Introducción a la correspondencia entre pruebas y programas:

El cálculo de construcciones y sus extensiones

Alexandre Miquel

junio de 2021

Historia

1971 J.-Y. Girard: *Une extension de l'interprétation fonctionnelle de Gödel à l'analyse et son application à l'élimination des coupures dans l'analyse et la théorie des types* (Sistema F)

1971 P. Martin-Löf: *A theory of types* (Sistema «Type : Type»)

1972 J.-Y. Girard: *Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur* (Sistema Fω)

1975 P. Martin-Löf: *An Intuitionistic Theory of Types: Predicative Part* (Teoría de tipos intensional)

1985 T. Coquand & G. Huet: *Constructions: A Higher Order Proof System for Mechanizing Mathematics* (Cálculo de construcciones)

1990 C. Paulin: *Le calcul des constructions inductives* (CIC, Coq)

1990 Z. Luo: *The Extended Calculus of Constructions* (ECC)

1990 S. Berardi, H. Barendregt: *Pure Type Systems*

Plan

Plan

Sistemas de tipos puros

- Un marco general para definir múltiples sistemas de tipos basados en el producto dependiente (sin tipos inductivos)
- Factoriza la mayoría de las propiedades sintácticas de esos sistemas (*subject reduction* por ejemplo, pero no la normalización)

Definición (Sistema de tipos puros) [Berardi & Barendregt 1990]

Un sistema de tipos puros (PTS) está definido por:

- Un conjunto de suertes $\mathcal{S}$ (tipos de los tipos)
- Un conjunto de axiomas $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$ (tipado de las suertes)
- Un conjunto de reglas $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$ (y de los productos dependientes)

- Cada PTS define:
  - Un conjunto de términos (parametrizado por $\mathcal{S}$)
  - Un sistema de tipos (parametrizado por $\mathcal{S}$, $\mathcal{A}$ y $\mathcal{R}$)

Sintaxis

Definición (Términos y tipos)

$$M, N, A, B \quad ::= \quad x \quad | \quad \lambda x : A \,.\, M \quad | \quad MN \qquad \text{(cálculo } \lambda \text{ a la Church)}$$
$$\quad | \quad s \quad | \quad \Pi x : A \,.\, B \qquad \text{(suerte + producto dependiente)}$$

- Ninguna distinción sintáctica entre los términos y los tipos
- Un tipo es un término $T : s$, donde $s \in \mathcal{S}$:

  término : *tipo* : *suerte*
- $\Pi x : A \,.\, B$ escrito $A \to B$ cuando $x \notin FV(\Gamma)$
- **Reducción:** β-reducción (confluente)
- **Contextos:** $\Gamma \equiv x_1 : A_1, \ldots, x_n : A_n$ (listas ordenadas)
- Sistema de tipado basado en dos juicios:
  - $\vdash \Gamma$ context «El contexto Γ está bien formado»
  - $\Gamma \vdash M : A$ «En el contexto Γ, el término M tiene tipo A»

Reglas de tipado

- Reglas del juicio $\vdash \Gamma$ context (buena formación de contexto)

$$\frac{}{\vdash \varnothing \text{ context}} \qquad \frac{\Gamma \vdash A : s}{\vdash \Gamma, x : A \text{ context}}\ x \notin \mathrm{dom}(\Gamma)$$

- Reglas del juicio $\Gamma \vdash M : A$ (tipado)

$$\frac{\vdash \Gamma \text{ context}}{\Gamma \vdash x : T}\ (x:A) \in \Gamma \qquad \frac{\Gamma \vdash \Pi x : A \,.\, B : s \qquad \Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A \,.\, M : \Pi x : A \,.\, B}$$

$$\frac{\vdash \Gamma \text{ context}}{\Gamma \vdash s_1 : s_2}\ \text{si}\ (s_1 : s_2) \in \mathcal{A} \qquad \frac{\Gamma \vdash M : \Pi x : A \,.\, B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$$

$$\frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash \Pi x : A \,.\, B : s_3}\ \text{si}\ (s_1, s_2, s_3) \in \mathcal{R}$$

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s}{\Gamma \vdash M : A'}\ \text{si}\ A \cong A'$$

Ejemplos

- **El sistema «Type : Type»:** [Martin-Löf 1971]

$$\begin{aligned} \mathcal{S} &= \{\text{Type}\} \\ \mathcal{A} &= \{(\text{Type} : \text{Type})\} \\ \mathcal{R} &= \{(\text{Type}, \text{Type}, \text{Type})\} \end{aligned}$$

- **El marco lógico de la teoría de tipos:** (versión polimórfica)

$$\begin{aligned} \mathcal{S} &= \{\text{Set},\ \text{Type}\} \\ \mathcal{A} &= \{(\text{Set} : \text{Type})\} \\ \mathcal{R} &= \{(\text{Set}, \text{Set}, \text{Set}),\ (\text{Set}, \text{Type}, \text{Type}), \\ &\quad\ (\text{Type}, \text{Set}, \text{Type}),\ (\text{Type}, \text{Type}, \text{Type})\} \end{aligned}$$

- **Los sistemas del cubo:** (véase más adelante)

$$\begin{aligned} \mathcal{S} &= \{\star,\ \Box\} \qquad \text{(notación alt.: } \star = \text{Prop}, \Box = \text{Type)} \\ \mathcal{A} &= \{(\star : \Box)\} \\ \mathcal{R} &\subseteq \{(\star, \star, \star),\ (\star, \Box, \Box),\ (\Box, \star, \star),\ (\Box, \Box, \Box)\} \end{aligned}$$

Propiedades sintácticas (1/4)

Recordatorio: $\Gamma' \sqsubseteq \Gamma \quad \equiv \quad \Gamma'$ es un prefijo de Γ

Lema (Buena formación)

1. Si $\Gamma \vdash M : A$, entonces $\vdash \Gamma$ context
2. Si $\vdash \Gamma$ context, entonces $\vdash \Gamma'$ context para todo $\Gamma' \sqsubseteq \Gamma$

Lema (Variables libres)

1. Si $\vdash x_1 : A_1, \ldots, x_n : A_n$ context, entonces
   - $FV(A_i) \subseteq \{x_1, \ldots, x_{i-1}\}$ para todo $i \in [1..n]$
2. Si $x_1 : A_1, \ldots, x_n : A_n \vdash M : A$, entonces
   - $FV(A_i) \subseteq \{x_1, \ldots, x_{i-1}\}$ para todo $i \in [1..n]$
   - $FV(M) \subseteq \{x_1, \ldots, x_n\}$ y $FV(A) \subseteq \{x_1, \ldots, x_n\}$

Propiedades sintácticas (2/4)

Recordatorio: Dados contextos Γ, Γ', se escribe $\Gamma \subseteq \Gamma'$ cuando $(x : A) \in \Gamma$ implica $(x : A) \in \Gamma'$ para toda declaración $(x : A)$

Lema (Debilitamiento)

Si $\Gamma \vdash M : A$, $\Gamma \subseteq \Gamma'$ y $\vdash \Gamma'$ context, entonces $\Gamma' \vdash M : A$

Lema (Sustitutividad)

1. Si $\vdash \Gamma, x : A, \Gamma'$ context y $\Gamma \vdash N : A$, entonces $\vdash \Gamma, \Gamma'[x := N]$ context
2. Si $\Gamma, x : A, \Gamma' \vdash M : B$ y $\Gamma \vdash N : A$, entonces $\Gamma, \Gamma'[x := N] \vdash M[x := N] : B[x := N]$

Propiedades sintácticas (3/4)

Lema de inversión

- $\Gamma \vdash x : C \quad \Rightarrow \quad \exists A \begin{cases} (x : A) \in \Gamma \\ C \cong A \end{cases}$
- $\Gamma \vdash s : C \quad \Rightarrow \quad \exists s' \begin{cases} (s, s') \in \mathcal{A} \\ C \cong s' \end{cases}$
- $\Gamma \vdash \lambda x : A \,.\, M : C \quad \Rightarrow \quad \exists s, B \begin{cases} \Gamma \vdash \Pi x : A \,.\, B : s \\ \Gamma, x : A \vdash M : B \\ C \cong \Pi x : A \,.\, B \end{cases}$
  (con $x \notin \mathrm{dom}(\Gamma)$)
- $\Gamma \vdash M\, N : C \quad \Rightarrow \quad \exists A, B \begin{cases} \Gamma \vdash M : \Pi x : A \,.\, B \\ \Gamma \vdash N : A \\ C \cong B[x := N] \end{cases}$
- $\Gamma \vdash \Pi x : A \,.\, B : C \quad \Rightarrow \quad \exists s_1, s_2, s_3 \begin{cases} \Gamma \vdash A : s_1 \\ \Gamma, x : A \vdash B : s_2 \\ (s_1, s_2, s_3) \in \mathcal{R} \\ C \cong s_3 \end{cases}$
  (con $x \notin \mathrm{dom}(\Gamma)$)

Propiedades sintácticas (4/4)

Proposición (Tipo de los tipos)

Si $\Gamma \vdash M : A$, entonces:

- o bien $\Gamma \vdash A : s$ para alguna suerte s
- o bien $A \equiv s$ es una suerte máxima (*top sort*)

Suerte máxima (*top sort*) $\equiv$ suerte s tal que $(s, s') \notin \mathcal{A}$ para todo $s' \in \mathcal{S}$

Proposición (*Subject reduction*)

Si $\Gamma \vdash M : A$ y $M \succ_\beta^* M'$, entonces $\Gamma \vdash M' : A$

¡Cuidado!

- Ningún resultado de normalización (fuerte o débil) en general

  Algunos sistemas normalizan, otros no

- Ningún resultado de consistencia lógica en general

  Se necesita definir una suerte de las proposiciones $\rightsquigarrow$ PTS lógicos

  [Coquand & Herbelin 1994]

Sistemas de tipos puros funcionales

Definición (PTS funcional)

Un PTS $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ es funcional si para todos $s_1, s_2, s_2', s_3, s_3' \in \mathcal{S}$:

1. $(s_1, s_2) \in \mathcal{A}, \quad (s_1, s_2') \in \mathcal{A} \quad \Rightarrow \quad s_2 = s_2'$
2. $(s_1, s_2, s_3) \in \mathcal{R}, \quad (s_1, s_2, s_3') \in \mathcal{R} \quad \Rightarrow \quad s_3 = s_3'$

En los PTS funcionales:

Proposición (Unicidad del tipo)

$\Gamma \vdash M : A, \quad \Gamma \vdash M : A' \quad \Rightarrow \quad A \cong A'$

Proposición (Refuerzo)

1. $\vdash \Gamma, x : A, \Gamma'$ context, $\quad x \notin FV(\Gamma'), \quad \Rightarrow \quad \vdash \Gamma, \Gamma'$ context
2. $\Gamma, x : A, \Gamma' \vdash M : B, \quad x \notin FV(\Gamma', M, B) \quad \Rightarrow \quad \Gamma, \Gamma' \vdash M : B$

Intuición: Se pueden eliminar las declaraciones de las variables no utilizadas

Verificación e inferencia de tipo

Dado un PTS $\mathcal{P}$, se consideran los siguientes tres problemas:

1. **Verificación de contexto:**
   Dado Γ, determinar si el juicio $\vdash \Gamma$ context es derivable o no
2. **Verificación de tipo:**
   Dados Γ, M, A, determinar si el juicio $\Gamma \vdash M : A$ es derivable o no
3. **Inferencia de tipo:**
   Dados Γ y M, determinar si existe A tal que $\Gamma \vdash M : A$
   (y devolver tal tipo A cuando existe)

Teorema

Sea $\mathcal{P}$ un PTS funcional en el cual:

1. los conjuntos $\mathcal{A}$ y $\mathcal{R}$ definen funciones (parciales) recursivas, y
2. todos los tipos bien formados (en $\mathcal{P}$) tienen una forma normal.

