

Gramáticas a tener en cuenta:

- Términos

$$M ::= x \mid \lambda x : \sigma. M \mid M M \mid \text{true} \mid \text{false} \mid \text{if } M \text{ then } M \text{ else } M \mid 0 \mid \text{succ}(M) \mid \text{pred}(M) \mid \text{isZero}(M)$$

Donde la letra  $x$  representa un *nombre de variable* arbitrario. Tales nombres se toman de un conjunto infinito dado  $\mathfrak{X} = \{w, w_1, w_2, \dots, x, x_1, x_2, \dots, y, y_1, y_2, \dots, z, z_1, z_2, \dots\}$

- Tipos

$$\sigma ::= \text{Bool} \mid \text{Nat} \mid \sigma \rightarrow \sigma$$

**Ejercicio 1** ★ [ej 1]

Determinar qué expresiones son sintácticamente válidas (es decir, pueden ser generadas con las gramáticas presentadas) y determinar a qué categoría pertenecen (expresiones de términos o expresiones de tipos):

- a)  $x$  ✓
- b)  $x x$  ✓ (  $M M$  )
- c)  $M$  ✗
- d)  $M M$  ✗
- e)  $\text{true false}$  ✓ (  $M M$  )
- f)  $\text{true succ(false true)}$  ✓ (  $M M, \text{succ}(M)$  )
- g)  $\lambda x. \text{isZero}(x)$  ✗ (  $\text{falta el tipo de } x$  )
- h)  $\lambda x : \sigma. \text{succ}(x)$  ✗

- i)  $\lambda x : \text{Bool}. \text{succ}(x)$  ✓ (  $\lambda x : \sigma. M, \text{succ}(M)$  )
- j)  $\lambda x : \text{if true then Bool else Nat}. x$  ✗
- k)  $\sigma$  ✗
- l)  $\text{Bool}$  ✗
- m)  $\text{Bool} \rightarrow \text{Bool}$  ✗
- n)  $\text{Bool} \rightarrow \text{Bool} \rightarrow \text{Nat}$  ✗
- ñ)  $(\text{Bool} \rightarrow \text{Bool}) \rightarrow \text{Nat}$  ✗
- o)  $\text{succ true}$  ✗
- p)  $\lambda x : \text{Bool}. \text{if } 0 \text{ then true else } 0 \text{ succ(true)}$  ✓

Gramáticas a tener en cuenta:

- Términos

$$M ::= x \mid \lambda x : \sigma. M \mid M M \mid \text{true} \mid \text{false} \mid \text{if } M \text{ then } M \text{ else } M \mid 0 \mid \text{succ}(M) \mid \text{pred}(M) \mid \text{isZero}(M)$$

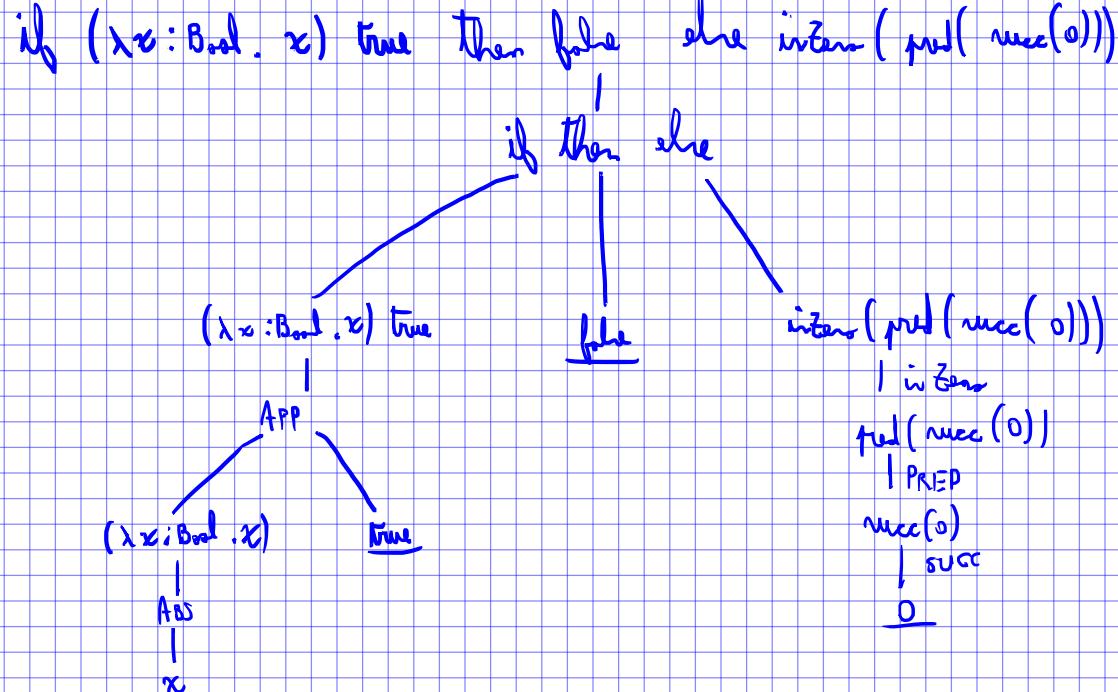
Donde la letra  $x$  representa un *nombre de variable* arbitrario. Tales nombres se toman de un conjunto infinito dado  $\mathfrak{X} = \{w, w_1, w_2, \dots, x, x_1, x_2, \dots, y, y_1, y_2, \dots, z, z_1, z_2, \dots\}$

- Tipos

$$\sigma ::= \text{Bool} \mid \text{Nat} \mid \sigma \rightarrow \sigma$$

**Ejercicio 2** [ej 2]

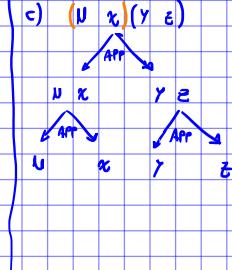
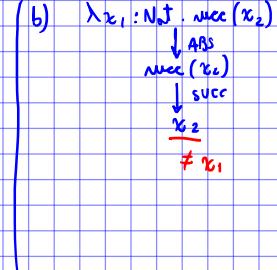
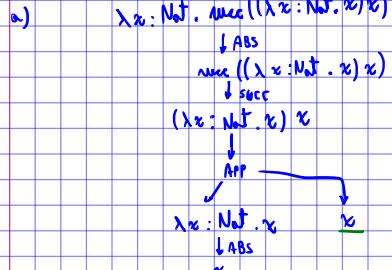
Mostrar un término que utilice al menos una vez **todas** las reglas de generación de la gramática y exhibir su **árbol sintáctico**.



**Ejercicio 3** ★ [ej 3] ??

??

