

Auxiliar N+1
CC41A - CC54A
Lenguajes de Programación

Profesor: Éric Tanter
Aux: Richard Ibarra, Oscar E. A. Callaú
{etanter,ribarra,oalvarez}@dcc.uchile.cl

Santiago - Chile, Jun/16/2008

1. Sistema de Tipos

Dada las siguientes reglas gramaticales

$$\begin{aligned} THs \rightarrow & num \\ & | \quad sym \\ & | \quad bool \\ & | \quad \{+ \ THs \ THs\} \\ & | \quad \{no \ THs\} \\ & | \quad \{fun \ sym \ \{THs\}\} \\ & | \quad \{THs \ THs\} \end{aligned}$$

Que define un lenguaje con tipos primitivos **num** que pertenece al conjunto de número enteros (**Z**) y **bool** que pertenece al conjunto de valores booleandos (**{true,false}**), con operaciones aritméticas y lógicas, declaración de funciones con recepción de un parámetro mediante la palabra reservada **fun**, manejo de variables (**sym**) y aplicación de funciones.

Se desea que Ud. como especialista de sistema de tipos:

1. Proponga todos los juicios necesarios para el lenguaje.
2. Proponga el juicio de tipo para la siguiente regla gramatical:

$$THs \rightarrow \{with \ \{symbol \ THs\} \ THs\}$$

3. Para las siguientes expresiones, complemente los tipos faltantes, obtenga el árbol de derivación para las siguientes expresiones y verifique si son válidas:

```
{fun x:int {fun {y:int} {+ x y}}}
```

```
{with {x {fun y:bool {y}}} {x {no true}}}
```

```
{with {x {fun x {x}}} {x x}}
```

```
{with {f {fun:int x:int {fun:int y:int {+ x y}}}}  
  {+ 3 {f 5}}}
```

2. Pauta: Sistema de Tipos

Juicio de tipos primitivos, los tipos primitivos que se definen en el lenguaje son: `num` y `bool`, como tipos para manejar los valores enteros como los valores de la lógica determinista de `Bool`.

$$\Gamma \vdash Z : \text{number}$$

$$\Gamma \vdash \{ \text{true}, \text{false} \} : \text{boolean}$$

Juicio de variables, la particularidad de este caso es la búsqueda de su tipo en el repositorio de tipos definidos, para nuestro caso inicialmente toda variable tiene un tipo polimórfico y según su contexto se restringue o no.

$$\Gamma \vdash i : \Gamma(i)$$

Juicio de operaciones Aritméticas, toda operación aritmética hecha en nuestro tipos primitivos tiene como resultado un valor determinista único constante que pertenece al conjunto de los enteros.

$$\frac{\Gamma \vdash i : \text{number} \quad \Gamma \vdash d : \text{number}}{\Gamma \vdash \{ + \ i \ d \} : \text{number}}$$

Juicio de operadores lógicos, como es expresado en el desarrollo de este trabajo el juicio de un operador lógico retornaria siempre un tipo lógico Booleando.

$$\frac{\Gamma \vdash b : \text{boolean}}{\Gamma \vdash \{ \text{no} \ b \} : \text{boolean}}$$

Juicio de funciones, la funciones como son bien conocidas están definidas como procesos que recibe un parámetro y retorna un cálculo, en otras palabras una entrada y una salida.

$$\frac{\Gamma[i : \leftarrow t_1] \vdash b : t_2}{\Gamma \vdash \{ \text{fun}(i : t_1) : t_2 \ b \} : (t_2 \rightarrow t_1)}$$

Juicio de aplicación de funciones, el caso de las aplicaciones es un estudio donde tanto la función deribada de un juicio anterior y su argumento real se encuentra para obtener un resultado concreto y estricto.

$$\frac{\Gamma \vdash f : \{ t_1 \rightarrow t_2 \} \quad \Gamma \vdash a : t_1}{\Gamma \vdash \{ f \ a \} : t_2}$$

Juicio de declaraciones(With), las declaraciones definidas en nuestro lenguaje como la utilización de la directiva *with* son procesadas como un híbrido de una función y su aplicación.

$$\frac{\Gamma \vdash \text{val} : t_1 \quad \Gamma[i : \leftarrow t_1] \vdash \text{body} : t_2}{\Gamma \vdash \{ \text{with} \ \{ x : t_1 \ \text{val} \} : t_2 \ \text{body} \} : t_2}$$

Árbol de derivación, para el último caso tenemos, que por el desarrollando de los juicios de tipos, que la expresión **no** es válida.

▪ (0.5pt)

$$\frac{\odot \vdash f : (\text{number} \rightarrow (\text{number} \rightarrow \text{number}))}{\odot \vdash \{ f \ 5 \} : (\text{number} \rightarrow \text{number})}$$

Luego, como:

$$\frac{\Gamma \vdash l : \text{number} \quad \Gamma \vdash r : \text{number}}{\Gamma \vdash \{ + \ l \ r \} : \text{number}}$$

Pero con la informacin `\{f 5\}` no tiene tipo `number`, luego el programa no es correcto.