

COMPROBACION DE TIPOS

Consiste en dos tareas:

1. INFERENCIA DE TIPOS $\left\langle \begin{array}{c} \text{Cálculo} \\ \text{Mantenimiento} \end{array} \right\rangle$ de la información de tipos de datos

2. VERIFICACION DE TIPOS \rightarrow El uso de esa información de tipos para asegurar que cada parte de un programa tenga sentido bajo las reglas de tipo del lenguaje.

Estas dos tareas están estrechamente relacionadas y se realizan juntas, y se hace referencia a ellas simplemente como COMPROBACION DE TIPOS

La información de tipos puede ser ESTATICA o DINAMICA, o una mezcla de las dos.

$\begin{array}{c} \text{tiempo de compilación} \\ \text{LISP y sus dialectos} \rightarrow \text{DINAMICA} \\ \text{Pascal, C y Ada} \rightarrow \text{ESTATICA} \end{array}$

$\begin{array}{c} \text{tiempo de ejecución} \end{array}$

* En principio cualquier verificación se puede realizar dinámicamente, si el código objeto carga el tipo de un elemento junto con el valor del elemento.

La información del tipo de datos puede presentarse en un programa en varias formas diferentes. Teóricamente, un tipo de datos es un conjunto de valores, o de manera más precisa, un conjunto de valores con ciertas operaciones sobre ellos.

Ej: El tipo de datos integer (entero) hace referencia a un subconjunto de los enteros matemáticos, junto con las operaciones aritméticas, tales como +, -, *, / que son proporcionadas por la definición del lenguaje.

Estos conjuntos por lo regular se describen mediante una EXPRESION DE TIPO, que es un NOMBRE DE TIPO, tal como integer, o una expresión estructurada tal como array[1..10] of real, cuyas operaciones generalmente se imponen

EJERCICIO DE CONVERSIÓN DE TIPOS

Sobre la siguiente gramática:

$E \rightarrow E + E$
 $| E - E$
 $| E * E$
 $| E / E$
 $| (E)$
 $| \text{numero}$
 $| \text{cadena}$

Realizar los siguientes controles semánticos:

1. Permitir todas las operaciones con números enteros y decimales
2. Realizar uso de promoción de tipos en el caso de que un operando sea entero y otro decimal (convertir el número de tipo entero a decimal)
3. Permitir sólo operaciones de suma (concatenación) y multiplicación (repetición) con cadenas de caracteres
4. Operaciones entre cadenas y números no están permitidas, exceptuando la multiplicación, la cual implica una cadena y un número que indica las veces que se repite: Ej: "ab" * 3 \Rightarrow "ababab"

$E_0 \rightarrow E_1 + E_2$

If ($E_1.\text{error} \neq \text{true}$) AND ($E_2.\text{error} \neq \text{true}$) then // Si las dos expresiones son correctas

If ($E_1.\text{tipo} == \text{cadena}$) AND ($E_2.\text{tipo} == \text{cadena}$) then

$E_0.\text{tipo} = E_1.\text{tipo}$

$E_0.\text{val} = E_1.\text{val} + E_2.\text{val}$ // (concatena las dos cadenas)

Else if ($E_1.\text{tipo} == \text{cadena}$) OR ($E_2.\text{tipo} == \text{cadena}$) then { $\begin{matrix} \text{cadena} \text{ AND } \text{cadena} \neq \text{cadena} \\ \text{cadena} \text{ VS } \text{numero} \neq \text{cadena} \end{matrix}$

$E_0.\text{error} = \text{true};$

Error ("No se puede sumar número con cadena")

Else if ($E_1.\text{tipo} == \text{entero}$) AND ($E_2.\text{tipo} == \text{entero}$) then

$E_0.\text{tipo} = E_1.\text{tipo};$

$E_0.\text{val} = E_1.\text{val} + E_2.\text{val};$

Else if ($E_1.\text{tipo} == \text{decimal}$) AND ($E_2.\text{tipo} == \text{decimal}$) then

$E_0.\text{tipo} = E_1.\text{tipo};$

$E_0.\text{val} = E_1.\text{val} + E_2.\text{val};$

Else if ($E_1.\text{tipo} == \text{decimal}$) AND ($E_2.\text{tipo} == \text{entero}$) then

$E_0.\text{tipo} = \text{decimal};$

$E_2.\text{val} = \text{entero2real}(E_2.\text{val});$

$E_0.\text{val} = E_1.\text{val} + E_2.\text{val};$

Else if ($E_1.\text{tipo} == \text{entero}$) AND ($E_2.\text{tipo} == \text{decimal}$) then

$E_0.\text{tipo} = \text{decimal};$

$E_1.\text{val} = \text{entero2real}(E_1.\text{val});$

$E_0.\text{val} = E_1.\text{val} + E_2.\text{val};$

End if;

Else // Hay error en E_1 o E_2 y no puede hacer comprobaciones, sólo propagar el error

$E_0.\text{error} = \text{true};$

$E_0.\text{tipo} = \text{vario}$

End if;