Entonces en el sistema $\mathcal{P}$, los tres problemas anteriores son decidibles

Obs.: No se necesita que todos los términos bien tipados en $\mathcal{P}$ sean normalizantes, sólo se necesita que los tipos (bien formados) sean normalizantes

Morfismos de PTS (1/2)

Definición (Morfismo de PTS)

Dados sistemas de tipos puros $\mathcal{P} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ y $\mathcal{P}' = (\mathcal{S}', \mathcal{A}', \mathcal{R}')$, un morfismo de $\mathcal{P}$ a $\mathcal{P}'$ es una función $\phi : \mathcal{S} \to \mathcal{S}'$ tal que:

1. $(s_1, s_2) \in \mathcal{A} \;\Rightarrow\; (\phi(s_1), \phi(s_2)) \in \mathcal{A}'$
2. $(s_1, s_2, s_3) \in \mathcal{R} \;\Rightarrow\; (\phi(s_1), \phi(s_2), \phi(s_3)) \in \mathcal{R}'$

(para todos $s_1, s_2, s_3 \in \mathcal{S}$)

Dado un morfismo de $\mathcal{P}$ a $\mathcal{P}'$, la función $\phi : \mathcal{S} \to \mathcal{S}'$ subyacente se extiende sintácticamente a los términos, a los contextos, a los juicios y a las reducciones de $\mathcal{P}$, de tal modo que:

1. Si $\vdash \Gamma$ context (en $\mathcal{P}$), entonces $\vdash \phi(\Gamma)$ context (en $\mathcal{P}'$)
2. Si $\Gamma \vdash M : A$ (en $\mathcal{P}$), entonces $\phi(\Gamma) \vdash \phi(M) : \phi(A)$ (en $\mathcal{P}'$)
3. Si $M \succ M'$ (en $\mathcal{P}$), entonces $\phi(M) \succ \phi(M')$ (en $\mathcal{P}'$)

Morfismos de PTS (2/2)

- Un morfismo $\phi : \mathcal{P} \to \mathcal{P}'$ transforma cada sucesión (in)finita de reducciones en $\mathcal{P}$ en una sucesión (in)finita de reducciones en $\mathcal{P}'$.

  Por lo tanto:

Dado un morfismo de $\mathcal{P}$ a $\mathcal{P}'$:

1. Si $\mathcal{P}$ no es normalizante, entonces $\mathcal{P}'$ tampoco es normalizante
2. Si $\mathcal{P}'$ es normalizante, entonces $\mathcal{P}$ también es normalizante

- Intuitivamente, los morfismos van de los PTS «más normalizantes» a los PTS «menos normalizantes»

- En particular, se observa que para todo sistema $\mathcal{P}$, existe un (único) morfismo de $\mathcal{P}$ al sistema «Type : Type» $(\phi(s) := \text{Type}, \ \ s \in \mathcal{S})$

  Es decir: «Type : Type» es el objeto terminal de la categoría de los PTS

- Por lo tanto, «Type : Type» es el PTS «menos normalizante»

  Girard dedujo la no normalización del sistema «Type : Type» (Martin-Löf 1971) de la no normalización del sistema U (Girard 1972)

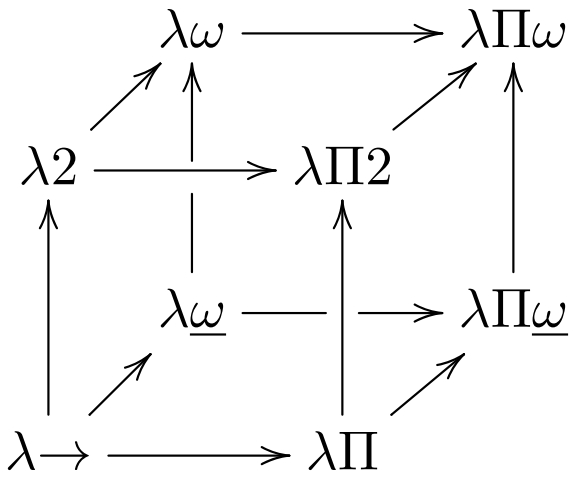
Plan

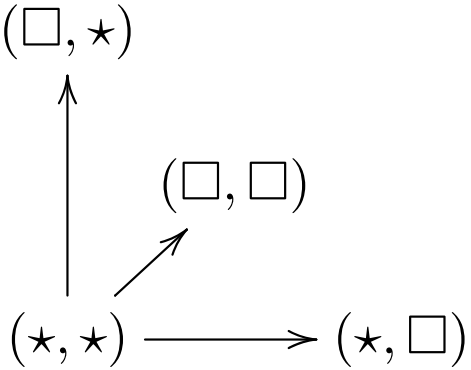
El cubo de Barendregt

8 sistemas (funcionales) formados a partir de:

- $\mathcal{S} = \{ \star, \ \Box \}$ (notación alt.: $\star = \text{Prop} = \text{Set}$, $\Box = \text{Type}$)
- $\mathcal{A} = \{ (\star : \Box) \}$
- $\mathcal{R} \subseteq \{ (\star, \star), \ (\Box, \star)?, \ (\star, \Box)?, \ (\Box, \Box)? \}$

(Convención: $(s_1, s_2) \in \mathcal{R}$ significa $(s_1, s_2, s_2) \in \mathcal{R}$)





Las flechas representan las inclusiones entre los 8 sistemas
Los 8 sistemas son fuertemente normalizantes y lógicamente consistentes

Estratificación de los términos

En cada uno de los 8 sistemas del cubo, se pueden dividir los términos (bien tipados) en 4 niveles distintos:

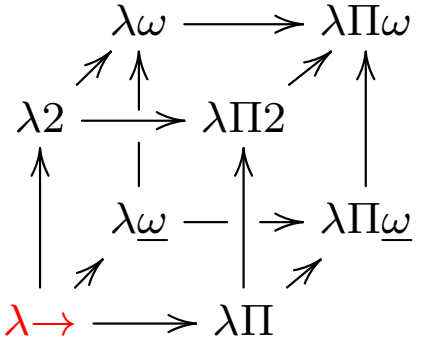
- Los géneros (*kinds*) $\quad K, L, \ldots : \Box$
- Los constructores (de tipos/proposiciones) $\quad \phi, \psi, \ldots : K : \Box$
- Las proposiciones $\quad A, B, \ldots : \star : \Box$
- Las pruebas $\quad t, u, \ldots : A : \star$

$$\begin{array}{ccccc} & & \text{Constructores} & : & \text{Géneros} & : & \Box \\ & & (\phi, \psi, \ldots) & & (K, L, \ldots) & & \\ & & \cup & & \cup & & \\ \text{Pruebas} & : & \text{Proposiciones} & : & \star & & \\ (t, u, \ldots) & & (A, B, \ldots) & & & & \end{array}$$

$\rightsquigarrow$ Presentación estratificada de la sintaxis

$\lambda\!\!\rightarrow$ = Cálculo lambda simplemente tipado

$$\begin{aligned} \mathcal{S} &= \{\star,\ \Box\} \\ \mathcal{A} &= \{(\star : \Box)\} \\ \mathcal{R} &= \{(\star,\star)\} \end{aligned}$$



Sintaxis estratificada de $\lambda\!\!\rightarrow$:

Géneros	K, L	$::=$	$\star$	$(: \Box)$
Proposiciones	A, B	$::=$	α	$(: \star : \Box)$
		$\mid$	$A \to B$	$(\star,\star)$
Pruebas	t, u	$::=$	x	$(: A : \star)$
		$\mid$	$\lambda x : A \,.\, u \quad \mid \quad t\, u$	$(\star,\star)$

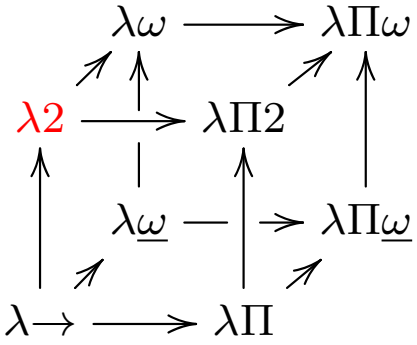
El único género es $\star$, entonces los únicos constructores son las proposiciones

$\lambda 2 \;=\;$ Sistema F

[Girard 1971]

$$\begin{array}{lcl} \mathcal{S} & = & \{\star,\ \Box\} \\ \mathcal{A} & = & \{(\star : \Box)\} \\ \mathcal{R} & = & \{(\star,\star),\ (\Box,\star)\} \end{array}$$

$(\Box,\star) =$ polimorfismo



Sintaxis estratificada de $\lambda 2$:

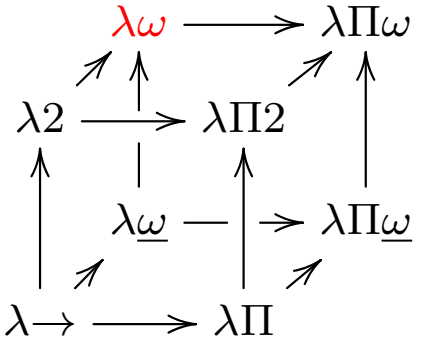
Géneros	K, L	$::=$	$\star$		$(: \Box)$
Proposiciones	A, B	$::=$	α		$(: \star : \Box)$
		$\mid$	$A \to B$		$(\star,\star)$
		$\mid$	$\forall \alpha : \star \,.\, A$		$(\Box,\star)$
Pruebas	t, u	$::=$	x		$(: A : \star)$
		$\mid$	$\lambda x : A \,.\, u$	$\mid\ t\,u$	$(\star,\star)$
		$\mid$	$\lambda \alpha : \star \,.\, u$	$\mid\ t\,A$	$(\Box,\star)$

El único género es $\star$, entonces los únicos constructores son las proposiciones

$\lambda\omega$ = Sistema Fω [Girard 1972]

$$\begin{array}{lcl} \mathcal{S} & = & \{\star,\ \Box\} \\ \mathcal{A} & = & \{(\star : \Box)\} \\ \mathcal{R} & = & \{(\star,\star),\ (\Box,\star),\ (\Box,\Box)\} \end{array}$$

$(\Box,\Box)$ = constructores de tipos



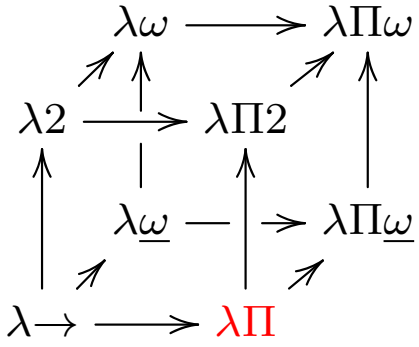
Sintaxis estratificada de $\lambda\omega$:

Géneros	K, L	::=	$\star$		$(: \Box)$
		|	$K \to L$		$(\Box,\Box)$
Constructores	ϕ, ψ, A, B	::=	α		$(: K : \Box)$
		|	$A \to B$		$(\star,\star)$
		|	$\forall \alpha : K \,.\, A$		$(\Box,\star)$
		|	$\lambda \alpha : K \,.\, \phi$	| $\phi\,\psi$	$(\Box,\Box)$
Pruebas	t, u	::=	x		$(: A : \star)$
		|	$\lambda x : A \,.\, u$	| $t\,u$	$(\star,\star)$
		|	$\lambda \alpha : K \,.\, u$	| $t\,\phi$	$(\Box,\star)$

$\lambda\Pi$ = *Logical framework* monomórfico [Martin-Löf 1975]

$$\begin{array}{lcl} \mathcal{S} & = & \{\star,\ \Box\} \\ \mathcal{A} & = & \{(\star : \Box)\} \\ \mathcal{R} & = & \{(\star,\star),\ (\star,\Box)\} \end{array}$$

$(\star,\Box)$ = tipos dependientes



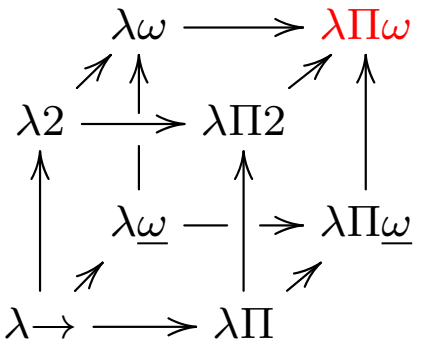
Sintaxis estratificada de $\lambda\Pi$:

