

Práctica 3: PCF tipado

1. Tipar los siguientes términos (si es posible).

- a) $\lambda x : \text{nat}.x$
- b) $\lambda x : \text{nat} \Rightarrow \text{nat}.x$
- c) $\lambda x : \text{nat} \Rightarrow \text{nat}.\lambda y : \text{nat} \Rightarrow \text{nat}.xy$
- d) $\lambda x : \text{nat} \Rightarrow \text{nat}.\lambda y : \text{nat}.xy$
- e) $\text{let } x : \text{nat} \Rightarrow \text{nat} = \lambda x : \text{nat}.x + 1 \text{ in } x^3$
- f) $\text{let } x : \text{nat} \Rightarrow \text{nat} = \lambda x : \text{nat}.x + 1 \text{ in } x(\lambda x.x)$
- g) $\text{fix } f : A.\lambda n : B.\lambda m : C.\text{ifz } m \text{ then } 1 \text{ else } n \times fn(m - 1)$
para algún A, B y C . ¿De qué función se trata?

2. Extender PCF con booleanos `true`, `false` e `if – then – else`.

- a) Dar la gramática, semántica operacional y reglas de tipado.
- b) Tipar:
 - 1) $\text{if } (\lambda x : A.x)\text{true then } 2 \text{ else } 1$, para algún A .
 - 2) $(\lambda x : \text{bool}.\text{if } x \text{ then } (\lambda x : \text{nat}.x) \text{ else } 0)\text{false}$
 - 3) $\lambda x : \text{bool}.\text{if } x \text{ then false else true}$

3. Extender PCF con constructores para pares: (t, u) , $\text{fst}(t, u)$ y $\text{snd}(t, u)$.

- a) Dar la gramática, semántica operacional y reglas de tipado.
- b) Escribir una suma que reciba un par y otra que reciba dos argumentos, y tiparlos.
- c) Definir una función `curry` que tome una función que espera un par, y dos argumentos, y ejecute esa función con esos argumentos y tiparla.
- d) Definir una función `uncurry` que tome una función que espera dos argumentos y un par, y ejecute esa función con los elementos del par, y tiparla.

4. Extender PCF con constructores de listas de números:

<code>nil</code>	(lista vacía)
<code>cons tu</code>	(agregar el elemento t a la lista u)
<code>ifnil t then u else v</code>	(if lista vacía)
<code>hd t</code>	(head)
<code>tl t</code>	(tail)

- a) Dar la gramática, semántica operacional y reglas de tipado.
- b) Escribir una función que sume los elementos de una lista de 4 elementos. Tiparla.
- c) Tipar la función del ítem anterior aplicada a

`cons 0(cons (($\lambda x : \text{nat}.x + 1$)2)(cons 2(cons 1 nil)))`

y dar su traza de reducción en CBN.