Introducción a la correspondencia entre pruebas y programas:

Sistema F y eliminación de cortes en la Aritmética intuicionista de segundo orden (HA2)

Alexandre Miquel

mayo de 2021

Introducción

- **Sistema F:** descubierto independientemente por

J.-Y. Girard:	El sistema F	(1970)
J. C. Reynolds:	El cálculo lambda polimórfico	(1974)

- Motivaciones bastante diferentes...

Girard:	Interpretación de la lógica de segundo orden
Reynolds:	Programación funcional

  ... relacionadas por la correspondencia de Curry-Howard

- Influencia importante en el desarrollo de la teoría de tipos

– Interpretación de la lógica de alto orden	[Girard, Martin-Löf]
– Type:Type	[Martin-Löf 1971]
– Teoría de tipos de Martin-Löf	[1972, 1984, 1990, ...]
– El Cálculo de Construcciones	[Coquand 1984]

Plan

1. Introducción
2. Sistema F en el estilo de Church
3. Tipos de datos en el sistema F
4. Sistema F en el estilo de Curry
5. El teorema de normalización fuerte
6. Lógica de 2^{do} orden: sistema NJ2
7. Aritmética intuicionista de 2^{do} orden: sistema HA2
8. Eliminación de cortes en $HA2^{-}$

Plan

1. Introducción
2. Sistema F en el estilo de Church
3. Tipos de datos en el sistema F
4. Sistema F en el estilo de Curry
5. El teorema de normalización fuerte
6. Lógica de 2^{do} orden: sistema NJ2
7. Aritmética intuicionista de 2^{do} orden: sistema HA2
8. Eliminación de cortes en $HA2^{-}$

Sintaxis

Dos formas de variables:

- Variables de términos: x, y, z, etc.
- Variables de tipos: α, β, γ, etc.

Definición (Tipos y términos)

$$
\begin{array}{llllllll}
\textbf{Tipos} & A, B & ::= & \alpha & \mid & A \to B & \mid & \forall \alpha \,.\, B \\
\textbf{Términos} & M, N & ::= & x & & & & \\
 & & \mid & \lambda x : A \,.\, M & \mid & M\,N & & \text{(abstr./apl. de término)} \\
 & & \mid & \lambda \alpha \,.\, M & \mid & M\,A & & \text{(abstr./apl. de tipo)}
\end{array}
$$

Notaciones:

- Conjunto de las variables (de términos) libres: $FV(M)$
- Conjunto de las variables de tipos libres: $TV(M)$, $TV(A)$
- Sustitución de término: $M[x := N]$
- Sustitución de tipo: $M[\alpha := A]$, $B[\alpha := A]$

Se consideran términos y tipos a menos de α-conversión

Reducción

- Relación $\succ$ de reducción definida por 2 reglas:

$$
\begin{array}{lrcl}
(\beta_1) & (\lambda x : A \, . \, M)\, N & \succ & M[x := N] \\
(\beta_2) & (\lambda \alpha \, . \, M)\, A & \succ & M[\alpha := A]
\end{array}
$$

$+$ clausura contextual

- **Obs.:** Las otras combinaciones de una abstracción con una aplicación no tienen sentido, y estarán rechazadas por el tipado

- **Notaciones:** $(\succ^*) \ := \ (\succ)^*, \qquad (\cong) \ := \ (\succ \cup \prec)^*$

Proposición (Confluencia)

La reducción $M \succ M'$ es confluente

Demostración: Ejercicio.

Reglas de tipado

Contextos $\Gamma \quad ::= \quad x_1 : A_1, \ \ldots, \ x_n : A_n \qquad (x_i \not\equiv x_j \text{ si } i \neq j)$

Juicio de tipado $\Gamma \vdash t : A$

$$\frac{}{\Gamma \vdash x : A}\ (x{:}A)\in\Gamma$$

$$\frac{\Gamma,\ x : A \vdash M : B}{\Gamma \vdash \lambda x{:}A\,.\,M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\,N : B}$$

$$\frac{\Gamma \vdash M : B}{\Gamma \vdash \lambda\alpha\,.\,M : \forall\alpha\,.\,B}\ \text{si } \alpha \notin TV(\Gamma) \qquad \frac{\Gamma \vdash M : \forall\alpha\,.\,B}{\Gamma \vdash M\,A : B[\alpha := A]}$$

- Declaración implícita de las variables de tipo (para cada $\alpha \in TV(\Gamma)$)
- También se podrían declarar explícitamente: $\alpha : *$ (cf PTS)
- Una regla para cada construcción sintáctica
  $\Rightarrow$ sistema dirigido por la sintaxis

Ejemplo: la identidad polimórfica (1/2)

- Sea $\quad \mathsf{id} \;:\equiv\; \lambda\alpha\,.\,\lambda x\!:\!\alpha\,.\,x$

- Tenemos que:

$$\begin{array}{lcll} \mathsf{id} & : & \forall\alpha\,.\,\alpha\to\alpha & \\ \mathsf{id}\,B & : & B\to B & \text{para todo tipo } B \\ \mathsf{id}\,B\,N & : & B & \text{para todo término } N : B \end{array}$$

- En particular, tomando $\quad B \;:\equiv\; \forall\alpha\,.\,\alpha\to\alpha \quad$ y $\quad N \;:\equiv\; \mathsf{id}$

$$\begin{array}{lcl} \mathsf{id}\,(\forall\alpha\,.\,\alpha\to\alpha) & : & (\forall\alpha\,.\,\alpha\to\alpha)\;\to\;(\forall\alpha\,.\,\alpha\to\alpha) \\ \mathsf{id}\,(\forall\alpha\,.\,\alpha\to\alpha)\,\mathsf{id} & : & \forall\alpha\,.\,\alpha\to\alpha \end{array}$$

$\Rightarrow$ El sistema de tipado es impredicativo (o cíclico)

Ejemplos (2/2)

- **Identidad polimórfica, continuación**

$$\mathsf{id}\,B\,N \quad\equiv\quad (\lambda\alpha\,.\,\lambda x:\alpha\,.\,x)\,B\,N \quad\succ\quad (\lambda x:B\,.\,x)\,N \quad\succ\quad N$$

$$\mathsf{id}\;(\forall\alpha\,.\,\alpha\to\alpha)\;\mathsf{id}\;(\forall\alpha\,.\,\alpha\to\alpha)\;\cdots\;\mathsf{id}\;(\forall\alpha\,.\,\alpha\to\alpha)\;\mathsf{id}\,B\,N \quad\succ^*\quad N$$

- **Un poco más complicado...**

$$\begin{array}{l}\big(\lambda\alpha\,.\,\lambda x:\alpha\,.\,\lambda f:\alpha{\to}\alpha\,.\,\overbrace{f\,(\cdots\,(f}^{\text{32 veces}}\,x)\cdots)\big)\\ \quad\big(\forall\alpha\,.\,\alpha{\to}(\alpha{\to}\alpha){\to}\alpha\big)\;\big(\lambda\alpha\,.\,\lambda x:\alpha\,.\,\lambda f:\alpha{\to}\alpha\,.\,f\,x\big)\\ \quad\big(\lambda n:\forall\alpha\,.\,\alpha{\to}(\alpha{\to}\alpha){\to}\alpha\,.\,\lambda\alpha\,.\,\lambda x:\alpha\,.\,\lambda f:\alpha{\to}\alpha\,.\,n\,\alpha\,(n\,\alpha\,x\,f)\,f\big)\end{array}$$

$$\succ^*\quad \lambda\alpha\,.\,\lambda x:\alpha\,.\,\lambda f:\alpha{\to}\alpha\,.\,\underbrace{(f\,\cdots\,(f}_{4\,294\,967\,296\ \text{veces}}\,x)\cdots)$$

Propiedades básicas (1/4)

Dado $\Gamma \equiv x_1 : A_1, \ldots, x_n : A_n$, se escribe $\mathrm{dom}(\Gamma) := \{x_1, \ldots, x_n\}$

Lema (Declaración de las variables libres)

Si $\Gamma \vdash M : A$, entonces $FV(M) \subseteq \mathrm{dom}(\Gamma)$

Demostración. Por inducción sobre la derivación de $\Gamma \vdash M : A$. □

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)

La siguiente regla es admisible: $$\frac{\Gamma \vdash M : A}{\Gamma' \vdash M : A}\ \text{si } \Gamma \subseteq \Gamma'$$

Demostración. Por inducción sobre la derivación de $\Gamma \vdash M : A$. □

Propiedades básicas

Lema (Sustitutividad de tipo)

La siguiente regla es admisible:

$$\frac{\Gamma \vdash M : B}{\Gamma[\alpha := A] \vdash M[\alpha := A] : B[\alpha := A]}$$

Demostración. Por inducción sobre la derivación de $\Gamma \vdash M : B$. $\square$

Lema (Sustitutividad de término)

La siguiente regla es admisible:

$$\frac{\Gamma, x : A, \Delta \vdash M : B \qquad \Gamma \vdash N : A}{\Gamma, \Delta \vdash M[x := N] : B}$$

Demostración. Por inducción sobre la derivación de $\Gamma, x : A, \Delta \vdash M : B$, usando la regla de debilitamiento para tratar el caso donde $M \equiv x$. $\square$

Propiedades básicas (3/4)

Lema de inversión

1. Si $\Gamma \vdash x : C$, entonces $(x : C) \in \Gamma$
2. Si $\Gamma \vdash \lambda x{:}A\,.\,M : C$ (con $x \notin \mathrm{dom}(\Gamma)$), entonces $\Gamma, x : A \vdash M : B$ para algún tipo B tal que $C \equiv A \to B$
3. Si $\Gamma \vdash M\,N : C$, entonces $\Gamma \vdash M : A \to C$ y $\Gamma \vdash N : A$ para algún tipo A
4. Si $\Gamma \vdash \lambda\alpha\,.\,M : C$ (con $\alpha \notin TV(\Gamma)$), entonces $\Gamma \vdash M : B$ para algún tipo B tal que $C \equiv \forall\alpha\,.\,B$
5. Si $\Gamma \vdash M\,A : C$, entonces $\Gamma \vdash M : \forall\alpha\,.\,B$ para algún tipo B tal que $C \equiv B[\alpha := A]$

Demostración. Ejercicio.

Propiedades básicas (4/4)

Proposición (Unicidad del tipo)

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

Demostración. Por inducción sobre el término M, usando el lema de inversión. $\square$

Proposición (*Subject Reduction*)

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

Demostración. Ejercicio.

