Cálculo Lambda

PLP

Cálculo Lambda “puro” (sin tipos)

Sintaxis (CL puro)

```
M ::= x
      | λx.M
      | M M
```

Semántica (CL puro)

$$(\lambda x.M)\ N \rightarrow M\{x := N\}$$

Si $M \rightarrow M'$, entonces:

$$M\ N \rightarrow M'\ N$$

$$N\ M \rightarrow N\ M'$$

$$\lambda x.M \rightarrow \lambda x.M'$$

Cálculo Lambda puro

- No hay tipos (todas las expresiones son funciones)
- La única operación es la aplicación de funciones
- ¡ES TURING COMPLETO!

Cálculo Lambda no tipado con booleanos

Sintaxis (CL no tipado con booleanos)

```
M ::= x
      | λx.M
      | M M
      | true
      | false
      | if M then M else M
```

Semántica (CL no tipado con booleanos)

Se agregan:

if true then M else N $\rightarrow$ **M**

if false then M else N $\rightarrow$ **M**

Si **M** $\rightarrow$ **M′**, entonces:

if M then N else O $\rightarrow$

if M′ then N else O

Cálculo Lambda no tipado con booleanos

- Sigue sin haber tipos
  - **`true false`** es un programa "válido" (aunque "se cuelga")
- Lo único que hicimos fue agregar formas de expresar nuevos valores
  - Sigue siendo Turing Completo

Cálculo Lambda tipado
(lo que vamos a usar en la práctica)

Cálculo Lambda tipado

- Ahora sí hay tipos
  - Por ahora sólo funciones, booleanos y naturales (λ^{BN})
- Determinista
- Call-by-value

Sintaxis (λ^{BN})

```
M ::= x
    | λx:σ.M
    | M M
    | true
    | false
    | if M then M else M
    | zero
    | succ(M)
    | pred(M)
    | isZero(M)
```

```
σ ::= Bool
    | Nat
    | σ → σ
```

Tipado (λ^{BN})

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma}\ ax_v$$

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x : \sigma.M : \sigma \to \tau} \to_i \qquad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M\ N : \tau} \to_e$$

$$\frac{}{\Gamma \vdash \mathsf{true} : \mathsf{Bool}}\ ax_{\mathsf{true}} \qquad \frac{}{\Gamma \vdash \mathsf{false} : \mathsf{Bool}}\ ax_{\mathsf{false}}$$

$$\frac{\Gamma \vdash M : \mathsf{Bool} \quad \Gamma \vdash P : \sigma \quad \Gamma \vdash Q : \sigma}{\Gamma \vdash \mathsf{if}\ M\ \mathsf{then}\ P\ \mathsf{else}\ Q : \sigma}\ \mathsf{if}$$

$$\frac{}{\Gamma \vdash \mathsf{zero} : \mathsf{Nat}}\ \mathsf{zero} \qquad \frac{\Gamma \vdash M : \mathsf{Nat}}{\Gamma \vdash \mathsf{succ}(M) : \mathsf{Nat}}\ \mathsf{succ}$$

$$\frac{\Gamma \vdash M : \mathsf{Nat}}{\Gamma \vdash \mathsf{pred}(M) : \mathsf{Nat}}\ \mathsf{pred} \qquad \frac{\Gamma \vdash M : \mathsf{Nat}}{\Gamma \vdash \mathsf{isZero}(M) : \mathsf{Bool}}\ \mathsf{isZero}$$

Semántica (λ^{BN}) (1)

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

{β} $(\lambda \mathbf{x}{:}\sigma.\mathbf{M})\ \mathbf{V} \to \mathbf{M}\{\mathbf{x} := \mathbf{V}\}$

{if_t} **if true then M else N** $\to$ **M**

{if_f} **if false then M else N** $\to$ **N**

{pred} **pred(succ(V))** $\to$ **V**

{isZero_0} **isZero(zero)** $\to$ **true**

{isZero_n} **isZero(succ(V))** $\to$ **false**

Semántica (λ^{BN}) (2)