- a) Marcar las ocurrencias del término  $x$  como subtérmino en  $\lambda x : \text{Nat}. \text{succ}((\lambda x : \text{Nat}. x) x)$ .
- b) ¿Ocurre  $x_1$  como subtérmino en  $\lambda x_1 : \text{Nat}. \text{succ}(x_2)$ ? no
- c) ¿Ocurre  $x$  ( $y z$ ) como subtérmino en  $u x (y z)$ ? no



Ejercicio 4 ★

Para los siguientes términos:

- $u \ x \ (y \ z) \ (\lambda v: \text{Bool}. \ v \ y)$
- $(\lambda x: \text{Bool} \rightarrow \text{Nat} \rightarrow \text{Bool}. \ \lambda y: \text{Bool} \rightarrow \text{Nat}. \ \lambda z: \text{Bool}. \ x \ z \ (y \ z)) \ u \ v \ w$
- $w \ (\lambda x: \text{Bool} \rightarrow \text{Nat} \rightarrow \text{Bool}. \ \lambda y: \text{Bool} \rightarrow \text{Nat}. \ \lambda z: \text{Bool}. \ x \ z \ (y \ z)) \ u \ v$

Se pide:

I Insertar todos los paréntesis de acuerdo a la convención usual.

II Dibujar el árbol sintáctico de cada una de las expresiones.

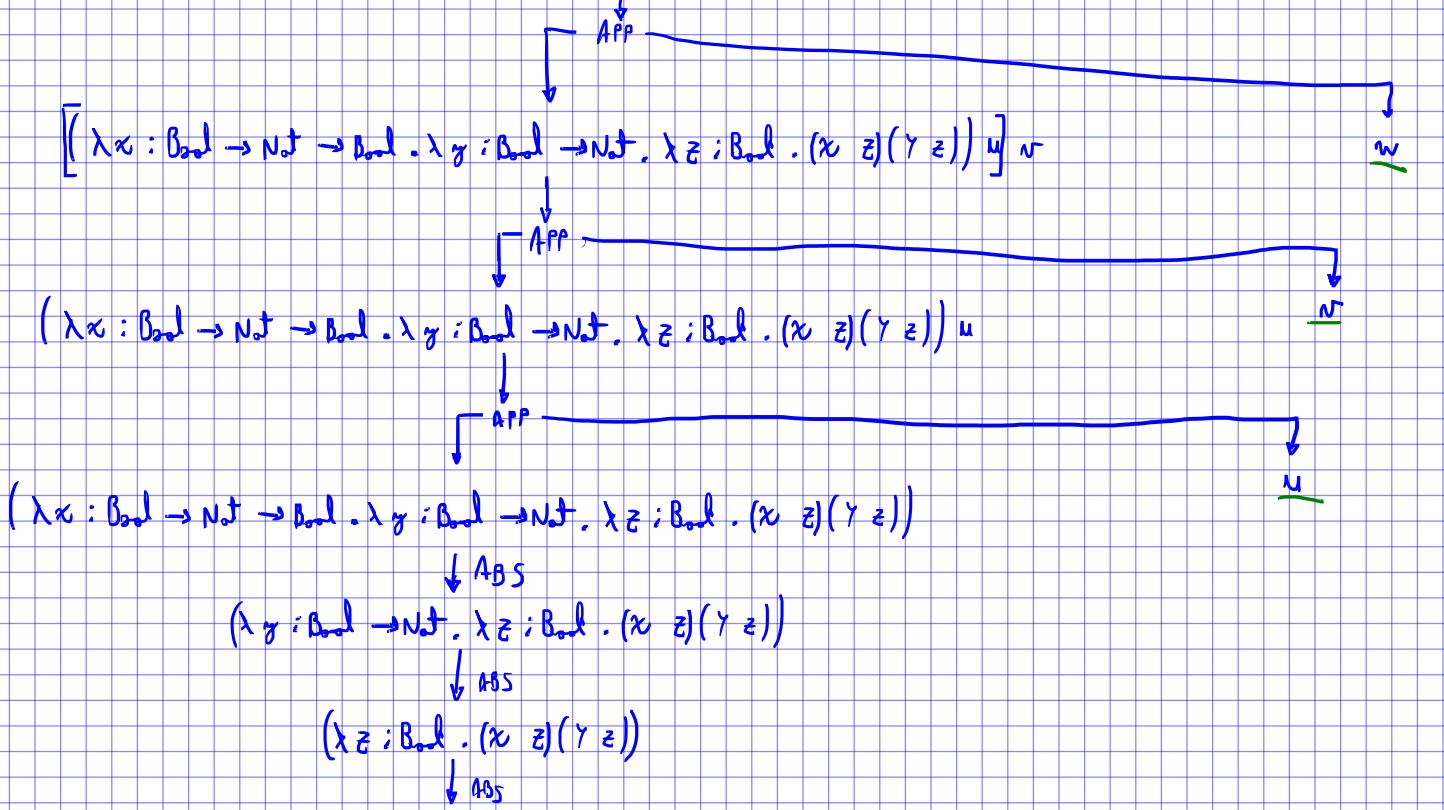
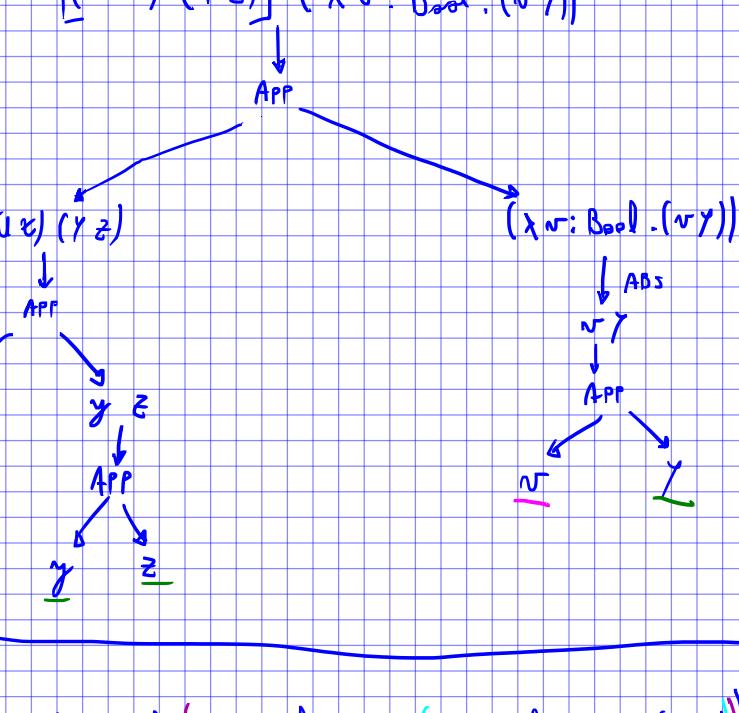
III Indicar en el árbol cuáles ocurrencias de variables aparecen ligadas y cuáles libres.

IV ¿En cuál de los términos anteriores ocurre la siguiente expresión como subtérmino?