Corolario

1. Si $\Gamma \vdash M : A$ y $M \succ^* M'$, entonces $\Gamma \vdash M' : A$
2. En particular, la forma de normal de M (cuando existe) tiene el mismo tipo que M (cuando existe)

Normalización fuerte

Teorema (Normalización fuerte)

Si $\Gamma \vdash M : A$, entonces M es fuertemente normalizante

Demostración: Postergada

Corolario (Convertibilidad entre términos tipados)

1. Dos términos de mismo tipo son β-convertibles si y sólo si tienen la misma forma normal:

$$\begin{tikzcd} M_1 \arrow[rr, leftrightarrow, "*"] \arrow[dr, dotted, "*"'] & & M_2 \arrow[dl, dotted, "*"] \\ & N \ \text{(normal)} & \end{tikzcd}$$

2. La relación $M_1 \cong M_2$ entre términos tipados es decidible

Verificación e inferencia de tipo

Se consideran los siguientes dos problemas:

1. **El problema de la verificación de tipo:**

   Dados Γ, M, A, determinar si el juicio $\Gamma \vdash M : A$ es derivable o no

2. **El problema de la inferencia de tipo:**

   Dados Γ, M, determinar si existe un tipo A tal que $\Gamma \vdash M : A$
   (y devolver tal tipo A cuando existe)

Proposición (Decidabilidad)

En el sistema F (a la Church), los problemas de la verificación y de la inferencia de tipo son decidibles

Demostración. Ejercicio.

Plan

1. Introducción
2. Sistema F en el estilo de Church
3. Tipos de datos en el sistema F
4. Sistema F en el estilo de Curry
5. El teorema de normalización fuerte
6. Lógica de 2^{do} orden: sistema NJ2
7. Aritmética intuicionista de 2^{do} orden: sistema HA2
8. Eliminación de cortes en $HA2^-$

Booleanos

- Codificación de los booleanos:

$$\begin{array}{rcll}
\mathsf{Bool} & :\equiv & \forall\gamma\,.\,\gamma \to \gamma \to \gamma & \\
\mathsf{true} & :\equiv & \lambda\gamma\,.\,\lambda x, y : \gamma\,.\,x & : \ \mathsf{Bool} \\
\mathsf{false} & :\equiv & \lambda\gamma\,.\,\lambda x, y : \gamma\,.\,y & : \ \mathsf{Bool} \\
\mathtt{if} & :\equiv & \lambda\alpha\,.\,\lambda b : \mathsf{Bool}\,.\,\lambda x, y : \alpha\,.\,b\ \alpha\ x\ y & \\
 & : & \forall\alpha.\ \mathsf{Bool} \to \alpha \to \alpha \to \alpha &
\end{array}$$

- Reducción:

$$\mathtt{if}\ A\ \mathsf{true}\ M\ M' \ \succ^* \ M \qquad\qquad \mathtt{if}\ A\ \mathsf{false}\ M\ M' \ \succ^* \ M'$$

Lema (Formas canónicas de tipo Bool)

Los términos $\mathsf{true} \equiv \lambda\gamma\,.\,\lambda x, y : \gamma\,.\,x$ y $\mathsf{false} \equiv \lambda\gamma\,.\,\lambda x, y : \gamma\,.\,y$ son los únicos términos cerrados y en forma normal de tipo $\mathsf{Bool} \equiv \forall\gamma\,.\,\gamma \to \gamma \to \gamma$

Demostración. Por análisis de caso sobre la derivación (Ejercicio). $\square$

Producto cartesiano

- Codificación del producto cartesiano $A \times B$:

$$
\begin{array}{rcl}
A \times B & :\equiv & \forall\gamma\,.\,(A \to B \to \gamma) \to \gamma \\
\langle M_1, M_2 \rangle_{A,B} & :\equiv & \lambda\gamma\,.\,\lambda f : A \to B \to \gamma\,.\,f\,M_1\,M_2 \quad : \quad A \times B \\
 & & \hfill (\text{si} \;\; M_1 : A \;\; \text{y} \;\; M_2 : B) \\
\text{fst} & :\equiv & \lambda\alpha, \beta\,.\,\lambda p : \alpha \times \beta\,.\,p\,\alpha\,(\lambda x : \alpha\,.\,\lambda y : \beta\,.\,x) \\
 & : & \forall\alpha, \beta\,.\,\alpha \times \beta \to \alpha \\
\text{snd} & :\equiv & \lambda\alpha, \beta\,.\,\lambda p : \alpha \times \beta\,.\,p\,\beta\,(\lambda x : \alpha\,.\,\lambda y : \beta\,.\,y) \\
 & : & \forall\alpha, \beta\,.\,\alpha \times \beta \to \beta
\end{array}
$$

- Reducción:

$$
\text{fst}\;A\;B\;\langle M_1, M_2 \rangle \;\; \succ^* \;\; M_1 \qquad\qquad \text{snd}\;A\;B\;\langle M_1, M_2 \rangle \;\; \succ^* \;\; M_2
$$

- **¡Cuidado!** Cuando A, B son tipos funcionales, pueden existir términos cerrados y en forma normal de tipo $A \times B$ que no son de la forma $\langle M_1, M_2 \rangle_{A,B}$

Suma directa

- Codificación de la suma directa $A + B$:

$$
\begin{array}{rcll}
A + B & :\equiv & \forall\gamma\,.\,(A \to \gamma) \to (B \to \gamma) \to \gamma & \\
\\
\iota_1^{A,B}(M) & :\equiv & \lambda\gamma\,.\,\lambda f : A \to \gamma\,.\,\lambda g : B \to \gamma\,.\, f\ M \quad : \quad A + B & (\text{si } M : A) \\
\iota_2^{A,B}(M) & :\equiv & \lambda\gamma\,.\,\lambda f : A \to \gamma\,.\,\lambda g : B \to \gamma\,.\, g\ M \quad : \quad A + B & (\text{si } M : B) \\
\\
\mathsf{case} & :\equiv & \lambda\alpha,\beta,\gamma\,.\,\lambda s : \alpha + \beta\,.\,\lambda f : \alpha \to \gamma\,.\,\lambda g : \beta \to \gamma\,.\, s\ \gamma\ f\ g & \\
 & : & \forall\alpha,\beta,\gamma\,.\,\alpha + \beta \to (\alpha \to \gamma) \to (\beta \to \gamma) \to \gamma &
\end{array}
$$

- Reducción:

$$
\begin{array}{lcl}
\mathsf{case}\ A\ B\ C\ (\iota_1^{A,B}(N))\ (\lambda x_1 : A\,.\,M_1)\ (\lambda x_2 : B\,.\,M_2) & \succ^* & M_1[x_1 := N] \\
\mathsf{case}\ A\ B\ C\ (\iota_2^{A,B}(N))\ (\lambda x_1 : A\,.\,M_1)\ (\lambda x_2 : B\,.\,M_2) & \succ^* & M_2[x_2 := N]
\end{array}
$$

- **¡Cuidado!** Cuando A, B son tipos funcionales, pueden existir términos cerrados y en forma normal de tipo $A + B$ que no son de la forma $\iota_1^{A,B}(N)$ o $\iota_2^{A,B}(N)$

Tipos finitos

- Codificación de Fin_n $(n \geq 0)$:

$$\mathrm{Fin}_n \quad :\equiv \quad \forall\gamma\,.\,\underbrace{\gamma \to \cdots \to \gamma}_{n \text{ times}} \to \gamma$$

$$\mathbf{e}_i \quad :\equiv \quad \lambda\gamma\,.\,\lambda x_1 : \gamma \ldots \lambda x_n : \gamma\,.\,x_i \quad : \quad \mathrm{Fin}_n \qquad (1 \leq i \leq n)$$

Lema (Formas canónicas de tipo Fin_n)

$\mathbf{e}_1, \ldots, \mathbf{e}_n$ son los únicos términos cerrados y en forma normal de tipo Fin_n

- En particular:

$$\begin{array}{llllll}
\mathrm{Fin}_2 & \equiv & \forall\gamma\,.\,\gamma \to \gamma \to \gamma & \equiv & \mathrm{Bool} & \text{(tipo de los booleanos)} \\
\mathrm{Fin}_1 & \equiv & \forall\gamma\,.\,\gamma \to \gamma & \equiv & \mathrm{Unit} & \text{(tipo unitario)} \\
\mathrm{Fin}_0 & \equiv & \forall\gamma\,.\,\gamma & \equiv & \bot & \text{(tipo vacío)}
\end{array}$$

(**Obs.:** No hay ningún término cerrado y en forma normal de tipo $\bot$)

Enteros naturales

- Codificación de los enteros naturales:

$$
\begin{array}{lcl}
\mathsf{Nat} & :\equiv & \forall\gamma\,.\,\gamma \to (\gamma \to \gamma) \to \gamma \\
\\
\overline{0} & \equiv & \lambda\gamma\,.\,\lambda x{:}\gamma\,.\,\lambda f{:}\gamma{\to}\gamma\,.\,x \\
\overline{1} & \equiv & \lambda\gamma\,.\,\lambda x{:}\gamma\,.\,\lambda f{:}\gamma{\to}\gamma\,.\,f\ x \\
\overline{2} & \equiv & \lambda\gamma\,.\,\lambda x{:}\gamma\,.\,\lambda f{:}\gamma{\to}\gamma\,.\,f\ (f\ x) \\
 & \vdots & \\
\overline{n} & \equiv & \lambda\gamma\,.\,\lambda x{:}\gamma\,.\,\lambda f{:}\gamma{\to}\gamma\,.\,\underbrace{f(\cdots(f}_{n\ \text{veces}}\ x)\cdots) \quad : \quad \mathsf{Nat} \\
 & \vdots &
\end{array}
$$

Lema (Formas canónicas de tipo Nat)

Los términos $\overline{0}$, $\overline{1}$, $\overline{2}$, ... son los únicos términos cerrados y en forma normal de tipo Nat

Calculando con los enteros de Church (1/3)

Intuición: El entero de Church $\overline{n}$ funciona como un iterador:

$$\overline{n}\ A\ M\ M' \quad \succ^* \quad \underbrace{M'\ (\cdots(M'}_{n}\ \ M)\cdots) \qquad (M : A, \quad M' : A \to A)$$

- Función $\quad \mathsf{succ} \ : \ \mathsf{Nat} \to \mathsf{Nat} \quad$ (sucesor)

$$\mathsf{succ} \quad :\equiv \quad \lambda n : \mathsf{Nat} \,.\, \lambda\gamma \,.\, \lambda x : \gamma \,.\, \lambda f : \gamma{\to}\gamma \,.\, f\ (n\ \gamma\ x\ f)$$

- Funciones $\quad \mathsf{plus}, \mathsf{plus}' \ : \ \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat} \quad$ (suma)

$$\begin{aligned} \mathsf{plus} \quad &:\equiv \quad \lambda n, m : \mathsf{Nat} \,.\, \lambda\gamma \,.\, \lambda x : \gamma \,.\, \lambda f : \gamma{\to}\gamma \,.\, m\ \gamma\ (n\ \gamma\ x\ f)\ f \\ \mathsf{plus}' \quad &:\equiv \quad \lambda n, m : \mathsf{Nat} \,.\, m\ \mathsf{Nat}\ n\ \mathsf{succ} \end{aligned}$$

