

Clase Práctica 6 : Cálculo λ

Tomás Felipe Melli

July 11, 2025

Índice

1	Primera extensión :	pares	2
2	Segunda extensión :	uniones disjuntas	3
3	Tercera extensión :	árboles binarios	4
3.1	árbol binario		4
3.2	árboles bis		5

1 Primera extensión : pares

Sintaxis: tipos y términos

Nos dan la siguiente sintaxis con sus tipos y términos:

$$\sigma, \tau ::= \dots \mid \sigma \times \tau$$

$$M, N ::= \dots \mid \langle M, N \rangle \mid \pi_1(M) \mid \pi_2(M)$$

Y nos piden definir como macro la función $\text{curry}_{\sigma, \tau, \psi}$ que sirve para currificar funciones que reciben pares como argumentos.

$$\text{curry}_{\sigma, \tau, \psi} \stackrel{\text{def}}{=} \lambda f : \sigma \times \tau \rightarrow \psi. \lambda x : \sigma. \lambda y : \tau. f \langle x, y \rangle$$

Reglas de tipado

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

$$\frac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \pi_1(M) : \sigma} \text{T-}\pi_1$$

$$\frac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \pi_2(M) : \tau} \text{T-}\pi_2$$

Y por tanto, extender el conjunto de valores y determinar las nuevas reglas de semántica. Las reglas de semántica son : o de **congruencia** o de **cómputo**. Las de congruencia sirven para decir qué parte de un término reduce primero cuando tenemos varias cosas que podrían reducirse. O sea, **qué reduce primero**. Por ello, estas tienen como premisa que un subtérmino reduce y nos dicen cómo meter la reducción dentro del término más grande. Las de cómputo nos dicen **a qué reduce**

Extendiendo valores

$$V ::= \dots \mid \langle V, V \rangle$$

Reglas de semántica

$$\frac{}{\pi_1(\langle V_1, V_2 \rangle) \rightarrow V_1} \text{E-}\pi_1 - 1 \quad \text{cómputo}$$

$$\frac{M \rightarrow M'}{\pi_1(M) \rightarrow \pi_1(M')} \text{E-}\pi_1 - 2 \quad \text{congruencia}$$

$$\frac{}{\pi_2(\langle V_1, V_2 \rangle) \rightarrow V_2} \text{E-}\pi_2 - 1 \quad \text{cómputo}$$

$$\frac{M \rightarrow M'}{\pi_2(M) \rightarrow \pi_2(M')} \text{E-}\pi_2 - 2 \quad \text{congruencia}$$

$$\frac{M \rightarrow M'}{\langle M, N \rangle \rightarrow \langle M', N \rangle} \text{E-Par1} \quad \text{congruencia}$$

$$\frac{N \rightarrow N'}{\langle V, N \rangle \rightarrow \langle V, N' \rangle} \text{E-Par2} \quad \text{congruencia}$$

Qué problema introduce la siguiente regla ?

$$\pi_1(\langle M, N \rangle) \rightarrow M$$

El problema que introduce es que podemos aplicar o esa regla o la que reduce $M \rightarrow M'$ nos queda $\pi_1(\langle M', N \rangle)$ con lo cuál estaríamos rompiendo el determinismo. Como se ve en este ejemplo

$$\pi_1(\langle (\lambda x. \text{Bool}. x) \text{ true}, \text{false} \rangle) \xrightarrow{\text{MAL}} \begin{matrix} (\lambda x. \text{Bool}. x) \text{ true} \\ \times \\ \pi_1(\langle \text{true}, \text{false} \rangle) \end{matrix}$$

Nos piden a continuación verificar el siguiente juicio de tipado :

