

Análisis semántico

Estrella Pulido Cañabate

¿Qué es el análisis semántico?

- ◆ Fase del compilador en la que se **comprueba la corrección semántica** del programa
- ◆ Conjunto de **subrutinas independientes** que pueden ser invocadas por los analizadores morfológico y sintáctico

Funcionamiento

- ◆ Recibe como **entrada el árbol de derivación** del programa
- ◆ **Añade** al árbol de derivación una serie de **anotaciones**, que permiten determinar la corrección semántica del programa y preparar la generación de código
- ◆ La **salida** que genera es un **árbol de derivación con anotaciones semánticas**

Comprobaciones semánticas

1. Cuando se utiliza un identificador, éste ha sido declarado previamente.
2. Se ha asignado valor a las variables antes de su uso.
3. Los índices para acceder a los *arrays* están dentro del rango válido.
4. En las expresiones aritméticas, los operandos respetan las reglas sobre los tipos de datos permitidos por los operadores.

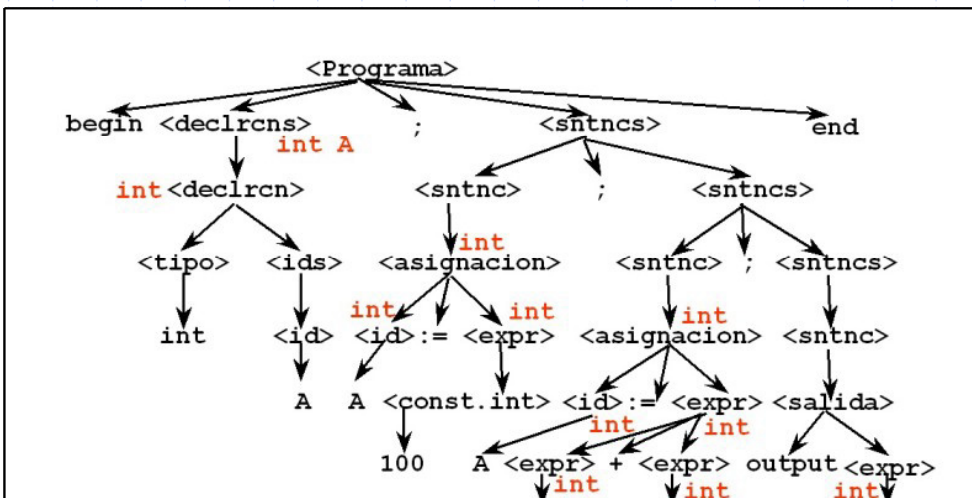
Comprobaciones semánticas (II)

1. Cuando se invoca un procedimiento, éste ha sido declarado adecuadamente. Además, el número, tipo y posición de cada uno de sus argumentos debe ser compatible con la declaración.
2. Las funciones contienen al menos una instrucción en la que se devuelve su valor al programa que las invocó.

begin

```
int A;  
A := 100;  
A := A + A;  
output A;
```

end



Gramáticas de atributos

- ◆ Se añaden **atributos semánticos** a cada símbolo de la gramática (ejemplos: tipo, valor, ...)
 - Para referirse a un atributo se utiliza la notación `símbolo.atributo`
- ◆ Se añaden **acciones semánticas** a cada regla
 - Expresión que calcula el valor de alguno de los atributos de los símbolos de la regla

Ejemplo 1

$\Sigma_T = \{+, *, (,), c(\text{valor}, \text{tipo}), i(\text{valor}, \text{tipo})\}$

$\Sigma_N = \{E(\text{valor}, \text{tipo})\}$

$P = \{$

$E ::= E_i + E_d \{E.\text{valor} = E_i.\text{valor} + E_d.\text{valor};$
 $E.\text{tipo} = E_i.\text{tipo};\}$