- Funciones $\quad \mathsf{mult}, \mathsf{mult}' \ : \ \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat} \quad$ (producto)

$$\begin{aligned} \mathsf{mult} \quad &:\equiv \quad \lambda n, m : \mathsf{Nat} \,.\, \lambda\gamma \,.\, \lambda x : \gamma \,.\, \lambda f : \gamma{\to}\gamma \,.\, n\ \gamma\ x\ (\lambda y : \gamma \,.\, m\ \gamma\ y\ f) \\ \mathsf{mult}' \quad &:\equiv \quad \lambda n, m : \mathsf{Nat} \,.\, n\ \mathsf{Nat}\ \overline{0}\ (\mathsf{plus}\ m) \end{aligned}$$

Calculando con los enteros de Church (2/3)

- Función predecesor $\textcolor{red}{\mathsf{pred}} \;:\; \mathsf{Nat} \to \mathsf{Nat}$

$$\begin{array}{lcl} \mathsf{pred}\ \overline{0} & \cong & \overline{0} \\ \mathsf{pred}\ (\overline{n+1}) & \cong & \overline{n} \end{array}$$

$$\begin{array}{rcll} \mathsf{fst} & :\equiv & \lambda p : \mathsf{Nat}{\times}\mathsf{Nat} \,.\, p\ \mathsf{Nat}\ (\lambda x, y : \mathsf{Nat} \,.\, x) & : \ \mathsf{Nat}{\times}\mathsf{Nat} \to \mathsf{Nat} \\ \mathsf{snd} & :\equiv & \lambda p : \mathsf{Nat}{\times}\mathsf{Nat} \,.\, p\ \mathsf{Nat}\ (\lambda x, y : \mathsf{Nat} \,.\, y) & : \ \mathsf{Nat}{\times}\mathsf{Nat} \to \mathsf{Nat} \\ \mathsf{step} & :\equiv & \lambda p : \mathsf{Nat}{\times}\mathsf{Nat} \,.\, \langle \mathsf{snd}\ p,\ \mathsf{succ}\ (\mathsf{snd}\ p) \rangle & : \ \mathsf{Nat}{\times}\mathsf{Nat} \to \mathsf{Nat}{\times}\mathsf{Nat} \\ \mathsf{pred} & :\equiv & \lambda n : \mathsf{Nat} \,.\, \mathsf{fst}\ (n\ (\mathsf{Nat}{\times}\mathsf{Nat})\ \langle \overline{0}, \overline{0} \rangle\ \mathsf{step}) & : \ \mathsf{Nat} \to \mathsf{Nat} \end{array}$$

- La función de Ackermann $\textcolor{red}{\mathsf{ack}} \;:\; \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat}$

$$\begin{array}{llllcl} \mathsf{ack} & \overline{0} & \overline{m} & \cong & \overline{m+1} \\ \mathsf{ack} & (\overline{n+1}) & \overline{0} & \cong & \mathsf{ack}\ \overline{n}\ \overline{1} \\ \mathsf{ack} & (\overline{n+1}) & (\overline{m+1}) & \cong & \mathsf{ack}\ \overline{n}\ (\mathsf{ack}\ (\overline{n+1})\ \overline{m}) \end{array}$$

$$\begin{array}{rcll} \mathsf{next} & :\equiv & \lambda f : (\mathsf{Nat}{\to}\mathsf{Nat}) \,.\, \lambda p : \mathsf{Nat} \,.\, p\ \mathsf{Nat}\ (f\ \overline{1})\ f & : \ (\mathsf{Nat}{\to}\mathsf{Nat}) \to (\mathsf{Nat}{\to}\mathsf{Nat}) \\ \mathsf{ack} & :\equiv & \lambda n, m : \mathsf{Nat} \,.\, n\ (\mathsf{Nat}{\to}\mathsf{Nat})\ \mathsf{succ}\ \mathsf{next}\ m & : \ \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat} \end{array}$$

Calculando con los enteros de Church (3/3)

- El recursor del sistema T (versión polimórfica):

$$\begin{array}{rcl} \texttt{rec} & : & \forall\alpha\,.\,\alpha \to (\mathsf{Nat} \to \alpha \to \alpha) \to \mathsf{Nat} \to \alpha \\ \texttt{rec}\ A\ M\ M'\ \overline{0} & \cong & M_0 \\ \texttt{rec}\ A\ M\ M'\ (\overline{n+1}) & \cong & M'\ \overline{n}\ (\texttt{rec}\ A\ M\ M'\ \overline{n}) \end{array}$$

$$\begin{array}{rcl} \texttt{fst}_\alpha & :\equiv & \lambda p : \mathsf{Nat} \times \alpha\,.\,p\ \mathsf{Nat}\ (\lambda x : \mathsf{Nat}\,.\,\lambda y : \alpha\,.\,x) \quad : \quad \mathsf{Nat} \times \alpha \to \mathsf{Nat} \\ \texttt{snd}_\alpha & :\equiv & \lambda p : \mathsf{Nat} \times \alpha\,.\,p\ \alpha\ (\lambda x : \mathsf{Nat}\,.\,\lambda y : \alpha\,.\,y) \quad : \quad \mathsf{Nat} \times \alpha \to \alpha \\ \texttt{iter}_\alpha & :\equiv & \lambda f : \mathsf{Nat} \to \alpha \to \alpha\,.\,\lambda p : \mathsf{Nat} \times \alpha\,. \\ & & \quad \langle \texttt{succ}\ (\texttt{fst}_\alpha\ p),\ f\ (\texttt{fst}_\alpha\ p)\ (\texttt{snd}_\alpha\ p) \rangle \\ & : & (\mathsf{Nat} \to \alpha \to \alpha) \to \mathsf{Nat} \times \alpha \to \mathsf{Nat} \times \alpha \\ \texttt{rec} & :\equiv & \lambda\alpha\,.\,\lambda x : \alpha\,.\,\lambda f : \mathsf{Nat} \to \alpha \to \alpha\,.\,\lambda n : \mathsf{Nat}\,. \\ & & \quad \texttt{snd}_\alpha\ \big(n\ (\mathsf{Nat} \times \alpha)\ \langle \overline{0}, x \rangle\ (\texttt{iter}_\alpha\ f)\big) \\ & : & \forall\alpha\,.\,\alpha \to (\mathsf{Nat} \to \alpha \to \alpha) \to \mathsf{Nat} \to \alpha \end{array}$$

Teorema (Funciones definibles en el sistema F)

Toda función $f : \mathbb{N}^k \to \mathbb{N}$ $(k \geq 1)$ definible en el sistema T también es definible en el sistema F

Plan

1. Introducción
2. Sistema F en el estilo de Church
3. Tipos de datos en el sistema F
4. Sistema F en el estilo de Curry
5. El teorema de normalización fuerte
6. Lógica de 2^{do} orden: sistema NJ2
7. Aritmética intuicionista de 2^{do} orden: sistema HA2
8. Eliminación de cortes en $HA2^{-}$

El polimorfismo del sistema F

Polimorfismo de ML/Haskell ($*$)

Tipos $A, B \quad ::= \quad \alpha \mid A \to B \mid \cdots$ (tipos del usuario)

Esquemas $S \quad ::= \quad \forall\vec{\alpha}\,.\,B$

El esquema de tipo $\forall\alpha\,.\,B$ está définido después de sus instancias $B[\alpha := A]$

$\Rightarrow$ El sistema de tipos es predicativo

($*$) Al menos en las versiones básicas de Haskell

Polimorfismo del sistema F

Tipos $A, B \quad ::= \quad \alpha \mid A \to B \mid \forall\alpha\,.\,B$

El tipo $\forall\alpha\,.\,B$ y sus instancias $B[\alpha := A]$ están définidos simultáneamente

$$\forall\alpha\,.\,\alpha \to \alpha \qquad \text{y} \qquad (\forall\alpha\,.\,\alpha \to \alpha) \to (\forall\alpha\,.\,\alpha \to \alpha)$$

$\Rightarrow$ El sistema de tipos es impredicativo, o cíclico

Extracción de términos lambda puros

En el sistema F a la Church, el polimorfismo es explícito:

$$\mathsf{id} \;\equiv\; \lambda\alpha\,.\,\lambda x : \alpha\,.\,x \qquad \text{e} \qquad \mathsf{id}\ \mathsf{Nat}\ 2$$

- Dos formas de redexes $\quad (\lambda x : A\,.\,t)u \quad$ y $\quad (\lambda\alpha\,.\,t)A$

Idea: Borrar las abstracciones/aplicaciones/anotaciones de tipo

Definición (Función de borrado $M \mapsto |M|$)

$$\begin{array}{rcl@{\qquad\qquad}rcl}
|x| & :\equiv & x & & & \\
|\lambda x : A\,.\,M| & :\equiv & \lambda x\,.\,|M| & |M\,N| & :\equiv & |M||N| \\
|\lambda\alpha\,.\,M| & :\equiv & |M| & |M\,A| & :\equiv & |M|
\end{array}$$

- Lenguaje de destino: el cálculo lambda puro
- Redexes de 2da forma borradas, redexes de 1ra forma mantenidas

Extensión de la función de borrado

Los términos borrados tienen buenas propiedades computacionales...

- Una única forma de redex, fácil de ejecutar
- Los cálculos inútiles (sobre los tipos) están borrados
- La esencia del cálculo está mantenida (cf justificación posterior)

... pero ¿qué estatus con respecto al tipado?

