

Apellido y Nombre:

email:

nota

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

Lenguajes y Compiladores

Segundo Examen Parcial

2018

1. Considere la expresión $e = (\lambda x. \lambda y. x(e_0 y))(\lambda x. \lambda y. y(xy))(\lambda x. \lambda y. y)$. Para cada evaluación (normal o eager) determine si se puede definir e_0 de manera que la expresión e no se pueda evaluar (o sea no está definida la semántica big-step).
2. Determine si es Verdadero o Falso. Justifique su respuesta. Enuncie todo resultado teórico que utilice.
 - a) Si $e \rightarrow^* z$ entonces $e \Rightarrow_N z$ o $e \Rightarrow_E z$. (Aquí z es una forma canónica).
 - b) Si $e \rightarrow^* e'$ entonces $\llbracket e \rrbracket^E = \llbracket e' \rrbracket^E$.
 - c) Si existe z' tal que $e' \Rightarrow_E z'$, entonces vale la regla β , o sea las expresiones $(\lambda v. e)e'$ y $(e/v \mapsto e')$ tienen la misma semántica eager.
3. Considere la expresión $e = \langle (\lambda v. e_0)e_1, \langle e_2, e_3 \rangle.1, e_4 \rangle$, donde $\llbracket e_0 \rrbracket \eta = \iota_{norm} z$ y $\llbracket e_4 \rrbracket \eta = \perp_D$. Calcule la semántica denotacional eager y normal de $e.0$, suponiendo que $v \notin FV e_0$.
4. Considere el programa aplicativo normal $e = \mathbf{rec}(\lambda w. \langle 1, \langle w, 1 \rangle \rangle)$. Calcule de manera detallada la semántica denotacional de $e.0$.
5. Considere el programa aplicativo eager
$$e = \mathbf{letrec} \ w \equiv \lambda x. \mathbf{if} \ e_0 \ \mathbf{then} \ 1 \ \mathbf{else} \ w \ x \ \mathbf{in} \ w \ (2 + e_1),$$
donde e_0 y e_1 son expresiones cerradas. Analice todos los resultados posibles que puede asumir $\llbracket e \rrbracket \eta$, en función de los resultados que adopte $\llbracket e_0 \rrbracket \eta$, y $\llbracket e_1 \rrbracket \eta$.
6. Enuncie y demuestre el Teorema de Sustitución para el Cálculo Lambda Normal .