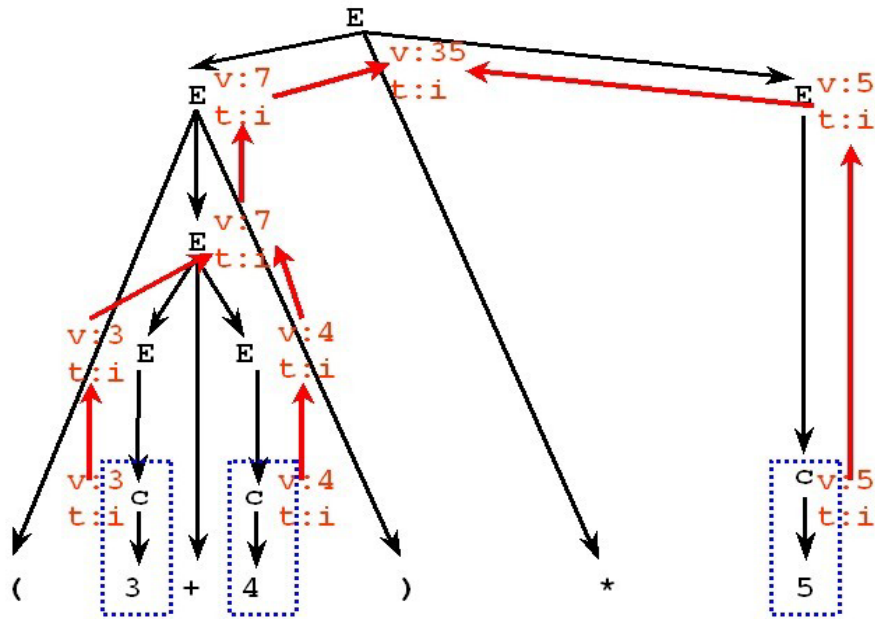
$E ::= -E_d \quad \{E.\text{valor} = - E_d.\text{valor};$
 $E.\text{tipo} = E_d.\text{tipo};\}$

$E ::= E_i * E_d \{E.\text{valor} = E_i.\text{valor} * E_d.\text{valor};$
 $E.\text{tipo} = E_i.\text{tipo};\}$

$E ::= (E_d) \quad \{E.\text{valor} = E_d.\text{valor};$
 $E.\text{tipo} = E_d.\text{tipo}\}$

$E ::= i \quad \{E.\text{valor} = i.\text{valor},$
 $E.\text{tipo} = i.\text{tipo}\}$

Ejemplo 1



- ◆ Todas las flechas que aparecen en la figura, que indican el orden de realización de las operaciones, son *ascendentes*, es decir: *la información que se asigna a la parte izquierda de la regla se ha calculado utilizando únicamente información procedente de la parte derecha de la misma regla.*

Ejemplo 2

$\Sigma_T = \{\text{int}(\text{tipo}), \text{real}(\text{tipo}), \text{i}(\text{tipo})\},$

$\Sigma_N = \{D(\text{tipo}), T(\text{tipo}), L(\text{tipo})\},$

$P = \{$

$D ::= TL \quad \{L.\text{tipo} = T.\text{tipo};$
 $\quad \quad \quad D.\text{tipo} = L.\text{tipo};\}$

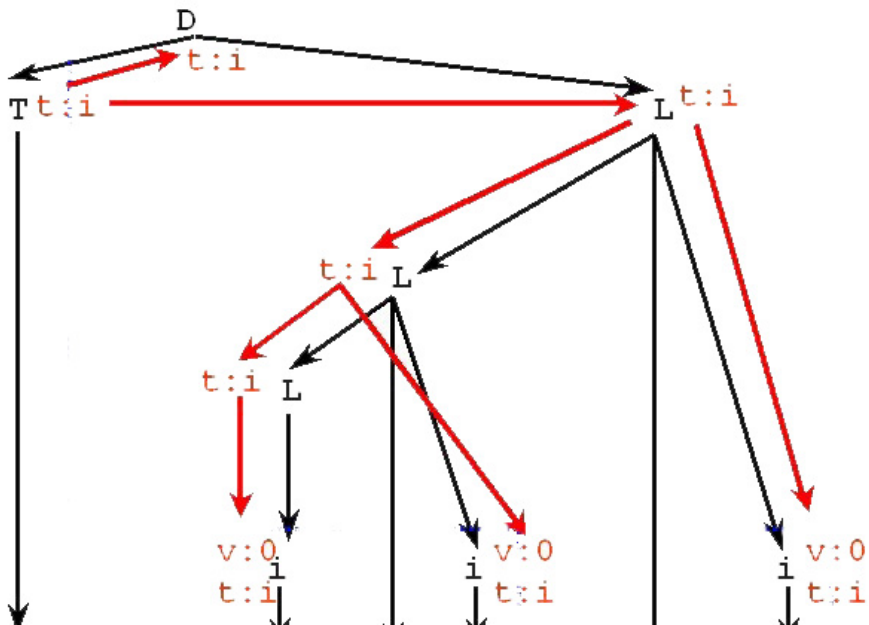
$T ::= \text{int} \quad \{T.\text{tipo} = \text{entero};\}$