$(\lambda x: \text{Bool} \rightarrow \text{Nat} \rightarrow \text{Bool}. \ \lambda y: \text{Bool} \rightarrow \text{Nat}. \ \lambda z: \text{Bool}. \ x \ z \ (y \ z)) \ u$

• libre

• ligada



$(\lambda x: \text{Bool} \rightarrow \text{Nat} \rightarrow \text{Bool}. \ \lambda y: \text{Bool} \rightarrow \text{Nat}. \ \lambda z: \text{Bool}. \ (x \ z) \ (y \ z)) \ u$

$\downarrow \text{ABS}$

$(\lambda y: \text{Bool} \rightarrow \text{Nat}. \ \lambda z: \text{Bool}. \ (x \ z) \ (y \ z))$

$\downarrow \text{ABS}$

$(\lambda z: \text{Bool}. \ (x \ z) \ (y \ z))$

$\downarrow \text{ABS}$

$(x \ z) \ (y \ z)$

$\downarrow \text{APP}$

$\boxed{x \ z} \ \boxed{y \ z}$

$\downarrow \text{APP}$

$x \ \boxed{z} \ y \ \boxed{z}$

$\downarrow \text{APP}$

$x \ \boxed{z} \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

$x \ z \ y \ z$

$\downarrow \text{APP}$

**Ejercicio 5 (Derivaciones ★) 4 5)**

Demostrar – o explicar por qué no puede demostrarse – cada uno de los siguientes juicios de tipado.

- a)  $\emptyset \triangleright \text{if true then } 0 \text{ else succ}(0) : \text{Nat}$
- b)  $\{x : \text{Nat}, y : \text{Bool}\} \triangleright \text{if true then false else } (\lambda z : \text{Bool}. z) \text{ true} : \text{Bool}$
- c)  $\emptyset \triangleright \text{if } \lambda x : \text{Bool}. x \text{ then } 0 \text{ else succ}(0) : \text{Nat}$
- d)  $\{x : \text{Bool} \rightarrow \text{Nat}, y : \text{Bool}\} \triangleright x y : \text{Nat}$

$$\text{a)} \frac{\emptyset \triangleright \text{true} : \text{Bool}, \emptyset \triangleright 0 : \text{Nat}, \emptyset \triangleright \text{succ}(0) : \text{Nat}}{\emptyset \triangleright \text{if true then } 0 \text{ else succ}(0) : \text{Nat}} \quad T\text{-if}$$

$$\frac{\Gamma, [z : \text{Bool}] \triangleright z : \text{Bool} \quad T\text{-MS}}{\Gamma \triangleright (\lambda z : \text{Bool}. z) : \text{Y} \rightarrow \Gamma, \Gamma \triangleright \text{true} : \text{Bool}} \quad T\text{-APP}$$

$$\text{b)} \frac{\Gamma \triangleright \text{true} : \text{Bool}, \Gamma \triangleright \text{false} : \text{Bool}, \Gamma \triangleright (\lambda z : \text{Bool}. z) \text{ true} : \Gamma}{\Gamma = \{x : \text{Nat}, y : \text{Bool}\} \triangleright \text{if true then false else } (\lambda z : \text{Bool}. z) \text{ true} : \text{Bool}} \quad T\text{-if}$$

$$\begin{aligned} \Gamma &= \text{Bool} \\ Y &= \text{Bool} \end{aligned}$$

$$\text{c)} \frac{\cancel{x \triangleright \lambda x : \text{Bool}. x : \text{Bool}} \quad \emptyset \triangleright 0 : \text{Nat} \quad \emptyset \triangleright \text{succ}(0) : \text{Nat}}{\emptyset \triangleright \text{if } \lambda x : \text{Bool}. x \text{ then } 0 \text{ else succ}(0) : \text{Nat}} \quad T\text{-if}$$

d)

$$\frac{\begin{array}{c} \frac{T\text{-APP}}{\Gamma \triangleright x : \text{B} \rightarrow \text{N}} \\ \frac{T\text{-VAR}}{x : \text{B} \rightarrow \text{N} \in \Gamma} \end{array} \quad \begin{array}{c} \Gamma = \{x : \text{B} \rightarrow \text{N}, y : \text{B}\} \triangleright x y : \text{N} \\ \frac{T\text{-VAr}}{y : \text{B} \in \Gamma} \end{array}}{\checkmark}$$

**Ejercicio 7 ★**

Determinar qué tipos representan  $\sigma$  y  $\tau$  en cada uno de los siguientes juicios de tipado. Si hay más de una solución, o si no hay ninguna, indicarlo.

- a)  $\{x : \sigma\} \triangleright \text{isZero}(\text{succ}(x)) : \tau, \Gamma = N, Y = \beta$
- b)  $\emptyset \triangleright (\lambda x : \sigma. x)(\lambda y : \text{Bool}. 0) : \sigma$
- c)  $\{y : \tau\} \triangleright \text{if } (\lambda x : \sigma. x) \text{ then } y \text{ else succ}(0) : \sigma$
- d)  $\{x : \sigma\} \triangleright x y : \tau, \tau = t_1 \rightarrow \tau, Y = t_2$
- e)  $\{x : \sigma, y : \tau\} \triangleright x y : \tau, \tau = \tau \rightarrow \tau, Y = t_1$
- f)  $\{x : \sigma\} \triangleright x \text{ true} : \tau, \tau = B \rightarrow \gamma, Y = t_1$
- g)  $\{x : \sigma\} \triangleright x \text{ true} : \sigma, \forall \alpha \exists \alpha$
- h)  $\{x : \sigma\} \triangleright x x : \tau, \forall \alpha \exists \alpha$

$$\text{b)} \frac{\emptyset \triangleright (\lambda x : \tau. x)(\lambda y : \text{B}. 0) : \tau}{\frac{\emptyset \triangleright \lambda x : \tau. x : (\text{B} \rightarrow \text{N}) \rightarrow \sigma}{\frac{\emptyset \triangleright x : \text{N}}{\Gamma = \text{B} \rightarrow \text{N}}}} \quad T\text{-APP}$$

$$\text{c)} \frac{T\text{-IF}}{\frac{\Gamma \triangleright (\lambda x : \tau. x) : B \cancel{\times \text{succ}}}{\frac{\Gamma \triangleright y : \text{N} \quad T\text{-VAR} \quad T\text{-SUC}}{\frac{\Gamma \triangleright \text{succ}(0) : \text{N}}{\Gamma = \sigma = N}}}}} \quad \Rightarrow \sigma = N$$

**Ejercicio 10 ★**

Sean  $\sigma, \tau, \rho$  tipos. Según la definición de sustitución, calcular:

- a)  $(\lambda y : \sigma. x)(\lambda x : \tau. x)\{x \leftarrow (\lambda y : \rho. x y)\}$
- b)  $(y(\lambda v : \sigma. x v))\{x \leftarrow (\lambda y : \tau. v y)\}$

