

Análisis de Lenguajes de Programación

Trabajo Práctico 3

Nicolás Felipe Del Piano

Observaciones:

Tabla de precedencia

Operador Precedencia

Abs	1
Let	1
Suc	2
Fst	2
Snd	2
Rec	2
App	3
As	4

Los operadores que faltan, o bien llevan paréntesis obligatoriamente o no los necesitan.

Los ejercicios no citados en este informe se encuentran resueltos en el código impreso al final.

Ejercicio 1.

Dar una derivación de tipo para el término S definido en Prelude.lam.

$$\frac{\frac{\frac{\frac{\frac{\frac{\frac{\Gamma' \vdash x: B \rightarrow B \rightarrow B}{\Gamma' \vdash (x \ z): B \rightarrow B} \text{ T - VAR} \quad \frac{\frac{\frac{\frac{\Gamma' \vdash z: B}{\Gamma' \vdash (y \ z): B} \text{ T - VAR} \quad \frac{\frac{\frac{\Gamma' \vdash y: B \rightarrow B}{\Gamma' \vdash (y \ z): B} \text{ T - VAR} \quad \frac{\frac{\Gamma' \vdash z: B}{\Gamma' \vdash (y \ z): B} \text{ T - VAR}}{\Gamma' \vdash (y \ z): B} \text{ T - APP}}{\Gamma' \vdash (x \ z): B \rightarrow B} \text{ T - APP}}{\Gamma, x: B \rightarrow B \rightarrow B, y: B \rightarrow B, z: B \vdash (x \ z) (y \ z): B} \text{ T - APP}}{\Gamma, x: B \rightarrow B \rightarrow B, y: B \rightarrow B \vdash \lambda z: B. (x \ z) (y \ z): B \rightarrow B} \text{ T - ABS}}{\Gamma, x: B \rightarrow B \rightarrow B \vdash \lambda y: B \rightarrow B. \lambda z: B. (x \ z) (y \ z): (B \rightarrow B) \rightarrow B \rightarrow B} \text{ T - ABS}}{\Gamma \vdash \lambda x: B \rightarrow B \rightarrow B. \lambda y: B \rightarrow B. \lambda z: B. (x \ z) (y \ z): (B \rightarrow B \rightarrow B) \rightarrow (B \rightarrow B) \rightarrow B \rightarrow B} \text{ T - ABS}$$

Donde $\Gamma' = \Gamma, x: B \rightarrow B \rightarrow B, y: B \rightarrow B, z: B$

Ejercicio 2.

La función infer retorna un valor de tipo Either String Type, porque su trabajo es inferir los tipos de los términos y devolver algo de tipo Type, pero si existiese un término atascado o una forma normal que no sea un valor, debería dar retornar un error. Ese es el lugar que ocupa String, en el tipo Either String Type.

(>>=) funciona de la siguiente manera. Primero recibe un argumento de tipo Either String Type. Luego una función que tiene tipo Type → Either String Type. Si el primer argumento es "Left s", aplica Left a s (s:String). En cambio, si es del tipo "Right t", devuelve la función que recibió como segundo argumento aplicada a t (t:Type). Esto nos permite hacer una secuencia de operaciones (>>=), donde se van aplicando sucesivamente los resultados obtenidos.

Ejercicio 5.

De una derivación de tipo para el término (let z = ((λ x : B. x) as B → B) in z) as B → B. Verificar que este término tipa correctamente en el intérprete.

$$\begin{array}{c}
\frac{}{\Gamma, x \vdash x:B} T - VAR \\
\frac{}{\Gamma \vdash \lambda x:B .x : B \rightarrow B} T - ABS \\
\frac{}{\Gamma \vdash ((\lambda x:B .x) \text{ as } B \rightarrow B): B \rightarrow B} T - ASCRIBE \quad \frac{}{\Gamma, z:B \rightarrow B \vdash z:B \rightarrow B} T - VAR \\
\frac{}{\Gamma \vdash \text{let } z = ((\lambda x:B .x) \text{ as } B \rightarrow B) \text{ in } z : B \rightarrow B} T - LET \\
\frac{}{\Gamma \vdash \text{let } z = ((\lambda x:B .x) \text{ as } B \rightarrow B) \text{ in } z) \text{ as } B \rightarrow B: B \rightarrow B} T - ASCRIBE
\end{array}$$

Verificación:

```
ST> :t (let z = ((\x:B.x) as B->B) in z) as B -> B
B -> B
```

Ejercicio 7.

Extender la relación de evaluación para contemplar los pares. A la hora de reducir (t₁,t₂), reducir primero por completo t₁ y luego reducir t₂.

$$\frac{t_1 \rightarrow t'_1}{(t_1, t_2) \rightarrow (t'_1, t_2)} E - PAIR1$$

$$\frac{v \rightarrow t'_2}{(v, t_2) \rightarrow (v, t'_2)} E - PAIR2$$

$$\frac{t \rightarrow t'}{fst\ t \rightarrow fst\ t'} E - FST1$$

$$fst(v_1, v_2) \rightarrow v_1 \quad E - FST2$$

$$\frac{t \rightarrow t'}{snd\ t \rightarrow snd\ t'} \quad E - SND1$$

$$snd(v_1, v_2) \rightarrow v_2 \quad E - SND2$$

Ejercicio 9.

De una derivación de tipo para el término $fst\ (unit\ as\ Unit, \lambda x : (B, B).\ snd\ x)$. Verificar que este término tipa correctamente en el intérprete.

$$\frac{\frac{\frac{\overline{\Gamma \vdash unit:Unit}}{\Gamma \vdash unit\ as\ Unit:Unit} \quad T - UNIT \quad \frac{\frac{\overline{\Gamma, x:(B,B) \vdash x:(B,B)}}{\Gamma, x:(B,B) \vdash snd\ x:B} \quad T - VAR \quad T - SND}{\Gamma \vdash \lambda x : (B,B).\ snd\ x : (B,B) \rightarrow B} \quad T - ABS}{\Gamma \vdash (unit\ as\ Unit, \lambda x : (B,B).\ snd\ x) : (Unit, (B,B) \rightarrow B)} \quad T - PAIR}{\Gamma \vdash fst\ (unit\ as\ Unit, \lambda x : (B,B).\ snd\ x) : Unit} \quad T - FST$$

Verificación :

```
ST> :t (fst (unit as Unit, \x:(B,B).snd x))
Unit
```

Ejercicio 11.

Definir la función Ack dentro del cálculo.

```
def A = (\n1:Nat.\n2:Nat. R (\n:Nat.suc n) (\x:Nat->Nat.
\y:Nat.\z:Nat. R (x (suc 0)) (\w:Nat.\y:Nat.(x w)) z) n1 n2)
```

Verificación:

```
ST> A
\x:Nat.\y:Nat.(R (\z:Nat.suc z) (\z:Nat -> Nat.\a:Nat.\b:Nat.R
z (suc 0)) (\c:Nat.\d:Nat.z c) b) x) y
```