

Proyecto 3 - Analizador de Tipos

1. Descripción de la Actividad

Un analizador de tipos es un algoritmo que asigna un tipo a una expresión en función del uso de las subexpresiones que utiliza. Utilizaremos para este proyecto un subconjunto muy básico de un lenguaje con funciones.

Expresiones: $E ::= \text{entero} \mid \text{booleano} \mid \text{var} \mid E_1 + E_2 \mid E_1 < E_2 \mid E_1 \wedge E_2 \mid E_1 E_2 \mid \text{lambda } \text{var}.E \mid (E)$

donde:

- **entero** representa cualquier número entero
- **booleano** representa uno de los valores True o False
- **var** representa el nombre de una variable.
- E, E_1, E_2 son expresiones
- $+$ es un operador que recibe un par de enteros y retorna un entero
- $<$ es un operador que compara enteros y retorna un booleano
- \wedge es un operador que recibe un par de booleanos y retorna un booleano.
- La aplicación de una función E_1 a su argumento E_2 se expresa como $E_1 E_2$.
- Una función se define con la expresión $(\text{lambda } \text{var}.E)$ donde var es el parámetro formal y E es la expresión resultante.

Los tipos de las expresiones vienen dados por los términos siguientes:

Tipo $T ::= \text{int} \mid \text{bool} \mid a \mid T_1 \rightarrow T_2 \mid (T)$

donde:

- **int** y **bool** son dos constantes de tipo.
- **a** es una variable de tipo.
- T, T_1, T_2 son tipos
- $T_1 \rightarrow T_2$ es el tipo de las funciones que reciben un argumento de tipo T_1 y retornan un argumento de tipo T_2 .

1.1. Primera parte

Seleccionar una representación para expresiones y tipos. Se recomienda definir una clase *Expresion* de la que hereden las diferentes formas de expresiones y una clase *Tipo* de la que hereden las diferentes formas de tipos. No se requiere escribir un analizador sintáctico para las expresiones.

Una sustitución es una lista de pares (variable de tipo, tipo). La aplicación de una sustitución a un tipo consiste en remplazar en el tipo cada una de las variables sustituidas por su correspondiente valor. Por ejemplo, $\text{sustituir}([(a, b \rightarrow \text{int})], a \rightarrow b) = (b \rightarrow \text{int}) \rightarrow b$.

La composición de sustituciones es una sustitución: Sean s_1 y s_2 dos sustituciones: $\text{componer}(s_1, s_2)$ es una sustitución tal que para cualquier tipo T :

$$\text{sustituir}(\text{componer}(s_1, s_2), T) = \text{sustituir}(s_2, \text{sustituir}(s_1, T)).$$

Para construir una sustitución se incluyen los pares $(x_1, \text{sustituir}(s_2, T_1))$ por cada par (x_1, T_1) de s_1 junto con los pares (x_2, T_2) de s_2 tales que x_2 no es la parte izquierda de un par de s_1 .

Para asignar tipos a expresiones hace falta utilizar la operación de unificación $\text{unif}(T_1, T_2)$ que puede producir una excepción cuando los tipos no son unificables o puede retornar una sustitución tal que al aplicarla a cada uno de los términos unificados se obtiene el mismo resultado. Por ejemplo $\text{unif}(a, a \rightarrow a)$ produce una excepción mientras que $\text{unif}(a \rightarrow a, b \rightarrow (c \rightarrow c))$ produce (por ejemplo) la sustitución $[(a, c \rightarrow c), (b, c \rightarrow c)]$.

1.2. Segunda parte

Representar sustituciones como listas de pares (variable, tipo) y las operaciones de sustituir , componer y unif (la segunda generalmente hace falta para implementar la tercera, en <http://ldc.usb.ve/~suarez/lenguajes1/Prolog.pdf> hay una descripción del algoritmo de unificación de términos).

Un ambiente de tipos para variables es una función parcial de variables de tipo en tipos. Ella indica cual tipo corresponde a cada variable de una expresión. Un ambiente puede ser vacío, es decir, cualquiera sea la variable que reciba retorna una excepción. la extensión de un ambiente de tipos con un nuevo par (x, T) se define como

$$\begin{aligned} ((x : T) + \text{Amb})(x) &= T \\ ((x : T) + \text{Amb})(y) &= \text{Amb}(y) \text{ cuando } x \neq y \end{aligned}$$

Un juicio sobre el tipo de una expresión es una tripleta (Amb, E, T) donde Amb es un ambiente de tipos, E es una expresión y T es un tipo. Un juicio de tipos generalmente se escribe: $\text{Amb} \vdash E : T$.