Renombrar variables en ambos términos para no cambiar el significado del término.

$$\text{a)} \quad \lambda y : \sigma. (\lambda z : \rho. x z)(\lambda w : \tau. x w)$$

$$\text{b)} \quad y(\lambda u : \sigma. (\lambda v : \tau. w v) w)$$

**Ejercicio 11 (Valores) ★**

Dado el conjunto de valores visto en clase:

$$V := \lambda x : \sigma. M \mid \text{true} \mid \text{false} \mid 0 \mid \text{succ}(V)$$

Determinar si cada una de las siguientes expresiones es o no un valor:

- a)  $(\lambda x : \text{Bool}. x) \text{ true} \quad \times \text{APP}$
- b)  $(\lambda x : \text{Bool}. 2) \quad \checkmark \text{ ABS}$
- c)  $(\lambda x : \text{Bool}. \text{pred}(2)) \quad \checkmark \text{ ABS}$
- d)  $(\lambda y : \text{Nat}. (\lambda x : \text{Bool}. \text{pred}(2)) \text{ true}) \quad \checkmark \text{ APP}$
- e)  $x \ x$
- f)  $\text{succ}(\text{succ}(0)) \quad \times \text{succ}$

### Ejercicio 12 (Programa, Forma Normal) ★

Para el siguiente ejercicio, considerar el cálculo **sin** la regla  $\text{pred}(0) \rightarrow 0$

Un **programa** es un término que tipo en el contexto vacío (es decir, no puede contener variables libres).  
 Para cada una de las siguientes expresiones

a) Determinar si puede ser considerada un **programa**.

b) Si vale (a), ¿Cuál es el resultado de su evaluación? Determinar si se trata de una forma normal, y en caso de serlo, si es un **valor** o un **error**.

i)  $(\lambda x: \text{Bool}. x) \text{ true}$

II)  $\lambda x: \text{Nat}. \text{ pred}(\text{succ}(x))$

III)  $\lambda x: \text{Nat}. \text{ pred}(\text{succ}(y))$

IV)  $(\lambda x: \text{Bool}. \text{ pred}(\text{isZero}(x))) \text{ true}$

V)  $(\lambda f: \text{Nat} \rightarrow \text{Bool}. f \ 0) (\lambda x: \text{Nat}. \text{ isZero}(x))$

VI)  $(\lambda f: \text{Nat} \rightarrow \text{Bool}. x) (\lambda x: \text{Nat}. \text{ isZero}(x))$

VII)  $(\lambda f: \text{Nat} \rightarrow \text{Bool}. f \ \text{pred}(0)) (\lambda x: \text{Nat}. \text{ isZero}(x))$

VIII)  $\text{fix } (\lambda y: \text{Nat}. \text{ succ}(y))$

1)  $\frac{(\lambda x: \text{B}. x) \text{ true}}{\text{true}}$  E-APPABs  $\rightsquigarrow$  no es programa, no es FN

2)  $\lambda x: \text{N}. \text{ pred}(\text{succ}(x))$   $\rightsquigarrow$  es programa, no es FN

3)  $\lambda z: \text{N}. \text{ pred}(\text{succ}(z))$   $\rightsquigarrow$  no es programa, no es FN

4)  $\frac{(\lambda z: \text{B}. \text{ pred}(\text{isZero}(z))) \text{ true}}{\text{pred}(\text{isZero}(\text{true}))}$  E-APPABs  $\rightsquigarrow$  no es programa porque no tiene tipo

5)  $\frac{(\lambda f: \text{N} \rightarrow \text{B}. \text{ f } 0) (\lambda x: \text{N}. \text{ isZero}(x))}{(\lambda x: \text{N}. \text{ isZero}(x)) \ 0}$  E-APPABs  $\rightsquigarrow$  no es programa, no es FN  
 $\frac{\text{isZero}(0)}{\text{true}}$  E-INTRO<sub>isZero</sub>  
 true

6)  $\frac{(\lambda f: \text{N} \rightarrow \text{B}. x) (\lambda x: \text{N} \vdash \text{ isZero}(x))}{x}$  E-APPABs  $\rightsquigarrow$  no es programa, no es FN

7)  $\frac{(\lambda f: \text{N} \rightarrow \text{B}. \text{ f } 0) (\lambda x: \text{N}. \text{ isZero}(x))}{(\lambda x: \text{N}. \text{ isZero}(x)) \ \text{pred}(0)}$  E-APPABs  $\rightsquigarrow$  no es programa porque no tiene tipo  
 X error

8)  $\frac{\text{fix } (\lambda y: \text{N}. \text{ succ}(y))}{\text{succ}(\text{fix } (\lambda y: \text{N}. \text{ succ}(y)))}$  E-FIX<sub>BETA</sub>  $\rightsquigarrow$  no es programa, no es FN  
 $\vdots$   
 00

## Ejercicio 17 ★

Este ejercicio extiende el Cálculo Lambda tipado con listas. Comenzamos ampliando el conjunto de tipos:

$$\sigma ::= \dots | [\sigma]$$

donde  $[\sigma]$  representa el tipo de las listas cuyas componentes son de tipo  $\sigma$ . El conjunto de términos ahora incluye:

$$M, N, O ::= \dots | []_\sigma | M :: N | \text{case } M \text{ of } \{[] \rightsquigarrow N | h :: t \rightsquigarrow O\} | \text{foldr } M \text{ base} \rightsquigarrow N; \text{rec}(h, r) \rightsquigarrow O$$

donde

- $[]_\sigma$  es la lista vacía cuyos elementos son de tipo  $\sigma$ ;
- $M :: N$  agrega  $M$  a la lista  $N$ ;
- $\text{case } M \text{ of } \{[] \rightsquigarrow N | h :: t \rightsquigarrow O\}$  es el observador de listas. Por su parte, los nombres de variables que se indiquen luego del  $|$  ( $h$  y  $t$  en este caso) son variables que pueden aparecer libres en  $O$  y deberán ligarse con la cabeza y cola de la lista respectivamente;
- $\text{foldr } M \text{ base} \rightsquigarrow N; \text{rec}(h, r) \rightsquigarrow O$  es el operador de recursión estructural (no currificado). Los nombres de variables indicados entre parentesis ( $h$  y  $r$  en este caso) son variables que pueden aparecer libres en  $O$  y deberán ser ligadas con la cabeza y el resultado de la recursión respectivamente.

Por ejemplo,

- $\text{case } 0 :: \text{succ}(0) :: []_{\text{Nat}} \text{ of } \{[] \rightsquigarrow \text{false} | x :: xs \rightsquigarrow \text{isZero}(x)\} \Rightarrow \text{true}$
- $\text{foldr } \underline{1} :: 2 :: 3 :: (\lambda x : [\text{Nat}]. x) []_{\text{Nat}} \text{ base} \rightsquigarrow 0; \text{rec}(\text{head}, \text{rec}) \rightsquigarrow \text{head} + \text{rec} \Rightarrow 6$

a) Mostrar el árbol sintáctico para los dos ejemplos dados.