Géneros	K, L	$::=$	$\star$			$(: \Box)$
		$\mid$	$\Pi x : A \,.\, K$			$(\star, \Box)$
Constructores	ϕ, ψ, A, B	$::=$	α			$(: K : \Box)$
		$\mid$	$\Pi x : A \,.\, B$			$(\star,\star)$
		$\mid$	$\lambda x : A \,.\, \phi$	$\mid$	$\phi\, t$	$(\star, \Box)$
Pruebas	t, u	$::=$	x			$(: A : \star)$
		$\mid$	$\lambda x : A \,.\, u$	$\mid$	$t\, u$	$(\star,\star)$

Obs.: *Logical framework* polimórfico = $\lambda\Pi$ + reglas $(\Box,\Box)$, $(\Box,\star,\Box)$ ($\notin$ Cubo)

$\lambda\Pi\omega$ = Cálculo de construcciones (CC) [Coquand & Huet 1985]

$$\begin{aligned}\mathcal{S} &= \{\star,\ \Box\}\\ \mathcal{A} &= \{(\star : \Box)\}\\ \mathcal{R} &= \{(\star,\star),\ (\Box,\star),\ (\star,\Box),\ (\Box,\Box)\}\end{aligned}$$

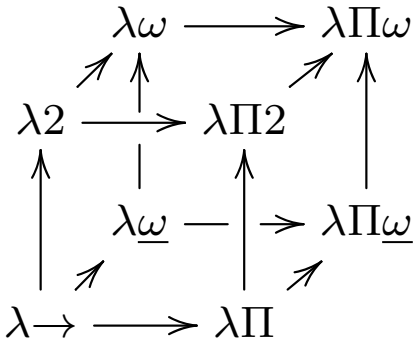


Sintaxis estratificada de $\lambda\Pi\omega$:

Géneros	K, L	$::=$	$\star$			$(: \Box)$
		$\mid$	$\Pi x : A \,.\, K$			$(\star, \Box)$
		$\mid$	$\Pi \alpha : K \,.\, L$			$(\Box, \Box)$
Constructores	ϕ, ψ, A, B	$::=$	α			$(: K : \Box)$
		$\mid$	$\Pi x : A \,.\, B$			$(\star, \star)$
		$\mid$	$\forall \alpha : K \,.\, A$			$(\Box, \star)$
		$\mid$	$\lambda x : A \,.\, \phi$	$\mid$	$\phi\, t$	$(\star, \Box)$
		$\mid$	$\lambda \alpha : K \,.\, \phi$	$\mid$	$\phi\, \psi$	$(\Box, \Box)$
Pruebas	t, u	$::=$	x			$(: A : \star)$
		$\mid$	$\lambda x : A \,.\, u$	$\mid$	$t\, u$	$(\star, \star)$
		$\mid$	$\lambda \alpha : K \,.\, u$	$\mid$	$t\, \phi$	$(\Box, \star)$

Borrado de las dependencias (1/4)

Existe una función de borrado de las dependencias $M \mapsto |M|$ que va de los 4 sistemas de la faz derecha (i.e. con tipos dependientes) a los 4 sistemas de la faz izquierda (i.e. sin tipos dependientes)



$$M \mapsto |M| \ : \begin{cases} \lambda\Pi & \to & \lambda{\to} \\ \lambda\Pi 2 & \to & \lambda 2 \\ \lambda\Pi\underline{\omega} & \to & \lambda\underline{\omega} \\ \lambda\Pi\omega & \to & \lambda\omega \end{cases}$$

Proposición (Corrección)

Dado un sistema $\mathcal{P} \in \{\lambda\Pi,\ \lambda\Pi 2,\ \lambda\Pi\underline{\omega},\ \lambda\Pi\omega\}$ (faz derecha) y escribiendo $|\mathcal{P}|$ al correspondiente sistema en la faz izquierda:

1. Si $\vdash \Gamma$ context (en $\mathcal{P}$), entonces $\vdash |\Gamma|$ context (en $|\mathcal{P}|$)
2. Si $\Gamma \vdash M : A$ (en $\mathcal{P}$), entonces $|\Gamma| \vdash |M| : |A|$ (en $|\mathcal{P}|$)

Borrado de las dependencias (2/4)

Definición de la función $M \mapsto |M|$ [Paulin-Mohring 1989]

Géneros	$\lvert\star\rvert$	$:\equiv$	$\star$	$(:\Box)$
	$\lvert\Pi x:A\,.\,K\rvert$	$:\equiv$	$\lvert K\rvert$	$(\star,\Box)$
	$\lvert\Pi\alpha:K\,.\,L\rvert$	$:\equiv$	$\lvert K\rvert \to \lvert L\rvert$	$(\Box,\Box)$
Constructores	$\lvert\alpha\rvert$	$:\equiv$	α	$(:K:\Box)$
	$\lvert\Pi x:A\,.\,B\rvert$	$:\equiv$	$\lvert A\rvert \to \lvert B\rvert$	$(\star,\star)$
	$\lvert\forall\alpha:K\,.\,A$	$:\equiv$	$\forall\alpha:\lvert K\rvert\,.\,\lvert A\rvert$	$(\Box,\star)$
	$\lvert\lambda x:A\,.\,\phi\rvert$	$:\equiv$	$\lvert\phi\rvert$	$(\star,\Box)$
	$\lvert\phi\,t\rvert$	$:\equiv$	$\lvert\phi\rvert$	$(\star,\Box)$
	$\lvert\lambda\alpha:K\,.\,\phi\rvert$	$:\equiv$	$\lambda\alpha:\lvert K\rvert\,.\,\lvert\phi\rvert$	$(\Box,\Box)$
	$\lvert\phi\,\psi\rvert$	$:\equiv$	$\lvert\phi\rvert\,\lvert\psi\rvert$	$(\Box,\Box)$
Pruebas	$\lvert x\rvert$	$:\equiv$	x	$(:A:\star)$
	$\lvert\lambda x:A\,.\,u\rvert$	$:\equiv$	$\lambda x:\lvert A\rvert\,.\,\lvert u\rvert$	$(\star,\star)$
	$\lvert t\,u\rvert$	$:\equiv$	$\lvert t\rvert\,\lvert u\rvert$	$(\star,\star)$
	$\lvert\lambda\alpha:K\,.\,u\rvert$	$:\equiv$	$\lambda\alpha:\lvert K\rvert\,.\,\lvert u\rvert$	$(\Box,\star)$
	$\lvert t\,\phi\rvert$	$:\equiv$	$\lvert t\rvert\,\lvert\phi\rvert$	$(\Box,\star)$

Borrado de las dependencias (3/4)

La función $M \mapsto |M|$ es una retracción del cubo de Barendregt en su faz izquierda (en la cual $|M| \equiv M$).

Por lo tanto, dado $\mathcal{P} \in \{\lambda\Pi,\ \lambda\Pi 2,\ \lambda\Pi\underline{\omega},\ \lambda\Pi\omega\}$ (faz derecha) y escribiendo $|\mathcal{P}|$ al correspondiente sistema en la faz izquierda:

Proposición (Extensión conservativa) [Paulin-Mohring 1989]

El sistema $\mathcal{P}$ es una extensión conservativa de $|\mathcal{P}|$

Demostración. En efecto, si $\Gamma \vdash M : A$ (con $\Gamma, A \in |\mathcal{P}|$ y $M \in \mathcal{P}$), entonces $\Gamma \vdash |M| : A$ (con $|M| \in |\mathcal{P}|$), pues $|\Gamma| \equiv \Gamma$ y $|A| \equiv A$. □

Corolario (Equiconsistencia)

Los sistemas $|\mathcal{P}|$ y $\mathcal{P}$ son equiconsistentes

Obs.: Un sistema $\mathcal{P}$ del cubo es consistente cuando no existe ningún término de prueba t tal que: $\alpha : \star \vdash t : \alpha$ ($\alpha : \star$ = «proposición cualquiera»)

Borrado de las dependencias (4/4)

- La función de borrado induce las siguientes retracciones:

$$\begin{array}{lcll} \lambda\Pi{\to} & \rightsquigarrow & \lambda{\to} & = \text{ cálculo lambda simplemente tipado} \\ & & & = \text{ cálculo proposicional mínimo} \\ \lambda\Pi 2 & \rightsquigarrow & \lambda 2 & = \text{ sistema F} \\ & & & = \text{ cálculo proposicional de } 2^{\text{do}} \text{ orden} \\ \lambda\Pi\omega \text{ (CC)} & \rightsquigarrow & \lambda\omega & = \text{ sistema F}\omega \\ & & & = \text{ cálculo proposicional de alto orden} \end{array}$$

- Usando una retracción modificada que preserva todas las redexes, también se puede demostrar que:

Teorema (Equinormalización) [Geuvers & Nederpelt 1991]

Para cada sistema $\mathcal{P}$ de la faz derecha del cubo, tenemos que:

$$\begin{array}{rl} & \mathcal{P} \text{ fuertemente normalizante} \\ \text{si y sólo si} & |\mathcal{P}| \text{ fuertemente normalizante} \end{array}$$

Plan

Los sistemas U y U^- (1/4)

- El sistema U está definido por: [Girard 1972]
  - $\mathcal{S} = \{ \star, \ \Box, \ \triangle \}$ (notación alt.: $\triangle =$ Kind)
  - $\mathcal{A} = \{ (\star : \Box), \ (\Box : \triangle) \}$
  - $\mathcal{R} = \{ (\star,\star), \ (\Box,\star), \ (\Box,\Box), \ (\triangle,\Box), \ (\triangle,\star) \}$

  (Como siempre: $(s_1, s_2) \in \mathcal{R}$ significa $(s_1, s_2, s_2) \in \mathcal{R}$)

- Sistema U^- := sistema U $-$ regla $(\triangle,\star)$

- **Intuición:** Sistema U^- = sistema $F\omega$ + polimorfismo impredicativo al nivel de los géneros (= impredicatividad^2)

$$\mathcal{R} = \{ \underbrace{\overbrace{(\star,\star), \ (\Box,\star)}^{F}, \ \overbrace{(\Box,\Box), \ (\triangle,\Box)}^{F'}}_{F\omega}, \ (\triangle,\star) \}_{U^-}$$

Los sistemas U y U^- (2/4)

Como en los sistemas del cubo, se pueden dividir los términos de los sistemas U y U^- en 4 niveles distintos:

- Los géneros (*kinds*) $\quad K, L, \ldots : \Box : \triangle$
- Los constructores (de tipos/proposiciones) $\quad \phi, \psi, \ldots : K : \Box$
- Las proposiciones $\quad A, B, \ldots : \star : \Box$
- Las pruebas $\quad t, u, \ldots : A : \star$

$$\begin{array}{ccccccc} & \text{Constructores} & : & \text{Géneros} & : & \Box & : & \triangle \\ & (\phi, \psi, \ldots) & & (K, L, \ldots) & & & & \\ & \cup & & \cup & & & & \\ \text{Pruebas} & : & \text{Proposiciones} & : & \star & & & \\ (t, u, \ldots) & & (A, B, \ldots) & & & & & \end{array}$$

Obs.: La nueva suerte máxima $\triangle$ sólo sirve para enriquecer el nivel de los géneros

Los sistemas U y U^- (3/4)

Sintaxis estratificada de los sistemas U y U^-

Géneros	K, L	$::=$	$\star$		$(: \Box)$
		$\mid$	κ		$(: \Box : \triangle)$
		$\mid$	$K \to L$		$(\Box, \Box)$
		$\mid$	$\forall \kappa : \Box \,.\, K$		$(\triangle, \Box)$
Constructores	ϕ, ψ, A, B	$::=$	α		$(: K : \Box)$
		$\mid$	$A \to B$		$(\star, \star)$
		$\mid$	$\forall \alpha : K \,.\, A$		$(\Box, \star)$
		$\mid$	$\forall \kappa : \Box \,.\, A$		$(\triangle, \star)$
		$\mid$	$\lambda \alpha : K \,.\, \phi$	$\mid \; \phi\, \psi$	$(\Box, \Box)$
		$\mid$	$\lambda \kappa : \Box \,.\, \phi$	$\mid \; \phi\, K$	$(\triangle, \Box)$
Pruebas	t, u	$::=$	x		$(: A : \star)$
		$\mid$	$\lambda x : A \,.\, u$	$\mid \; t\, u$	$(\star, \star)$
		$\mid$	$\lambda \alpha : K \,.\, u$	$\mid \; t\, \phi$	$(\Box, \star)$
		$\mid$	$\lambda \kappa : \Box \,.\, u$	$\mid \; t\, K$	$(\triangle, \star)$

