Cubo λ y Cálculo de Construcciones

Cristian Sottile
ICC (UBA/CONICET) & UNQ

Ponencias de doctorandes
Tópicos de Reescritura, Cálculo Lambda y Tipos
DC, Exactas, UBA

Buenos Aires, Argentina

13 de Noviembre de 2023

El cubo λ

Extensiones del $\lambda^{\rightarrow}$ que permiten parametrizar las posibles combinaciones de términos y tipos:

$$
\begin{tikzcd}
 & \lambda\omega \arrow[rr] & & \lambda C \\
\lambda 2 \arrow[ur] \arrow[rr] & & \lambda\Pi 2 \arrow[ur] & \\
 & \lambda\underline{\omega} \arrow[uu] \arrow[rr] & & \lambda\Pi\underline{\omega} \arrow[uu] \\
\lambda^{\rightarrow} \arrow[uu] \arrow[ur] \arrow[rr] & & \lambda\Pi \arrow[uu] \arrow[ur] &
\end{tikzcd}
$$

$\uparrow$ **Polimorfismo**: términos dependiendo de tipos

e.g. $\vdash \Lambda X.\lambda x^{X}.x : \forall X.X \to X$

$\nearrow$ **Constructores de tipos**: tipos dependiendo de tipos

e.g. $\vdash \lambda x^{\tau}.x : (\lambda\alpha^{*}.\alpha \to \alpha)\tau$

$\rightarrow$ **Tipos dependientes**: tipos dependiendo de términos

e.g. $\vdash \lambda x^{A}.t : \Pi x^{A}.\,B$ con $x \in fv(A)$

El cubo λ

Extensiones del $\lambda^{\to}$ que permiten parametrizar las posibles combinaciones de términos y tipos:

$$\begin{tikzcd}
 & \lambda\omega \arrow[rr] & & \lambda C \\
\boxed{\lambda 2} \arrow[ur] \arrow[rr] & & \lambda\Pi 2 \arrow[ur] & \\
 & \lambda\underline{\omega} \arrow[uu] \arrow[rr] & & \lambda\Pi\underline{\omega} \arrow[uu] \\
\lambda^{\to} \arrow[uu] \arrow[ur] \arrow[rr] & & \lambda\Pi \arrow[uu] \arrow[ur] &
\end{tikzcd}$$

$\uparrow$ **Polimorfismo**: términos dependiendo de tipos

e.g. $\vdash \Lambda X.\lambda x^{X}.x : \forall X.X \to X$

$\nearrow$ **Constructores de tipos**: tipos dependiendo de tipos

e.g. $\vdash \lambda x^{\tau}.x : (\lambda\alpha^{*}.\alpha \to \alpha)\tau$

$\to$ **Tipos dependientes**: tipos dependiendo de términos

e.g. $\vdash \lambda x^{A}.t : \Pi x^{A}.B$ con $x \in fv(A)$

El cubo λ

Extensiones del $\lambda^{\to}$ que permiten parametrizar las posibles combinaciones de términos y tipos:

$$
\begin{tikzcd}
& \lambda\omega \arrow[rr] & & \lambda C \\
\lambda 2 \arrow[ur] \arrow[rr] & & \lambda\Pi 2 \arrow[ur] & \\
& \boxed{\lambda\underline{\omega}} \arrow[uu] \arrow[rr] & & \lambda\Pi\underline{\omega} \arrow[uu] \\
\lambda^{\to} \arrow[uu] \arrow[ur] \arrow[rr] & & \lambda\Pi \arrow[uu] \arrow[ur] &
\end{tikzcd}
$$

$\uparrow$ **Polimorfismo**: términos dependiendo de tipos

e.g. $\vdash \Lambda X.\lambda x^X.x : \forall X.X \to X$

$\nearrow$ **Constructores de tipos**: tipos dependiendo de tipos

e.g. $\vdash \lambda x^\tau.x : (\lambda\alpha^*.\alpha \to \alpha)\tau$

$\to$ **Tipos dependientes**: tipos dependiendo de términos

e.g. $\vdash \lambda x^A.t : \Pi x^A.B$ con $x \in fv(A)$

El cubo λ

Extensiones del $\lambda^{\rightarrow}$ que permiten parametrizar las posibles combinaciones de términos y tipos:

$$\begin{tikzcd}
 & \lambda\omega \arrow[rr] & & \lambda C \\
\lambda 2 \arrow[ur] \arrow[rr] & & \lambda\Pi 2 \arrow[ur] & \\
 & \lambda\underline{\omega} \arrow[uu] \arrow[rr] & & \lambda\Pi\underline{\omega} \arrow[uu] \\
\lambda^{\rightarrow} \arrow[uu] \arrow[ur] \arrow[rr] & & \boxed{\lambda\Pi} \arrow[uu] \arrow[ur] &
\end{tikzcd}$$