La función de borrado, definida sobre los términos, se puede extender a:

- Toda la sintaxis
- Los juicios
- Las reglas de tipado
- Las derivaciones

$\Rightarrow$ Define un nuevo formalismo: el sistema F a la Curry

Sistema F a la Curry [Leivant '83]

Sintaxis

Tipos	A, B	$::=$	$\alpha \mid A \to B \mid \forall\alpha\,.\,B$
Términos	M, N	$::=$	$x \mid \lambda x\,.\,M \mid M\,N$
Contextos	Γ	$::=$	$\varnothing \mid \Gamma,\ x : A$

Reducción $\quad (\lambda x\,.\,M)\,N \succ M[x := N]$

Observaciones:

- Los tipos (y los contextos) no cambian
- Los términos son ahora los términos lambda puros
- Una única forma de redex

Sistema F a la Curry: reglas de tipado

Definición (Reglas de tipado)

$$\frac{}{\Gamma \vdash x : A} \text{ si } (x{:}A) \in \Gamma$$

$$\frac{\Gamma, \ x : A \vdash M : B}{\Gamma \vdash \lambda x \, . \, M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M \, N : B}$$

$$\frac{\Gamma \vdash M : B}{\Gamma \vdash M : \forall \alpha \, . \, B} \text{ si } \alpha \notin TV(\Gamma) \qquad \frac{\Gamma \vdash M : \forall \alpha \, . \, B}{\Gamma \vdash M : B[\alpha := A]}$$

$\Rightarrow$ Las reglas ya no son dirigidas por la sintaxis

Sistema F a la Curry: propiedades

Lo que no cambia:

- Propiedades básicas (sustitutividad, etc.) + *subject reduction*
- La normalización fuerte (cf más adelante)

Lo que cambia:

- Un término puede tener múltiples tipos:

$$
\begin{array}{rclcll}
\Delta & \equiv & \lambda x\,.\,x\;x & : & (\forall\alpha\,.\,\alpha \to \alpha) \to (\forall\alpha\,.\,\alpha \to \alpha) & \\
 & & & : & (\forall\alpha\,.\,\alpha) \to (\forall\alpha\,.\,\alpha) & \\
 & & & : & (\forall\alpha\,.\,\alpha) \to (\forall\alpha\,.\,\alpha \to \alpha) & \\
 & & & : & \mathsf{Bool} \to \mathsf{Bool} \to \mathsf{Bool} & (\text{‘o’ booleano}) \\
 & & & : & \mathsf{Nat}' \to \mathsf{Nat}' & (n \mapsto n^n) \\
 & & & & (\text{con } \mathsf{Nat}' :\equiv \forall\gamma\,.\,(\gamma \to \gamma) \to (\gamma \to \gamma)) &
\end{array}
$$

- Ningún tipo principal
- Verificación e inferencia de tipo se vuelven indecidibles [Wells '94]

Borrado y tipado

Equivalencia entre las presentaciones a la Church y a la Curry

1. Si $\Gamma \vdash M_0 : A$ (Church), entonces $\Gamma \vdash |M_0| : A$ (Curry)
2. Si $\Gamma \vdash M : A$ (Curry), entonces $\Gamma \vdash M_0 : A$ (Church) para algún M_0 tal que $|M_0| \equiv M$

- La función de borrado $M \mapsto |M|$ transforma:

Mundo de Church		**Mundo de Curry**	
derivaciones	en	derivaciones	(isomorfismo)
juicios derivables	en	juicios derivables	(sobreyección)

- **¡Cuidado!** No es inyectiva sobre los juicios derivables:

$$\begin{array}{rcl}
\lambda f:(\forall\alpha\,.\,\alpha\to\alpha)\,.\,f\,(\forall\alpha\,.\,\alpha\to\alpha)\,f & : & (\forall\alpha\,.\,\alpha\to\alpha)\to(\forall\alpha\,.\,\alpha\to\alpha)\\
\lambda f:(\forall\alpha\,.\,\alpha\to\alpha)\,.\,\lambda\alpha\,.\,f\,(\alpha\to\alpha)\,(f\,\alpha) & : & (\forall\alpha\,.\,\alpha\to\alpha)\to(\forall\alpha\,.\,\alpha\to\alpha)\\
\rightsquigarrow \qquad \lambda f\,.\,f\,f & : & (\forall\alpha\,.\,\alpha\to\alpha)\to(\forall\alpha\,.\,\alpha\to\alpha)
\end{array}$$

Borrado y reducción

Redexes de 2da forma borradas / redexes de 1ra forma mantenidas:

$$\begin{array}{lccccc} \text{(Church)} & (\lambda\alpha\,.\,\lambda x:\alpha\,.\,x)\,B\,y & \succ & (\lambda x:B\,.\,x)\,y & \succ & y \\ \quad\downarrow \text{Borrado} & & & & & \\ \text{(Curry)} & (\lambda x\,.\,x)\,y & \equiv & (\lambda x\,.\,x)\,y & \succ & y \end{array}$$

Lema 1 (de Church a Curry):

Si $M_0, M_0' \in$ Church, entonces

$$M_0 \succ^n M_0' \quad \Rightarrow \quad |M_0| \succ^p |M_0'| \qquad \text{(con } p \leq n\text{)}$$

Demostración. Ejercicio.

Lema 2 (de Curry a Church):

Si $M_0 \in$ Church, $M' \in$ Curry y M_0 bien tipado, entonces

$$|M_0| \succ^p M' \quad \Rightarrow \quad \exists M_0' \; (|M_0'| = M' \;\wedge\; M_0 \succ^n M_0') \qquad \text{(con } n \geq p\text{)}$$

Demostración. Ejercicio.

Equivalencia de normalización

Lema 3 (Argumento combinatorio):

1. Durante la contracción de una redex de 1ra forma, el número de redexes de ambas formas puede crecer
2. Durante la contracción de una redex de 2da forma,
   - el número de redexes de 1ra forma puede crecer
   - el número de redexes de 2da forma no crece
   - el número de abstracciones de tipo $(\lambda\alpha\,.\,t)$ decrece

Demostración. Ejercicio.

Combinando los lemas 1, 2 and 3, se demuestra el:

Teorema (Equivalencia de normalización)

Los siguientes enunciados son combinatoriamente equivalentes:

1. Todo término tipado de F-Church es fuertemente normalizable
2. Todo término tipado de F-Curry es fuertemente normalizable

Demostración. Ejercicio.

Plan

1. Introducción
2. Sistema F en el estilo de Church
3. Tipos de datos en el sistema F
4. Sistema F en el estilo de Curry
5. El teorema de normalización fuerte
6. Lógica de 2^{do} orden: sistema NJ2
7. Aritmética intuicionista de 2^{do} orden: sistema HA2
8. Eliminación de cortes en $HA2^-$

Significado de la cuantificación de tipo (1/2)

Pregunta: ¿Cúal es el significado de $\forall\alpha\,.\,\alpha \to \alpha$?

Primer escenario: Un producto cartesiano infinito (a la Martin-Löf)

$$
\begin{aligned}
\forall\alpha\,.\,\alpha \to \alpha \quad &\approx \quad \prod_{\alpha\ \mathsf{type}} (\alpha \to \alpha) \\
&\approx \quad (\bot \to \bot) \times (\mathsf{Bool} \to \mathsf{Bool}) \times (\mathsf{Nat} \to \mathsf{Nat}) \times \cdots
\end{aligned}
$$

Como todos los tipos $A \to A$ son habitados:

1. El producto cartesiano $\forall\alpha\,.\,\alpha \to \alpha$ debería ser más grande que todos los tipos de la forma $A \to A$
2. En particular, $\forall\alpha\,.\,\alpha \to \alpha$ debería ser más grande que su propio espacio de funciones $(\forall\alpha\,.\,\alpha \to \alpha) \to (\forall\alpha\,.\,\alpha \to \alpha)\ldots$

. . . Un escenario muy paradójico

Significado de la cuantificación de tipo (2/2)

Segundo escenario: En F-Curry, las reglas $\forall$-intro y $\forall$-elim

$$\frac{\Gamma \vdash M : B}{\Gamma \vdash M : \forall\alpha\,.\,B}\ \text{si } \alpha \notin TV(\Gamma) \qquad\qquad \frac{\Gamma \vdash M : \forall\alpha\,.\,B}{\Gamma \vdash M : B[\alpha := A]}$$

sugieren que $\forall$ no es un producto cartesiano, sino una intersección

Considerando de vuelta el ejemplo anterior:

1. La intersección $\forall\alpha\,.\,\alpha \to \alpha$ es más pequeña que todos los $A \to A$
2. En particular, el tipo $\forall\alpha\,.\,\alpha \to \alpha$ es más pequeño que su propio espacio de funciones $(\forall\alpha\,.\,\alpha \to \alpha) \to (\forall\alpha\,.\,\alpha \to \alpha)\ldots$

... Un escenario mucho mejor

$\Rightarrow$ Vamos a demostrar la normalización fuerte para F-Curry

Recordatorio: SN(F-Church) $\Leftrightarrow$ SN(F-Curry) (equivalencia combinatoria)

Candidatos de reducibilidad (1/3)

- **SN** := conjunto de los términos lambda fuertemente normalizantes
- $\mathrm{Red}_1(M) \ := \ \{M' \ : \ M \succ M'\}$
- Un término neutro es un término lambda que no es una abstracción
  (En F-Curry, las únicas formas canónicas son las abstracciones $\lambda x \,.\, M$)

Definición (Candidato de reducibilidad)

Un conjunto de términos $C \subseteq \Lambda$ (posiblemente abiertos) es un candidato de reducibilidad cuando cumple los siguientes criterios:

(CR1) $C \subseteq \mathbf{SN}$

(CR2) Si $M \in C$, entonces $\mathrm{Red}_1(M) \subseteq C$

(CR3) Si un término neutro M es tal que $\mathrm{Red}_1(M) \subseteq C$, entonces $M \in C$

- Se escribe $\mathcal{CR}$ al conjunto de todos los candidatos de reducibilidad

Candidatos de reducibilidad (2/3)

Recordatorio: Todo candidato $C \in \mathcal{CR}$ contiene todas las variables: $x \in C$

Proposición (Estructura de retículo completo)

1. $\mathbf{SN} \in \mathcal{CR}$
2. $\mathcal{CR}$ está cerrado por intersección cualquiera (pero no vacía):

$$I \neq \varnothing, \quad (C_i)_{i \in I} \in \mathcal{CR}^I \quad \Rightarrow \quad \Big(\bigcap_{i \in I} C_i\Big) \in \mathcal{CR}$$

Demostración. Ejercicio.

Dicho de otro modo:

- $(\mathcal{CR}, \subseteq)$ es un retículo completo, donde

$$\top_{\mathcal{CR}} = \mathbf{SN} \quad \text{y} \quad \bot_{\mathcal{CR}} = \bigcap \mathcal{CR} = \{M \in \mathbf{SN} \ : \ M \succ^* x\, N_1 \cdots N_k\}$$

- $\mathcal{CR}$ también está cerrado por unión cualquiera (no vacía), pero la prueba es mucho más difícil [Riba '07]

Candidatos de reducibilidad (3/3)

Definición (Flecha de Kleene)

Dados conjuntos $C, D \subseteq \Lambda$, se define:

$$C \to D \ := \ \{M \in \Lambda \ : \ \forall N \in C, \ MN \in D\}$$

Lema (Clausura de $\mathcal{CR}$ por la flecha de Kleene)

Si $C, D \in \mathcal{CR}$, entonces $(C \to D) \in \mathcal{CR}$

Demostración. Ejercicio.

Lema (Clausura por expansíon de cabeza)

En cualquier candidato $C \in \mathcal{CR}$:
Si $M[x := N] \in C$ y $N \in \mathbf{SN}$, entonces $(\lambda x\,.\,M)N \in C$

Demostración. Ejercicio.