Recordatorio: $U^- = U -$ regla $(\triangle, \star)$

Los sistemas U y U^- (4/4)

- En el sistema U, se observa que:
  - Los géneros no contienen redexes
  - Los constructores (y proposiciones) son fuertemente normalizantes (por una traducción obvia en el sistema F, que preserva las redexes)
  - Por lo tanto, todos los tipos son fuertemente normalizantes

  $\Rightarrow$ Verificación e inferencia de tipo decidibles

- Sin embargo, existen pruebas cerradas $t : \forall \alpha : \star . \alpha$

  $\Rightarrow$ Sistema inconsistente y no normalizante (sólo al nivel de las pruebas)

- Algunas pruebas de inconsistencia:
  - Girard 1972: Paradoja de Burali-Forti en el sistema U (por codificación de un tipo universal de «todos los ordinales»)
  - Coquand 1986: Paradoja de Reynolds en el sistema U^-
  - Hurkens 1995: Inconsistencia muy compacta en el sistema U^-
  - Miquel 2001: Paradoja de Russell en el sistema U^- (por codificación de la teoría de conjuntos de Cantor-Frege)

El sistema λPRED

- El sistema λPRED está definido por: [Berardi 1988]
  - $\mathcal{S} = \{ \star^{\mathsf{T}},\ \star^{\mathsf{P}},\ \star^{\mathsf{F}},\ \Box^{\mathsf{T}},\ \Box^{\mathsf{P}} \}$
  - $\mathcal{A} = \{ (\star^{\mathsf{T}} : \Box^{\mathsf{T}}),\ (\star^{\mathsf{P}} : \Box^{\mathsf{P}}) \}$
  - $\mathcal{R} = \{ (\star^{\mathsf{P}}, \star^{\mathsf{P}}),\ (\star^{\mathsf{T}}, \star^{\mathsf{P}}),\ (\star^{\mathsf{T}}, \Box^{\mathsf{P}}),\ (\star^{\mathsf{T}}, \star^{\mathsf{T}}, \star^{\mathsf{F}}),\ (\star^{\mathsf{T}}, \star^{\mathsf{F}}, \star^{\mathsf{F}}) \}$

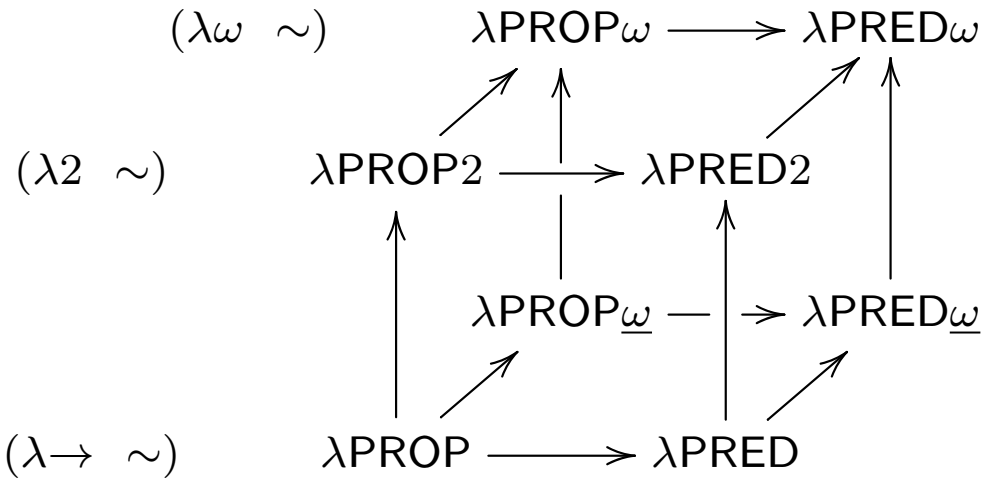
Sintaxis estratificada

Tipos	σ, τ	:	$\star^{\mathsf{T}} : \Box^{\mathsf{T}}$		(= variables)
Tipos de funciones	T_F	:	$\star^{\mathsf{F}}$	$::=$	$\tau \to \sigma \mid \tau \to T_F$
Funciones	F	:	$T_F : \star^{\mathsf{F}}$	$::=$	$f \mid \lambda x : \tau \,.\, t$
				$\mid$	$\lambda x : \tau \,.\, F \mid F\, t$
Términos	t, u	:	$\sigma : \star^{\mathsf{T}}$	$::=$	$x \mid F\, t$
Tipos de predicados	T_P	:	$\Box^{\mathsf{P}}$	$::=$	$\star^{\mathsf{P}} \mid \tau \to T_P$
Predicados	A, B, P, Q	:	$T_P : \Box^{\mathsf{P}}$	$::=$	$\alpha \mid A \Rightarrow B \mid \forall x : \tau \,.\, A$
				$\mid$	$\lambda x : \tau \,.\, P \mid P\, t$
Pruebas	p, q	:	$A : \star^{\mathsf{P}}$	$::=$	$\xi \mid \lambda \xi : A \,.\, p \mid p\, q$
				$\mid$	$\lambda x : \tau \,.\, p \mid p\, t$

$=$ Cálculo de predicados (lógica mínima de 1$^{\text{er}}$ orden)

El cubo lógico

El sistema λPRED pertenece al cubo lógico: [Berardi 1988]



λPROP	=	Cálculo proposicional mínimo de 1er orden
λPRED	=	Cálculo de predicados mínimo de 1er orden
λPROP2	=	Cálculo proposicional mínimo de 2do orden
λPRED2	=	Cálculo de predicados mínimo de 2do orden
λPROPω	=	Cálculo proposicional mínimo de alto orden
λPREDω	=	Cálculo de predicados mínimo de alto orden

+ función de borrado $(M \mapsto |M|) : \lambda\text{PRED}* \rightarrow \lambda\text{PROP}*$

El cálculo de construcciones con universos (CC^ω)

- El cálculo de construcciones con universos (CC^ω) está definido por: [Coquand & Paulin-Mohring $\sim$ 1990]
  - $\mathcal{S} = \{\star\} \cup \{\Box_i \ : \ i \geq 1\}$ (not. alt.: $\star = \mathsf{Prop}$, $\Box_i = \mathsf{Type}_i$))
  - $\mathcal{A} = \{(\star : \Box_1)\} \cup \{(\Box_i : \Box_{i+1}) \ : \ i \geq 1\}$
  - $\mathcal{R} = \{(\star, \star, \star)\} \cup \{(\Box_i, \star, \star) \ : \ i \geq 1\} \cup \{(\star, \Box_i, \Box_i) \ : \ i \geq 1\} \cup \{(\Box_i, \Box_j, \Box_{\max(i,j)}) \ : \ i, j \geq 1\}$

  $=$ CC + jerarquía predicativa de universos a la Martin-Löf

  $=$ PTS subyacente del cálculo de construcciones inductivas (Coq)
- Sistema fuertemente normalizante (prueba en ZF)
- Fuerza teórica con respecto a la teoría de conjuntos de Zermelo (Z):
  - CC^3 (= CC con 3 universos) $>$ Z [Miquel 2001]
  - $\mathrm{CC}^\omega \geq \mathrm{Z} + \mathcal{U}^{<\omega}$ (= Z con ω Z-universos) [ídem]
  - **Conjetura:** $\mathrm{CC}^\omega \approx \mathrm{Z} + \mathcal{U}^{<\omega}$ [M. 2021?]

El sistema $F\omega$ con universos ($F\omega^2$)

- El sistema $F\omega$ con universos ($F\omega^2$) está definido por: [M. 2001]
  - $\mathcal{S} = \{\star\} \cup \{\Box_i \ : \ i \geq 1\}$ (not. alt.: $\star = \mathsf{Prop}$, $\Box_i = \mathsf{Type}_i$))
  - $\mathcal{A} = \{(\star : \Box_1)\} \cup \{(\Box_i : \Box_{i+1}) \ : \ i \geq 1\}$
  - $\mathcal{R} = \{(\star, \star, \star)\} \cup \{(\Box_i, \star, \star) \ : \ i \geq 1\} \cup \{(\Box_i, \Box_j, \Box_{\max(i,j)}) \ : \ i, j \geq 1\}$

  $=$ $F\omega$ + jerarquía predicativa de universos a la Martin-Löf

  $=$ CC$^\omega$ sin tipos dependientes de pruebas (reglas $(\star, \Box_i, \Box_i)$, $i \geq 1$)

- Presentación estratificada en dos niveles:
  - Nivel «predicativo», formado por los términos objetos $M : T : \Box_i$ (que incluyen las proposiciones $A : \star : \Box_1$, sin depender de las pruebas)
  - Nivel «impredicativo», formado por los términos de prueba $t : A : \star$
- Resultados de fuerza teórica análogos:
  - $F\omega.3$ ($= F\omega$ con 3 universos) $>$ Z
  - $F\omega^2 \geq \mathsf{Z} + \mathcal{U}^{<\omega}$ ($=$ Z con ω Z-universos)
  - **Conjetura:** CC$^\omega \sim F\omega^2 \sim \mathsf{Z} + \mathcal{U}^{<\omega}$

El sistema λZ

- El sistema λZ está definido por: [M. 2005]
  - $\mathcal{S} = \{\star, \Box_1, \Box_2, \Box_3\}$
  - $\mathcal{A} = \{(\star : \Box_1), (\Box_1 : \Box_2), (\Box_2 : \Box_3)\}$
  - $\mathcal{R} = \{(\star, \star, \star)\} \cup \{(\Box_i, \star, \star) : i \in \{1, 2, 3\}\} \cup \{(\Box_i, \Box_j, \Box_{\max(i,j)}) : i, j \in \{1, 2\}\}$
- $\lambda Z \subset F\omega.3 \subset F\omega^2 \subset \mathsf{CC}^\omega$
- Presentación estratificada en dos niveles similar a la del sistema $F\omega^2$
  (Términos-objetos (predicativos) + términos de prueba (impredicativos))

Teorema [M. 2005]

El sistema λZ es equiconsistente a la teoría de Zermelo: $\lambda Z \sim$ (I)Z

- **Más precisamente:** existe una traducción de IZ en λZ a través de la cual λZ es una extensión conservativa de $\mathsf{IZ} + \mathsf{TC} + \mathsf{AFA}$ $(\sim \mathsf{IZ} \sim \mathsf{Z})$
  (TC = axioma de la clausura transitiva; AFA = axioma de antifundación)

El sistema V (1/2)

- El sistema V está definido por: [M. 2009]
  - $\mathcal{S} = \{ \star, \ \Box, \ \triangle \}$
  - $\mathcal{A} = \{ (\star : \Box), \ (\Box : \triangle) \}$ (notación alt. $\triangle = \Box_2 = \text{Kind}$)
  - $\mathcal{R} = \{ \underbrace{(\star, \star), \ (\Box, \star), \ (\Box, \Box)}_{F\omega}, \ (\triangle, \star) \}$

- Sistema V $=$ Sistema $U -$ regla $(\triangle, \Box)$
  $=$ Sistema $F_\omega +$ regla $(\triangle, \star)$
  $=$ Lógica de orden superior $+ \forall$ sobre todos los órdenes

Teorema [M. 2009]

$$V \sim \text{(I)Z} - \text{Infinito} \sim \text{HA/PA}$$
$$V + \texttt{Nat} \sim \text{(I)Z}$$

donde `Nat` $(: \Box)$ = género primitivo de los enteros naturales

$+$ Versión clásica con traducciones negativas (a la Gödel-Gentzen, a la Friedman)

El sistema V (2/2)