$\frac{}{x : \text{Nat} \vdash x : \text{Nat}} \text{T-VAR}$	$\frac{}{x : \text{Nat} \vdash \text{true} : \text{Bool}} \text{T-TRUE}$
$\frac{x : \text{Nat} \vdash x : \text{Nat} \quad x : \text{Nat} \vdash \text{true} : \text{Bool}}{x : \text{Nat} \vdash \langle x, \text{true} \rangle : \text{Nat} \times \text{Bool}} \text{T-PAR}$	
$\frac{}{\emptyset \vdash \lambda x : \text{Nat}. \langle x, \text{true} \rangle : \text{Nat} \rightarrow \text{Nat} \times \text{Bool}} \text{T-ABS}$	
$\frac{}{\emptyset \vdash 0 : \text{Nat}} \text{T-ZERO}$	
$\frac{\emptyset \vdash \lambda x : \text{Nat}. \langle x, \text{true} \rangle : \text{Nat} \rightarrow \text{Nat} \times \text{Bool} \quad \emptyset \vdash 0 : \text{Nat}}{\emptyset \vdash (\lambda x : \text{Nat}. \langle x, \text{true} \rangle) 0 : \text{Nat} \times \text{Bool}} \text{T-APP}$	
$\frac{\emptyset \vdash (\lambda x : \text{Nat}. \langle x, \text{true} \rangle) 0 : \text{Nat} \times \text{Bool}}{\emptyset \vdash \pi_1((\lambda x : \text{Nat}. \langle x, \text{true} \rangle) 0) : \text{Nat}} \text{T-}\pi_1$	

Y reducir

$\pi_1((\lambda x : \text{Nat}. \langle x, \text{true} \rangle) 0) \xrightarrow[\beta]{E-\pi_1,} \pi_1(\langle 0, \text{true} \rangle) \xrightarrow{E-\pi_1 \text{ PAR}} 0$
$\pi_1(\pi_1(\lambda x : \text{Nat}, \langle \langle x, x \rangle, \text{true} \rangle) 0) \xrightarrow[\beta]{E-\pi_1, E-\pi_1,} \pi_1(\pi_1(\langle \langle 0, 0 \rangle, \text{true} \rangle))$

2 Segunda extensión : uniones disjuntas

Sintaxis: tipos y términos

$$\sigma ::= \dots \mid \sigma + \sigma$$

$$M ::= \dots \mid \text{left}_\sigma(M) \mid \text{right}_\sigma(M) \mid \text{case } M \text{ of } \text{left}(x) \rightsquigarrow M ; \text{right}(y) \rightsquigarrow M$$

Donde $\sigma + \tau$ representa el tipo de la unión disjunta entre sigma y tau, similar al tipo `Either σ τ` en Haskell. Donde $\text{left}_\sigma(M)$ y $\text{right}_\sigma(M)$ inyectan un valor en la unión. Donde `case M of left(x) \rightsquigarrow N ; right(y) \rightsquigarrow 0` efectúa un análisis de casos del término M comparándolo con los patrones de `left` y `right`.

En esta línea nos piden marcar subtérminos y anotaciones de tipos,

$$\sigma ::= \dots \mid \sigma + \sigma$$

$$M ::= \dots \mid \text{left}_\sigma(M) \mid \text{right}_\sigma(M) \mid \text{case } M \text{ of } \text{left}(x) \rightsquigarrow M ; \text{right}(y) \rightsquigarrow M$$

Y enunciar las nuevas reglas de tipado y extender el conjunto de valores y las reglas de semántica de la nueva extensión.

Reglas de tipado

$$\frac{\Gamma \vdash M : \tau}{\Gamma \vdash \text{left}_\sigma(M) : \tau + \sigma} \text{T-Left}$$

$$\frac{\Gamma \vdash M : \tau}{\Gamma \vdash \text{right}_\sigma(M) : \sigma + \tau} \text{T-Right}$$

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x : \sigma. M : \sigma \rightarrow \tau} \text{T-Abs}$$

$$\frac{\Gamma \vdash M : \sigma + \tau \quad \Gamma, x : \sigma \vdash N : \rho \quad \Gamma, y : \tau \vdash O : \rho}{\Gamma \vdash \text{case } M \text{ of } \text{left}(x) \rightsquigarrow N ; \text{right}(y) \rightsquigarrow O : \rho} \text{T-Case}$$

