

## Lenguajes y Compiladores - Trabajo práctico 6 - Año 2007

- (1) Para el término  $(\lambda xy.yx((\lambda x.x) (\lambda z.z))) ((\lambda xy.xy) (\lambda x.x) (\lambda z.z))$ 
  - (a) Dar la reducción de orden normal.
  - (b) Dar la evaluación normal.
  - (c) Dar la evaluación eager.
  
- (2) Elimine los patrones de la siguiente expresión del lenguaje aplicativo eager: **let**  $\langle x, \langle \langle \rangle, y \rangle \rangle \equiv w, z \equiv \langle 0, 0 \rangle$  **in**  $x + z.0$ .
  
- (3) Suponga que  $e$  es una expresión cerrada. Para el término **letrec**  $f \equiv \lambda x.\mathbf{if} e \mathbf{then} 1 \mathbf{else} f x \mathbf{in} f 0$  en el lenguaje aplicativo eager se pide
  - (a) Calcular la semántica denotacional directa considerando por separado los casos  $\llbracket e \rrbracket_\eta$  verdadero o falso.
  - (b) Calcular la evaluación considerando por separado los casos  $e \Rightarrow \mathbf{true}$  y  $e \Rightarrow \mathbf{false}$ .
  
- (4) Suponga que  $e$  es una expresión cerrada. Considere la siguiente expresión del lenguaje aplicativo eager:  
**letrec**  $f = \lambda x.\mathbf{if} e_0 \mathbf{then} \mathbf{True} \mathbf{else} f x$   
**in**  $\langle f 0 + 1, \mathbf{True} + f 0 \rangle$ 
  - (a) Calcular la semántica denotacional directa considerando por separado los casos  $\llbracket e \rrbracket_\eta$  verdadero o falso.
  - (b) Calcular la evaluación considerando por separado los casos  $e \Rightarrow \mathbf{true}$  y  $e \Rightarrow \mathbf{false}$ .
  
- (5) Resolver el ejercicio 11.1 del libro de Reynolds.
  
- (6) Resolver el ejercicio 11.2 del libro de Reynolds.
  
- (7) Resolver el ejercicio 11.3 del libro de Reynolds.
  
- (8) Resolver el ejercicio 11.4 del libro de Reynolds.
  
- (9) Resolver el ejercicio 11.10 del libro de Reynolds.
  
- (10) Definir la semántica denotacional directa de **let** para el lenguaje aplicativo eager.

- (11) Para el lenguaje aplicativo normal, reescribir el término
- $$\mathbf{letrec} \quad \mathit{par} \equiv \lambda x. \mathbf{if} \ x = 0 \ \mathbf{then} \ \mathbf{true} \ \mathbf{else} \ \mathit{impar}(x - 1)$$
- $$\mathit{impar} \equiv \lambda x. \mathbf{if} \ x = 0 \ \mathbf{then} \ \mathbf{false} \ \mathbf{else} \ \mathit{par}(x - 1)$$
- in**  $e$
- eliminando la notación **letrec**
- utilizando **let**, patrones y **rec**.
  - utilizando patrones y **rec**.
  - utilizando **rec**.
- (12) Para cada una de las versiones del lenguaje aplicativo, dar las reglas de evaluación de las expresiones de la forma @ y **sumcase**.
- (13) Definir la semántica denotacional de continuaciones de **let** para el lenguaje aplicativo eager.
- (14) Resolver el ejercicio 12.1 (incisos (b) a (d)) del libro de Reynolds.
- (15) Resolver el ejercicio 12.2 del libro de Reynolds.
- (16) Para el lenguaje aplicativo normal, definir las operaciones de lista como syntactic sugar usando las operaciones @ y **sumcase**. Luego dar la semántica operacional.
- (17) Para el lenguaje aplicativo normal, definir las formas canónicas del tipo lista, y dar las reglas de evaluación correspondientes. Corroborar que lo hecho es consistente con el ejercicio anterior.
- (18) Para el lenguaje aplicativo normal dar la semántica directa y de continuaciones de **rec**  $e$ .
- (19) Para el lenguaje aplicativo eager, calcular y comparar la semántica denotacional eager de la expresión  $(\lambda x.x \ 3)(\lambda y.y + x)$  en un ambiente  $\eta$  tal que  $\eta \ x = \iota_{int} 1$  utilizando alcance estático y dinámico.
- (20) Definir la semántica denotacional directa de **let** y **letrec** para el lenguaje con alcance dinámico.
- (21) Definir la semántica denotacional de continuaciones de **let** y **letrec** para el lenguaje aplicativo eager con alcance dinámico.
- (22) Calcular la semántica de **callcc**  $\lambda k.(\lambda x.\mathbf{throw} \ k \ 3)(\mathbf{throw} \ k \ 4)$ .