

Apellido y Nombre:
email:

| |
|------|
| nota |
|------|

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

Lenguajes y Compiladores

Segundo Examen Parcial

23/6/2014

- Determine cuál es verdadera. Justifique su respuesta.
 - $\langle \mathbf{while\ true\ do\ } x := x - 1, [\sigma|x : 2] \rangle \rightarrow^* \langle \mathbf{while\ true\ do\ } x := x - 1, [\sigma|x : 0] \rangle$
 - $\langle \mathbf{newvar\ } x := 1 \mathbf{\ in\ } y := x; x := 0, \sigma \rangle \rightarrow \langle \mathbf{skip}, [\sigma|y : 0] \rangle$
 - $\langle \mathbf{newvar\ } x := 1 \mathbf{\ in\ } y := x; x := 0, \sigma \rangle \rightarrow \langle \mathbf{newvar\ } x := 1 \mathbf{\ in\ } x := 0, [\sigma|y : 0] \rangle$
- Evalúe (\Rightarrow) la siguiente expresión del lenguaje aplicativo eager:
letrec $f \equiv \lambda x. \mathbf{if\ } x \leq 0 \mathbf{\ then\ } x \mathbf{\ else\ } f(x - 1) \mathbf{\ in\ } f\ 2$
- Calcule la semántica denotacional del siguiente programa del lenguaje Imperativo con fallas e I/O:
newvar $x := 0 \mathbf{\ in\ if\ } x < y \mathbf{\ then\ } !x \mathbf{\ else\ } ?x; \mathbf{abort}$
- Para cada ítem, dé una expresión e del Cálculo Lambda tal que:
 - Su denotación en D_∞ sea \perp , pero que su semántica normal no sea \perp .
 - Tenga denotación \perp en la modalidad eager, pero no sea \perp ni en la semántica normal ni en la del Cálculo Lambda (D_∞).Recordar que los dominios involucrados en cada caso son:
 - Para el cálculo lambda: $D_\infty \cong [D_\infty \rightarrow D_\infty]$.
 - Para orden normal: $V \cong [D \rightarrow D]$ y $D = V_\perp$.
 - Para orden eager: $V \cong [V \rightarrow D]$ y $D = V_\perp$.
- Encuentre programas en los respectivos lenguajes que tengan como semántica denotacional el objeto dado. No es necesario que calcule la semántica.
 - En el lenguaje Imperativo con fallas e I/O:
 $\llbracket c \rrbracket \sigma = \iota_{in}(\lambda k. \iota_{out}(k, \iota_{out}(k + 1, \iota_{out}(k + 2, \dots))))$
 - En el lenguaje Iswim: $\llbracket e \rrbracket \eta \sigma = \iota_{norm}([\sigma|r_0 : \iota_{int}0 | r_1 : \iota_{ref}r_0], \iota_{ref}r_0)$, donde $r_0 = \mathbf{new\ } \sigma$ y $r_1 = \mathbf{new\ } [\sigma|r_0 : \iota_{int}0]$
- Determine si las siguientes afirmaciones son verdaderas o falsas; justifique su respuesta.
 - Si $e \in \langle \mathbf{exp} \rangle$ es una expresión del Lenguaje Aplicativo Eager tal que $\llbracket e \rrbracket \eta = \iota_{norm}z$, entonces considerada como expresión Iswim satisface $\llbracket e \rrbracket \eta \sigma = \iota_{norm}(\sigma, z)$, para todo $\sigma \in \Sigma$.
 - Existen $e \in \langle \mathbf{exp} \rangle$, $\eta \in Env$, $\sigma, \sigma' \in \Sigma$ y $z \in V$, tales que $\llbracket e \rrbracket \eta \sigma = \iota_{norm}(\sigma', z)$ y $dom(\sigma') \subset dom(\sigma)$. (Notar que la inclusión es estricta).