

## Lenguajes y Compiladores - Guía del 26/05/2016

**Objetivos:** Comprender las extensiones del cálculo lambda con constantes y operadores. Lograr definir mecanismos de evaluación alternativos. Identificar el origen de la diferencia en los órdenes de evaluación. Poder relacionar las nociones operacionales con las denotacionales.

1. Considere las expresiones  $\mathbf{True} \vee 0$ ,  $\mathbf{True} \vee \Delta\Delta$ , donde  $\Delta = (\lambda x.xx)$ 
  - a) Evalúe de modo eager y normal ambas expresiones.
  - b) De la semántica denotacional eager y normal de las expresiones.
2. Considere la expresión  $\mathbf{let} f \equiv \lambda x.\mathbf{True} \mathbf{in} f (\mathbf{True} + 0)$ . Recuerde que  $\mathbf{let} f \equiv e \mathbf{in} e' \doteq (\lambda f.e')e$ 
  - a) Explique, sin hacer ninguna evaluación, si ese término tiene o no forma canónica en evaluación eager y en evaluación normal.
  - b) Construya el árbol de la evaluación para cada uno de esos órdenes.

3. Extender la semántica de la evaluación para describir el tratamiento de errores. Para esto incorpore las expresiones **error** y **typeerror** como resultados posibles de una evaluación, al mismo nivel que las formas canónicas. Por ejemplo se deberán agregar (entre otras) la reglas:

$$\frac{e \Rightarrow [i] \quad e' \Rightarrow [0]}{e \div e' \Rightarrow \mathbf{error}} \quad \frac{e \Rightarrow [b] \quad e' \Rightarrow z'}{e \vee e' \Rightarrow \mathbf{typeerror}} \quad (z' \notin \langle \mathit{boolcnf} \rangle)$$

4. Analice qué régimen de evaluación de subexpresiones ha adoptado en la semántica del ejercicio anterior. Esto es, las subexpresiones (por ejemplo en  $e + e'$ ) se evalúan antes de chequear que los tipos sean correctos, o se obtiene **typeerror** frente a una inconsistencia de tipos, aunque no se hayan terminado de evaluar todas las subexpresiones?

De reglas que representen la opción no considerada en el ejercicio anterior.

5. Encuentre ecuaciones semánticas sencillas para las siguientes expresiones, considerando los casos eager y normal.

$$(a) \llbracket (\lambda x.e)e' \rrbracket \eta \quad (b) \llbracket we \rrbracket [\eta | w : \iota_{\mathit{fun}} f]$$

6. Considere las expresiones  $\langle \mathbf{True} + 0, \Delta\Delta \rangle$  y  $\langle \Delta\Delta, \mathbf{True} + 0 \rangle$ , donde  $\Delta = (\lambda x.xx)$

- a) Evalúe de modo eager y normal ambas expresiones.
- b) De la semántica denotacional eager y normal de las expresiones.

7. De una expresión  $e$  tal que  $e$  tenga forma canónica bajo orden normal y que también la tengan las siguientes (infinitas) expresiones:  $e.1$ ,  $(e.2).1$ ,  $((e.2).2).1$ , etc.

8. Suponga que  $e$  es una expresión cerrada. Considere las siguientes expresiones:

**letrec**  $f \equiv \lambda x.\mathbf{if} e \mathbf{then} 1 \mathbf{else} f x \mathbf{in} f 0$

**letrec**  $f \equiv \lambda x.\mathbf{if} e \mathbf{then} \mathbf{True} \mathbf{else} f x \mathbf{in} f 0 + 1$

Evaluar del modo eager y normal estos programas, considerando por separado los casos  $e \Rightarrow \mathbf{true}$  y  $e \Rightarrow \mathbf{false}$ .

9. Para los programas del ejercicio 8, calcular la semántica denotacional eager y normal directa considerando por separado los casos  $\llbracket e \rrbracket \eta = \iota_{\mathit{norm}} (\iota_{\mathit{bool}} \mathbf{V})$  y  $\llbracket e \rrbracket \eta = \iota_{\mathit{norm}} (\iota_{\mathit{bool}} \mathbf{F})$ .
10. Decida si la siguiente afirmación *Mmm?* es cierta o no y justifique su respuesta: “Si  $e \Rightarrow_E z$ , entonces toda subexpresión  $e'$  de  $e$  tiene forma canónica”.