$\uparrow$ **Polimorfismo**: términos dependiendo de tipos

e.g. $\vdash \Lambda X.\lambda x^{X}.x : \forall X.X \to X$

$\nearrow$ **Constructores de tipos**: tipos dependiendo de tipos

e.g. $\vdash \lambda x^{\tau}.x : (\lambda\alpha^{*}.\alpha \to \alpha)\tau$

$\rightarrow$ **Tipos dependientes**: tipos dependiendo de términos

e.g. $\vdash \lambda x^{A}.t : \Pi x^{A}.B$ con $x \in fv(A)$

El cubo λ

Extensiones del $\lambda^{\to}$ que permiten parametrizar las posibles combinaciones de términos y tipos:

$$\begin{tikzcd}
 & \lambda\omega \arrow[rr] & & \boxed{\lambda C} \\
\lambda 2 \arrow[ur] \arrow[rr] & & \lambda\Pi 2 \arrow[ur] & \\
 & \lambda\underline{\omega} \arrow[uu] \arrow[rr] & & \lambda\Pi\underline{\omega} \arrow[uu] \\
\lambda^{\to} \arrow[uu] \arrow[ur] \arrow[rr] & & \lambda\Pi \arrow[uu] \arrow[ur] &
\end{tikzcd}$$

$\uparrow$ **Polimorfismo**: términos dependiendo de tipos

e.g. $\vdash \Lambda X.\lambda x^{X}.x : \forall X.X \to X$

$\nearrow$ **Constructores de tipos**: tipos dependiendo de tipos

e.g. $\vdash \lambda x^{\tau}.x : (\lambda\alpha^{*}.\alpha \to \alpha)\tau$

$\to$ **Tipos dependientes**: tipos dependiendo de términos

e.g. $\vdash \lambda x^{A}.t : \Pi x^{A}.B$ con $x \in fv(A)$

El cubo λ

$\uparrow$ **Polimorfismo**: términos dependiendo de tipos — muchos términos en uno

- Un tipo de tipos

$$
\begin{array}{lcl}
K & ::= & * \qquad\qquad S ::= \square \\
A & ::= & X \mid A \to A \mid \forall X.A \\
t & ::= & x \mid \lambda x^A.t \mid tt \mid \Lambda X.t \mid tA
\end{array}
\qquad
\begin{array}{c}
\lambda x^\tau.x^\tau \\
\vdots \\
\lambda x^A.x^A
\end{array}
\Longrightarrow \Lambda X.\lambda x^X.x^X
$$

$\nearrow$ **Constructores de tipos**: tipos dependiendo de tipos — abstraemos estructura en tipos

- λ en tipos
- Cons. vs "valores"
- Tipos módulo $=_\beta^A$

$$
\begin{array}{lcl}
K & ::= & * \mid K \to K \qquad S ::= \square \\
A & ::= & \alpha \mid A \to A \mid \lambda \alpha^K.A \mid AA \\
t & ::= & x \mid \lambda x^A.t \mid tt
\end{array}
\qquad
\begin{array}{c}
\tau \to \tau \\
\vdots \\
A \to A
\end{array}
\Longrightarrow
\begin{array}{c}
(\lambda \alpha^*.\alpha \to \alpha)\,\tau \\
\vdots \\
(\lambda \alpha^*.\alpha \to \alpha)\,A
\end{array}
$$

$\rightarrow$ **Tipos dependientes**: tipos dependiendo de términos — indexamos tipos por términos

- Tipos en kinds
- Más especificidad
- Tipos módulo $=_\beta^{A,t}$

$$
\begin{array}{lcl}
K & ::= & * \mid A \to K \qquad S ::= \square \\
A & ::= & \alpha \mid \lambda x^A.A \mid \Pi x^A.A \mid At \\
t & ::= & x \mid \lambda x^A.t \mid tt
\end{array}
\qquad
\mathtt{List} \Longrightarrow
\begin{array}{c}
(\Pi x^{\mathbb{N}}.\ \mathtt{List}\ x)\ 0 \\
(\Pi x^{\mathbb{N}}.\ \mathtt{List}\ x)\ n \\
\vdots
\end{array}
$$

El cubo λ: formación de tipos en $\lambda\underline{\omega}$

$$S ::= \square \qquad K ::= * \mid K \to K \qquad A ::= \alpha \mid A \to A \mid \lambda\alpha^K.A \mid AA \qquad t ::= x \mid \lambda x^A.t \mid tt$$