Los juicios de tipos siguen las siguientes reglas:

$$\begin{aligned} \text{Amb} &\vdash \text{entero} : \text{int} \\ \text{Amb} &\vdash \text{booleano} : \text{bool} \\ \text{Amb} &\vdash x : T \text{ si } \text{Amb} \text{ contiene el par } (x, T) \\ \text{Amb} &\vdash E_1 + E_2 : \text{int} \text{ si } \text{Amb} \vdash E_1 : \text{int} \text{ y } \text{Amb} \vdash E_2 : \text{int} \\ \text{Amb} &\vdash E_1 < E_2 : \text{bool} \text{ si } \text{Amb} \vdash E_1 : \text{int} \text{ y } \text{Amb} \vdash E_2 : \text{int} \\ \text{Amb} &\vdash E_1 \wedge E_2 : \text{bool} \text{ si } \text{Amb} \vdash E_1 : \text{bool} \text{ y } \text{Amb} \vdash E_2 : \text{bool} \\ \text{Amb} &\vdash (\text{lambda } x.E) : T_1 \rightarrow T_2 \text{ si } (x : T_1) + \text{Amb} \vdash E : T_2 \\ \text{Amb} &\vdash \text{Amb} \vdash E_1 E_2 : T \text{ si existe } T' \text{ tal que } \text{Amb} \vdash E_1 : T' \rightarrow T \text{ y } \text{Amb} \vdash E_2 : T' \end{aligned}$$

Para implementar la asignación de tipos se puede definir la función $\text{asigTipo}(\text{Amb}, E, T)$ cuyo resultado es una sustitución s que cumple la relación $\text{Amb} \vdash E : \text{sustituir}(s, T)$.

Siguiendo los juicios anteriores, asigTipo puede implementarse siguiendo los lineamientos dados a continuación:

$$\begin{aligned} \text{asigTipo}(\text{Amb}, \text{entero}, T) &= \text{unif}(T, \text{int}) \\ \text{asigTipo}(\text{Amb}, \text{booleano}, T) &= \text{unif}(T, \text{bool}) \\ \text{asigTipo}(\text{Amb}, x, T) &= \text{unif}(T, \text{Amb}(x)) \end{aligned}$$

$$\begin{aligned} \text{asigTipo}(\text{Amb}, E_1 + E_2, T) &= \text{componer}(s_2, \text{unif}(T, \text{int})) \\ &\text{donde:} \\ s_2 &= \text{componer}(s_1, \text{asigTipo}(\text{Amb}, E_2, \text{int})) \\ s_1 &= \text{asigTipo}(\text{Amb}, E_1, \text{int}) \end{aligned}$$

$$\begin{aligned} \text{asigTipo}(\text{Amb}, E_1 < E_2, T) &= \text{componer}(s_2, \text{unif}(T, \text{bool})) \\ &\text{donde:} \\ s_2 &= \text{componer}(s_1, \text{asigTipo}(\text{Amb}, E_2, \text{int})) \\ s_1 &= \text{asigTipo}(\text{Amb}, E_1, \text{int}) \end{aligned}$$

$$\begin{aligned} \text{asigTipo}(\text{Amb}, E_1 \wedge E_2, T) &= \text{componer}(s_2, \text{unif}(T, \text{bool})) \\ &\text{donde:} \\ s_2 &= \text{componer}(s_1, \text{asigTipo}(\text{Amb}, E_2, \text{bool})) \\ s_1 &= \text{asigTipo}(\text{Amb}, E_1, \text{bool}) \end{aligned}$$

$$\begin{aligned} \text{asigTipo}(\text{Amb}, (\text{lambda } x.E), T) &= \text{componer}(s_1, \text{unif}(T, \text{sustituir}(s_1, a \rightarrow b))) \\ &\text{donde:} \\ s_1 &= \text{asigTipo}(\text{Amb}_1, E, b), \\ \text{Amb}_1 &= (x : a) + \text{Amb}, \\ b &= \text{nueva variable de tipos} \\ a &= \text{nueva variable de tipos} \end{aligned}$$

$$\begin{aligned} \text{asigTipo}(\text{Amb}, E_1 E_2, T) &= \text{componer}(s_1, \text{asigTipo}(\text{Amb}, E_1, \text{sustituir}(s_1, a \rightarrow T))) \\ &\text{donde:} \\ s_1 &= \text{asigTipo}(\text{Amb}, E_2, a) \\ a &= \text{nueva variable de tipos} \end{aligned}$$

2. Entrega de la implementación

- Debe entregar un archivo **.tar.gz** o **.tar.bz2** (no puede ser **.rar**, ni **.zip**) que contenga los archivos en Python de la implementación de la actividad descrita anteriormente.
- Debe incluir en su entrega un archivo llamado **Leeme.txt**, en el cual se debe indicar el estado actual del proyecto y las indicaciones de como ejecutarlo.
- Debe enviar el código de su proyecto a la dirección de correo **gpalma@ldc.usb.ve**
- La fecha de entrega del proyecto es el viernes 9 de abril del 2010 (semana 12), hasta las 3:30 pm.
- Se evaluará el estilo correcto de programación, debe emplear una indentación adecuada y consistente en cualquier editor. Su proyecto debe estar debidamente documentado. Se evaluará el uso de la programación orientada a objetos y de la programación funcional vista en clase. En proyecto debe ejecutarse correctamente en las máquinas del LDC.
- Este proyecto está sujeto a modificaciones/correcciones las cuales se publicarán página web del curso. Por lo tanto es su deber revisarla periódicamente.
- El valor del proyecto es 34 puntos.