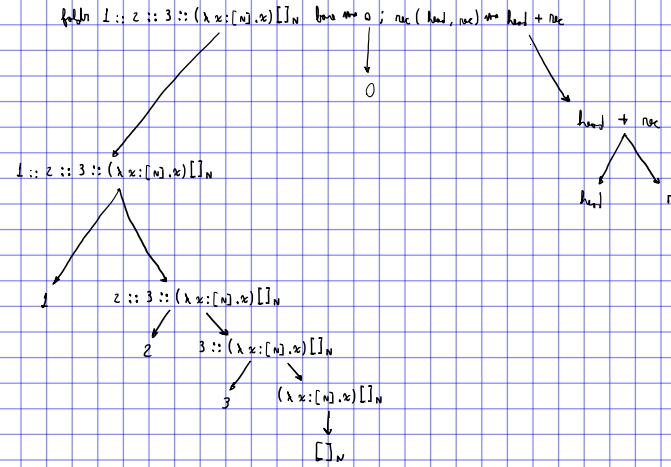
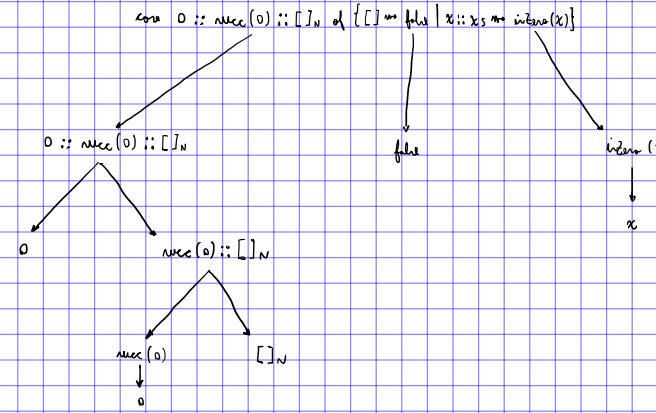
b) Agregar reglas de tipado para las nuevas expresiones.

c) Demostrar el siguiente juicio de tipado (recomendación: marcar variables libres y ligadas en el término antes de comenzar).

$$\{x : \text{Bool}, y : [\text{Bool}]\} \triangleright \text{foldr } x :: x :: y \text{ base} \rightsquigarrow y; \text{rec}(y, x) \rightsquigarrow \text{if } y \text{ then } x \text{ else } []_{\text{Bool}} : [\text{Bool}]$$

d) Mostrar cómo se extiende el conjunto de valores. Estos deben reflejar la forma de las listas que un programa podría devolver.

e) Agregar los axiomas y reglas de reducción asociados a las nuevas expresiones.



$$\frac{}{\Gamma \vdash []_r : [\tau]} T-\text{VACIA}$$

$$\frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : [\tau]}{\Gamma \vdash M :: N : [\tau]} T-\text{ADD}$$

$$\frac{\Gamma \vdash M : [\gamma] \quad \Gamma \vdash N : \tau \quad \Gamma \cup \{h : \gamma, t : [\gamma]\} \triangleright O : \tau}{\Gamma \vdash \text{let } M \text{ of } [] \text{ in } N \mid h :: t \triangleright O : \tau} T-\text{CASE}$$

$$\frac{\Gamma \vdash M : [\gamma] \quad \Gamma \vdash N : \tau \quad \Gamma \cup \{h : \gamma, r : \tau; r\} \triangleright O : \tau}{\Gamma \vdash \text{let } M \text{ base } \mapsto N; \text{ rec}(h, r) \mapsto O : \tau} T-\text{fUn}$$

$$\begin{array}{c} \frac{T-\text{fUn}}{\Gamma = \{x : B, y : [B]\} \triangleright \text{let } x :: x :: y \text{ in } \text{base } \mapsto y; \text{ rec}(h, r) \mapsto \text{if } h \text{ then } r \text{ else } []_B : [\alpha]} \\ \frac{\begin{array}{c} \frac{\Gamma \vdash x : x :: y : [\alpha]}{\Gamma \vdash y : [\alpha]} T-\text{VAR} & \frac{\Gamma \vdash y : [\alpha]}{y : [\alpha] \in \Gamma} T-\text{APP} \\ \frac{\Gamma \vdash x : B \quad \Gamma \vdash x : B}{x : B \in \Gamma} T-\text{VAR} & \frac{\Gamma \vdash x : B \quad \Gamma \vdash y : [\alpha]}{x : B \in \Gamma \quad \Gamma \vdash y : [\alpha]} T-\text{APP} \end{array}}{\begin{array}{c} \checkmark \\ \checkmark \end{array}} \\ \frac{\begin{array}{c} \frac{\Gamma \vdash h : B \quad \Gamma \vdash r : [\alpha]}{\Gamma \vdash h : B \in \Gamma \quad \Gamma \vdash r : [\alpha]} T-\text{VAR} & \frac{\Gamma \vdash h : B \quad \Gamma \vdash r : [\alpha]}{\Gamma \vdash h : B \in \Gamma \quad \Gamma \vdash r : [\alpha]} T-\text{VAR} \\ \frac{\Gamma \vdash h : B \quad \Gamma \vdash r : [\alpha]}{h : B \in \Gamma} T-\text{VAR} & \frac{\Gamma \vdash h : B \quad \Gamma \vdash r : [\alpha]}{r : [\alpha] \in \Gamma} T-\text{VACIA} \end{array}}{\begin{array}{c} \checkmark \\ \checkmark \end{array}} \end{array}$$

$V := \dots \mid [M_1, \dots, M_k] \mid []$

$$\frac{N \rightarrow N'}{M : N \rightarrow M : N'} E-\text{ADD}$$

$$\frac{M \rightarrow M'}{\text{let } M \text{ of } [] \text{ in } N \mid h :: t \triangleright O} E-\text{CASE}_1$$

$$\frac{\text{let } [] \text{ of } [] \text{ in } N \mid h :: t \triangleright O \rightarrow N}{E-\text{CASE}_2}$$

$$\frac{\Gamma \vdash M_1 : \tau \quad \Gamma \vdash M_2 : [\tau]}{\text{let } M_1 : M_2 \text{ of } [] \text{ in } N \mid h :: t \triangleright O \{h \leftarrow M_1, t \leftarrow M_2\}} E-\text{CASE}_3$$

$$\frac{M \rightarrow M'}{\text{let } M \text{ base } \mapsto N; \text{ rec}(h, r) \mapsto O \rightarrow \text{let } M' \text{ base } \mapsto N; \text{ rec}(h, r) \mapsto O} E-\text{fUn}_1$$

