



PROCESADORES DE LENGUAJE

Ingeniería Informática
Primer curso de segundo ciclo



Departamento de Informática y Análisis Numérico
Escuela Politécnica Superior
Universidad de Córdoba
Curso académico 2010 - 2011

Relación de ejercicios nº 5: Traducción basada en la sintaxis

1. Dada la siguiente definición basada en la sintaxis:

<i>REGLAS DE PRODUCCIÓN</i>	<i>REGLAS SEMÁNTICAS</i>
$S \rightarrow E$	imprimir(E.valor)
$E \rightarrow E_1 + T$	E.valor = E_1 .valor + T.valor
$E \rightarrow T$	E.valor = T.valor
$T \rightarrow T_1 * F$	T.valor = T_1 .valor * F.valor
$T \rightarrow F$	T.valor = F.valor
$F \rightarrow (E)$	F.valor = E.valor
$F \rightarrow \text{identificador}$	F.valor = identificador.valor
$F \rightarrow \text{número}$	F.valor = número.valor

y la cadena $w = (4 * 2 + 5) * 3$

- Construye el árbol sintáctico anotado asociado a la cadena w .
- Dibuja el grafo de dependencias y determina el valor de los atributos.

2. Considera la siguiente gramática:

$P = \{$
 $E \rightarrow E \text{ or } T$
 $E \rightarrow T$
 $T \rightarrow T \text{ and } F$
 $T \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{not } F$
 $F \rightarrow \text{true}$
 $F \rightarrow \text{false}$
 $\}$

- Construye la tabla de análisis sintáctico SLR.
- Diseña una definición s-atribuida y, basándose en dicha definición, construye un traductor ascendente (con su correspondiente pila Valor) que utilice la tabla de análisis SLR

c. Utiliza el traductor ascendente para evaluar la expresión lógica:
false or not (true and false)

3. La siguiente gramática genera expresiones formadas mediante la aplicación de un operador aritmético “+” a constantes reales y enteras. Cuando se suman dos enteros, el tipo obtenido es entero; en caso contrario, es real:

$$P = \{ \begin{array}{l} E \rightarrow E + T \\ E \rightarrow T \\ T \rightarrow \text{número.número} \\ T \rightarrow \text{número} \end{array} \}$$

- Diseña una definición basada en la sintaxis para determinar el tipo de cada subexpresión: entero o real.
- Amplía la definición dirigida en la sintaxis del apartado anterior para que traduzca expresiones a notación postfija, además de determinar los tipos.
 - Utiliza el operador unitario “entareal” para convertir un valor entero en un valor real equivalente, de manera que ambos operandos de “+” en la forma postfija tengan el mismo tipo.

Vg.: $2 + 3.5 + 7 \rightarrow \text{entareal}(2) 3.5 + \text{entareal}(7) +$

- Transforma la definición basada en la sintaxis del apartado anterior en un esquema de traducción equivalente.
- Elimina la recursión por la izquierda y factoriza por la izquierda el esquema de traducción.
- Genera la tabla predictiva correspondiente a la gramática del esquema de traducción obtenido en el apartado anterior.
- Utiliza la tabla predictiva para codificar un traductor descendente predictivo.

4. Sea el atributo sintetizado “val” el que da el valor del número binario generado por S en la siguiente gramática. Así, si la entrada es 101.101, entonces $S.\text{val} = 5.625$

$$P = \{ \begin{array}{l} S \rightarrow L . L \\ S \rightarrow L \\ L \rightarrow L B \\ L \rightarrow B \\ B \rightarrow 0 \\ B \rightarrow 1 \end{array} \}$$

- Diseña una definición basada en la sintaxis que permita determinar S.val.
- Construye un traductor ascendente utilizando la tabla de análisis sintáctico LR.

5. Considera la siguiente gramática:

$$P = \{ \begin{array}{l} S \rightarrow id L \\ L \rightarrow , id L \\ L \rightarrow : T \\ T \rightarrow integer \\ T \rightarrow real \end{array} \}$$

- a. Construye un esquema de traducción para introducir el tipo de cada identificador en la tabla de símbolos.
 - b. Construye un traductor descendente predictivo a partir del esquema de traducción del apartado anterior.
6. Diseña un esquema de traducción que asegure que un identificador no aparece dos veces en una lista de identificadores. Utiliza dicho esquema para obtener un traductor ascendente o descendente.
7. Escribe un esquema de traducción que permita generar las declaraciones de las estructuras en C:
- a. Los campos de las estructuras podrán ser de tipo int, float, char o un puntero (simple, doble, triple,...) a int, float, char o a otras estructuras.
 - b. El esquema ha de controlar que no se utilice un mismo identificador para dos o más estructuras.
 - c. Además, el esquema ha de guardar en la tabla de símbolos el número de campos que posee cada estructura.
8. Considera la siguiente definición L-atribuida:

REGLA DE PRODUCCIÓN	REGLA SEMÁNTICA
$E \rightarrow T E'$	$E'.h = T.s, E.s = E'.s$
$E' \rightarrow or T E'_1$	$E'_1.h = E'.h, E'.s = T.s \text{ or } E'_1.s$
$E' \rightarrow \epsilon$	$E'.s = E'.h$
$T \rightarrow F T'$	$T'.h = F.s, T.s = T'.s$
$T' \rightarrow and F T'_1$	$T'_1.h = T'.h, T'.s = F.s \text{ and } T'_1.s$
$T' \rightarrow \epsilon$	$T'.s = T'.h$
$F \rightarrow (E)$	$F.s = E.s$
$F \rightarrow true$	$F.s = verdadero$
$F \rightarrow false$	$F.s = falso$
$F \rightarrow not F_1$	$F.s = not F_1$

- a. Introduce símbolos marcadores en la gramática subyacente para poder realizar una evaluación ascendente de los atributos heredados.

- b. Construye la tabla de análisis sintáctico LR para la gramática construida en el paso anterior.
- c. Efectúa una evaluación ascendente, utilizando la pila Valor, de la siguiente cadena:
(true and false or true) or not (false and true)

9. Considera las siguientes “estructuras” definidas en el lenguaje de programación C:

```
struct ficha_punto
{
    double x, y, z;
    int a, b;
};
```

- a. Diseña una gramática que permita generar este tipo de estructuras.
 - **Observación:** los campos de la estructura sólo serán de tipo **double o int**.
- b. Utiliza dicha gramática para diseñar una **definición s-atribuida** que permita introducir en la tabla de símbolos el número de campos que posee cada estructura.
- c. Modifica la definición s-atribuida para que exprese las acciones semánticas utilizando la pila “Valor”.

10. Dada la gramática LR(1):

```
P = {
    L → L b
    L → a
}
```

considera la siguiente modificación de la misma:

```
P' = {
    L → M L b
    L → a
    M → ε
}
```

- a. ¿Cuál es el árbol sintáctico asociado a la cadena “abbb”?
- b. Comprueba que la gramática modificada **no** es LR(1).