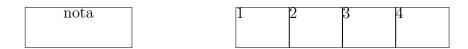
Apellido y Nombre: email:



Lenguajes y Compiladores

Parcial 3

18/6/2010

- 1. Enuncie y pruebe el Teorema de Sustitución para el cálculo lambda (para D_{∞}).
- 2. Considere la siguiente expresión

```
\begin{array}{ccc} \mathbf{letrec} & \mathbf{f} \equiv \lambda & \mathbf{x}.\mathbf{if} & \mathbf{x} = 0 \\ & \mathbf{then} \ \mathbf{true} \\ & \mathbf{else} \ \mathbf{if} & \mathbf{x} = 1 \\ & \mathbf{then} \ \mathbf{false} \\ & \mathbf{else} \ \mathbf{f} \ (\mathbf{x} - 2) \\ \mathbf{in} & \mathbf{f} \ (\langle \Delta \Delta, 2 \rangle.1) \end{array}
```

- a) Evalúe (\Rightarrow) la expresión en la modalidad eager.
- b) Elimine la notación letrec considerando el programa en el lenguaje aplicativo normal.
- c) Evalúe (\Rightarrow) la expresión resultante del inciso anterior en el lenguaje aplicativo normal.
- 3. Considere la siguiente expresión

```
(\lambda \times .sumcase \times of (\lambda \vee .\langle \vee, \vee \vee \rangle, \lambda \vee .error) (@ 0 5)
```

- a) Calcule la semántica denotacional en el lenguaje eager.
- b) Calcule la semántica denotacional en el lenguaje normal.
- 4. Considere los siguientes programas del lenguaje Small. Para cada uno de ellos analice el programa y la definición de la semántica de Small y proponga directamente la ecuación para $[prog]\eta\kappa f\sigma$. Justificar sin calcular.

```
a) let var x;
      const c = 3
in x := val x + c

b) let var x;
      const c = 3
in x := val x + c;
    !(val x)

c) let var x;
    proc p(x) {
    !x
    }
in p(val x)
```