Sintaxis estratificada del sistema $V + \texttt{Nat}$

Géneros	K, L	$::=$	$\star$	$(:\Box)$
		$\mid$	κ	$(:\Box:\triangle)$
		$\mid$	$\texttt{Nat}$	$(:\Box:\triangle)$
		$\mid$	$K \to L$	$(\Box,\Box)$
Constructores	ϕ, ψ, A, B	$::=$	α	$(:K:\Box)$
		$\mid$	$\texttt{0} \mid \texttt{S} \mid \texttt{rec}_K$	
		$\mid$	$A \to B$	$(\star,\star)$
		$\mid$	$\forall \alpha : K \,.\, A$	$(\Box,\star)$
		$\mid$	$\forall \kappa : \Box \,.\, A$	$(\triangle,\star)$
		$\mid$	$\lambda \alpha : K \,.\, \phi \mid \phi\,\psi$	$(\Box,\Box)$
Pruebas	t, u	$::=$	x	$(:A:\star)$
		$\mid$	$\lambda x : A \,.\, u \mid t\,u$	$(\star,\star)$
		$\mid$	$\lambda \alpha : K \,.\, u \mid t\,\phi$	$(\Box,\star)$
		$\mid$	$\lambda \kappa : \Box \,.\, u \mid t\,K$	$(\triangle,\star)$

Regla $(\Box,\star)$: $\forall \alpha : K \,.\, A \sim$ cuantificación acotada $(\forall x \in y)\phi(x)$

Regla $(\triangle,\star)$: $\forall \kappa : \Box \,.\, A \sim$ cuantificación no acotada $\forall x\, \phi(x)$

Plan

Cálculo de construcciones: presentación en forma de PTS

- **Suertes:**

$$\begin{array}{lcl} \text{Prop} & = & \text{suerte impredicativa (proposiciones)} \\ \text{Type} & = & \text{universo predicativo (tipos de datos)} \end{array}$$

- **Axioma:**

$$\text{Prop} \ : \ \text{Type}$$

- **Reglas:**

Prop,	Prop,	Prop	(implicación, tipo flecha)
Type,	Prop,	Prop	(polimorfismo impredicativo)
Prop,	Type,	Type	(tipos dependientes de pruebas)
Type,	Type,	Type	(constructores de tipos)

Recordatorio: En la teoría de tipos de Martin-Löf, la suerte «Prop» se llama «Set», y la regla (Type, Prop, Prop) está remplazada por la regla (Type, Set, Type).

Cálculo de construcciones: sintaxis y estratificación

Definición (Términos)

$$M, N, A, B \quad ::= \quad x \quad | \quad \lambda x : A \,.\, M \quad | \quad M\, N$$
$$| \quad \text{Prop} \quad | \quad \text{Type} \quad | \quad \Pi x : A \,.\, B$$

Tipado del producto dependiente:

$$\frac{\Gamma \vdash T : s_1 \qquad \Gamma, x : T \vdash U : s_2}{\Gamma \vdash \Pi x : T \,.\, U : s_2}$$

Recordatorio: En MLTT, la suerte en la conclusión sería $\max(s_1, s_2)$

Estratificación en 4 niveles:

1. Géneros = términos T : Type
2. Constructores de tipos = términos M : T : Type
3. Tipos (proposiciones) = términos A : Prop ($\subset$ nivel 2)
4. Términos de prueba = términos M : A : Prop

Cálculo de construcciones: ejemplos de términos

1. **Géneros** $(T : \mathsf{Type})$
   - $\mathsf{Prop} \to \mathsf{Prop} \;:\; \mathsf{Type}$
   - $\Pi A : \mathsf{Prop} \,.\, A \to A \to \mathsf{Prop} \;:\; \mathsf{Type}$
2. **Constructores de tipos** $(M : T : \mathsf{Type})$
   - $\neg \;:\equiv\; \lambda A : \mathsf{Prop} \,.\, A \to \bot \;:\; \mathsf{Prop} \to \mathsf{Prop}$
   - $\lambda A, B : \mathsf{Prop} \,.\, A \wedge B \;:\; \mathsf{Prop} \to \mathsf{Prop} \to \mathsf{Prop}$
   - $\lambda A : \mathsf{Prop} \,.\, \lambda x, y : A \,.\, (x =_A y) \;:\; \Pi A : \mathsf{Prop} \,.\, A \to A \to \mathsf{Prop}$
3. **Tipos proposicionales** $(A : \mathsf{Prop})$
   - $\Pi A : \mathsf{Prop} \,.\, A \to A \;:\; \mathsf{Prop}$
   - $\bot \;:\equiv\; \Pi X : \mathsf{Prop} \,.\, X \;:\; \mathsf{Prop}$
   - $A \wedge B \;:\equiv\; \Pi X : \mathsf{Prop} \,.\, (A \to B \to X) \to X \;:\; \mathsf{Prop} \quad (A, B : \mathsf{Prop})$
   - $x =_A y \;:\equiv\; \Pi Z : (A \to \mathsf{Prop}) \,.\, Z\, x \to Z\, y \;:\; \mathsf{Prop} \quad (A : \mathsf{Prop},\ x, y : A)$
4. **Términos de prueba** $(M : A : \mathsf{Prop})$
   - $\lambda A : \mathsf{Prop} \,.\, \lambda x : A \,.\, x \;:\; \Pi A : \mathsf{Prop} \,.\, A \to A$
   - $\lambda A : \mathsf{Prop} \,.\, \lambda x : A \,.\, \lambda Z : (A \to \mathsf{Prop}) \,.\, \lambda z : Z\, x \,.\, z$
     $:\; \Pi A : \mathsf{Prop} \,.\, \Pi x : A \,.\, x =_A x$

Historia (1/6)

- Inicialmente, el cálculo de construcciones fue diseñado como un marco lógico polimórfico similar al marco lógico de Martin-Löf, pero con una suerte Set/Prop impredicativa:

$$(\mathsf{Type}, \mathsf{Set}, \mathsf{Type}) \quad \rightsquigarrow \quad (\mathsf{Type}, \mathsf{Prop}, \mathsf{Prop})$$

- En la primera versión del sistema, la suerte Prop era a la vez:
  - La suerte de los tipos y de las estructuras de datos
    (usando las codificaciones del sistema F)

$$\mathtt{Nat} \;:\; \mathsf{Prop} \;:\equiv\; \Pi Z : \mathsf{Prop} \,.\, Z \to (Z \to Z) \to Z$$

  - La suerte de las proposiciones y de las pruebas
    (usando las codificaciones del sistema NJ2)

$$\begin{aligned} \mathtt{Eq} \;:\; & \Pi A : \mathsf{Prop} \,.\, A \to A \to \mathsf{Prop} \;:\equiv \\ & \lambda A : \mathsf{Prop} \,.\, \lambda x, y : A \,.\, \Pi Z : A \to \mathsf{Prop} \,.\, Z\, x \to Z\, y \\ \mathtt{refl} \;:\; & \Pi A : \mathsf{Prop} \,.\, \Pi x : A \,.\, \mathtt{Eq}\, A\, x\, x \;:\equiv \\ & \lambda A : \mathsf{Prop} \,.\, \lambda x : A \,.\, \lambda Z : A \to \mathsf{Prop} \,.\, \lambda z : Z\, x \,.\, z \end{aligned}$$

Historia (2/6)

- Ejemplo de formalización en CC: [Paulin-Mohring 1989]

$$\begin{aligned}
\bot &: \mathsf{Prop} \quad :\equiv \quad \Pi X : \mathsf{Prop} \,.\, X \\
(\neg) &: \mathsf{Prop} \to \mathsf{Prop} \quad :\equiv \quad \lambda A : \mathsf{Prop} \,.\, A \to \bot \\
(\times) &: \mathsf{Prop} \to \mathsf{Prop} \to \mathsf{Prop} \quad :\equiv \\
&\quad \lambda A, B : \mathsf{Prop} \,.\, \Pi X : \mathsf{Prop} \,.\, (A \to B \to X) \to X \\
\texttt{pair} &: \Pi A, B : \mathsf{Prop} \,.\, A \to B \to A \times B \quad :\equiv \\
&\quad \lambda A, B : \mathsf{Prop} \,.\, \lambda x : A \,.\, \lambda y : B \,.\, \lambda X : \mathsf{Prop} \,.\, \lambda f : A \to B \to X \,.\, f\, x\, y \\
\texttt{fst} &: \Pi A, B : \mathsf{Prop} \,.\, A \times B \to A \quad :\equiv \\
&\quad \lambda A, B : \mathsf{Prop} \,.\, \lambda p : A \times B \,.\, p\, A\, (\lambda x : A \,.\, \lambda _ : B \,.\, x) \\
\texttt{snd} &: \Pi A, B : \mathsf{Prop} \,.\, A \times B \to B \quad :\equiv \\
&\quad \lambda A, B : \mathsf{Prop} \,.\, \lambda p : A \times B \,.\, p\, B\, (\lambda _ : A \,.\, \lambda y : B \,.\, y) \\
\texttt{Eq} &: \Pi A : \mathsf{Prop} \,.\, A \to A \to \mathsf{Prop} \quad :\equiv \\
&\quad \lambda A : \mathsf{Prop} \,.\, \lambda x, y : A \,.\, \Pi Z : A \to \mathsf{Prop} \,.\, Z\, x \to Z\, y \\
\texttt{refl} &: \Pi A : \mathsf{Prop} \,.\, \Pi x : A \,.\, \texttt{Eq}\, A\, x\, x \quad :\equiv \\
&\quad \lambda A : \mathsf{Prop} \,.\, \lambda x : A \,.\, \lambda Z : A \to \mathsf{Prop} \,.\, \lambda z : Z\, x \,.\, z
\end{aligned}$$

Historia (3/6)

- Ejemplo de formalización en CC (continuación):

$$\begin{aligned}
\texttt{Nat} &: \mathsf{Prop} \equiv \Pi X : \mathsf{Prop} \,.\, X \to (X \to X) \to X \\
\texttt{0} &: \texttt{Nat} \equiv \lambda X : \mathsf{Prop} \,.\, \lambda x : X \,.\, \lambda f : X \to X \,.\, x \\
\texttt{S} &: \texttt{Nat} \to \texttt{Nat} \equiv \\
&\quad \lambda n : \texttt{Nat} \,.\, \lambda X : \mathsf{Prop} \,.\, \lambda x : X \,.\, \lambda f : X \to X \,.\, f\,(n\,X\,x\,f) \\
\texttt{NAT} &: \texttt{Nat} \to \mathsf{Prop} \equiv \\
&\quad \lambda n : \texttt{Nat} \,.\, \Pi Z : \texttt{Nat} \to \mathsf{Prop} \,. \\
&\qquad Z\,\texttt{0} \to (\Pi x : \texttt{Nat} \,.\, Z\,x \to Z\,(\texttt{S}\,x)) \to Z\,n \\
\texttt{NAT_0} &: \texttt{NAT}\,\texttt{0} \equiv \\
&\quad \lambda Z : \texttt{Nat} \to \mathsf{Prop} \,.\, \lambda z : Z\,\texttt{0} \,.\, \lambda f : (\Pi x : \texttt{Nat} \,.\, Z\,x \to Z\,(\texttt{S}\,x)) \,.\, z \\
\texttt{NAT_S} &: \Pi n : \texttt{Nat} \,.\, \texttt{NAT}\,n \to \texttt{NAT}\,(\texttt{S}\,n) \equiv \\
&\quad \lambda n : \texttt{Nat} \,.\, \lambda h : \texttt{NAT}\,n \,.\, \lambda Z : \texttt{Nat} \to \mathsf{Prop} \,. \\
&\qquad \lambda z : Z\,\texttt{0} \,.\, \lambda f : (\Pi x : \texttt{Nat} \,.\, Z\,x \to Z\,(\texttt{S}\,x)) \,.\, f\,n\,(h\,Z\,z\,f) \\
\texttt{NAT_ind} &: \Pi P : \texttt{Nat} \to \mathsf{Prop} \,. \\
&\qquad P\,\texttt{0} \to (\Pi x : \texttt{Nat} \,.\, \texttt{NAT}\,x \to P\,x \to P\,(\texttt{S}\,x)) \to \\
&\qquad\quad \Pi x : \texttt{Nat} \,.\, \texttt{NAT}\,x \to P\,x \\
&\quad \equiv \text{(ejercicio)}
\end{aligned}$$

Historia (4/6)

