

Apellido y Nombre:

email:

nota

1	2	3	4	5	6
---	---	---	---	---	---

Lenguajes y Compiladores

Segundo Examen Parcial

12/6/2013

1. Considere la expresión $(\lambda x.x(\lambda y.xy)x)(\lambda z.\lambda w.z)$.
 - a) Evalúe la expresión en orden normal (\Rightarrow).
 - b) El resultado obtenido, ¿es el mismo que el que se obtendría con la evaluación eager?
 - c) ¿Es la forma canónica una forma normal de la expresión?
2. Responda V o F. Justifique.
 - a) Si una expresión cualquiera e del cálculo lambda puede ser evaluada en ambos órdenes, eager y normal (o sea ambas terminan), entonces el resultado es el mismo en ambos órdenes.
 - b) Si $e \rightarrow^* e_0$ entonces $\llbracket e \rrbracket = \llbracket e' \rrbracket$ (en el Cálculo Lambda).
 - c) Si $e \rightarrow^* e_0$ entonces $\llbracket e \rrbracket = \llbracket e' \rrbracket$ (en el Cálculo Lambda Eager).
3. Considere el Lenguaje Aplicativo Eager.
 - a) Dé la ecuación semántica de $e + e'$.
 - b) Calcule la semántica de **true** + $\Delta\Delta$
 - c) Modifique la ecuación semántica de la suma de manera que si alguna de las expresiones tiene como semántica un valor atípico (fuera de V) y la otra es un valor de V , entonces el resultado sea el valor atípico. Por ejemplo en el caso anterior la semántica debería ser \perp , en vez de *tyerr*.
4. Considere los siguientes programas en el Lenguaje Aplicativo Normal:
$$e_0 = \mathbf{rec} (\lambda f.\lambda x.\langle f\ x, 1 \rangle) \qquad e_1 = \mathbf{rec} (\lambda f.\lambda x. (\langle f\ x, 1 \rangle.0))$$

Evalúe (o determine que no se puede evaluar):

(i) e_0 (ii) $e_0\ 0$ (iii) $e_1\ 0$.
5. Considere el siguiente programa en el lenguaje Iswim:
$$e =_{def} \mathbf{letrec}\ f \equiv \lambda x. \mathbf{if}\ x = 0 \mathbf{then\ skip\ else}\ (\mathbf{ref}\ 0; f(x - 1)) \mathbf{in}\ f\ [n]$$
 - a) Proponga sin calcular $\llbracket e \rrbracket \eta \llbracket \cdot \rrbracket$, asumiendo $n < 0$ y $n \geq 0$.
 - b) Calcule F (funcional del letrec) de la forma más sencilla posible.
 - c) Calcule $F^2 \perp \langle \llbracket \cdot \rrbracket, \text{!int}n \rangle$.
6. Se quiere utilizar semántica de continuaciones para calcular cuántas asignaciones hay en un programa; notar que esta es una propiedad estática y no tiene que ver con la ejecución. Definir una función

$$\llbracket _ \rrbracket^c: \langle \text{comm} \rangle \rightarrow (\mathbb{Z} \rightarrow \mathbb{Z}) \rightarrow \mathbb{Z}$$

que cuente la cantidad de asignaciones en un programa del lenguaje imperativo simple (incluyendo comandos para fallas, recuperación de fallas, input y output). Indique además cuál es la continuación inicial.