Reglas de formación

$$\frac{\Gamma \vdash A : *}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma \vdash A : * \quad \Gamma \vdash B : *}{\Gamma \vdash A \to B : *} \qquad \frac{\Gamma \vdash K : \square}{\Gamma, \alpha : K \vdash \alpha : K} \qquad \frac{\Gamma \vdash K : \square \quad \Gamma \vdash L : \square}{\Gamma \vdash K \to L : \square}$$

$$s := \{*, \square\} \qquad \frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma \vdash A : s \quad \Gamma \vdash B : s}{\Gamma \vdash A \to B : s}$$

Reglas de abstracción

$$\frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash A \to B : *}{\Gamma \vdash \lambda x^A.t : A \to B} \qquad \frac{\Gamma, \alpha : K \vdash A : L \quad \Gamma \vdash K \to L : \square}{\Gamma \vdash \lambda\alpha^K.A : K \to L}$$

$$\frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash A \to B : s}{\Gamma \vdash \lambda x^A.t : A \to B}$$

El cubo λ: formación de tipos en λP

$$S ::= \Box \qquad K ::= * \mid A \to K \qquad A ::= \alpha \mid \lambda x^A.A \mid \Pi x^A.A \mid At \qquad t ::= x \mid \lambda x^A.t \mid tt$$

Reglas de formación

$$\frac{\Gamma \vdash A : *}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma \vdash A : * \quad \Gamma, x : A \vdash B : *}{\Gamma \vdash \Pi x^A.\, B : *} \qquad \frac{\Gamma \vdash A : * \quad \Gamma, x : A \vdash K : \Box}{\Gamma \vdash \Pi x^A.\, K : \Box}$$

$$s := \{*, \Box\} \qquad \frac{\Gamma \vdash A : * \quad \Gamma, x : A \vdash B : s}{\Gamma \vdash \Pi x^A.\, B : s}$$

Reglas de abstracción

$$\frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash \Pi x^A.\, B : *}{\Gamma \vdash \lambda x^A.t : \Pi x^A.\, B} \qquad \frac{\Gamma, x : A \vdash B : K \quad \Gamma \vdash \Pi x^A.\, K : \Box}{\Gamma \vdash \lambda x^A.B : \Pi x^A.\, K}$$

$$\frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash \Pi x^A.\, B : s}{\Gamma \vdash \lambda x^A.t : \Pi x^A.\, B}$$

El Cálculo de Construcciones

[T. Coquand & G. Huet]

Citas

- *Formalismo de alto orden para pruebas constructivas en estilo de deducción natural*
- *Una prueba es un término λ tipado con proposiciones de la lógica subyacente*
- *Borrando los tipos tenemos el término que representa el algoritmo asociado*

Ideas e intuiciones

- **Aplastamos** el lenguaje: términos tipados con términos

$$\lambda x : N.\, M$$

- Agregamos otro **ligador**

$$[x : A]M$$

  que construye **productos**, donde A denota un tipo (es decir un término)

El cubo λ: formación de tipos en λC

$\nearrow$ **Constructores de tipos**: tipos dependiendo de tipos

$$S ::= \square \qquad K ::= * \mid K \to K \qquad A ::= \alpha \mid A \to A \mid \lambda\alpha^K.A \mid AA \qquad t ::= x \mid \lambda x^A.t \mid tt$$

$$s := \{*, \square\} \qquad \frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma \vdash A : s \quad \Gamma \vdash B : s}{\Gamma \vdash A \to B : s} \qquad \frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash A \to B : s}{\Gamma \vdash \lambda x^A.t : A \to B}$$

$\to$ **Tipos dependientes**: tipos dependiendo de términos

$$S ::= \square \qquad K ::= * \mid A \to K \qquad A ::= \alpha \mid \lambda x^A.A \mid \Pi x^A.A \mid At \qquad t ::= x \mid \lambda x^A.t \mid tt$$

$$s := \{*, \square\} \qquad \frac{\Gamma \vdash A : * \quad \Gamma, x : A \vdash B : s}{\Gamma \vdash \Pi x^A. B : s} \qquad \frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash \Pi x^A. B : s}{\Gamma \vdash \lambda x^A.t : \Pi x^A. B}$$

λC **Tipos dependientes**: términos y tipos dependiendo de términos y tipos

$$t ::= x \mid \square \mid * \mid tt \mid \lambda x^t.t \mid \Pi x^t. t$$

$$s := \{*, \square\} \qquad \frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash \Pi x^A. B : s_2} \qquad \frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash \Pi x^A. B : s}{\Gamma \vdash \lambda x^A.t : \Pi x^A. B}$$

El cubo λ: generalización

Instanciando la regla de formación con distintos s_1 y s_2, tenemos todas las posibles parametrizaciones del cubo:

- $(*, *)$: términos dependiendo de términos
- $(\square, *)$: términos dependiendo de tipos
- $(\square, \square)$: tipos dependiendo de tipos
- $(*, \square)$: tipos dependiendo de términos

$\lambda\!\rightarrow$	$(*, *)$			
$\lambda 2$	$(*, *)$	$(\square, *)$		
$\lambda\underline{\omega}$	$(*, *)$		$(\square, \square)$	
$\lambda\omega$	$(*, *)$	$(\square, *)$	$(\square, \square)$	
λP	$(*, *)$			$(*, \square)$
$\lambda P2$	$(*, *)$	$(\square, *)$		$(*, \square)$
$\lambda P\underline{\omega}$	$(*, *)$		$(\square, \square)$	$(*, \square)$
$\lambda P\omega = \lambda C$	$(*, *)$	$(\square, *)$	$(\square, \square)$	$(*, \square)$

El cubo λ: generalización

$$s_1,\ s_2 \quad \in \quad \{ \quad (*,*) \quad (\Box,*) \quad (\Box,\Box) \quad (*,\Box) \quad \}$$

$$(\mathit{sort}) \quad \emptyset \vdash * : \Box$$

$$(\mathit{var}) \ \frac{\Gamma \vdash A : s}{\Gamma,\ x : A \vdash x : A} \quad \text{if } x \notin \Gamma$$

$$(\mathit{weak}) \ \frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s}{\Gamma,\ x : C \vdash A : B} \quad \text{if } x \notin \Gamma$$

$$(\mathit{form}) \ \frac{\Gamma \vdash A : s_1 \qquad \Gamma,\ x : A \vdash B : s_2}{\Gamma \vdash \Pi x : A\,.\,B \ : \ s_2}$$

$$(\mathit{appl}) \ \frac{\Gamma \vdash M \ : \ \Pi x : A\,.\,B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN \ : \ B[x := N]}$$

$$(\mathit{abst}) \ \frac{\Gamma,\ x : A \vdash M : B \qquad \Gamma \vdash \Pi x : A\,.\,B \ : \ s}{\Gamma \vdash \lambda x : A\,.\,M \ : \ \Pi x : A\,.\,B}$$

$$(\mathit{conv}) \ \frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s}{\Gamma \vdash A : B'} \quad \text{if } B =_\beta B'$$

Propiedades de los sistemas del cubo λ

- Church-Rosser
- Subject Reduction
- Strong Normalization

$$\Gamma \vdash A : B \text{ implica } A, B \in \mathsf{SN}$$

- Unicidad de tipos

$$\Gamma \vdash A : B \text{ y } \Gamma \vdash A : B' \text{ implica } B =_\beta B'$$

Encoding de lógica de predicados mínima en λP

S is a set A is a proposition	$S : *$ $A : *$
$a \in S$ p proves A	$a : S$ $p : A$
P is a predicate on S	$P : S \to *$

$A \Rightarrow B$ $\forall_{x \in S}(P(x))$	$A \to B \ (= \Pi x : A \,.\, B)$ $\Pi x : S \,.\, Px$
($\Rightarrow$-*elim*) ($\Rightarrow$-*intro*)	(*appl*) (*abst*)
($\forall$-*elim*) ($\forall$-*intro*)	(*appl*) (*abst*)

Ejemplo

Sean

$$P : S \to * \qquad\qquad a : S$$

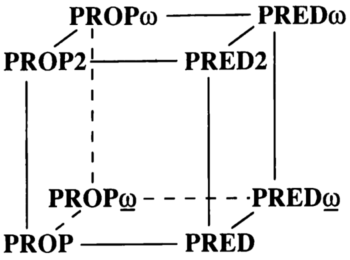
entonces

1. $P\,a : *$
2. $P\,a$ habitado (*i.e.* $\exists t : P\,a$) $\implies P(a)$
3. $P\,a$ no habitado (*i.e.* $\nexists t : P\,a$) $\implies \neg P(a)$

Propositions as Types

El cubo L

PROP	proposition logic;
PROP2	second order proposition logic;
PROP$\underline{\omega}$	weakly higher order proposition logic;
PROPω	higher order proposition logic;
PRED	predicate logic;
PRED2	second order predicate logic;
PRED$\underline{\omega}$	weakly higher order predicate logic;
PREDω	higher order predicate logic.



Bibliografía

- Thierry Coquand, Gérard Huet. **The calculus of constructions**. 1988
- Rob Nederpelt, Herman Geuvers. **Type Theory and Formal Proof: An Introduction**. 2014
- Henk Barendregt. **Introduction to generalized type systems**. 1991