

Compiladores: Sesión 15. Análisis semántico, traducción dirigida por sintaxis

Prof. Gloria Inés Alvarez V.

Departamento de Ciencias e Ingeniería de la Computación
Pontificia Universidad Javeriana Cali

13 de marzo de 2008

Traducción Dirigida por Sintaxis

- Asocia información a construcciones de un lenguaje de programación, asignándole atributos al símbolo de la gramática que representa dicha construcción.
- El valor de los atributos se calcula mediante reglas semánticas que se asocian a cada producción.
- Hay dos notaciones:
 - Definiciones dirigidas por sintaxis
 - Esquemas de traducción

Definiciones Dirigidas por Sintaxis

Definición

Es una generalización de una gramática incontextual en la cual cada símbolo tiene asociado un conjunto de atributos.

Hay dos clases de atributos posibles:

- Atributos Sintetizados
- Atributos Heredados

Atributos

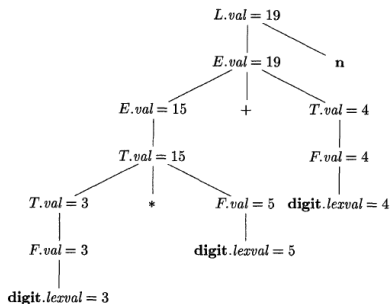
- En general los atributos de un nodo reciben valor mediante la evaluación de las reglas semánticas asociadas a la producción usada en ese nodo propietario del atributo.
- Los valores de los atributos sintetizados se calculan a partir de los valores de atributos de sus nodos hijos en el árbol de análisis sintáctico.
- Los valores de los atributos heredados se calculan a partir de los valores de atributos de su nodo padre o sus nodos hermanos.

Reglas semánticas

- Las reglas semánticas establecen dependencias entre los atributos de los diferentes símbolos.
- Esas dependencias se reflejan en un grafo con el fin de establecer un orden de evaluación de las reglas.
- La evaluación de las reglas asigna valor a los atributos.
- Un árbol de análisis sintáctico que contiene también los atributos y su valores se llama un árbol anotado o decorado.

Ejemplo Definición Dirigida por Sintaxis[Aho 2a Ed. p308]

| PRODUCTION | SEMANTIC RULES |
|-----------------------------------|---------------------------------|
| 1) $L \rightarrow E n$ | $L.val = E.val$ |
| 2) $E \rightarrow E_1 + T$ | $E.val = E_1.val + T.val$ |
| 3) $E \rightarrow T$ | $E.val = T.val$ |
| 4) $T \rightarrow T_1 * F$ | $T.val = T_1.val \times F.val$ |
| 5) $T \rightarrow F$ | $T.val = F.val$ |
| 6) $F \rightarrow (E)$ | $F.val = E.val$ |
| 7) $F \rightarrow \mathbf{digit}$ | $F.val = \mathbf{digit.lexval}$ |

Figure 5.3: Annotated parse tree for $3 * 5 + 4 n$

Atributos Sintetizados

- Los atributos sintetizados se utilizan ampliamente.
- Si una definición dirigida por sintaxis tiene únicamente atributos sintetizados se dice que es **S-atribuida**.
- El árbol de análisis sintáctico de una gramática S-atribuida puede decorarse mediante un recorrido en postorden.

Atributos Heredados

- Sirven para expresar la dependencia que hay entre una construcción del lenguaje de programación y su contexto.
- Siempre es posible reescribir una definición dirigida por sintaxis para que sea S-atribuida.
- En ocasiones es más natural utilizar atributos heredados.

Ejemplo Atributos Sintetizados y Heredados [Aho 2a.Ed. p.315]

| PRODUCTION | SEMANTIC RULES |
|-------------------------------------|--|
| 1) $D \rightarrow T L$ | $L.inh = T.type$ |
| 2) $T \rightarrow \mathbf{int}$ | $T.type = \text{integer}$ |
| 3) $T \rightarrow \mathbf{float}$ | $T.type = \text{float}$ |
| 4) $L \rightarrow L_1, \mathbf{id}$ | $L_1.inh = L.inh$ $addType(\mathbf{id.entry}, L.inh)$ |
| 5) $L \rightarrow \mathbf{id}$ | $addType(\mathbf{id.entry}, L.inh)$ |

Grafo de Dependencias

Para construir el grafo asociado a una definición dirigida por sintaxis se debe:

- 1 Convertir todas las reglas semánticas a la forma
$$b = f(c_1, c_2, \dots, c_k)$$
- 2 Definir un nodo del grafo por cada atributo y arco de b a c siempre que el atributo b dependa para su cálculo del atributo c
- 3 A medida que se van aplicando las producciones se deben ir adicionando los arcos al grafo de dependencias

Ejemplo Grafo de Dependencias [Aho 2a.Ed. p.316]

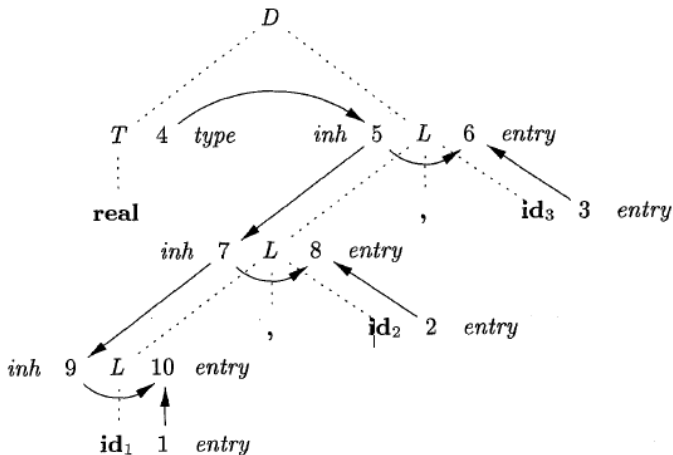


Figure 5.9: Dependency graph for a declaration **float** id_1 , id_2 , id_3

Uso del Grafo de Dependencias

- Para establecer un orden correcto para evaluar las reglas semánticas, se debe hacer un ordenamiento topológico del grafo de dependencias.
- Notar que si el grafo de dependencias es cíclico no es posible encontrar un orden de evaluación de las reglas.

Dada una definición dirigida por sintaxis es muy difícil saber si existe un árbol de análisis sintáctico cuyo grafo de dependencias sea cíclico. Por eso se procura utilizar subclases de definiciones que garanticen que siempre existe un orden de evaluación de las reglas semánticas.

Subclases de DDS con Grafo de Dependencias Acíclico

- Gramáticas S-atribuidas
- Gramáticas L-atribuidas: Entre los atributos asociados al cuerpo de una producción, los arcos del grafo de dependencia van siempre de izquierda a derecha.

Reglas Semánticas con Efecto de Borde Controlado

Hay dos maneras de controlar los efectos de borde:

- Permitirlos únicamente en caso que no afecten la evaluación de los atributos
- Restringir los órdenes de evaluación permitidos de modo que todos produzcan la misma traducción

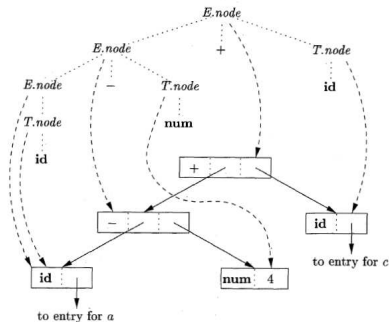
Aplicaciones de la traducción dirigida por sintaxis

- Construcción del árbol de sintaxis
 - Cada nodo corresponde a un elemento de la entrada
 - Las hojas almacenan el valor léxico del elemento que contienen
 - Las reglas semánticas asociadas a las producciones van indicando la manera como se enlazan los nodos
- Chequeo de tipos
- Generación de código intermedio

Ejemplo Generación del Árbol de Sintaxis [Aho 2a.Ed. p. 320]

| PRODUCTION | SEMANTIC RULES |
|-------------------------------|--|
| 1) $E \rightarrow E_1 + T$ | $E.node = \text{new Node}('+', E_1.node, T.node)$ |
| 2) $E \rightarrow E_1 - T$ | $E.node = \text{new Node}('-', E_1.node, T.node)$ |
| 3) $E \rightarrow T$ | $E.node = T.node$ |
| 4) $T \rightarrow (E)$ | $T.node = E.node$ |
| 5) $T \rightarrow \text{id}$ | $T.node = \text{new Leaf}(\text{id}, \text{id.entry})$ |
| 6) $T \rightarrow \text{num}$ | $T.node = \text{new Leaf}(\text{num}, \text{num.val})$ |

Figure 5.10: Constructing syntax trees for simple expressions

Figure 5.11: Syntax tree for $a - 4 + c$