Extensión de valores

$$V ::= \dots \mid left_{\sigma}(V) \mid right_{\sigma}(V)$$

Reglas de semántica

$$\frac{M \rightarrow M'}{left_{\sigma}(M) \rightarrow left_{\sigma}(M')} \text{ E-Left}$$

$$\frac{M \rightarrow M'}{right_{\sigma}(M) \rightarrow right_{\sigma}(M')} \text{ E-Right}$$

$$\frac{M \rightarrow M'}{\text{case } M \text{ of } left(x) \rightsquigarrow N ; right(y) \rightsquigarrow 0 \rightarrow \text{case } M' \text{ of } left(x) \rightsquigarrow N ; right(y) \rightsquigarrow 0} \text{ E-Case}$$

$$\frac{}{\text{case } left_{\sigma}(V) \text{ of } left(x) \rightsquigarrow N ; right(y) \rightsquigarrow 0 \rightarrow N \{ x := V \}} \text{ E-CaseLeft}$$

$$\frac{}{\text{case } right_{\sigma}(V) \text{ of } left(x) \rightsquigarrow N ; right(y) \rightsquigarrow 0 \rightarrow 0 \{ y := V \}} \text{ E-CaseRight}$$

Detalle importante de por qué tenemos en la regla de tipado el $y : \tau$ a la derecha de todo :

3 Tercera extensión : árboles binarios

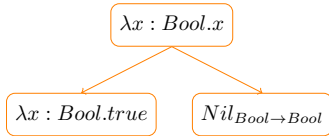
3.1 árbol binario

Sintaxis: tipos y términos

$$\sigma ::= \dots \mid AB_{\sigma}$$

$$M, N, O ::= \dots \mid Nil_{\sigma} \mid Bin(M, N, O) \mid raiz(M) \mid der(M) \mid izq(M) \mid esNil(M)$$

Árbol de funciones



Este árbol sería

$$Bin(Bin(Nil_{Bool \rightarrow Bool}, \lambda x : Bool.true, Nil_{Bool \rightarrow Bool}), \lambda x : Bool.x, Nil_{Bool \rightarrow Bool})$$

Reglas de tipado

$$\frac{}{\Gamma \vdash Nil_{\sigma} : AB_{\sigma}} \text{ T-Nil}$$

$$\frac{\Gamma \vdash M : AB_{\sigma} \quad \Gamma \vdash N : \sigma \quad \Gamma \vdash O : AB_{\sigma}}{\Gamma \vdash Bin(M, N, O) : AB_{\sigma}} \text{ T-Bin}$$

$$\frac{\Gamma \vdash M : AB_{\sigma}}{\Gamma \vdash raiz(M) : AB_{\sigma}} \text{ T-Raiz}$$

$$\frac{\Gamma \vdash M : AB_{\sigma}}{\Gamma \vdash esNil(M) : AB_{\sigma}} \text{ T-EsNil}$$

$$\frac{\Gamma \vdash M : AB_{\sigma}}{\Gamma \vdash izq(M) : AB_{\sigma}} \text{ T-Izq}$$

$$\frac{\Gamma \vdash M : AB_{\sigma}}{\Gamma \vdash der(M) : AB_{\sigma}} \text{ T-Der}$$

Extensión de valores

$$V ::= \dots \mid Nil_\sigma \mid Bin(V, V, V)$$

Reglas de semántica

$$\frac{M \rightarrow M'}{Bin(M, N, O) \rightarrow Bin(M', N, O)} \text{ E-Bin1}$$

$$\frac{N \rightarrow N'}{Bin(V, N, O) \rightarrow Bin(V, N', O)} \text{ E-Bin2}$$

$$\frac{O \rightarrow O'}{Bin(V_1, V_2, O) \rightarrow Bin(V_1, V_2, O')} \text{ E-Bin3}$$

$$\frac{M \rightarrow M'}{raiz(M) \rightarrow raiz(M')} \text{ E-Raiz}$$

$$\frac{M \rightarrow M'}{raiz(Bin(V_1, V_2, V_3)) \rightarrow V_2} \text{ E-RaizBin}$$

