

Prueba puntuable del tema 4: análisis semántico

Procesadores de lenguaje - II26 - TE2

25 de marzo de 2010

Normativa La duración de esta prueba es de 40 minutos. Puedes consultar libros y apuntes. No puedes utilizar dispositivos electrónicos de ningún tipo para resolverla.

Enunciado Considera la siguiente gramática de un lenguaje de programación ficticio:

$\langle \text{Programa} \rangle \rightarrow \text{programa } (\langle \text{Función} \rangle)^* (\langle \text{Sentencia} \rangle)^+ \text{ finPrograma}$

$\langle \text{Función} \rangle \rightarrow \text{función } \langle \text{Tipo} \rangle \text{ identificador } \langle \text{Parámetros} \rangle (\langle \text{Sentencia} \rangle)^+ \text{ finFunción}$

$\langle \text{Tipo} \rangle \rightarrow \text{entero} \mid \text{real}$

$\langle \text{Parámetros} \rangle \rightarrow \text{abreParéntesis } (\langle \text{Tipo} \rangle \text{ identificador } (\text{coma } \langle \text{Tipo} \rangle \text{ identificador })^*)? \text{ cierraParéntesis}$

$\langle \text{Sentencia} \rangle \rightarrow \text{escribe } \langle \text{Expresión} \rangle \text{ puntoYComa}$

$\langle \text{Sentencia} \rangle \rightarrow \text{devuelve } \langle \text{Expresión} \rangle \text{ puntoYComa}$

$\langle \text{Sentencia} \rangle \rightarrow \text{identificador asigna } \langle \text{Expresión} \rangle \text{ puntoYComa}$

$\langle \text{Sentencia} \rangle \rightarrow \text{si abreParéntesis } \langle \text{Expresión} \rangle \text{ cierraParéntesis } (\langle \text{Sentencia} \rangle)^+ (\text{siNo } (\langle \text{Sentencia} \rangle)^+)? \text{ finSi}$

$\langle \text{Expresión} \rangle \rightarrow \langle \text{Término} \rangle (\text{operador } \langle \text{Término} \rangle)?$

$\langle \text{Término} \rangle \rightarrow \text{identificador}$

$\langle \text{Término} \rangle \rightarrow \text{constante}$

$\langle \text{Término} \rangle \rightarrow \text{llama identificador abreParéntesis } (\langle \text{Expresión} \rangle (\text{coma } \langle \text{Expresión} \rangle)^*)? \text{ cierraParéntesis}$

$\langle \text{Término} \rangle \rightarrow \text{abreParéntesis } \langle \text{Expresión} \rangle \text{ cierraParéntesis}$

Añade acciones semánticas a esa gramática para convertirla en un esquema de traducción dirigido por la sintaxis que permita sintetizar, mientras se lleva a cabo un análisis RLL(1) de la entrada, un atributo $\langle \text{Programa} \rangle.cantidadRecursivas$ con la cantidad de funciones que contienen alguna llamada (de forma directa) a sí mismas.

Ten en cuenta las siguientes restricciones: no puedes modificar las producciones de la gramática; no puedes utilizar variables globales; todos los atributos que utilices han de ser de tipo entero, lógico o cadena; y ningún atributo puede comportarse como heredado en un sitio y como sintetizado en otro.

Indica, para cada atributo que utilices, si es heredado o sintetizado, su tipo, y qué representa.

Supón que el analizador léxico dota de un atributo *lexema* a todo componente de la categoría **identificador**.

Ejemplo Si la entrada a analizar fuese esta:

programa

```
funcion entero Suma (entero a, entero b)
    devuelve a + b;
fin_funcion

funcion entero Combinaciones (entero n, entero k)
    si (k == 0)
        devuelve 1;
    fin_si
    si (k == n)
        devuelve 1;
    fin_si
    devuelve llama Suma (llama Combinaciones (n - 1, k - 1),
                        llama Combinaciones (n - 1, k));
fin_funcion

funcion entero Factorial (entero n)
    si (n <= 1)
        devuelve 1;
    si_no
        devuelve n * llama Factorial (n - 1);
    fin_si
fin_funcion

escribe llama Combinaciones (5, 3);

escribe llama Factorial (5) / (llama Factorial (3) * llama Factorial (2));

fin_programa
```

el resultado en el atributo que se pide sería 2, puesto que tanto la función **Combinaciones** como la función **Factorial** contienen alguna llamada a sí mismas.