- Con el fin de diseñar un mecanismo de extracción de programas (hacia $F\omega$ o Caml), las versiones siguientes de Coq (años 1990) distinguían dos suertes impredicativas:
  - una suerte Set de los tipos de datos, con un contenido computacional relevante (i.e. preservado durante la extracción)
  - una suerte Prop de las proposiciones, con un contenido computacional irrelevante (i.e. borrado durante la extracción)
- En paralelo, se extendió el sistema con una jerarquía de universos predicativos Type_i $(i \geq 1)$ a la Martin-Löf:

Suertes: $\mathsf{Prop}, \quad \mathsf{Set}, \quad \mathsf{Type}_i$

Axiomas: $(\mathsf{Prop} : \mathsf{Type}_1), \quad (\mathsf{Set} : \mathsf{Type}_1), \quad (\mathsf{Type}_i : \mathsf{Type}_{i+1})$

Reglas:

$$\begin{array}{ll} (\mathsf{Prop}, \mathsf{Prop}, \mathsf{Prop}), & (\mathsf{Set}, \mathsf{Set}, \mathsf{Set}), \\ (\mathsf{Set}, \mathsf{Prop}, \mathsf{Prop}), & (\mathsf{Prop}, \mathsf{Set}, \mathsf{Set}), \\ (\mathsf{Type}_i, \mathsf{Prop}, \mathsf{Prop}), & (\mathsf{Type}_i, \mathsf{Set}, \mathsf{Set}), \\ (\mathsf{Prop}, \mathsf{Type}_i, \mathsf{Type}_i), & (\mathsf{Set}, \mathsf{Type}_i, \mathsf{Type}_i), \\ (\mathsf{Type}_i, \mathsf{Type}_j, \mathsf{Type}_{\max(i,j)}) & (i, j \geq 1) \end{array}$$

Historia (5/6)

- Sin embargo, los tipos de datos (en Set) codificados en el estilo del sistema F tenían múltiples defectos:
  1. Los correspondientes principios de inducción no eran derivables
     (corrección: introducir predicados de relativización, por ej. `NAT`)
  2. Los tipos $A \times B$ y $A + B$ tenían demasiados elementos
     (corrección: ídem)
  3. La proposición $\mathtt{S}\, x \neq \mathtt{0}$ no era derivable
     (corrección: añadir un axioma, sin contenido computacional)

- Para corregir estos defectos, Coquand y Paulin-Mohring remplazaron las definiciones impredicativas (estilo sistema F) por un mecanismo primitivo de definiciones inductivas (estilo Martin-Löf):

  $\Rightarrow$ Cálculo de construcciones inductivas (CIC)

```
Inductive nat : Set :=
|  0 : nat
|  S : nat → nat
```

Historia (6/6)

- En 2002, Chicli descubrió que la suerte Set (impredicativa y con contenido computacional relevante[1]) era incompatible con la ley del tercer excluido y el axioma de elección:

Teorema [Chicli 2002]

$\mathsf{CIC} + \mathsf{EM} + \mathsf{AC} \vdash \bot$, donde:

- CIC = Cálculo de construcciones inductivas (con Set impredicativa)
- EM $\equiv \forall A : \mathsf{Prop} \,.\, A \vee \neg A$
- AC $\equiv \forall A, B : \mathsf{Set} \,.\, \forall R : A \to B \to \mathsf{Set} \,.$
  $(\forall x : A \,.\, \exists y : B \,.\, R\, x\, y) \to \exists f : A \to B \,.\, \forall x : A \,.\, R\, x\, (f\, x)$

- Para mantener la compatibilidad del sistema Coq con la matemática clásica, se eliminó la suerte Set impredicativa

  $\Rightarrow$ *Predicative Inductive Calculus of Constructions* (pCIC)

  En pCIC, se mantiene «Set» como abreviatura para «Type_0»

[1] Al contrario de Prop, en la cual siempre se puede suponer que todas las pruebas son iguales (*proof irrelevance*): $\forall A : \mathsf{Prop} \,.\, \forall x, y : A \,.\; x =_A y$

El cálculo de construcciones con universos (CC^ω)

CC^ω = PTS subyacente del Cálculo de construcciones inductivas (pCIC)

- **Suertes:**

Prop	=	suerte impredicativa (proposiciones)
$\text{Type}_i \quad (i \geq 1)$	=	$i$ésimo universo predicativo (tipos de datos)

- **Axiomas:**

  $$\text{Prop} : \text{Type}_1$$
  $$\text{Type}_i : \text{Type}_{i+1}$$

- **Reglas:**

Prop,	Prop,	Prop	(implicación, tipo flecha)
Type_i,	Prop,	Prop	(polimorfismo impredicativo)
Prop,	Type_i,	Type_i	(tipos dependientes de pruebas)
Type_i,	Type_j,	$\text{Type}_{\max(i,j)}$	(producto dependiente predicativo)

En la implementación (Coq), se mantiene una suerte Set $:\equiv$ Type_0 (predicativa)

Extensión con una relación de cumulatividad $(\mathrm{CC}^{\omega}_{\preceq})$

- Se introduce una relación de cumulatividad $A \preceq B$ sobre los tipos:

$$\frac{}{\mathsf{Prop} \preceq \mathsf{Type}_1} \qquad \frac{}{\mathsf{Type}_i \preceq \mathsf{Type}_j}\ (i \leq j)$$

$$\frac{A \cong A'}{A \preceq A'} \qquad \frac{A \preceq B \qquad B \preceq C}{A \preceq C} \qquad \frac{B \preceq C}{\Pi x : A \,.\, B \preceq \Pi x : A \,.\, C}$$

Formalmente, la relación de cumulatividad $A \preceq B$ (que contiene la relación de conversión $A \cong A'$) está definida sobre los términos no tipados

- Se remplaza la regla de conversión por una regla de subtipado:

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash B : s}{\Gamma \vdash M : B}\ (A \preceq B)$$

- $\mathrm{CC}^{\omega}_{\preceq}$ cumple la *subject reduction* y la normalización fuerte

  Además la relación $A \preceq B$ es decidible sobre los tipos A, B bien formados

- Se pierde la unicidad del tipo, pero siempre existe un tipo principal (= un mínimo tipo con respecto a $\preceq$)

Extensión con Σ-tipos (*Extended Calculus of Constructions*)

ECC $=$ CC $+$ universos $+$ cumulatividad $+$ Σ-tipos [Luo 1990]

Definición (Sintaxis y reducción de ECC)

$$M, N, A, B \quad ::= \quad \cdots \quad | \quad \Sigma x : A \,.\, B \quad | \quad \langle M, N \rangle_C \quad | \quad \pi_1(M) \quad | \quad \pi_2(M)$$

Reducción: $\beta \quad + \quad \pi_1(\langle M, N \rangle_C) \succ M \quad + \quad \pi_2(\langle M, N \rangle_C) \succ N$

$$\frac{\Gamma \vdash A : \mathsf{Type}_i \qquad \Gamma, x : A \vdash B : \mathsf{Type}_i}{\Gamma \vdash \Sigma x : A \,.\, B : \mathsf{Type}_i}$$

$$\frac{\Gamma \vdash \Sigma x : A \,.\, B : \mathsf{Type}_i \qquad \Gamma \vdash M : A \qquad \Gamma \vdash N : B[x := M]}{\Gamma \vdash \langle M, N \rangle_{\Sigma x : A.B} : \Sigma x : A \,.\, B}$$

$$\frac{\Gamma \vdash M : \Sigma x : A \,.\, B}{\Gamma \vdash \pi_1(M) : A} \qquad \frac{\Gamma \vdash M : \Sigma x : A \,.\, B}{\Gamma \vdash \pi_2(M) : B[x := \pi_1(M)]}$$

- **Obs.:** La cumulatividad $\mathsf{Prop} \preceq \mathsf{Type}_i$ permite formar el tipo

$$\{x : A \mid P\,x\} \;:\equiv\; \Sigma x : A \,.\, P\,x \qquad \text{con} \quad A : \mathsf{Type}_i, \quad P : A \to \mathsf{Prop}$$

El cálculo de construcciones inductivas (pCIC) (1/2)

El cálculo de construcciones inductivas (pCIC) es $\mathrm{CC}^{\omega}_{\preceq}$ extendido con:

1. **Un mecanismo de definición de familias inductivas:**

$$\begin{array}{l}\texttt{Inductive}\ I\ (\vec{a} : \vec{A})\ :\ \Pi\vec{x} : \vec{T}\,.\,s\ :=\\ |\quad c_1\ :\ \Pi\vec{y}_1 : \vec{T}_1\,.\ \ I\,\vec{a}\,\vec{N}_1\\ \qquad\vdots\\ |\quad c_n\ :\ \Pi\vec{y}_n : \vec{T}_n\,.\ \ I\,\vec{a}\,\vec{N}_n\end{array}$$

(posiblemente mutuas) donde:

- $\vec{a} : \vec{A}$ son los parámetros de I
- $\vec{x} : \vec{T}$ son los argumentos de I
- $c_i\ :\ \Pi\vec{y}_i : \vec{T}_i\,.\,I\,\vec{a}\,\vec{N}_i\ \ (1 \leq i \leq n)$ son los constructores de I

$+$ condiciones sobre s y las suertes de $\vec{T}$, $\vec{T}_i$

$+$ condiciones de positividad sobre los tipos $\vec{T}_i$

El cálculo de construcciones inductivas (pCIC) (2/2)

2. **Un mecanismo de *pattern-matching*:**

$$\begin{array}{l}\texttt{match}\ M\ (\texttt{as}\ x)^?\ (\texttt{return}\ T)^?\ \texttt{with}\\ \mid\ \ c_1\ \vec{x}_1\ \Rightarrow\ M_1(\vec{x}_1)\\ \qquad\vdots\\ \mid\ \ c_n\ \vec{x}_n\ \Rightarrow\ M_n(\vec{x}_n)\\ \texttt{end}\end{array}$$

3. **Un mecanismo de definición de funciones recursivas:**

$$\texttt{Fixpoint}\ f\ (\vec{x} : \vec{A})\ \{\texttt{struct}\ x_i\}\ :\ B\ :=\ M(f, \vec{x})$$

(+ condiciones de decrecimiento estructural)

4. + definición simple («Definition»)
   + declaración de axioma («Axiom»)
   + definición de familias coinductivas (Coinductive)
   + definición de funciones correcursivas (Cofixpoint)

Plan

El diccionario

Los términos (cerrados y en forma normal) de CC constan de:

- funciones $\lambda x : A \,.\, M$
- tipos funcionales $\Pi x : A \,.\, B$
- suertes (tipos de tipos) Prop, Type

Teoría de tipos	**Teoría de conjuntos**
función (λ)	función ($\mapsto$)
tipo	conjunto
tipo funcional	conjunto de funciones
suerte (tipo de tipos)	conjunto de conjuntos
$M : A$	$[\![M]\!] \in [\![A]\!]$
relación de tipado	relación de pertenencia

Esquema general de la interpretación

Función de interpretación $[\![\_]\!]\_ \;:\; \mathsf{Term} \times \mathsf{Val}_{\mathcal{M}} \rightharpoonup \mathcal{M}$ (parcial)

$$
\begin{aligned}
[\![x]\!]_\rho &:= \rho(x) \\
[\![\lambda x : A \,.\, M]\!]_\rho &:= (v \in [\![A]\!]_\rho \mapsto [\![M]\!]_{\rho;x\leftarrow v}) \\
[\![\Pi x : A \,.\, B]\!] &:= \prod_{v \in [\![A]\!]_\rho} [\![B]\!]_{\rho;x\leftarrow v} \\
[\![\mathsf{Prop}]\!]_\rho,\ [\![\mathsf{Type}]\!]_\rho &:= \text{a definir}
\end{aligned}
$$

- Term = conjunto de los términos (no tipados)
- $\mathcal{X}$ = conjunto de las variables
- $\mathcal{M}$ = espacio de las denotaciones (a definir)
- $\mathsf{Val}_{\mathcal{M}} = \mathcal{X} \underset{\text{fin}}{\rightharpoonup} \mathcal{M} =$ espacio de las valuaciones en $\mathcal{M}$

Propiedades de clausura de las suertes

- Formación de un tipo en Prop:

$$\frac{\vdash A : s \qquad x : A \vdash B : \mathsf{Prop}}{\vdash \Pi x : A \,.\, B : \mathsf{Prop}} \qquad (s \in \{\mathsf{Prop}, \mathsf{Type}\})$$

- Formación de un tipo en Type:

$$\frac{}{\vdash \mathsf{Prop} : \mathsf{Type}} \qquad \frac{\vdash A : \mathsf{Prop}}{\vdash A : \mathsf{Type}}\ (*)$$

$$\frac{\vdash A : s \qquad x : A \vdash B : \mathsf{Type}}{\vdash \Pi x : A \,.\, B : \mathsf{Type}} \qquad (s \in \{\mathsf{Prop}, \mathsf{Type}\})$$

$(*)$ Regla de cumulatividad Prop $\subset$ Type, opcional

Dificultad: Circularidad inducida por la impredicatividad de Prop

Modelación de la impredicatividad

$$\frac{\vdash A : s \qquad x : A \vdash B : \mathsf{Prop}}{\vdash \Pi x : A \,.\, B : \mathsf{Prop}} \; s \in \{\mathsf{Prop}, \mathsf{Type}\}$$

- **Traducción conjuntista:** Hallar un predicado $Prop(\_)$ tal que:

Para toda familia de conjuntos $(B_x)_{x \in A}$ queremos que:

$$\big((\forall x \in A)\ Prop(B_x)\big) \quad \Rightarrow \quad Prop\Big(\prod_{x \in A} B_x\Big)$$

... independientemente del tamaño del conjunto A

- **Problema:** ¿Cómo definir el predicado $Prop(X)$?