No definimos reglas para $raiz(Nil_\sigma)$, $izq(Nil_\sigma)$, $der(Nil_\sigma)$ ya que perdemos progreso.

$$\frac{M \rightarrow M'}{izq(M) \rightarrow izq(M')} \text{ E-Izq}$$

$$\frac{M \rightarrow M'}{der(M) \rightarrow der(M')} \text{ E-Der}$$

$$\frac{}{izq(Bin(V_1, V_2, V_3)) \rightarrow V_1} \text{ E-IzqBin}$$

$$\frac{}{der(Bin(V_1, V_2, V_3)) \rightarrow V_3} \text{ E-DerBin}$$

$$\frac{M \rightarrow M'}{esNil(M) \rightarrow esNil(M')} \text{ E-esNil}$$

$$\frac{}{esNil(Bin(V_1, V_2, V_3)) \rightarrow false} \text{ E-esNilBin}$$

$$\frac{}{esNil(Nil_\sigma) \rightarrow true} \text{ E-esNilNil}$$

3.2 árboles bis

Sintaxis : tipos y términos

$$\sigma ::= \dots \mid AB_\sigma$$

$$M, N, O ::= \dots \mid Nil_\sigma \mid Bin(M, N, O) \mid \text{case } M \text{ of } Nil \rightsquigarrow N ; Bin(i, r, d) \rightsquigarrow 0$$

Reglas de tipado

$$\frac{}{\Gamma \vdash Nil_\sigma : AB_\sigma} \text{ T-Nil}$$

$$\frac{\Gamma \vdash M : AB_\sigma \quad \Gamma \vdash N : \sigma \quad \Gamma \vdash O : AB_\sigma}{\Gamma \vdash Bin(M, N, O) : AB_\sigma} \text{ T-Bin}$$

$$\frac{\Gamma \vdash M : AB_\tau \quad \Gamma \vdash N : \sigma \quad \Gamma, i : AB_\tau, r : \tau, d : AB_\tau \vdash O : \sigma}{\Gamma \vdash \text{case } M \text{ of } Nil \rightsquigarrow N ; Bin(i, r, d) \rightsquigarrow 0 : \sigma} \text{ T-CaseAB}$$

Extensión de valores

$$V ::= \dots \mid Nil_\sigma \mid Bin(V, V, V)$$

Reglas de semántica

$$\begin{array}{c}
\frac{M \rightarrow M'}{Bin(M, N, O) \rightarrow Bin(M', N, O)} \text{E-Bin1} \\
\\
\frac{N \rightarrow N'}{Bin(V, N, O) \rightarrow Bin(V, N', O)} \text{E-Bin2} \\
\\
\frac{O \rightarrow O'}{Bin(V_1, V_2, O) \rightarrow Bin(V_1, V_2, O')} \text{E-Bin3} \\
\\
\frac{M \rightarrow M'}{\text{case } M \text{ of Nil } \rightsquigarrow N ; Bin(i, r, d) \rightsquigarrow 0 \rightarrow \text{case } M' \text{ of Nil } \rightsquigarrow N ; Bin(i, r, d) \rightsquigarrow 0} \text{E-CaseAB} \\
\\
\frac{}{\text{case Nil}_\sigma \text{ of Nil } \rightsquigarrow N ; Bin(i, r, d) \rightsquigarrow 0 \rightarrow N} \text{E-CaseNil} \\
\\
\frac{}{\text{case Bin}(V_1, V_2, V_3) \text{ of Nil } \rightsquigarrow N ; Bin(i, r, d) \rightsquigarrow 0 \rightarrow 0\{i := V_1, r := V_2, d := V_3\}} \text{E-CaseBin}
\end{array}$$