$$\frac{\text{let } [] \text{ base } \mapsto N; \text{ rec}(h, r) \mapsto O \rightarrow N}{E-\text{fUn}_2}$$

$$\frac{\Gamma \vdash M_1 : \tau \quad \Gamma \vdash M_2 : [\tau]}{\text{let } M_1 : M_2 \text{ base } \mapsto N; \text{ rec}(h, r) \mapsto O \{h \leftarrow M_1, r \leftarrow \text{let } M_2 \text{ base } \mapsto N; \text{ rec}(h, r) \mapsto O\}} E-\text{fUn}_3$$

$$\frac{\text{case } O :: \text{succ}(0) :: []_N \text{ of } [] \mapsto \text{fold} \mid x :: x :: \text{succ}(x) \text{ on } \text{succ}(0)}{\frac{\text{succ}(0) \text{ type}}{E-\text{case}_3}}$$

$$\begin{array}{c} \frac{\text{fold} \mid 1 :: 2 :: 3 :: []_N \text{ base } \mapsto 0; \text{ rec}(\text{head}, \text{rec}) \mapsto \text{head} + \text{rec}}{E-\text{fUn}_1, E-\text{add}, E-\text{add}, E-\text{add}, E-\text{APPAB}} \\ \frac{\text{fold} \mid 1 :: 2 :: 3 :: []_N \text{ base } \mapsto 0; \text{ rec}(\text{head}, \text{rec}) \mapsto \text{head} + \text{rec}}{E-\text{fUn}_2} \\ \frac{1 + \text{fold} \mid 2 :: 3 :: []_N \text{ base } \mapsto 0; \text{ rec}(\text{head}, \text{rec}) \mapsto \text{head} + \text{rec}}{E-\text{fUn}_3} \\ \frac{1 + 2 + \text{fold} \mid 3 :: []_N \text{ base } \mapsto 0; \text{ rec}(\text{head}, \text{rec}) \mapsto \text{head} + \text{rec}}{E-\text{fUn}_4} \\ \frac{3 + \text{fold} \mid 3 :: []_N \text{ base } \mapsto 0; \text{ rec}(\text{head}, \text{rec}) \mapsto \text{head} + \text{rec}}{E-\text{fUn}_5} \\ \frac{3 + 3 + \text{fold} \mid []_N \text{ base } \mapsto 0; \text{ rec}(\text{head}, \text{rec}) \mapsto \text{head} + \text{rec}}{E-\text{fUn}_6} \\ \frac{6 + \text{fold} \mid []_N \text{ base } \mapsto 0; \text{ rec}(\text{head}, \text{rec}) \mapsto \text{head} + \text{rec}}{E-\text{fUn}_7} \\ \frac{6 + 0}{6} E-\text{fUn}_8 \end{array}$$

### Ejercicio 18 ★

A partir de la extensión del ejercicio 17, definir una nueva extensión que incorpore expresiones de la forma  $\text{map}(M, N)$ , donde  $N$  es una lista y  $M$  una función que se aplicará a cada uno de los elementos de  $N$ .

Importante: tener en cuenta las anotaciones de tipos al definir las reglas de tipado y semántica.

$$\frac{\Gamma \vdash M : \gamma \rightarrow \tau \quad \Gamma \vdash N : [\gamma]}{\Gamma \vdash \text{map}(M, N) : [\tau]} T-\text{MAP}$$

$$\frac{M_2 \rightarrow M_2' \quad \Gamma \vdash M_2 : \tau}{\text{map}(M_1, M_2) \rightarrow M_2'} E-\text{map}_1$$

$$\frac{\Gamma \vdash M : [] \rightarrow \tau \quad \Gamma \vdash N : [\tau]}{\text{map}(M, N) \rightarrow []} E-\text{map}_2$$

$$\frac{\Gamma \vdash M : h :: t \rightarrow \tau \quad \Gamma \vdash N : [\tau]}{\text{map}(M, N) \rightarrow M \cdot h :: \text{map}(M, t)} E-\text{map}_3$$

## Ejercicio 19 ★

A partir de la extensión del ejercicio 17, agregaremos términos para representar listas por comprensión, con un selector y una guarda, de la siguiente manera:  $[M \mid x \leftarrow S, P]$ , donde  $x$  es el nombre de una variable que puede aparecer libre en los términos  $M$  y  $P$ . La semántica es análoga a la de Haskell: para cada valor de la lista representada por el término  $S$ , se sustituye  $x$  en  $P$  y, de resultar verdadero, se agrega  $M$  con  $x$  sustituido al resultado. Definir las reglas de tipado, el conjunto de valores y las reglas de semántica para esta extensión.

$$\frac{\Gamma' = \Gamma \cup \{x:\tau\} \quad \Gamma \vdash M : \sigma \quad \Gamma \vdash S : [\tau] \quad \Gamma \vdash P : \text{Bool} \quad \Gamma \vdash t : \sigma}{\Gamma \vdash [M \mid x \leftarrow S, P] : [\sigma]} \text{ E-COMP}_1$$

$$\frac{}{[\Gamma_1 \mid x \leftarrow []] \rightarrow []} \text{ E-COMP}_2$$

$$\frac{M_3 \rightarrow M'_3 \quad \Gamma \vdash h : \sigma \quad \Gamma \vdash t : [\sigma]}{[\Gamma_1 \mid x \leftarrow h \#_1 t, M_2] \rightarrow [\Gamma_1 \mid x \leftarrow h \#_1 \text{if } M_2[x \leftarrow h] \text{ then } M_2[x \leftarrow h] :: [\Gamma_1 \mid x \leftarrow t, M_2] \text{ else } [\Gamma_1 \mid x \leftarrow t, M_2]]} \text{ E-COMP}_3$$

## Ejercicio 20 ★

La aplicación parcial sobre funciones currificadas es una de las ventajas de los lenguajes funcionales, como el cálculo lambda tipado. Sin embargo, el mecanismo del cálculo lambda (que se repite en la mayoría de los lenguajes funcionales como Haskell) es limitado, ya que la aplicación parcial debe hacerse siempre en el orden de los argumentos. Por ejemplo, si tenemos la función potencia, podemos usarla con aplicación parcial para definir la función cuadrado, si su primer parámetro es el exponente, o la función dosALA si su primer parámetro es la base, pero no podemos hacer ambas cosas con la misma función potencia.

Para solucionar este problema introduciremos el cálculo  $\mu$ , que es igual al cálculo lambda en todo, excepto en que el mecanismo para construir funciones  $(\lambda x.M)$  y el mecanismo para aplicarlas  $(M N)$  serán sustituidos por un nuevo mecanismo de construcción  $(\mu x_1, \dots, x_n.M)$  y de aplicación  $(M \#_i N)$ . Estos cambios también introducen un cambio en el sistema de tipos: en lugar de tener  $\sigma \rightarrow \tau$  tendremos  $\{\sigma_1, \dots, \sigma_n\} \rightarrow \tau$ . Notar que  $\{\sigma_1, \dots, \sigma_n\}$  no es un nuevo tipo, sino sólo una parte del nuevo tipo para funciones.