Interpretación de los tipos: intuiciones

Se trata de definir una interpretación de los tipos sintácticos A por candidatos de reducibilidad $[\![A]\!] \in \mathcal{CR}$, usando:

- La flecha de Kleene $C \to D$ para interpretar el tipo flecha $A \to B$
- La intersección $\bigcap_{C \in \mathcal{CR}} \cdots$ para interpretar la cuantificación de tipo $\forall \alpha \ldots$

Ejemplo: $\forall \alpha . (\alpha \to \alpha)$ tiene que ser interpretado por $\bigcap_{C \in \mathcal{CR}} (C \to C)$

Observación. La definición del candidato

$$\bigcap_{C \in \mathcal{CR}} (C \to C)$$

es impredicativa, pues la intersección involucra todos los candidatos $C \in \mathcal{CR}$, incluso el candidato que estamos definiendo (definición cíclica, legal en ZF)

Interpretación de los tipos: definiciones

Para interpretar las variables de tipo, se utilizan valuaciones de tipo:

Definición (Valuaciones de tipo)

Una valuación de tipo es una función $\rho \in \mathcal{CR}^{\mathsf{TVar}}$
(donde TVar es el conjunto de todas las variables de tipo)

Se escribe $\mathsf{TVal} := \mathcal{CR}^{\mathsf{TVar}}$ al conjunto de las valuaciones de tipo

Definición (Interpretación de los tipos)

Por inducción sobre A se define una función $[\![A]\!] : \mathsf{TVal} \to \mathcal{CR}$:

$$\begin{array}{rcl} [\![\alpha]\!]_\rho & = & \rho(\alpha) \\ [\![A \to B]\!]_\rho & = & [\![A]\!]_\rho \to [\![B]\!]_\rho \\ [\![\forall\alpha\,.\,B]\!]_\rho & = & \displaystyle\bigcap_{C \in \mathcal{CR}} [\![B]\!]_{\rho;\alpha\leftarrow C} \end{array}$$

Obs: La valuación $(\rho;\alpha\leftarrow C)$ está definida por $\begin{cases} (\rho;\alpha\leftarrow C)(\alpha) = C \\ (\rho;\alpha\leftarrow C)(\beta) = \rho(\beta) & \text{para todo } \beta \neq \alpha \end{cases}$

Interpretación de los contextos

Definición (Sustitución, recordatorio)

Una sustitución es un conjunto finito de la forma

$$\sigma \; :\equiv \; \{x_1 := N_1, \ldots, x_n := N_n\} \qquad \text{(con } x_i \not\equiv x_j \text{ si } i \neq j\text{)}$$

Dada una sustitución $\sigma = \{x_1 := N_1, \ldots, x_n := N_n\}$, se escriben:

- $\mathrm{dom}(\sigma) = \{x_1, \ldots, x_n\}$ a su dominio
- $FV(\sigma) = FV(N_1) \cup \cdots \cup FV(N_n)$ a su conjunto de variables libres
- $M[\sigma]$ a la aplicación de σ a un término M

Definición (Interpretación de los contextos)

Dados un contexto Γ y una valuación $\rho \in \mathrm{TVal}$, se define:

$$[\![\Gamma]\!]_\rho \; := \; \left\{ \sigma \text{ sustitución } : \begin{array}{l} \mathrm{dom}(\sigma) = \mathrm{dom}(\Gamma) \quad \text{y} \\ \sigma(x) \in [\![A]\!]_\rho \text{ para todo } (x : A) \in \Gamma \end{array} \right\}$$

El invariante de normalización fuerte

Proposición (Invariante de normalización fuerte)

Si $\Gamma \vdash M : A$ en el sistema F a la Curry, entonces

$$\forall \rho \in \mathsf{TVal} \qquad \forall \sigma \in [\![\Gamma]\!]_\rho \qquad M[\sigma] \in [\![A]\!]_\rho$$

Demostración. Por inducción sobre la derivación de $\Gamma \vdash M : A$ (Ejercicio).

Teorema (Normalización fuerte en F-Curry)

Todo término tipado en F-Curry es fuertemente normalizante

Demostración. Ejercicio.

Corolario (Normalización fuerte en F-Church)

Todo término tipado en F-Church es fuertemente normalizante

Observación sobre la impredicatividad

En la demostración de la propiedad de normalización fuerte, la interpretación de la cuantificación de tipo se basa en la propiedad:

$$\text{Si} \quad (C_i)_{i\in I} \in \mathcal{CR}^I, \quad \text{entonces} \quad \bigcap_{i\in I} C_i \ \in \ \mathcal{CR}$$

en el caso particular donde $I = \mathcal{CR}$ (intersección impredicativa)

- En matemática «clásica», esta construcción es legal:

  $\Rightarrow$ Teorías de conjuntos usuales (Z, ZF, ZFC) impredicativas, debido al axioma del conjunto potencia

- En matemática constructiva en el estilo de Bishop o Martin-Löf, este principio está rechazado por razones filosóficas:
  - Ninguna explicación constructiva convincente
  - Sospecha en cuanto a esta forma particular de definición cíclica

Impredicatividad: un ejemplo (1/2)

Dados un K-espacio vectorial V y un subconjunto $X \subseteq V$,
¿cómo definir el subespacio vectorial $\overline{X} \subseteq V$ generado por X en V?

Método «abstracto»:

1. Sea el conjunto $\mathfrak{S}_X := \left\{ S \in \mathfrak{P}(V) \ : \ S \text{ s.e.v y } X \subseteq S \right\}$
2. Se observa que $\mathfrak{S}_X \neq \varnothing$, pues $V \in \mathfrak{S}_X$
3. Se define: $\overline{X} := \bigcap_{S \in \mathfrak{S}_X} S$
4. Por def., $\overline{X}$ está incluido en todos los s.e.v. de V que contienen X
5. Pero $\overline{X}$ sí mismo es un s.e.v. que contiene X (entonces $\overline{X} \in \mathfrak{S}_X$)
6. Por lo tanto, $\overline{X}$ es el mínimo de $\mathfrak{S}_X$ (con respecto a $\subseteq$)

Esta construcción es impredicativa... pero legal en ZF

Conjunto $\overline{X}$ definido *a partir de* $\mathfrak{S}_X$, $\underbrace{\text{que ya contiene } \overline{X} \text{ como elemento}}_{\text{descubierto a fortiori}}$

Impredicatividad: un ejemplo (2/2)

Pero hay otros métodos para definir el subespacio $\overline{X} \subseteq V$...

- **Definición concreta estándar, por combinaciones lineales:**

  Sea $\overline{X}$ el conjunto de los vectores de la forma $v = \alpha_1 \cdot v_1 + \cdots + \alpha_n \cdot v_n$

  donde (v_i) recorre todas las familias finitas de elementos de X
  (α_i) recorre todas las familias finitas de escalares

- **Definición inductiva:**

  Sea $\overline{X}$ el conjunto definido inductivamente por las reglas:

$$\frac{}{\vec{0} \in \overline{X}} \qquad \frac{x \in X}{x \in \overline{X}} \qquad \frac{v \in \overline{X}}{\alpha \cdot v \in \overline{X}} \qquad \frac{v_1 \in \overline{X} \quad v_2 \in \overline{X}}{v_1 + v_2 \in \overline{X}}$$

$\Rightarrow$ Ambas definiciones son predicativas... y definen el mismo objeto

Plan

1. Introducción
2. Sistema F en el estilo de Church
3. Tipos de datos en el sistema F
4. Sistema F en el estilo de Curry
5. El teorema de normalización fuerte
6. **Lógica de 2$^{\text{do}}$ orden: sistema NJ2**
7. Aritmética intuicionista de 2$^{\text{do}}$ orden: sistema HA2
8. Eliminación de cortes en HA2^{-}

El lenguaje de la lógica (mínima) de segundo orden

- La lógica de segundo orden manipula dos tipos de objetos:
  - Objetos de 1$^{\text{er}}$ orden = individuos (i.e. objetos básicos de la teoría)
  - Objetos de 2$^{\text{do}}$ orden = relaciones sobre los individuos
    = conjuntos de k-uplas de individuos

Definición (Términos y fórmulas de la lógica mínima de segundo orden)

Términos $\quad t, t' \quad ::= \quad x \quad | \quad f(t_1, \dots, t_k)$

Fórmulas $\quad A, B \quad ::= \quad X(t_1, \dots, t_k) \quad | \quad A \Rightarrow B \quad | \quad \forall x\, A \quad | \quad \forall X\, A$

- Dos tipos de variables (y de cuantificación):
  - 1$^{\text{er}}$ orden: $x, y, z, \dots$
  - 2$^{\text{do}}$ orden: $X, Y, Z, \dots$ de todas aridades $k \geq 0$

- Dos tipos de sustitución:
  - de 1$^{\text{er}}$ orden: $t[x := u]$, $A[x := u]$ (definida de modo usual)
  - de 2$^{\text{do}}$ orden: $A[X := P]$, $Q[X := P]$ (def. postergada)

Términos de primer orden

- Definidos a partir de un vocabulario $\mathcal{V}$ (de modo usual):

Términos $\quad t, u \quad ::= \quad x \quad | \quad f(t_1, \ldots, t_k)$

  - donde f recorre los símbolos de función de aridad k en $\mathcal{V}$

- También se pueden suponer los términos equipados con una relación de reducción $t \succ t'$ convergente (= confluente + f. normalizante)

- **Ejemplo:** Términos de la aritmética computacional $\mathrm{HA}^{\cong}$:

Términos $\quad t, u \quad ::= \quad x \quad | \quad 0 \quad | \quad s(t) \quad | \quad \mathrm{pred}(t) \quad | \quad t + u \quad | \quad t \times u$

$+$ relación de reducción $t \succ t'$ de $\mathrm{HA}^{\cong}$

Fórmulas de segundo orden

- Fórmulas de la lógica mínima de 2^{do} orden

$$\textbf{Fórmulas} \qquad A, B \quad ::= \quad X(t_1, \ldots, t_k) \quad | \quad A \Rightarrow B \quad | \quad \forall x\, A \quad | \quad \forall X\, A$$

sólo basadas en «$\Rightarrow$» y «$\forall$» (1^{er} y 2^{do} orden)