$T ::= \text{real} \quad \{T.\text{tipo} = \text{real};\}$

$L ::= L_d, i \quad \{L_d.\text{tipo} = L.\text{tipo};$
 $\quad \quad \quad i.\text{tipo} = L.\text{tipo};\}$

$L ::= i \quad \{i.\text{tipo} = L.\text{tipo};\}$

Ejemplo 2



Tipos de atributos

◆ Atributos sintetizados

- su valor se calcula utilizando los atributos de los símbolos de la parte derecha
- Ejemplo: atributos *tipo* y *valor*

```
E ::= Ei+Ed {E.valor = Ei.valor + Ed.valor;  
                E.tipo = Ei.tipo;}
```

Tipos de atributos (II)

◆ Atributos heredados

- su valor se calcula utilizando los atributos de la parte izquierda o los de otros símbolos de la parte derecha.
- Atributos heredados *por la derecha*
 - ◆ el cálculo del valor de un atributo utiliza atributos de los símbolos que están situados a su derecha.
- Atributos heredados *por la izquierda*
 - ◆ se utilizan los que están a su izquierda y

Tipos de atributos (III)

- ◆ L hereda el atributo `tipo` de su hermano izquierdo (T)

```
D ::= TL {L.tipo = T.tipo;  
          D.tipo = L.tipo;}
```

- ◆ i hereda el atributo `tipo` de su padre (L)

```
L ::= i {i.tipo = L.tipo;}
```

Algunas extensiones

- ◆ Las acciones semánticas pueden aparecer en cualquier posición de la parte derecha de la regla

```
A ::= id {comprobar que id ha sido  
         declarado y recuperar su tipo} := E  
      {comprobar compatibilidad de tipos;  
       id.valor = E.valor}
```


Algunas extensiones (II)

- ◆ Siempre es posible transformar una gramática de atributos con acciones semánticas en cualquier posición en otra equivalente con todas las acciones semánticas al final de las reglas.

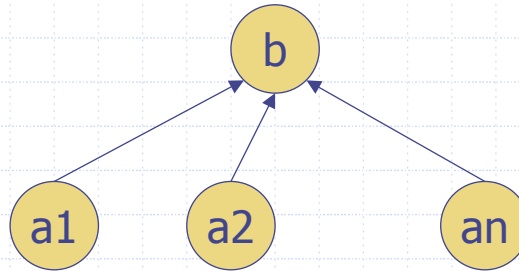
Algunas extensiones (III)

```
A ::= V := E {comprobar compatibilidad de  
tipos;  
id.valor = E.valor}
```

```
V ::= id {comprobar que id ha sido  
declarado y recuperar su tipo  
V.tipo = id.tipo}
```

Dependencias entre atributos

$$b = f(a_1, a_2, \dots, a_n)$$



Grafo de dependencias

Si aparecen dependencias circulares – modificar la gramática

Ejercicio 1

$$\begin{aligned} S ::= ABC & \quad \{B.u = S.u \\ & \quad A.u = B.v + C.v \\ & \quad S.v = A.v\} \end{aligned}$$

$$A ::= a \quad \{A.v = 2 * A.u\}$$

$$B ::= b \quad \{B.v = B.u\}$$

$$C ::= c \quad \{C.v = 1\}$$

- ◆ Dibujar árbol de derivación para la cadena abc
- ◆ Dibujar el grafo de dependencias
- ◆ Suponiendo que se asigna el valor 3 a S.u antes de comenzar la propagación de atributos, ¿cuál será el valor de S.v al terminar la propagación?

Ejercicio 2

$$S ::= ABC \quad \{B.u = S.u$$

$$C.u = A.v$$

$$A.u = B.v + C.v$$

$$S.v = A.v\}$$

$$A ::= a \quad \{A.v = 2 * A.u\}$$

$$B ::= b \quad \{B.v = B.u\}$$

$$C ::= c \quad \{C.v = C.u - 2\}$$

- ◆ Dibujar el grafo de dependencias
- ◆ Suponiendo que se asigna el valor 3 a S.u antes de comenzar la propagación de atributos, ¿cuál será el valor de S.v al terminar la propagación?

¿Dónde se guardan los valores de los atributos semánticos?

- ◆ Los algoritmos descritos en análisis sintáctico manipulan, en lugar de símbolos, estructuras que contienen la información semántica de cada símbolo.
 - ◆ De igual forma, la estructura de cada unidad sintáctica generada por el analizador morfológico incorpora la información semántica asociada
 - Nombre del identificador
 - Valor de una constante
- } Ya lo vimos