La sintaxis del cálculo  $\mu$  y su conjunto de tipos, entonces, serán los siguientes:

$$M, N ::= \dots \mid \mu x_1 : \sigma_1, \dots, x_n : \sigma_n.M \mid M \#_i N \quad \sigma_1 \dots \sigma_n, \tau ::= \dots \mid \{\sigma_1, \dots, \sigma_n\} \rightarrow \tau$$

El término  $\mu x_1 : \sigma_1, \dots, x_n : \sigma_n.M$  sirve para construir una nueva función de  $n$  parámetros ordenados y el operador  $\#_i$  sirve para aplicar el  $i$ -ésimo parámetro. Notar que si la cantidad de parámetros de una función es mayor a 1, al aplicarla se obtiene una nueva función con un parámetro menos, pero si la cantidad de parámetros es exactamente 1, al aplicarla se obtiene su valor de retorno. Notar además que el orden de los tipos de los argumentos es importante: por ejemplo,  $\{nat, nat, bool\} \rightarrow nat$  y  $\{bool, nat, nat\} \rightarrow nat$  no son el mismo tipo.

- Introducir las reglas de tipado para la extensión propuesta.
- Dar formalmente la extensión de los valores e introducir las reglas de semántica para la extensión propuesta.
- Escribir las construcciones básicas del cálculo lambda ( $\lambda$  y aplicación) como macros del cálculo  $\mu$  para mostrar que este último puede emularlo.

$$\frac{\Gamma \cup \{x_1 : \tau_1, \dots, x_k : \tau_k\} \vdash M : \gamma}{\Gamma \vdash (\mu x_1 : \tau_1, \dots, x_k : \tau_k.M) : \{\tau_1, \dots, \tau_k\} \rightarrow \gamma} \text{ T-ABS}$$

$$\frac{\Gamma \vdash M : \{\tau_1, \dots, \tau_k\} \rightarrow \gamma \quad k > 1 \quad \Gamma \vdash N : \tau_i}{\Gamma \vdash M \#_i N : \{\tau_1, \dots, \tau_{i-1}, \tau_{i+1}, \dots, \tau_k\} \rightarrow \gamma} \text{ T-APP}_1$$

$$\frac{\Gamma \vdash M : \{\tau\} \rightarrow \gamma \quad \Gamma \vdash N : \tau}{\Gamma \vdash M \#_1 N : \gamma} \text{ T-APP}_2$$

$$V ::= \dots \mid \mu x_1 : \tau_1, \dots, x_k : \tau_k, M$$

$$\frac{M_1 \rightarrow M'_1 \quad M_2 \#_1 M_1 \rightarrow M'_2 \#_1 M_2}{M_2 \#_1 M_1 \rightarrow M'_2 \#_1 M'_1} \text{ E-APP}_1$$

$$\frac{M \rightarrow M'}{V \#_1 M \rightarrow V \#_1 M'} \text{ E-APP}_2$$

$$\frac{(\mu x_1 : \tau_1, \dots, x_k : \tau_k, M) \#_1 V \rightarrow (\mu x_1 : \tau_1, \dots, x_k : \tau_k, M[x_1 \leftarrow V])}{(\mu x : \sigma) \#_1 V \rightarrow M[x \leftarrow V]} \text{ E-APPABS}_1$$

$$\frac{K > 1}{(\mu x : \sigma) \#_1 V \rightarrow M[x \leftarrow V]} \text{ E-APPABS}_2$$

MACROS

$$(\lambda x . M) = (\mu x_1 : \sigma_1 . M)$$

$$M \#_1 N = M \#_1 N$$

### Ejercicio 23 ★

Se desea extender el cálculo lambda tipado agregando *unión de funciones*. Para ello, extenderemos el conjunto de términos y el de tipos de la siguiente manera:

$$M_1 \dots M_k ::= \dots | [(M_1, \dots, M_k)] \quad \sigma ::= \dots | \text{Union}(\sigma_1, \dots, \sigma_k)_\tau$$

Cada  $M_i$  dentro de “[ ]” es una función con distinto dominio del resto pero con la misma imagen.

En el tipo  $\text{Union}(\sigma_1, \dots, \sigma_k)_\tau$ , cada  $\sigma_i$  representa el tipo del dominio de  $M_i$  (la función en la posición  $i$ ), y  $\tau$  el tipo de la imagen de todas las funciones.

Al aplicarse esta unión sobre un valor de tipo  $\sigma$ , el término reduce utilizando la función de esta unión cuyo tipo para el dominio sea  $\sigma$ . Es decir, aplicando la función que corresponda según el dominio.

Por ejemplo, sea

$$l \stackrel{\text{def}}{=} [(\lambda x : \text{Nat}. x + 2, \lambda x : \text{Bool}. \text{if } x \text{ then } 4 \text{ else } 3, \lambda f : \text{Bool} \rightarrow \text{Nat}. (f \text{ true}) + 3)]$$

$l$  tiene tipo  $\text{Union}(\text{Nat}, \text{Bool}, \text{Bool} \rightarrow \text{Nat})_{\text{Nat}}$ .

Luego,  $l (\lambda b : \text{Bool}. \text{if } b \text{ then } 3 \text{ else } 4) \rightarrow ((\lambda b : \text{Bool}. \text{if } b \text{ then } 3 \text{ else } 4) \text{ true}) + 3 \rightarrow 6$ .

Se pide:

- Extender las reglas de tipado acordeamente.
- Mostrar el árbol de derivación para el juicio:  $\{y : \text{Nat}\} \triangleright [(\lambda x : \text{Bool}. y, \lambda x : \text{Nat}. x)] y : \text{Nat}$ .
- Indicar cómo se modifica el conjunto de valores. Justificar.
- Modificar o extender las reglas de semántica operacional para la extensión propuesta.