Si $\mathtt{M} \rightarrow \mathtt{M'}$, entonces:

$$\{\mu\} \quad \mathtt{M\ N} \rightarrow \mathtt{M'\ N}$$

$$\{\nu\} \quad \mathtt{V\ M} \rightarrow \mathtt{V\ M'}$$

$$\{\mathrm{if}_c\} \quad \mathtt{if\ M\ then\ N\ else\ O} \rightarrow \mathtt{if\ M'\ then\ N\ else\ O}$$

$$\{\mathrm{succ}_c\} \quad \mathtt{succ(M)} \rightarrow \mathtt{succ(M')}$$

$$\{\mathrm{pred}_c\} \quad \mathtt{pred(M)} \rightarrow \mathtt{pred(M')}$$

$$\{\mathrm{isZero}_c\} \quad \mathtt{isZero(M)} \rightarrow \mathtt{isZero(M')}$$

Sustituciones (1)

$$\mathbf{x}\ \{\mathbf{x} := \mathbf{N}\} = \mathbf{N}$$

$$\mathbf{y}\ \{\mathbf{x} := \mathbf{N}\} = \mathbf{y}$$

$$(\lambda\mathbf{x}:\sigma.\mathbf{M})\ \{\mathbf{x} := \mathbf{N}\} = \lambda\mathbf{x}:\sigma.\mathbf{M}$$

$$(\lambda\mathbf{y}:\sigma.\mathbf{M})\ \{\mathbf{x} := \mathbf{N}\} = \lambda\mathbf{y}:\sigma.(\mathbf{M}\ \{\mathbf{x} := \mathbf{N}\})$$

si $\mathbf{y} \notin FV(\mathbf{N})$

$$(\lambda\mathbf{y}:\sigma.\mathbf{M})\ \{\mathbf{x} := \mathbf{N}\} = \lambda\mathbf{z}:\sigma.(\mathbf{M}\ \{\mathbf{y} := \mathbf{z}\}\ \{\mathbf{x} := \mathbf{N}\})$$

si $\mathbf{y} \in FV(\mathbf{N})$, con $\mathbf{z} \notin FV(\mathbf{N})$

$$(\mathbf{M}\ \mathbf{O})\ \{\mathbf{x} := \mathbf{N}\} = (\mathbf{M}\ \{\mathbf{x} := \mathbf{N}\})(\mathbf{O}\ \{\mathbf{x} := \mathbf{N}\})$$

Sustituciones (2)

```
      true {x := N} = true
     false {x := N} = false
      zero {x := N} = zero
(if M then O \
   else P) {x := N} = if (M {x := N}) then \
                        (O {x := N}) else (P {x := N})
   succ(M) {x := N} = succ(M{x := N})
   pred(M) {x := N} = pred(M{x := N})
 isZero(M) {x := N} = isZero(M{x := N})
```

Juicio de tipado

$$\Gamma \vdash M : \sigma$$

con:

- Γ contexto $\{x_1{:}\sigma_1,\ x_1{:}\sigma_1,\ \ldots\ x_n{:}\sigma_n\}$
- M término de λ^{BN}
- σ tipo de λ^{BN}

Demostrar un juicio de tipado

- Es "dirigido por sintaxis" así que siempre sabemos qué regla usar
  - En general hay una única regla por término

`M ::= x`	ax_v
`	λx:σ.M`	$\to_i$
`	M M`	$\to_e$
`	true`	ax_t
`	false`	ax_f
`	if M then M else M`	if
`	zero`	ax_z
`	succ(M)`	succ
`	pred(M)`	pred
`	isZero(M)`	isZero

Ejercicio

Ejercicio

Demostrar los siguientes juicios de tipado.

a. $x : \text{Bool} \rightarrow \text{Nat} \vdash x\ \text{True} : \text{Nat}$

b. $z : \text{Nat} \vdash \text{succ}(\text{if true then zero else pred}(z)) : \text{Nat}$

