Parseo y Generación de Código – 2$^{\text{do}}$ semestre 2016
Licenciatura en Informática con Orientación en Desarrollo de Software
Universidad Nacional de Quilmes

Práctica 7
Tipos y unificación

Ejercicio 1. En el cálculo-λ extendido con booleanos (True, False, $\mathsf{if}\ a\ \mathsf{then}\ b\ \mathsf{else}\ c$), números naturales ($0$, $\mathsf{Succ}(n)$, $\mathsf{IsZero}(n)$, $\mathsf{Pred}(n)$), y las siguientes reglas para pares:

$$\frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash \langle M, N \rangle : \tau \times \sigma}$$

$$\frac{\Gamma \vdash M : \tau \times \sigma}{\Gamma \vdash \pi_1(M) : \tau} \qquad \frac{\Gamma \vdash M : \tau \times \sigma}{\Gamma \vdash \pi_2(M) : \sigma}$$

dar derivaciones de tipos para los siguientes juicios, determinando en cada caso el valor del tipo α. Para este ejercicio se recomienda determinarlo manualmente, es decir, sin utilizar un algoritmo de inferencia de tipos. Recordar que el constructor de funciones asocia a derecha, es decir "$\tau \to \sigma \to \rho$" equivale a "$\tau \to (\sigma \to \rho)$".

1. $f : \mathsf{Nat} \to \mathsf{Bool} \vdash f\,0 : \alpha$
2. $g : \mathsf{Nat} \to \mathsf{Bool} \vdash \mathsf{if}\ g\,0\ \mathsf{then}\ g\,\mathsf{Succ}(0)\ \mathsf{else}\ g\,\mathsf{Succ}(\mathsf{Succ}(0)) : \alpha$
3. $f : \mathsf{Nat} \to \mathsf{Nat} \vdash \lambda x : \mathsf{Nat}.\ \mathsf{IsZero}(f\,\mathsf{Succ}(x)) : \alpha$
4. $f : (\mathsf{Nat} \to \mathsf{Bool}) \times (\mathsf{Nat} \to \mathsf{Nat}) \vdash \lambda x : \mathsf{Nat}.\ \langle \pi_1(f)\,x, \pi_2(f)\,x \rangle : \alpha$
5. $\vdash \lambda f : \tau \to \tau.\ f\,(f\,(\lambda x : \mathsf{Bool}.\ x)) : \alpha$
6. $\vdash \lambda x : \tau \times (\sigma \times \rho).\ \langle \langle \pi_1(x), \pi_1(\pi_2(x)) \rangle, \pi_2(\pi_2(x)) \rangle : \alpha$
7. $\vdash \lambda x : \tau \to \sigma \to \rho.\ \lambda y : \sigma.\ \lambda z : \tau.\ xzy : \alpha$
8. $\vdash \lambda x : (\tau \times \sigma) \to \rho.\ \lambda y : \tau.\ \lambda z : \sigma.\ x\,\langle y, z \rangle : \alpha$
9. $\vdash \lambda x : \tau \to \sigma \to \rho.\ \lambda y : \tau \to \sigma.\ \lambda z : \tau.\ x\,z\,(y\,z) : \alpha$
10. $\vdash \lambda x : \tau \to \sigma \to \rho.\ \lambda y : \tau \times \sigma.\ x\,\pi_1(y)\,\pi_2(y) : \alpha$

Ejercicio 2. Justificar por qué los siguientes términos no son tipables (en ningún contexto posible):

1. $\mathsf{if}\ x\ \mathsf{then}\ 0\ \mathsf{else}\ x$
2. $\langle x\,0, x\,\mathsf{True} \rangle$
3. $x\,\mathsf{Succ}(x)$

4. if $f(f\,0)$ then True else False
5. if True then x else $\pi_1(x)$
6. $x\,x$

Ejercicio 3. Decidir en cada caso si existe el unificador más general para los siguientes problemas de unificación, y en tal caso exhibirlo. *Convención:* las letras f, g, h representan constructores, las letras a, b, c representan constantes (constructores de aridad 0), y las letras x, y, z representan variables.

1. $x \stackrel{?}{=} x$
2. $f(y) \stackrel{?}{=} x$
3. $f(x) \stackrel{?}{=} x$
4. $f(x, a) \stackrel{?}{=} f(y, z)$
5. $f(x, a) \stackrel{?}{=} f(x, g(z))$
6. $f(x, y, z) \stackrel{?}{=} f(g(y), g(z), g(a))$
7. $f(x, y, z) \stackrel{?}{=} f(g(y), g(z), g(x))$
8. $f(x, a) \stackrel{?}{=} f(y, x)$
9. $f(x, x) \stackrel{?}{=} f(g(y), b)$
10. $f(g(y), y) \stackrel{?}{=} f(x, g(x))$

Ejercicio 4. Decidir en cada caso si existe el unificador más general (mgu) para los siguientes problemas de unificación de tipos. En este caso los constructores de tipos son Nat (naturales), Bool (booleanos), $\tau \to \sigma$ (funciones), $\tau \times \sigma$ (pares), $[\tau]$ (listas). Las variables de tipos se denotan con las letras a, b, c, etc.

1. $S_1 = \mathsf{mgu}(\{a \times b \stackrel{?}{=} \mathsf{Nat} \times \mathsf{Bool}\})$
   Calcular $S_1(a)$.
2. $S_2 = \mathsf{mgu}(\{a \to b \to [a] \stackrel{?}{=} \mathsf{Nat} \to [a] \to c\})$
   Calcular $S_2(a \to b \to [a])$ y $S(\mathsf{Nat} \to [a] \to c)$ y verificar que sean iguales.
3. $S_3 = \mathsf{mgu}(a \to [a] \stackrel{?}{=} b \to b)$
4. $S_4 = \mathsf{mgu}(a \to [a] \stackrel{?}{=} b \times b)$
5. $S_5 = \mathsf{mgu}(a \to b \to a \stackrel{?}{=} \mathsf{Bool} \to c)$

6. $S_6 = \mathsf{mgu}(a \to (a \times [a]) \to [[a]] \stackrel{?}{=} [b] \to c)$

7. $S_7 = \mathsf{mgu}(c \to c \stackrel{?}{=} (a \to b) \to [a] \to [b])$

8. $S_8 = \mathsf{mgu}((\mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Bool}) \to c \stackrel{?}{=} (a \to b) \to [a] \to [b])$

9. $S_9 = \mathsf{mgu}((a \to b \to c) \to d \stackrel{?}{=} (\mathsf{Nat} \to e) \to (b \to a \to c))$

Ejercicio 5. Aplicar el algoritmo de inferencia de tipos sobre los siguientes términos, obteniendo así el juicio de tipado más general posible que les asigna un tipo:

1. $\langle \mathsf{Succ}(x), x \rangle$
2. $\mathsf{if}\ x\,y\ \mathsf{then}\ x\,0\ \mathsf{else}\ x\,\mathsf{Succ}(0)$
3. $\langle f\,x\,\mathsf{True}, f\,0\,(y\,x) \rangle$
4. $\lambda x.\, \mathsf{if}\ x\ \mathsf{then}\ \mathsf{False}\ \mathsf{else}\ \mathsf{True}$
5. $\lambda x.\, \lambda y.\, x$
6. $\lambda x.\, \lambda y.\, \lambda z.\, xz(yz)$
7. $\lambda f.\, \lambda g.\, \lambda x.\, f(gx)$

Ejercicio 6. Aplicar el algoritmo de inferencia de tipos sobre los términos del Ejercicio 2 y comprobar que el algoritmo falla en cada uno de esos casos.