$$\frac{\Gamma \triangleright M_i : \tau_i \rightarrow \gamma \quad \forall i \in \{1, \dots, k\} \quad \forall i, j \in \{1, \dots, k\}, i \neq j \Rightarrow \tau_i \neq \tau_j}{\Gamma \triangleright [(M_1, \dots, M_k)] : \text{Union}(\sigma_1, \dots, \sigma_k)_\gamma} \text{ T-UNION}$$

$$\frac{\Gamma \triangleright M : \text{Union}(\sigma_1, \dots, \sigma_k)_\gamma \quad \exists i \in \{1, \dots, k\} \triangleright N : \tau_i}{\Gamma \triangleright M N : \gamma} \text{ T-APP-UNION}$$

$$\frac{\begin{array}{c} \checkmark \quad \checkmark \\ \vdots \quad \vdots \\ \checkmark \quad \checkmark \\ \vdots \quad \vdots \end{array}}{\Gamma \triangleright \lambda x : B. y : B \rightarrow B} \quad \frac{\Gamma \triangleright \lambda x : N. x : N \rightarrow N}{\Gamma \triangleright \lambda x : N. x : N \rightarrow N} \quad \frac{\checkmark}{B \neq N} \quad \frac{\checkmark}{x(N) \in \Gamma} \quad \frac{\checkmark}{\Gamma \triangleright y : \text{Nat}} \text{ T-VAR}$$

$$\frac{\Gamma \triangleright \lambda x : B. y, \lambda x : N. x : \text{Union}(\sigma_1, \dots, \sigma_k)_N}{\Gamma \vdash \{y : \text{Nat}\} \triangleright [(\lambda x : B. y, \lambda x : N. x)] y : N} \text{ T-APP-UNION}$$

$$V = \dots | [(M_1, \dots, M_k)]$$

$$\frac{\text{Dado } i \in \{1, \dots, k\} \quad \Gamma \triangleright M_i : \tau_i \rightarrow \gamma \quad \Gamma \triangleright V : \top}{[(M_1, \dots, M_k)] V \rightarrow M_i V} \text{ E-UNION}$$

## Ejercicio 25 ★

Se desea extender el cálculo lambda tipado para tener un mayor control sobre el proceso de reducción. Para esto, se introducen expresiones capaces de detener la reducción de un término, o de continuar una reducción que estaba detenida.

El conjunto de tipos será:  $\sigma ::= \dots | \det(\sigma)$  donde  $\det(\sigma)$  es el tipo de los términos que resultan de detener la reducción de términos de tipo  $\sigma$ .

El conjunto de términos será:  $M ::= \dots | \detener(M) | \continuar(M)$

El comportamiento de estas expresiones es el siguiente: sea  $M$  un término tipable cualquiera,  $\detener(M)$  detiene la reducción de  $M$ . Es decir, no reduce por más que  $M$  pueda reducirse. Por otro lado, si  $N$  es un término detenido,  $\continuar(N)$  reanuda la reducción de  $N$ . Por ejemplo:

$\continuar((\lambda x : \det(\text{Nat}).x) \detener(\text{pred}(\text{succ}(0)))) \rightarrow \continuar(\detener(\text{pred}(\text{succ}(0)))) \rightarrow \text{pred}(\text{succ}(0)) \rightarrow 0$ .

Además, las funciones que esperan argumentos detenidos, pueden recibir argumentos del tipo correspondiente sin detener. En ese caso, en lugar de reducir el argumento hasta obtener un valor, lo detienen. Esto permite definir funciones que toman parámetros por nombre (call-by-name). Por ejemplo:

```
(\lambda x : \det(\text{Nat}).\text{if true then } \continuar(x) \text{ else } 0) \text{ succ}(\text{pred}(\text{succ}(0))) \rightarrow
(\lambda x : \det(\text{Nat}).\text{if true then } \continuar(x) \text{ else } 0) \detener(\text{succ}(\text{pred}(\text{succ}(0)))) \rightarrow
\text{if true then } \continuar(\detener(\text{succ}(\text{pred}(\text{succ}(0))))) \text{ else } 0 \rightarrow
\continuar(\detener(\text{succ}(\text{pred}(\text{succ}(0))))) \rightarrow \text{succ}(\text{pred}(\text{succ}(0))) \rightarrow \text{succ}(0).
```

a) Introducir las reglas de tipado para la extensión propuesta.

b) Exhibir la derivación de tipado para el juicio:

$\{y : \text{Bool}\} \triangleright (\lambda x : \det(\text{Bool}).\text{if } y \text{ then } \continuar(x) \text{ else false}) \text{ isZero}(0) : \text{Bool}$ .

c) Indicar formalmente cómo se modifica el conjunto de valores, y dar la semántica operacional de a un paso para la extensión propuesta. Notar que puede ser necesario modificar alguna de las reglas preexistentes.

$$\frac{\Gamma \vdash M : \tau}{\Gamma \vdash \detener(M) : \det(\tau)} \text{ T-DET} \quad \frac{\Gamma \vdash M : \det(\tau)}{\Gamma \vdash \continuar(M) : \tau} \text{ T-CONT} \quad \frac{\Gamma \vdash M : \det(\tau) \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M N : \tau} \text{ T-APPDET}$$

$$\begin{array}{c} \Gamma = \{y : \text{Bool}\} \triangleright (\lambda x : \det(\text{Bool}).\text{if } y \text{ then } \continuar(x) \text{ else false}) \text{ isZero}(0) : \text{Bool} \\ \hline \begin{array}{l} \text{T-ABS: } \frac{\Gamma \vdash \lambda x : \det(\text{Bool}).\text{if } y \text{ then } \continuar(x) \text{ else false} : \det(\text{Bool}) \rightarrow \text{Bool}}{\Gamma' \vdash \lambda x : \det(\text{Bool}).\text{if } y \text{ then } \continuar(x) \text{ else false} : \det(\text{Bool}) \rightarrow \text{B}} \\ \text{T-IF: } \frac{\Gamma \vdash \text{if } y \text{ then } \continuar(x) \text{ else false} : \det(\text{Bool}) \rightarrow \text{B}}{\Gamma' \vdash y : \text{B}} \\ \text{T-VAR: } \frac{\Gamma \vdash y : \text{B}}{\Gamma' \vdash y : \text{B}} \end{array} \quad \begin{array}{l} \text{T-APPDET: } \frac{\Gamma \vdash \text{isZero}(0) : \text{B} \quad \Gamma \vdash 0 : \text{N}}{\Gamma \vdash 0 \mid N : \text{Bool}} \\ \text{T-else: } \frac{\Gamma \vdash 0 \mid N : \text{Bool}}{\Gamma \vdash \text{else} : \text{Bool}} \end{array} \end{array}$$

$V ::= \dots | \detener(M)$

$$\begin{array}{c} \frac{M \rightarrow M'}{\detener(M) \rightarrow \detener(M')} \text{ E-CONT}_1 \\ \frac{\Gamma \vdash V : \det(\tau) \rightarrow \gamma \quad \Gamma \vdash M : \tau}{V M \rightarrow V \detener(M)} \text{ E-APPDET} \\ \frac{\detener(\detener(M)) \rightarrow M}{\detener(\detener(M)) \rightarrow M} \text{ E-CONT}_2 \\ \frac{\Gamma \vdash V : \tau \rightarrow \gamma \quad \Gamma \vdash M : \tau \quad M \rightarrow M'}{V M \rightarrow V M'} \text{ E-APP}_2 \text{ (multifijo)} \\ \frac{\Gamma \vdash V : \tau}{(\lambda x : \tau. M) V \rightarrow M[x \rightarrow V]} \text{ E-APPABs (multifijo)} \end{array}$$