- Otras construcciones definidas mediante codificación de 2^{do} orden:

$$\begin{array}{rcll}
\top & :\equiv & \forall Z\,(Z \Rightarrow Z) & \text{(obviedad)} \\
\bot & :\equiv & \forall Z\, Z & \text{(absurdidad)} \\
\neg A & :\equiv & A \Rightarrow \bot & \text{(negación)} \\
A \wedge B & :\equiv & \forall Z\,((A \Rightarrow B \Rightarrow Z) \Rightarrow Z) & \text{(conjunción)} \\
A \vee B & :\equiv & \forall Z\,((A \Rightarrow Z) \Rightarrow (B \Rightarrow Z) \Rightarrow Z) & \text{(disyunción)} \\
\exists x\, A(x) & :\equiv & \forall Z\,(\forall x\,(A(x) \Rightarrow Z) \Rightarrow Z) & (\exists,\ 1^{\text{er}}\ \text{orden}) \\
\exists X\, A(X) & :\equiv & \forall Z\,(\forall X\,(A(X) \Rightarrow Z) \Rightarrow Z) & (\exists,\ 2^{\text{do}}\ \text{orden}) \\
t = u & :\equiv & \forall Z\,(Z(t) \Rightarrow Z(u)) & \text{(Igualdad de Leibniz)}
\end{array}$$

Predicados (1/2)

- Las variables de 2$^{\text{do}}$ orden representan relaciones abstractas
- Relaciones concretas representadas por predicados:

Predicados $\quad P, Q \quad ::= \quad \hat{x}_1 \cdots \hat{x}_k \ A \quad$ (de aridad k)

- Dado un predicado $P \equiv \hat{x}_1 \cdots \hat{x}_k \ A$:
  - Las variables $x_1, \ldots, x_k$ son los argumentos de P
  - Las otras variables libres de A son los parámetros de P
  - **Notación:** $FV(P) := FV(A) \setminus \{x_1, \ldots, x_k\}$
- Los predicados están considerados a menos de α-conversión

Definición (Aplicación de predicado)

Dados un predicado $P \equiv \hat{x}_1 \cdots \hat{x}_k \ A$ y términos $t_1, \ldots, t_k$, se escribe:

$$P(t_1, \ldots, t_k) \ :\equiv \ A[x_1 := t_1, \ldots, x_k := t_k]$$

Predicados (2/2)

Definición (Sustitución de 2^{do} orden)

Dados una variable de 2do orden X y un predicado P de misma aridad $k \geq 0$, se define la operación $A \mapsto A[X := P]$ de sustitución de 2do orden por:

$$
\begin{array}{rcll}
(X(t_1, \ldots, t_k))[X := P] & :\equiv & P(t_1, \ldots, t_k) & \\
(Y(t_1, \ldots, t_k))[X := P] & :\equiv & Y(t_1, \ldots, t_k) & (\text{si } Y \not\equiv X) \\
(A \Rightarrow B)[X := P] & :\equiv & A[X := P] \Rightarrow B[X := P] & \\
(\forall x\, A)[X := P] & :\equiv & \forall x\, A[X := P] & (\text{si } x \notin FV(P)) \\
(\forall Y\, A)[X := P] & :\equiv & \forall Y\, A[X := P] & (\text{si } Y \not\equiv X,\ Y \notin FV(P))
\end{array}
$$

(definición a menos de α-conversión)

Ejercicio: Enunciar y demostrar el correspondiente lema de sustitución

- **Obs.:** Cada variable X de 2do orden y de aridad k puede ser considerada como el predicado:

$$X \;\equiv\; \hat{x}_1 \cdots \hat{x}_k\ X(x_1, \ldots, x_k)$$

Los predicados unarios como conjuntos

- Los predicados unarios representan conjuntos de individuos

Notaciones: $\{x : A\} \;:\equiv\; \hat{x}\,A, \qquad t \in P \;:\equiv\; P(t)$

Ejemplo: El conjunto $\mathbb{N}$ de los enteros de Dedekind

$$\mathbb{N} \;:\equiv\; \{x \;:\; \forall Z\,(0 \in Z \Rightarrow \forall y\,(y \in Z \Rightarrow s(y) \in Z) \Rightarrow x \in Z\}$$

- Cuantificaciones relativizadas:

$$\begin{array}{rcl}(\forall x \in P)\,A(x) & \equiv & \forall x\,(x \in P \Rightarrow A(x)) \\ (\exists x \in P)\,A(x) & \equiv & \forall Z\,(\forall x\,(x \in P \Rightarrow A(x) \Rightarrow Z) \Rightarrow Z) \\ & \Leftrightarrow & \exists x\,(x \in P \wedge A(x))\end{array}$$

- Inclusión e igualdad extensional:

$$\begin{array}{rcl}P \subseteq Q & \equiv & \forall x\,(x \in P \Rightarrow x \in Q) \\ P = Q & \equiv & \forall x\,(x \in P \Leftrightarrow x \in Q)\end{array}$$

- Operaciones conjuntistas: $P \cup Q \;\equiv\; \{x \;:\; x \in P \vee x \in Q\}$ (etc.)

Reglas de deducción

- Reglas de la lógica intuicionista de 2$^{\text{do}}$ orden (sistema NJ2):

$$\frac{}{\Gamma \vdash A} \text{ si } A \in \Gamma$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \qquad \frac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x\, A} \text{ si } x \notin FV(\Gamma) \qquad \frac{\Gamma \vdash \forall x\, A}{\Gamma \vdash A[x := u]}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall X\, A} \text{ si } X \notin FV(\Gamma) \qquad \frac{\Gamma \vdash \forall X\, A}{\Gamma \vdash A[X := P]} \text{ si } \sharp X = \sharp P$$

- Cuando se trabaja con una congruencia $A \cong A'$ sobre las fórmulas, se remplazan las reglas (axioma), ($\forall^1$-el) y ($\forall^2$-el) por:

$$\frac{}{\Gamma \vdash A'} \text{ si } A' \cong A \in \Gamma \qquad \frac{\Gamma \vdash \forall x\, A}{\Gamma \vdash A'} \text{ si } A' \cong A[x := u] \qquad \frac{\Gamma \vdash \forall X\, A}{\Gamma \vdash A'} \text{ si } A' \cong A[X := P]$$

Reglas admisibles

Proposición (Debilitamiento generalizado + conversión)

La siguientes reglas de inferencia son admisibles en el sistema NJ2:

$$\frac{\Gamma \vdash A}{\Gamma' \vdash A}\ \text{si}\ \Gamma \subseteq \Gamma' \qquad\qquad \frac{\Gamma \vdash A}{\Gamma' \vdash A'}\ \text{si}\ \Gamma \cong \Gamma',\ A \cong A'$$

Demostración. Ejercicio

Proposición (Sustitutividad, 1^er^ y 2^do^ orden)

La siguientes reglas de inferencias son admisibles en el sistema NJ2:

$$\frac{\Gamma \vdash A}{\Gamma[x := u] \vdash A[x := u]} \qquad\qquad \frac{\Gamma \vdash A}{\Gamma[X := P] \vdash A[X := P]}$$

Demostración. Ejercicio

Reglas derivadas (1/2)

Las reglas de deducción de la lógica mínima de 2$^{\text{do}}$ orden sólo tratan las construcciones primitivas «$\Rightarrow$» y «$\forall$»

En este marco, vimos que las otras construcciones ($\top$, $\bot$, $\wedge$, $\vee$, $\exists$ etc.) son definibles; además sus correspondientes reglas son derivables:

- Conectivas lógicas: $\top$, $\bot$, $\wedge$ y $\vee$

$$\dfrac{}{\Gamma \vdash \top} \qquad \dfrac{\Gamma \vdash \bot}{\Gamma \vdash A}$$

$$\dfrac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \qquad \dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}$$

$$\dfrac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \dfrac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$$

$$\dfrac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}$$

Ejercicio. Derivar estas reglas

Reglas derivadas (2/2)

- Cuantificación existencial: 1^{er} y 2^{do} orden

$$\frac{\Gamma \vdash A[x := u]}{\Gamma \vdash \exists x\, A} \qquad \frac{\Gamma \vdash \exists x\, A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}\ \text{si } x \notin FV(\Gamma, B)$$

$$\frac{\Gamma \vdash A[X := P]}{\Gamma \vdash \exists X\, A}\ \text{si } \sharp X = \sharp P \qquad \frac{\Gamma \vdash \exists X\, A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}\ \text{si } X \notin FV(\Gamma, B)$$

Ejercicio. Derivar estas reglas

- Igualdad de Leibniz $t = u \;:\equiv\; \forall Z\,(Z(t) \Rightarrow Z(u))$

$$\frac{}{\Gamma \vdash t = t} \qquad \frac{\Gamma \vdash t = u \qquad \Gamma \vdash A[x := t]}{\Gamma \vdash A[x := u]}$$

Ejercicio. Derivar estas reglas

El esquema de comprensión

Proposición (Esquema de comprensión)

Para cada fórmula $A \equiv A(x_1, \ldots, x_k)$ $^{(*)}$, el axioma de comprensión

$$\exists Y\ \forall x_1 \cdots \forall x_k\ \left(Y(x_1, \ldots, x_k)\ \Leftrightarrow\ A(x_1, \ldots, x_k)\right)$$

es derivable en el sistema NJ2

$(*)$ La fórmula $A(x_1, \ldots, x_k)$ también puede depender de otras variables $\vec{z}$, $\vec{Z}$

Demostración. Basta con aplicar la regla (derivable) ($\exists$-in) de 2$^{\text{do}}$ orden con el predicado $P :\equiv \hat{x}_1 \cdots \hat{x}_k\ A(x_1, \ldots, x_k)$:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \vdash A(x_1, \ldots, x_k) \Leftrightarrow A(x_1, \ldots, x_k)\end{array}}{\vdash \forall x_1 \cdots \forall x_k\ \left(A(x_1, \ldots, x_k)\ \Leftrightarrow\ A(x_1, \ldots, x_k)\right)}}{\vdash \exists Y\ \forall x_1 \cdots \forall x_k\ \left(Y(x_1, \ldots, x_k)\ \Leftrightarrow\ A(x_1, \ldots, x_k)\right)}\ (\exists\text{-in con } Y{:=}P)$$

$\square$

- **Intuición:** Esquema de comprensión = «eslabón perdido» entre las lógicas de 1$^{\text{er}}$ y de 2$^{\text{do}}$ orden (véase más adelante)

La lógica de 2^{do} orden como teoría de 1^{er} orden (1/4)

- **Observación.** Sintácticamente, el lenguaje de la lógica de 2^{do} orden es un lenguaje de 1^{er} orden con múltiples tipos:
  - Un tipo ι de los individuos (con las variables x, y, z, etc.)
  - Para cada $k \geq 0$, un tipo o_k de las relaciones de aridad k
    $= $ conjuntos de k-uplas
    (con las variables X, Y, Z, etc.)
- Formalmente, dicho lenguaje viene con:
  - Varios símbolos de función de tipo $\iota^k \to \iota$ (definidos en $\mathcal{V}$)
  - Ningún símbolo de función de tipo $\cdots \to o_k$
    $\Rightarrow$ Los únicos términos de tipo o_k son las variables $(X, Y, Z$, etc.)
  - Para cada $k \geq 0$, un símbolo de predicado $@_k$ de tipo $o_k \times \iota^k$
    Notación: $X(t_1, \ldots, t_k) \; :\equiv \; @_k(X, t_1, \ldots, t_k)$
- ¿ Cúal es la diferencia entre la lógica de 2^{do} orden y la lógica de 1^{er} orden basada en el lenguaje anterior ?