- **Una solución sencilla:** $Prop(X) \quad \equiv \quad X$ tiene 0 ou 1 elemento

  $\Rightarrow$ Semántica de la proof-irrelevance

Semántica de la *proof-irrelevance*

- Pruebas interpretadas por un único objeto: prf (a definir)
- Proposición = conjunto vacío ("falso") o unitario ("verdadero")
- $[\![\text{Prop}]\!] := \{\varnothing, \{\text{prf}\}\} = \mathfrak{P}(\{\text{prf}\})$ (clásicamente)

Lema (Interpretación conjuntista de la impredicatividad)

Si A es un conjunto cualquiera y si $(B_x)_{x\in A}$ es una familia tal que $B_x \in \{\varnothing, \{\text{prf}\}\}$ para todo $x \in A$, entonces:

$$\prod_{x\in A} B_x = \begin{cases} \varnothing & \text{si } B_x = \varnothing \text{ para algún } x \in A \\ \{(x_{\in A} \mapsto \text{prf})\} & \text{si } B_x = \{\text{prf}\} \text{ para todo } x \in A \end{cases}$$

- **Problema:** $(x_{\in A} \mapsto \text{prf}) \neq \text{prf}$ (en general)
- No se puede definir $[\![\text{Prop}]\!]$ como la clase (propia) formada por el conjunto vacío y todos los conjuntos unitarios

  $\Rightarrow$ Necesidad de identificar todas las funciones $(x_{\in A} \mapsto \text{prf})$ con prf

Funciones: codificación estándar (en teoría de conjuntos)

- **Pares:**

$$(x, y) \quad := \quad \{\{x\}, \{x, y\}\}$$

- **Funciones:**

$$\begin{aligned}
f \text{ función} \quad &:\equiv \quad \forall c\ (c \in f \ \Rightarrow\ \exists x\ \exists y\ c = (x, y)) \ \wedge \\
&\qquad \forall x\ \forall y\ \forall y'\ ((x, y) \in f \wedge (x, y') \in f \Rightarrow y = y') \\
\mathrm{dom}(f) \quad &:= \quad \{x \mid \exists y\ (x, y) \in f\} \\
(x_{\in D} \mapsto E_x) \quad &:= \quad \{(x, y) \mid x \in D \wedge y = E_x\} \\
f(x) \quad &:= \quad \bigcup\{y \ \mid \ (x, y) \in f\}
\end{aligned}$$

- **Producto dependiente:**

$$\prod_{x \in A} B_x \quad := \quad \left\{ f \ \mid \ f \text{ función} \wedge \mathrm{dom}(f) = A \wedge (\forall x \in A)\ f(x) \in B_x \right\}$$

Funciones: codificación alternativa

- **Funciones:**

$$\begin{array}{rcl} f \text{ función} & :\equiv & \forall c\ (c \in f\ \Rightarrow\ \exists x\ \exists z\ c = (x,z)) \\ \text{supp}(f) & := & \{x \mid \exists z\ (x,z) \in f\} \qquad \text{(soporte)} \\ (x_{\in D} \mapsto E_x) & := & \{(x,z) \mid x \in D \wedge z \in E_x\} \\ f(x) & := & \{z \mid (x,z) \in f\} \end{array}$$

- **Producto dependiente:**

$$\prod_{x \in A} B_x \ := \ \left\{ f \mid f \text{ función} \wedge \text{supp}(f) \subseteq A \wedge \forall x \in A\ \ f(x) \in B_x \right\}$$

Tomando prf $:= \varnothing$:

1. Para todo conjunto A tenemos que: $(x_{\in A} \mapsto \text{prf}) = \text{prf}$
2. Si $B_x \in \{\varnothing, \{\text{prf}\}\}$ para todo $x \in A$, entonces $\prod_{x \in A} B_x \in \{\varnothing, \{\text{prf}\}\}$

Interpretación de la suerte Type

$$\frac{}{\vdash \mathsf{Prop} : \mathsf{Type}} \qquad \frac{\vdash A : \mathsf{Prop}}{\vdash A : \mathsf{Type}} \qquad \frac{\vdash A : s \qquad x : A \vdash B : \mathsf{Type}}{\vdash \Pi x : A \,.\, B : \mathsf{Type}}$$

Interpretación de los tipos por conjuntos finitos:

- $[\![\mathsf{Type}]\!] := \mathsf{HF}\ (= V_\omega)$ (conj. de los conjuntos hereditariamente finitos)
- $[\![\mathsf{Prop}]\!] := \{\varnothing, \{\mathsf{prf}\}\} \in \mathsf{HF}$ (y $[\![\mathsf{Prop}]\!] \subset \mathsf{HF}$)
- HF está cerrado por los productos $\prod_{x \in A} B_x$

Modelo booleano:

$$\mathcal{M} := \mathsf{HF} \cup \{\mathsf{HF}\}$$

Interpretación de los términos

Definición de la función $(M, \rho) \mapsto [\![M]\!]_\rho \quad : \quad \mathsf{Term} \times \mathsf{Val}_{\mathcal{M}} \rightharpoonup \mathcal{M}$

$$
\begin{aligned}
[\![x]\!]_\rho &:= \rho(x) \\
[\![\mathsf{Prop}]\!]_\rho &:= \{\varnothing, \{\mathsf{prf}\}\} \\
[\![\mathsf{Type}]\!]_\rho &:= \mathsf{HF} \\
[\![\lambda x : A \,.\, M]\!]_\rho &:= (v \in [\![A]\!]_\rho \mapsto [\![M]\!]_{\rho; x \leftarrow v}) \\
[\![\Pi x : A \,.\, B]\!] &:= \prod_{v \in [\![A]\!]_\rho} [\![B]\!]_{\rho; x \leftarrow v}
\end{aligned}
$$

Extensión a los contextos: conjunto $[\![\Gamma]\!]$ de las valuaciones adaptadas

$$[\![\Gamma]\!] := \left\{ \rho \in \mathsf{Val}_{\mathcal{M}} \mid \forall (x : A) \in \Gamma \quad \rho(x) \in [\![A]\!]_\rho \right\}$$

Proposición (Invariante de corrección)

Si $\Gamma \vdash M : A$ y $\rho \in [\![\Gamma]\!]$, entonces $[\![M]\!]_\rho \in [\![A]\!]_\rho$

Un invariante de corrección problemático

El modelo no depende de las reglas de tipado...

... pero la dificultad de la prueba del invariante de corrección sí depende fuertemente de la formulación de las reglas

1. **Presentación con regla de conversión no tipada (estilo PTS)**
   - CC puro, sin cumulatividad Prop $\subset$ Type:
     El invariante se demuestra usando una presentación estratificada
   - CC con cumulatividad Prop $\subset$ Type:
     Ya no se sabe demostrar el invariante [Miquel-Werner 2003]
2. **Presentación con juicio de igualdad (a la Martin-Löf)**
   - El invariante de corrección se demuestra sin dificultad
   - ¿Qué hay de la equivalencia con la primera presentación?

Solución: Probar la equivalencia de ambas presentaciones

- Caso particular de los PTS funcionales: Adams 2006
- Caso general de los PTS (sin cumulatividad): Siles 2010

Lectura en las tripas del modelo

Principio: El cálculo de las denotaciones en el modelo permite demostrar resultados negativos sobre la sintaxis

(Resultado negativo $=$ Algo no es demostrable, derivable, etc.)

En particiliar, para todo tipo cerrado A : Prop:

$$\begin{array}{lll} [\![A]\!] = \varnothing & \Rightarrow & A \text{ no es demostrable en CC} \\ & & \equiv \neg A \text{ es consistente relativamente a CC} \\ & & \text{(se puede añadir } \neg A \text{ como axioma a CC)} \\ [\![A]\!] = \{\mathsf{prf}\} & \Rightarrow & \neg A \text{ no es demostrable en CC} \\ & & \equiv A \text{ es consistente relativamente a CC} \\ & & \text{(se puede añadir } A \text{ como axioma a CC)} \end{array}$$

Ejemplo: $$[\![\bot]\!] = [\![\Pi X : \mathsf{Prop} \,.\, X]\!] = \prod_{X \in \{\varnothing, \{\mathsf{prf}\}\}} X = \varnothing$$

$\Rightarrow$ $\bot$ no es demostrable $\equiv$ CC es consistente

Denotaciones de algunos términos cerrados

$$
\begin{array}{rcl}
[\![\lambda A, B : \mathsf{Prop} \,.\, A \to B]\!] & = & \text{tabla de verdad de la implicación} \\
[\![\lambda A, B : \mathsf{Prop} \,.\, A \wedge B]\!] & = & \text{tabla de verdad de la conjunción} \\
[\![\lambda A, B : \mathsf{Prop} \,.\, A \vee B]\!] & = & \text{tabla de verdad de la disyunción} \\
\ldots & & \ldots \\
[\![\Pi A : \mathsf{Prop} \,.\, (A \vee \neg A)]\!] & = & \{\mathsf{prf}\} \quad \text{(consistencia del tercero excluido)}
\end{array}
$$

Escribiendo $\mathsf{nat} :\equiv \Pi X : \mathsf{Prop} \,.\, X \to (X \to X) \to X$, tenemos que:

$$
\begin{array}{rcll}
[\![\mathsf{nat}]\!] & = & \{\mathsf{prf}\} & \\
[\![0 =_{\mathsf{nat}} 1]\!] & = & \{\mathsf{prf}\} & \text{(pues } 0 = 1 = \mathsf{prf} \text{ en el modelo)}
\end{array}
$$

De modo análogo se justifica la consistencia relativa de...

- Extensionalidad proposicional $\quad (A \Leftrightarrow B) \quad \Rightarrow \quad (A =_{\mathsf{Prop}} B)$
- Extensionalidad funcional $\quad (\forall x \; f(x) = g(x)) \quad \Rightarrow \quad f = g$
- *Proof-irrelevance*, axioma de elección, axioma K de Streicher, etc.

