiente gramática:

$$S \rightarrow E (num)$$

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow num \mid (E) \mid x$$

Sintetizar el valor de la expresión, reemplazando x por el número indicado entre paréntesis.

Atributos:

T.x	int	heredado
E.x	int	heredado
F.x	int	heredado
T.val	int	sintetizado
E.val	int	sintetizado
F.val	int	sintetizado
S.val	int	sintetizado

Gramática:

$S \rightarrow E (num)$	$\{E.x = num.val, S.val = E.val\}$
$E_1 \rightarrow E_2 + T$	$\{E_2.x = E_1.x, T.x = E_1.x, E_1.val = E_2.val + T.val\}$
$E \rightarrow T$	$\{T.x = E.x, E.val = T.val\}$
$T_1 \rightarrow T_2 * F$	$\{T_2.x = T_1.x, F.x = T_1.x, T_1.val = T_2 * F\}$
$T \rightarrow F$	$\{F.x = T.x, T.val = F.val\}$
$F \rightarrow num$	$\{F.val = num.val\}$
$F \rightarrow (E)$	$\{E.x = F.x, F.val = E.val\}$
$F \rightarrow x$	$\{F.val = F.x\}$

$$\begin{aligned}D &\rightarrow T \text{ id}[num] = \{LV\} \\T &\rightarrow int \mid float \\LV &\rightarrow num \mid LV, num\end{aligned}$$

Validar:

- 1 Que la dimensión sea de tipo `int`.
- 2 Que la dimensión sea la declarada.
- 3 Que todos los elementos sean del tipo declarado.

Atributos:

LV.cant	int	sintetizado
LV.tipo	string	heredado
T.tipo	string	sintetizado

Gramática:

$$\begin{aligned} D \rightarrow T \text{ id}[num] = \{LV\} & \quad \{ \text{CONDITION}(num.tipo = "int"), \\ & \quad \text{CONDITION}(num.val = LV.cant), \\ & \quad LV.tipo = T.tipo \} \\ T \rightarrow int & \quad \{ T.tipo = "int" \} \\ T \rightarrow float & \quad \{ T.tipo = "float" \} \\ LV \rightarrow num & \quad \{ LV.cant = 1, \\ & \quad \text{CONDITION}(LV.tipo = num.tipo) \} \\ LV_1 \rightarrow LV_2, num & \quad \{ LV_1.cant = LV_2.cant + 1, \\ & \quad \text{CONDITION}(LV_1.tipo = num.tipo), \\ & \quad LV_2.tipo = LV_1.tipo \} \end{aligned}$$

Atributos:

LV.cant	int	sintetizado
LV.tipo	string	sintetizado
T.tipo	string	sintetizado

Gramática:

$$\begin{aligned} D \rightarrow T \text{ id}[num] = \{LV\} & \quad \{ \text{CONDITION}(num.tipo = "int"), \\ & \quad \text{CONDITION}(num.val = LV.cant), \\ & \quad \text{CONDITION}(LV.tipo = T.tipo) \} \\ T \rightarrow int & \quad \{ T.tipo = "int" \} \\ T \rightarrow float & \quad \{ T.tipo = "float" \} \\ LV \rightarrow num & \quad \{ LV.cant = 1, LV.tipo = num.tipo \} \\ LV_1 \rightarrow LV_2, num & \quad \{ LV_1.cant = LV_2.cant + 1, \\ & \quad \text{CONDITION}(LV_2.tipo = num.tipo), \\ & \quad LV_1.tipo = LV_2.tipo \} \end{aligned}$$