

Práctica 3: Inferencia de tipos + Polimorfismo

Sección I: Inferencia de tipos

1. Dar el tipo principal, y todos los tipos, de los siguientes términos.

a) $\lambda x.x$

b) $(\lambda x.x + 1)2$

c) $\lambda x.\lambda y.xy$

d) $\text{let } x = \lambda x.x + 1 \text{ in } x3$

e) $\text{let } x = \lambda x.x + 1 \text{ in } x(\lambda x.x)$

f) $\mu f.\lambda n.\lambda m.\text{ifz } m \text{ then } 1 \text{ else } n \times fn(m - 1)$

g) $\lambda x.\lambda y.x(y + 1) + 2$

h) $\text{let } x = \lambda x.x \text{ in } xx$

i) $(\lambda x.x)(\lambda x.x)$

j) $\lambda x.xx$

2. Extender los algoritmos de Hindley y Robinson para la extensión de PCF con booleanos de la práctica anterior.
3. Extender los algoritmos de Hindley y Robinson para la extensión de PCF con pares de la práctica anterior.
4. Extender los algoritmos de Hindley y Robinson para la extensión de PCF con listas de la práctica anterior.
5. Extender PCF con árboles binarios, extendiendo: la gramática, la semántica operacional, las reglas de tipado y el algoritmo de inferencia de tipos. Los constructores deberían ser:

L_n : Una hoja con contenido n

$Tntu$: Un árbol binario, con subárboles t y u y n en su nodo

$\text{ifLeaf } t \text{ then } u \text{ else } v$ Testea si es una hoja

$\text{content } t$: Contenido del nodo

$\text{left } t$: Subárbol izquierdo (reduce a t si es una hoja)

$\text{right } t$: Subárbol derecho (reduce a t si es una hoja)

6. En el caso general de un lenguaje sin variables ligadas, reemplazamos las primeras tres reglas del algoritmo de Robinson por las siguientes dos:

- Si una ecuación tiene la forma

$$f(U_1, \dots, U_n) = f(V_1, \dots, V_n)$$

reemplazarla por las ecuaciones $U_1 = V_1, \dots, U_n = V_n$.

- Si una ecuación tiene la forma

$$f(U_1, \dots, U_n) = g(V_1, \dots, V_n)$$

done f y g son símbolos distintos, devolver error.

- En un lenguaje formado por un símbolo $+$ que toma dos argumentos y las constantes enteras:
 - ¿Tiene solución la ecuación $(2 + (3 + X)) = (X + (Y + 2))$?
 - ¿Tiene solución la ecuación $X + 2 = 4$?
 - ¿Cuál es la diferencia entre las dos ecuaciones anteriores y lo que aprendimos en la escuela?
- Usar la semántica operacional de PCF para definir otra noción de solución que coincida con lo que vimos en la escuela, y mostrar que $X + 2 = 4$ tiene solución en ese caso.

Sección II: Polimorfismo

- Dar el tipo más general a cada uno de los siguientes términos (sin usar el algoritmo de inferencia).
 - $\lambda x.x$
 - $(\lambda x.x + 1)2$
 - $\lambda x.\lambda y.xy$
 - $\text{let } x = \lambda x.x + 1 \text{ in } x3$
 - $\text{let } x = \lambda x.x + 1 \text{ in } x(\lambda x.x)$
 - $\mu f.\lambda n.\lambda m.\text{ifz } m \text{ then } 1 \text{ else } n \times fn(m - 1)$
 - $\lambda x.\lambda y.x(y + 1) + 2$
 - $\text{let } x = \lambda x.x \text{ in } xx$
 - $(\lambda x.x)(\lambda x.x)$
 - $\lambda x.xx$
- Extender PCF polimórfico con listas, como en la práctica 3, sólo que ahora usaremos el tipo `list` con un tipo como argumento. Así, escribimos `list[nat]` al tipo de listas de naturales, `list[nat \Rightarrow nat]` al tipo de las listas de funciones de naturales en naturales, y `list[list[nat]]` al tipo de las listas que como elementos tienen listas de naturales.
Escribir la función `map` que toma una función f y una lista, y aplica f a cada elemento de la lista. Dar un tipo para `map`
- System F es un sistema de tipos polimórfico más general que el visto en clase. En System F el polimorfismo puede aparecer en cualquier lugar, sin restricciones (a excepción de la restricción de la introducción del cuantificador universal, que implica derivar el tipo $\forall X.X$). En consecuencia, la inferencia se vuelve indecidible.

La gramática de tipos de System F es la siguiente:

$$A ::= X \mid \text{nat} \mid A \Rightarrow A \mid \forall X.A$$

- a)* Dar las reglas de tipado de System F.
- b)* Dar los tipos más generales, en System F, para los términos del punto 1.