Modelo booleano: el balance

- Modelo mínimo (numerable: mismo tamaño que la sintaxis)
- Prueba barata de la consistencia de CC (+ muchos axiomas)
- Desaparición del cálculo en el modelo $\Rightarrow$
  desaparición de los problemas de normalización

También demuestra las limitaciones de CC (en su versión de base)

- Indiscernibilidad (sintáctica) de las pruebas
- Ningún tipo infinito (en el modelo todos los tipos son finitos)

Sugiere algunas extensiones de la sintaxis:

- Jerarquía de universos Type_i (por encima de Type)
- Tipos inductivos primitivos (estructuras de datos)

La jerarquía de los universos predicativos

$$\frac{}{\vdash \mathsf{Prop} : \mathsf{Type}_1} \qquad \frac{}{\vdash \mathsf{Type}_i : \mathsf{Type}_{i+1}} \qquad \frac{\vdash A : \mathsf{Type}_i}{\vdash A : \mathsf{Type}_{i+1}}$$

$$\frac{\vdash A : \mathsf{Type}_i \qquad x : A \vdash B : \mathsf{Type}_i}{\vdash \Pi x : A \,.\, B : \mathsf{Type}_i} \qquad (i \geq 1)$$

Jerarquía de universos modelada por une familia $(\mathcal{U}_i)_{i \geq 1}$ tal que:

- $\mathcal{U}_i \in \mathcal{U}_{i+1}$
- $\mathcal{U}_i \subset \mathcal{U}_{i+1}$
- $\mathcal{U}_i$ cerrado por los productos $\prod_{x \in A} B_x$

Se mantienen $[\![\mathsf{Prop}]\!] := \{\varnothing, \{\mathsf{prf}\}\}$ y $[\![\mathsf{Type}_1]\!] := \mathsf{HF} = \mathcal{U}_1$

Pero para modelar Type_i $(i \geq 2)$ se utilizan ZF-universos encajados

ZF-universos

Definición (ZF-universo)

Un ZF-universo es un conjunto $\mathcal{U}$ tal que:

1. $\mathcal{U}$ es transitivo $\quad (i.e.\ X \in \mathcal{U} \ \Rightarrow\ X \subseteq \mathcal{U})$
2. $\mathbb{N} \in \mathcal{U}$
3. $X \in \mathcal{U} \ \Rightarrow\ \mathfrak{P}(X) \in \mathcal{U}$
4. $\forall (Y_x)_{x\in X} \qquad X \in \mathcal{U},\quad Y_x \in \mathcal{U} \ \Rightarrow\ \bigcup_{x\in X} Y_x \in \mathcal{U}$

- Cada ZF-universo $\mathcal{U}$ está cerrado por producto dependiente:

Si $A \in \mathcal{U}$ y $B_x \in \mathcal{U}$ para todo $x \in A$, alors $\prod_{x\in A} B_x \in \mathcal{U}$.

(Y más generalmente por todas las construcciones de ZF)

- Se interpreta la jerarquía Type_i $(i \geq 2)$ por una familia de ZF-universos encajados: $\qquad \mathcal{U}_2 \in \mathcal{U}_3 \in \mathcal{U}_4 \in \mathcal{U}_5 \in \cdots$

Para Type_1, basta con tomar $[\![\mathsf{Type}_1]\!] = \mathsf{HF}$

ZF-universos y cardinales inaccesibles

- Cada ZF-universo es un modelo de ZF, cuya existencia no puede ser demostrada en ZF $\Rightarrow$ necesidad de suponer su existencia
- Los ZF-universos están vinculados con los cardinales inaccesibles:

Definición (Cardinal inaccesible)

Un cardinal inaccesible es un cardinal μ tal que:

1. $\aleph_0 < \mu$
2. $\kappa < \mu \quad \Rightarrow \quad 2^\kappa < \mu$
3. $\lambda < \mu, \quad \kappa_\alpha < \mu$ (para todo $\alpha < \lambda$) $\quad \Rightarrow \quad \sup_{\alpha<\lambda} \kappa_\alpha < \mu$

Lema (Equivalencia entre ZF-universos y cardinales inaccesibles)

- Si $\mathcal{U}$ es un ZF-universo, entonces su cardinal es inaccesible
- Si μ es un cardinal inaccesible, entonces V_μ es un ZF-universo

  Recordatorio: $(V_\alpha)_{\alpha \in On}$ = jerarquía cumulativa de la teoría de conjuntos

- **Axioma:** Existen infinitos cardinales inaccesibles

Interpretación de los términos de $\mathrm{CC}^{\omega}_{\preceq}$

Definición de la función $(M, \rho) \mapsto [\![M]\!]_\rho \quad : \quad \mathsf{Term} \times \mathsf{Val}_{\mathcal{M}} \rightharpoonup \mathcal{M}$

$$
\begin{aligned}
[\![x]\!]_\rho &:= \rho(x) \\
[\![\mathsf{Prop}]\!]_\rho &:= \{\varnothing, \{\mathsf{prf}\}\} \\
[\![\mathsf{Type}_i]\!]_\rho &:= \begin{cases} \mathsf{HF} & \text{si} \quad i = 1 \\ \mathcal{U}_i & \text{si} \quad i \geq 2 \end{cases} \\
[\![\lambda x : A \,.\, M]\!]_\rho &:= (v \in [\![A]\!]_\rho \mapsto [\![M]\!]_{\rho; x \leftarrow v}) \\
[\![\Pi x : A \,.\, B]\!] &:= \prod_{v \in [\![A]\!]_\rho} [\![B]\!]_{\rho; x \leftarrow v}
\end{aligned}
\qquad \left(\mathcal{M} = \bigcup_{i \in \mathbb{N}} \mathcal{U}_i \right)
$$

Proposición (Invariante de corrección en $\mathrm{CC}^{\omega}_{\preceq}$)

Si $\Gamma \vdash M : A$ y $\rho \in [\![\Gamma]\!]$, entonces $[\![M]\!]_\rho \in [\![A]\!]_\rho$

- Mismas dificultades que en CC
- Ningún tipo infinito en Type_1 (en $\mathrm{CC}^{\omega}_{\preceq}$ sin tipos inductivos)

¿Se necesitan los cardinales inaccesibles?

- La hipótesis de inaccesibilidad: una hipótesis simple y estándar en teoría de conjuntos / teoría de modelos
- Ingrediente de la teoría de conjuntos de Tarski-Grothendieck:

Axioma de Grothendieck:

Para todo conjunto X, existe un ZF-universo $\mathcal{U} \notin X$

(Permite construir categorías de categorías de categorías...)

Pregunta: ¿Existe un modelo en ZF? (sin cardinales inaccesibles)

- Para $\mathrm{CC}^{\omega}_{\preceq}$ (sin tipos inductivos): Sí [Melliès-Werner 1997]
  ... pero la construction del modelo es mucho más difícil
- **Conjetura:** $\mathrm{CC}^{\omega}_{\preceq} \approx \mathrm{Z} + \mathrm{ZU}^{<\omega}$ [M. 2021?]
- Para CIC (Coq): ?

Plan

Normalización fuerte de $\mathrm{CC}^{\omega}_{\prec}$: las líneas generales

- Construcción de un modelo de normalización conjuntista de $\mathrm{CC}^{\omega}_{\prec}$ muy similar al modelo booleano
- Dos diferencias esenciales:
  1. La denotación de cada tipo encapsula información de reducibilidad para mantener el invariante de normalización

     En particular:
     - la denotación de Prop contiene todos los candidatos de reducibilidad
     - la suerte Type_1 está interpretada por el primer ZF-universo $\mathcal{U}_1$
  2. Todos los tipos (en el modelo) están habitados

     Para interpretar todos los contextos, inclusive inconsistentes

Se utiliza aquí una variante de la prueba presentada en [Altenkirch 1993]

Normalización fuerte (1/4)

- $\mathcal{CR}$ = conjunto de los candidatos de reducibilidad (a definir)
- Cada tipo está interpretado por un $\mathcal{CR}$-conjunto:

Definición ($\mathcal{CR}$-conjunto)

Un $\mathcal{CR}$-conjunto es un par $X = (|X|, \Vdash_X)$ formado por:

1. Un conjunto no vacío $|X|$ (el soporte de X)
2. Una relación binaria $\Vdash_X \subseteq \Lambda \times |X|$ (la realizabilidad local de X) tal que $\{M \in \Lambda \mid M \Vdash_X x\} \in \mathcal{CR}$ para todo $x \in |X|$

Intuición: la relación $M \Vdash_X x$ se lee: «M realiza x (en X)» y significa: «M es un término adecuado para representar la denotación x en el $\mathcal{CR}$-set X»

- A cada conjunto $S \neq \varnothing$ se asocia el $\mathcal{CR}$-conjunto trivial

$$\mathrm{triv}(S) := (S,\ \mathbf{SN} \times S)$$

= $\mathcal{CR}$-conjunto que asocia el candidato $\mathbf{SN}$ a cada elemento $x \in S$

Normalización fuerte (2/4)

Definición (Producto dependiente de una familia de $\mathcal{CR}$-conjuntos)

Dados

- un $\mathcal{CR}$-conjunto X, y
- una familia de $\mathcal{CR}$-conjuntos $Y = (Y_x)_{x \in |X|}$ (indizada por $|X|$)

se define el $\mathcal{CR}$-conjunto $\Pi(X, Y)$ por:

- $\left|\Pi(X, Y)\right| := \prod_{x \in |X|} |Y_x|$
- $M \Vdash_{\Pi(X,Y)} f \quad \Leftrightarrow$
$$(\forall N \in \Lambda)(\forall x \in |X|)\big(N \Vdash_X x \;\Rightarrow\; M\,N \Vdash_{Y_x} f(x)\big)$$
para todos $M \in \Lambda$ y $f \in \prod_{x \in |X|} Y_x$

Por construcción, $\Pi(X, Y)$ es un $\mathcal{CR}$-conjunto

Normalización fuerte (3/4)

- Dado un conjunto $\mathcal{C}$ de conjuntos no vacíos, se escribe $\mathcal{C}^{\langle \mathcal{CR} \rangle}$ al conjunto de los $\mathcal{CR}$-conjuntos cuyos soportes recorren $\mathcal{C}$:

$$\mathcal{C}^{\langle \mathcal{CR} \rangle} := \{X \ \mathcal{CR}\text{-conjunto} \mid |X| \in \mathcal{C}\}$$

Definición de la función $(M, \rho) \mapsto [\![M]\!]_\rho : \mathsf{Term} \times \mathsf{Val}_{\mathcal{M}} \rightharpoonup \mathcal{M}$

$$\begin{aligned}
[\![x]\!]_\rho &:= \rho(x) \\
[\![\mathsf{Prop}]\!]_\rho &:= \mathsf{triv}\big(\{\{\bullet\}\}^{\langle \mathcal{CR} \rangle}\big) \\
[\![\mathsf{Type}_i]\!]_\rho &:= \mathsf{triv}\big((\mathcal{U}_i \setminus \{\varnothing\})^{\langle \mathcal{CR} \rangle}\big) \\
[\![\lambda x : A \,.\, M]\!]_\rho &:= (v \in |[\![A]\!]_\rho| \mapsto [\![M]\!]_{\rho; x \leftarrow v}) \\
[\![\Pi x : A \,.\, B]\!] &:= \Pi([\![A]\!]_\rho,\ ([\![B]\!]_{\rho; x \leftarrow v})_{v \in |[\![A]\!]_\rho|})
\end{aligned}$$

Extensión a los contextos: conjunto $[\![\Gamma]\!]$ de las valuaciones adaptadas:

$$[\![\Gamma]\!] := \{\rho \in \mathsf{Val}_{\mathcal{M}} \mid \forall (x : A) \in \Gamma \quad \rho(x) \in |[\![A]\!]_\rho|\}$$

Normalización fuerte (4/4)

Proposición (Invariantes de corrección)

- Si $A \preceq A'$ y $[\![A]\!]_\rho, [\![A']\!]_\rho$ definidos, entonces $[\![A]\!]_\rho \subseteq [\![A']\!]_\rho$
- Si $\Gamma \vdash M : A$, entonces para toda valuación $\rho \in [\![\Gamma]\!]$:

$$[\![M]\!]_\rho,\ [\![A]\!]_\rho \text{ definidos} \quad \text{y} \quad [\![M]\!]_\rho \in |[\![A]\!]_\rho|$$

Invariante de normalización

Si $\Gamma \vdash M : A$, entonces para toda valuación $\rho \in [\![\Gamma]\!]$ y para toda sustitución σ tal que $\sigma(x_i) \Vdash_{[\![A_i]\!]_\rho} \rho(x_i)$ (para todos $(x_i : A_i) \in \Gamma$):

$$M[\sigma] \Vdash_{[\![A]\!]_\rho} [\![M]\!]_\rho$$

Corolario (Normalización fuerte)

Todos los términos bien tipados de $\mathrm{CC}^\omega_\preceq$ son fuertemente normalizantes