Nos piden reducir el siguiente término

`case if($\lambda x : Bool.x$) True then Bin(NilNat, 1, NilNat) else NilNat of Nil \rightsquigarrow False ; Bin(i, r, d) \rightsquigarrow isZero(r)`

Recordemos las siguiente reglas de cómputo :

$$\begin{array}{c}
\frac{}{\text{if true then N else } 0 \rightarrow N} \text{E-IfTrue} \\
\\
\frac{}{\text{if false then N else } 0 \rightarrow 0} \text{E-IfFalse}
\end{array}$$

y de congruencia :

$$\frac{M \rightarrow M'}{\text{if M then N else } 0 \rightarrow \text{if } M' \text{ then N else } 0} \text{E-If}$$

Con esto, ...

$$\begin{array}{l}
\frac{E-CaseAB, E-If, \beta}{\text{case if True then Bin(Nil}_{Nat}, \underline{1}, Nil_{Nat}) \text{ else Nil}_{Nat} \text{ of Nil } \rightsquigarrow \text{False ; Bin}(i, r, d) \rightsquigarrow \text{isZero(r)}} \\
\frac{E-CaseAB, E-IfTrue}{\text{case Bin(Nil}_{Nat}, \underline{1}, Nil_{Nat}) \text{ of Nil } \rightsquigarrow \text{False ; Bin}(i, r, d) \rightsquigarrow \text{isZero(r)}} \\
\frac{E-CaseBin}{\text{isZero(r) } \{ i := Nil_{Nat}, r := \underline{1}, d := Nil_{Nat} \} = \text{isZero}(\underline{1}) = \text{isZero}(\text{succ}(\text{zero}))} \\
\frac{E-isZero_n}{\text{false}}
\end{array}$$

Definir macros para raiz, der, izq y esNil

`esNilσ = λx : ABσ . case x of Nil \rightsquigarrow true ; Bin(i, r, d) \rightsquigarrow false`
`raizσ = λx : ABσ . case x of Nil \rightsquigarrow ⊥σ ; Bin(i, r, d) \rightsquigarrow r usa fix (punto fijo)`
`izqσ = λx : ABσ . case x of Nil \rightsquigarrow ⊥ABσ ; Bin(i, r, d) \rightsquigarrow i`

Definir una nueva extensión que incorpore expresiones de la forma `map(M, N)`, donde N es un árbol y M una función que se aplicará a cada elemento de N

Extendemos términos

$M, N ::= \dots \mid \text{map}(M, N)$

Reglas de tipado

$$\frac{\Gamma \vdash M : \sigma \rightarrow \rho \quad \Gamma \vdash N : AB_\sigma}{\Gamma \vdash \text{map}(M, N) : AB_\rho} \text{T-Map}$$

Extendemos valores

$V ::= \dots \mid$ no se agregan valores

Reglas de semántica

$$\frac{M \rightarrow M'}{\text{map}(M, N) \rightarrow \text{map}(M', N)} \text{E-Map1}$$

$$\frac{N \rightarrow N'}{\text{map}(V, N) \rightarrow \text{map}(V, N')} \text{E-Map2}$$

$$\frac{}{\text{map}(V, \text{Bin}(V_1, V_2, V_3)) \rightarrow \text{Bin}(\text{map}(V, V_1), V_2, \text{map}(V, V_3))} \text{E-MapBin}$$

$$\frac{\vdash V : \tau \rightarrow \rho}{\text{map}(V, \text{Nil}_\tau) \rightarrow \text{Nil}_\rho} \text{E-MapNil}$$

Como comentario :

$$\begin{aligned} & \text{map}(\lambda x : \text{Bool}. \text{ zero}, \text{ Nil}_{\text{Bool}}) \xrightarrow{E\text{-MapNil}} \text{ Nil}_{\text{Nat}} \\ & \vdash \lambda x : \text{Bool}. \text{ zero} : \text{ Bool} \rightarrow \text{ Nat} \end{aligned}$$