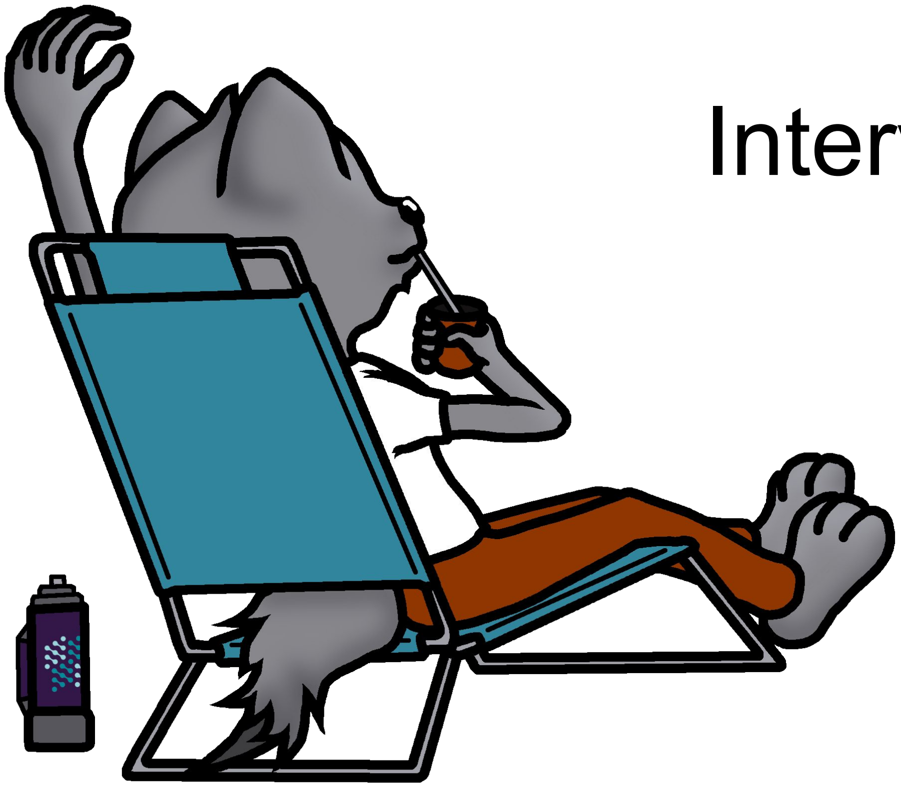
c. $\vdash (\lambda f{:}\text{Nat}{\rightarrow}\text{Bool}.\ \text{false})\ (\lambda x{:}\text{Nat}.\ \text{isZero}(x)) : \text{Bool}$

Ejercicio

Ejercicio

Reducir las siguientes expresiones.

a. pred(succ(pred(succ(cero))))
b. succ(succ(if isZero(zero) then zero else zero))
c. if (λf:Nat → Nat. false) (λx:Nat. succ(x))
   then true
   else (λx:Nat. isZero(x)) (succ(succ(zero)))



Intervalo

Extensiones

Ejercicio 20 (Pares, o produtos)

Puesta en común

Solución: reglas de tipado

$$\frac{\Gamma \vdash M \ : \ \sigma \qquad \Gamma \vdash N \ : \ \tau}{\Gamma \vdash \langle M, N \rangle \ : \ \sigma \times \tau} \texttt{T-PAR}$$

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

Ejercicio 20 (Pares, o produtos)

Demostrar el siguiente juicio de tipado.

$x : Bool \vdash \pi_1(<\pi_2(<cero, x>), true>) : Bool$

Solución: valores y reglas semánticas

$$V ::= \ldots | \langle V, V \rangle$$

$$\frac{M \to M'}{\langle M, N \rangle \to \langle M', N \rangle}\ \texttt{E-PAR1} \qquad \frac{N \to N'}{\langle V, N \rangle \to \langle V, N' \rangle}\ \texttt{E-PAR2}$$

$$\frac{M \to M'}{\pi_1(M) \to \pi_1(M')}\ \texttt{E-}\pi_1 \qquad \frac{M \to M'}{\pi_2(M) \to \pi_2(M')}\ \texttt{E-}\pi_2$$

$$\frac{}{\pi_1(\langle V_1, V_2 \rangle) \to V_1}\ \texttt{E-}\pi_1(V) \qquad \frac{}{\pi_2(\langle V_1, V_2 \rangle) \to V_2}\ \texttt{E-}\pi_2(V)$$

Ejercicio 20 (Pares, o produtos)

Reducir la siguiente expresión.

$$
\pi_2(
\quad (\lambda x : Nat \times Nat.\ x)
\quad \langle \pi_1(\langle zero, true \rangle), (\lambda x : Nat.\ succ(x))\ zero \rangle
)
$$

Ejercicio 21 (Uniones disjuntas, co-productos o sumas)

