

Apellido y Nombre:  
email (@mi.unc.edu.ar):  
Nota:

## Lenguajes y Compiladores

2do Parcial 2023

1. Considerá el lenguaje imperativo con input/output pero sin fallas. Sea  $\omega \in \Omega$  con

$$\begin{aligned}\omega &= \iota_{in}(f) \\ f(z) &= \iota_{out}\langle z, \iota_{in}(g_z) \rangle \\ g_z(z') &= \iota_{out}\langle z * z', \perp \rangle\end{aligned}$$

Dar tres programas  $c_1, c_2, c_3$  que satisfagan:

- $\perp \sqsubset \llbracket c_1 \rrbracket \sigma \sqsubset \omega$ .
  - $\omega \sqsubset \llbracket c_2 \rrbracket \sigma$ .
  - $\llbracket c_3 \rrbracket \sigma = \iota_{in}(h)$ , con  $h(0)$  incomparable con  $f(0)$  y  $h(n) = f(n)$  si  $n \neq 0$ .
2. Considerá el cálculo lambda puro.
- Proponé una expresión  $e$  que cumpla las siguientes condiciones simultáneamente:
    - bajo evaluación eager,  $e e'$  diverge para cualquiera expresión  $e'$ .
    - bajo evaluación normal,  $e e'$  evalúa a una forma canónica para cualquier expresión  $e'$ .
  - Realizá una reducción de  $e I$  hasta llegar a una abstracción. Recordá que una reducción es una secuencia de  $\beta$ -contracciones.
3. La contracción ( $\eta$ ) se especifica como  $\frac{}{(\lambda x.e x) \rightarrow_{\eta} e} x \notin FV(e)$ .

Considerá la semántica denotacional normal del cálculo lambda. Recordá que  $V^N \cong [D^N \rightarrow D^N]$  y  $D^N = V_{\perp}^N$ . Tu tarea es dar una expresión  $e$  concreta y mostrar que la semántica denotacional normal no respeta la contracción  $\eta$ . Para ello calculá la semántica de  $\lambda x.e x$  y la de  $e$  para ver que son distintas.

4. Ahora nos pasamos al lenguaje aplicativo eager.

- Proponé una regla de evaluación para expresiones con pattern-matching muy sencillo:

$$\mathbf{let} \langle \langle var \rangle, \langle var \rangle \rangle \equiv \langle exp \rangle \mathbf{in} \langle exp \rangle$$

- Evaluá la siguiente expresión usando esa regla.

$$\mathbf{let} \langle x, y \rangle \equiv (\lambda n. \langle n, n * (-1) \rangle)(-4) \mathbf{in} \mathbf{if} \ x < y \ \mathbf{then} \ -1 \ \mathbf{else} \ \mathbf{if} \ y = x \ \mathbf{then} \ 0 \ \mathbf{else} \ 1$$

5. Considerá la siguiente serie de naturales:

$$S = 1, 2, 3, 1, 2, 3, \dots$$

Dar una expresión  $e$  en el lenguaje aplicativo eager (sin división ni módulo) tal que la denotación de  $e e'$  sea  $\iota_{int}(S(n))$  si la semántica de  $e'$  es  $\iota_{int}(n)$  con  $n > 0$ .

6. Calcular la semántica denotacional eager de la expresión  $e$ .

Si necesitás compañía acá están algunos amigos de la cátedra:

$$\begin{aligned}K &= \lambda x y.x \\ S &= \lambda f g x.f x (g x) \\ I &= \lambda x.x \\ \Delta &= \lambda x.x x\end{aligned}$$