La lógica de 2$^{\text{do}}$ orden como teoría de 1$^{\text{er}}$ orden (2/4)

- Como siempre en lógica de 1$^{\text{er}}$ orden con múltiples tipos, el lenguaje de fórmulas inducido por el vocabulario anterior introduce cuantificaciones ($\forall$ y $\exists$) para cada tipo:

Fórmulas

$$
\begin{array}{rcl}
A, B & ::= & @_k(X, t_1, \ldots, t_k) \mid \top \mid \bot \\
& \mid & A \wedge B \mid A \vee B \mid A \Rightarrow B \\
& \mid & \underbrace{\forall x\, A \mid \exists x\, A}_{\text{cuant. de tipo } \iota} \mid \underbrace{\forall X\, A \mid \exists X\, A}_{\text{cuant. de tipo } o_k} \quad (k \geq 0)
\end{array}
$$

Recordatorio: Los únicos términos de tipo o_k son las variables (X, Y, Z, etc.)

- Correspondientes reglas para las cuantificaciones universales:

$$
\frac{\Gamma \vdash A}{\Gamma \vdash \forall x\, A} \text{ si } x \notin FV(\Gamma) \qquad \frac{\Gamma \vdash \forall x\, A}{\Gamma \vdash A[x := u]}
$$

$$
\frac{\Gamma \vdash A}{\Gamma \vdash \forall X\, A} \text{ si } X \notin FV(\Gamma) \qquad \frac{\Gamma \vdash \forall X\, A}{\Gamma \vdash A[X := Y]} \text{ si } X, Y : o_k
$$

Pues los únicos términos de tipo o_k son las variables (X, Y, Z, etc.)

La lógica de 2$^{\text{do}}$ orden como teoría de 1$^{\text{er}}$ orden (3/4)

- La diferencia entre la lógica de 1$^{\text{er}}$ orden (basada en los tipos ι y o_k) y la lógica de 2$^{\text{do}}$ orden yace en las reglas ($\forall$-el) de tipo o_k:

Regla ($\forall$-el) de 1$^{\text{er}}$ orden	**Regla ($\forall$-el) de 2$^{\text{do}}$ orden**
$\dfrac{\Gamma \vdash \forall X\, A}{\Gamma \vdash A[X := Y]}$ si $\sharp X{=}\sharp Y$	$\dfrac{\Gamma \vdash \forall X\, A}{\Gamma \vdash A[X := P]}$ si $\sharp X{=}\sharp P$

$+$ diferencia análoga en las reglas ($\exists$-in) de tipo o_k

1$^{\text{er}}$ orden: Sólo se puede sustituir $X : o_k$ por otra variable $Y : o_k$

2$^{\text{do}}$ orden: Se puede sustituir $X : o_k$ por cualquier $P :\equiv \hat{x}_1 \cdots \hat{x}_k\ A_0$

- **Eslabón perdido:** Se puede simular la regla de eliminación de 2$^{\text{do}}$ orden «$X := P$» mediante el axioma de comprensión:

$$\exists Y\ \forall x_1 \cdots \forall x_k\ \big(Y(x_1, \ldots, x_k) \ \Leftrightarrow\ P(x_1, \ldots, x_k)\big)$$

La lógica de 2do orden como teoría de 1er orden (4/4)

Teorema (La lógica de 2do orden como teoría de 1er orden)

Para toda fórmula A, los siguientes enunciados son equivalentes:

1. A es derivable en lógica de 2do orden (sin axiomas)
2. A es derivable en la teoría de 1er orden (con tipos ι y o_k, $k \geq 0$) cuyos axiomas son todos los axiomas de comprensión

$$\exists Y\ \forall x_1 \cdots \forall x_k\ \left(Y(x_1, \ldots, x_k)\ \Leftrightarrow\ A(x_1, \ldots, x_k)\right)$$

Demostración. Ejercicio.

- Para resumir:

Lógica de 2do orden $=$ **Lógica de 1er orden** $+$ **esquema de comprensión**

- A partir de esta caracterización (al 1er orden) se deduce la noción correcta (y completa) de modelo de la lógica de 2do orden (Ejercicio)

Plan

1. Introducción
2. Sistema F en el estilo de Church
3. Tipos de datos en el sistema F
4. Sistema F en el estilo de Curry
5. El teorema de normalización fuerte
6. Lógica de 2^{do} orden: sistema NJ2
7. **Aritmética intuicionista de 2^{do} orden: sistema HA2**
8. Eliminación de cortes en HA2^{-}

Sintaxis de HA2$^-$

- HA2 = Aritmética intuicionista de 2$^{\text{do}}$ orden
  (Individuos = enteros naturales)

Definición (Términos y fórmulas de HA2)

Términos	t, t'	$::=$	$x \mid 0 \mid s(t) \mid \mathrm{pred}(t) \mid t + u \mid t \times u$
Fórmulas	A, B	$::=$	$X(t_1, \ldots, t_k) \mid A \Rightarrow B \mid \forall x\, A \mid \forall X\, A$

- Relación de reducción $t \succ t'$ definida por las 6 reglas:

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
\mathrm{pred}(0) & \succ & 0 & t+0 & \succ & t & t \times 0 & \succ & 0 \\
\mathrm{pred}(s(t)) & \succ & t & t+s(u) & \succ & s(t+u) & t \times s(u) & \succ & (t \times u) + t
\end{array}$$

$+$ clausura contextual

- Relación de reducción $A \succ A'$ sólo inducida por $t \succ t'$
  (Por razones técnicas, no se considera aquí ningún predicado «null»)

Reglas de deducción de HA2

Reglas de deducción del sistema NJ2:

$$\frac{}{\Gamma \vdash A'} \ \text{si } A' \cong A \in \Gamma$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \qquad \frac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x\, A} \ \text{si } x \notin FV(\Gamma) \qquad \frac{\Gamma \vdash \forall x\, A}{\Gamma \vdash A'} \ \text{si } A' \cong A[x := u]$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall X\, A} \ \text{si } X \notin FV(\Gamma) \qquad \frac{\Gamma \vdash \forall X\, A}{\Gamma \vdash A'} \ \text{si } A' \cong A[X := P]$$

Axiomas específicos de HA2:

- Axioma de no confusión: $\forall x\,(s(x) = 0 \Rightarrow \bot)$
- Axioma de inducción: $\forall x\,(x \in \mathbb{N})$, o de modo equivalente:

$$\forall Z\,(0 \in Z \Rightarrow \forall y\,(y \in Z \Rightarrow s(y) \in Z) \Rightarrow \forall x\,(x \in Z))$$

Eliminación del axioma de inducción (1/2)

En lógica de 2$^{\text{do}}$ orden, no se necesita el axioma de inducción. En efecto:

- El conjunto $\mathbb{N}$ de los enteros naturales es definible:

$$\mathbb{N} \;:\equiv\; \{x \;:\; \forall Z\,(0 \in Z \Rightarrow \forall y\,(y \in Z \Rightarrow s(y) \in Z) \Rightarrow x \in Z)\}$$

- Se puede trabajar con cuantificaciones de 1$^{\text{er}}$ orden relativizadas a $\mathbb{N}$:

$$\begin{array}{rcl}
(\forall x \in \mathbb{N})A(x) & :\equiv & \forall x\,(x \in \mathbb{N} \Rightarrow A(x)) \\
(\exists x \in \mathbb{N})A(x) & :\equiv & \forall Z\,(\forall x\,(x \in \mathbb{N} \Rightarrow A(x) \Rightarrow Z) \;\Rightarrow\; Z) \\
 & \Leftrightarrow & \exists x\,(x \in \mathbb{N} \land A(x))
\end{array}$$

Lema (Inducción relativizada)

El axioma de inducción relativizado a $\mathbb{N}$ es derivable en el sistema NJ2:

$$\forall Z\,(0 \in Z \Rightarrow (\forall y \in \mathbb{N})(y \in Z \Rightarrow s(y) \in Z) \Rightarrow (\forall x \in \mathbb{N})(x \in Z))$$

Demostración. Ejercicio.

Eliminación del axioma de inducción (2/2)

- Formalmente, se define la operación de relativización $A \mapsto A^{\mathbb{N}}$ por:

$$
\begin{array}{rcl}
(X(t_1,\ldots,t_n))^{\mathbb{N}} & :\equiv & X(t_1,\ldots,t_n) \\
(A \Rightarrow B)^{\mathbb{N}} & :\equiv & A^{\mathbb{N}} \Rightarrow B^{\mathbb{N}} \\
(\forall x\, A)^{\mathbb{N}} & :\equiv & \forall x\,(x \in \mathbb{N} \Rightarrow A^{\mathbb{N}}) \\
(\forall X\, A)^{\mathbb{N}} & :\equiv & \forall X\, A^{\mathbb{N}}
\end{array}
$$

- Escribiendo $\mathrm{HA2}^- \;:=\; \mathrm{HA2} - \text{Inducción}$, se demuestra que:

Teorema (Simulación de HA2 en $\mathrm{HA2}^-$)

Para toda fórmula cerrada A: $\quad \mathrm{HA2} \vdash A \quad$ sii $\quad \mathrm{HA2}^- \vdash A^{\mathbb{N}}$

Demostración. Ejercicio.