Puesta en común

Solución: reglas de tipado

$$\frac{\Gamma \vdash M : \sigma}{\Gamma \vdash \mathsf{left}(M) : \sigma + \tau}\ \texttt{T-LEFT} \qquad \frac{\Gamma \vdash M : \tau}{\Gamma \vdash \mathsf{right}(M) : \sigma + \tau}\ \texttt{T-RIGHT}$$

$$\frac{\Gamma \vdash M : \sigma + \tau \qquad \Gamma, \boxed{x : \sigma} \vdash M_1 : \rho \qquad \Gamma, \boxed{y : \tau} \vdash M_2 : \rho}{\Gamma \vdash \mathsf{case}\ M\ \mathsf{of}\ \mathsf{left}(x) \hookrightarrow M_1 \parallel \mathsf{right}(y) \hookrightarrow M_2 : \rho}\ \texttt{T-CASE}$$

Ejercicio 21 (Uniones disjuntas, co-productos o sumas)

Demostrar el siguiente juicio de tipado.

$\vdash$ case left(zero) of
left(x) $\hookrightarrow$ right(zero) |
right(x) $\hookrightarrow$ left(true) : Bool + Nat

Solución: valores y reglas semánticas

$$V ::= \ldots \mid \mathsf{left}(V) \mid \mathsf{right}(V)$$

$$\frac{M \to M'}{\mathsf{left}(M) \to \mathsf{left}(M')}\ \texttt{E-LEFT} \qquad \frac{M \to M'}{\mathsf{right}(M) \to \mathsf{right}(M')}\ \texttt{E-RIGHT}$$

$$\frac{M \to M'}{\mathsf{case}\ M\ \mathsf{of}\ \mathsf{left}(x) \hookrightarrow M_1 \parallel \mathsf{right}(y) \hookrightarrow M_2 \to \mathsf{case}\ M'\ \mathsf{of}\ \mathsf{left}(x) \hookrightarrow M_1 \parallel \mathsf{right}(y) \hookrightarrow M_2}\ \texttt{E-CASE}$$

$$\frac{}{\mathsf{case}\ \mathsf{left}(V)\ \mathsf{of}\ \mathsf{left}(x) \hookrightarrow M_1 \parallel \mathsf{right}(y) \hookrightarrow M_2 \to M_1\boxed{\{x \leftarrow V\}}}\ \texttt{E-CASE-L}$$

$$\frac{}{\mathsf{case}\ \mathsf{right}(V)\ \mathsf{of}\ \mathsf{left}(x) \hookrightarrow M_1 \parallel \mathsf{right}(y) \hookrightarrow M_2 \to M_2\boxed{\{y \leftarrow V\}}}\ \texttt{E-CASE-R}$$

Ejercicio 21 (Uniones disjuntas, co-productos o sumas)

Reducir la siguiente expresión.

left(case right((λx:Nat. succ(x)) zero) of
    left(x) ↪ zero |
    right(x) ↪ succ(x)
)

EXTRA

Extensión con árboles binarios

M ::= …, Nil_σ, AB(M, M, M), map(M, M)

σ ::= …, AB_σ V ::= …, Nil_σ, AB(V, V, V)

Tipado:

$$\frac{}{\Gamma \vdash \mathrm{Nil}_\sigma : \mathrm{AB}_\sigma}\ (\text{T-}\textsc{Nil}) \qquad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \mathrm{AB}_\sigma}{\Gamma \vdash map(M, N) : \mathrm{AB}_\tau}\ (\text{T-}\textsc{Map})$$

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

Extensión con árboles binarios

$M ::= \ldots, \mathrm{Nil}_\sigma, \mathrm{AB}(M, M, M), \mathrm{map}(M, M)$

$\sigma ::= \ldots, \mathrm{AB}_\sigma \quad V ::= \ldots, \mathrm{Nil}_\sigma, \mathrm{AB}(V, V, V)$

Semántica (del map):

$$\frac{M \to M'}{map(M, N) \to map(M', N)} \text{(E-MAP1)} \qquad \frac{N \to N'}{map(V, N) \to map(V, N')} \text{(E-MAP2)}$$

$$\frac{\boxed{\vdash V : \sigma \to \tau}}{map(V, \mathrm{Nil}_\sigma) \to \mathrm{Nil}_\tau} \text{(E-MAPNIL)}$$

$$\frac{}{map(V, \mathrm{Bin}(U, X, W)) \to \mathrm{Bin}(map(V, U), V\ X, map(V, W))} \text{(E-MAPBIN)}$$