- En lo siguiente, trabajaremos en el sistema $\mathrm{HA2}^-$ donde el axioma de no confusión está expresado con la regla:

$$
\frac{\Gamma \vdash s(t) = 0}{\Gamma \vdash A} \qquad \text{donde} \quad s(t) = 0 \;:\equiv\; \forall Z\,(Z(s(t)) \Rightarrow Z(0))
$$

La noción de corte (1/2)

- **Recordatorio:** corte = trozo de derivación formado por una introducción de cierta construccón inmediatamente seguida por una eliminación de la misma construcción

  El sistema HA2$^-$ sólo tiene tres formas de corte:

- **Corte de implicación:**

$$\dfrac{\dfrac{\begin{array}{c}\vdots\; d\\ \Gamma, A \vdash B\end{array}}{\Gamma \vdash A \Rightarrow B} \qquad \begin{array}{c}\vdots\; d'\\ \Gamma \vdash A\end{array}}{\Gamma \vdash B} \quad \leadsto \quad \begin{array}{c}\vdots\; d'\\ \Gamma \vdash A\\ \vdots\; d[\mathrm{ax}(A){:=}d']\\ \Gamma \vdash B\end{array}$$

La noción de corte (2/2)

- **Corte de $\forall$ de 1$^{\text{er}}$ orden:** (con $A' \cong A[x := u]$)

$$\dfrac{\dfrac{\begin{array}{c}\vdots\ d\\ \Gamma \vdash A\end{array}}{\Gamma \vdash \forall x\, A}\,(\forall^1\text{-in})}{\Gamma \vdash A'}\,(\forall^1\text{-el}) \quad \rightsquigarrow \quad \dfrac{\begin{array}{c}\vdots\ d[x{:=}u]\\ \Gamma \vdash A[x := u]\end{array}}{\Gamma \vdash A'}\,(\text{Conv})$$

- **Corte de $\forall$ de 2$^{\text{do}}$ orden:**: (con $A' \cong A[X := P]$, $\sharp X = \sharp P$)

$$\dfrac{\dfrac{\begin{array}{c}\vdots\ d\\ \Gamma \vdash A\end{array}}{\Gamma \vdash \forall X\, A}\,(\forall^2\text{-in})}{\Gamma \vdash A'}\,(\forall^2\text{-el}) \quad \rightsquigarrow \quad \dfrac{\begin{array}{c}\vdots\ d[X{:=}P]\\ \Gamma \vdash A[X := P]\end{array}}{\Gamma \vdash A'}\,(\text{Conv})$$

Cortes derivados

Las codificaciones de $\wedge$, $\vee$, $\exists$, etc. permiten derivar los otros cortes:

- **Cortes de conjunción:**

$$\dfrac{\dfrac{\dfrac{\vdots\, d_1}{\Gamma \vdash A} \quad \dfrac{\vdots\, d_2}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\,(\wedge\text{-in})}{\Gamma \vdash A}\,(\wedge\text{-el}_1) \quad \rightsquigarrow \quad \dfrac{\vdots\, d_1}{\Gamma \vdash A} \qquad (+\text{ corte simétrico con } (\wedge\text{-el}_2))$$

- **Cortes de disyunción:**

$$\dfrac{\dfrac{\dfrac{\vdots\, d}{\Gamma \vdash A}}{\Gamma \vdash A \vee B}\,(\vee\text{-in}_1) \quad \dfrac{\vdots\, d'_1}{\Gamma, A \vdash C} \quad \dfrac{\vdots\, d'_2}{\Gamma, B \vdash C}}{\Gamma \vdash C}\,(\vee\text{-el}) \quad \rightsquigarrow \quad \dfrac{\dfrac{\vdots\, d}{\Gamma \vdash A}}{\vdots\, d'_1[\mathsf{ax}(A) := d]}\;\Gamma \vdash C$$

$(+\text{ corte simétrico con } (\vee\text{-in}_2))$

- **Cortes de $\exists$ (1er y 2do orden):** Ejercicio

Eliminación de cortes

Teorema (Eliminación de los cortes) [Girard '70]

El sistema formado por las 3 reglas de reducción anteriores es fuertemente normalizante, en el sentido de que no existe ninguna sucesión infinita de reducciones (entre derivaciones de un mismo secuente):

$$\nexists(d_0 \rightsquigarrow d_1 \rightsquigarrow d_2 \rightsquigarrow \cdots \rightsquigarrow d_i \rightsquigarrow d_{i+1} \rightsquigarrow \cdots)$$

Demostración. Próxima sección.

Corolario (Derivaciones sin cortes + consistencia)

1. Todo secuente derivable en $HA2^-$ tiene una derivación sin cortes
2. El secuente $\vdash \perp$ (con $\perp :\equiv \forall Z\, Z$) no es derivable en $HA2^-$
3. Si la fórmula $t = u$ (definida como $\forall Z\,(Z(t) \Rightarrow Z(u))$) es derivable sin contexto en el sistema $HA2^-$, entonces $t \cong u$

Demostración. Ejercicio.

Plan

1. Introducción
2. Sistema F en el estilo de Church
3. Tipos de datos en el sistema F
4. Sistema F en el estilo de Curry
5. El teorema de normalización fuerte
6. Lógica de 2do orden: sistema NJ2
7. Aritmética intuicionista de 2do orden: sistema HA2
8. Eliminación de cortes en HA2$^-$

Arquitectura de la demostración

Idea: Deducir el teorema de eliminación de cortes para el sistema $\mathrm{HA2}^-$ del teorema de normalización fuerte para el sistema F

1. Definir traducciones:

HA2⁻		**Systema F**
fórmula A	$\mapsto$	tipo A^*
(contexto lógico	$\mapsto$	contexto de tipado)
derivación d de A	$\mapsto$	término $d^* : A^*$

2. Verificar que cada reducción de corte (en $\mathrm{HA2}^-$) corresponde a uno o múltiples pasos de reducción (en el sistema F)

3. Normalización fuerte (F) $\Rightarrow$ Eliminación de cortes ($\mathrm{HA2}^-$)

Para ello, se trabaja en un sistema F «inconsistente», enriquecido con una constante $\Omega : \forall\alpha\,.\,\alpha$ (Esto no afecta la propiedad de norm. fuerte)

Traducción de las fórmulas de HA2^{-} (1/2)

- A cada variable de 2$^{\text{do}}$ orden X se asocia una variable de tipo α_X
- Se traduce cada fórmula A de HA2^{-} en un tipo A^* del sistema F:

$$\begin{array}{rcl} (X(t_1,\ldots,t_n))^* & :\equiv & \alpha_X \\ (A \Rightarrow B)^* & :\equiv & A^* \to B^* \\ (\forall x\, A)^* & :\equiv & A^* \\ (\forall X\, A)^* & :\equiv & \forall \alpha_X\,.\, A^* \end{array}$$

- **Obs.:** Todas las construcciones de 1$^{\text{er}}$ orden desaparecen

Proposición (Sustitutividad y conversión)

1. $(A[x := u])^* \equiv A^*$
2. $(A[X := P])^* \equiv A^*[\alpha_X := B^*]$ (si $P \equiv \hat{x}_1 \cdots \hat{x}_k\ B$)
3. Si $A \cong A'$, entonces $A^* \equiv A'^*$

Demostración. Ejercicio.

Traducción de las fórmulas de HA2^{-} (2/2)

- Test: traducción de las fórmulas definidas:

$$
\begin{array}{rcl}
(A \wedge B)^* & \equiv & A^* \times B^* \qquad \text{(producto cartesiano del sistema F)} \\
(A \vee B)^* & \equiv & A^* + B^* \qquad \text{(suma directa)} \\
(t = u)^* & \equiv & (\forall X\,(X(t) \Rightarrow X(u)))^* \quad \equiv \quad \forall \alpha_X \,.\, \alpha_X \to \alpha_X \quad \equiv \quad \mathrm{Unit}
\end{array}
$$

$\Rightarrow$ Pruebas de igualdad sin contenido computacional

- **Traducción de los contextos:** Cada contexto lógico

$$\Gamma \quad \equiv \quad A_1,\ \ldots,\ A_n$$

está traducido como un contexto de tipado del sistema F

$$\Gamma^* \quad :\equiv \quad \xi_1 : A_1^*,\ \ldots,\ \xi_n : A_n^*$$

asociando a cada hipótesis A_i una variable $\xi_i : A_i^*$

Traducción de las derivaciones (1/4)

Definición (Traducción $d \mapsto d^*$)

A cada derivación $d : (\Gamma \vdash A)$ en el sistema HA2$^-$ se asocia un término d^* del sistema F tal que $FV(d^*) \subseteq \{\xi_1, \ldots, \xi_n\}$, donde $\xi_1, \ldots, \xi_n$ son las variables asociadas a las hipótesis en Γ.

Formalmente:

- **Regla axioma:** (con $A' \cong A \in \Gamma$)

$$\left(\frac{}{\Gamma \vdash A'}\,{}^{(\mathrm{ax})}\right)^* \quad :\equiv \quad \xi$$

donde ξ es la variable asociada a la hipótesis A en el contexto Γ

- **No confusión:**

$$\left(\frac{\begin{array}{c}\vdots\ d\\ \Gamma \vdash s(t) = 0\end{array}}{\Gamma \vdash A}\right)^* \quad :\equiv \quad \Omega\ (\mathrm{Unit} \to A^*)\ d^*$$

Traducción de las derivaciones (2/4)

Definición (Traducción $d \mapsto d^*$, continuación)

- **Introducción de $\Rightarrow$:**

$$\left(\dfrac{\begin{matrix} \vdots\ d \\ \Gamma, A \vdash B \end{matrix}}{\Gamma \vdash A \Rightarrow B}\ (\Rightarrow\text{-in}) \right)^* \ :\equiv\ \lambda\xi : A^* \,.\, d^*$$

donde ξ es la variable asociada a la hipótesis A en el contexto Γ, A

- **Eliminación de $\Rightarrow$:**

$$\left(\dfrac{\begin{matrix} \vdots\ d \\ \Gamma \vdash A \Rightarrow B \end{matrix} \quad \begin{matrix} \vdots\ d' \\ \Gamma \vdash A \end{matrix}}{\Gamma \vdash B}\ (\Rightarrow\text{-el}) \right)^* \ :\equiv\ d^*\, d'^*$$

Obs.: Los cortes de $\Rightarrow$ se traducen en redexes de 1$^{\text{ra}}$ forma

Traducción de las derivaciones (3/4)

Definición (Traducción $d \mapsto d^*$, continuación)

- **Introducción de $\forall$ de 1^er^ orden:** (con $x \notin FV(\Gamma)$)

$$\left(\dfrac{\begin{array}{c}\vdots\, d\\ \Gamma \vdash A\end{array}}{\Gamma \vdash \forall x\, A}\ {\scriptstyle(\forall^1\text{-in})}\right)^* \;:\equiv\; (\lambda\xi : A^* \,.\, \xi)\, d^*$$

- **Eliminación de $\forall$ de 1^er^ orden:** (con $A' \cong A[x := t]$)

$$\left(\dfrac{\begin{array}{c}\vdots\, d\\ \Gamma \vdash \forall x\, A\end{array}}{\Gamma \vdash A'}\ {\scriptstyle(\forall^1\text{-el})}\right)^* \;:\equiv\; d^*$$

Obs.: Se inserta una identidad «trucha» en la traducción de la regla ($\forall^1$-in) para que los cortes de $\forall$ de 1^er^ orden se traduzcan en redexes (de 1^ra^ forma)

Traducción de las derivaciones (4/4)

Definición (Traducción $d \mapsto d^*$, fin)

- **Introducción de $\forall$ de 2$^{\text{do}}$ orden:** (con $X \notin FV(\Gamma)$)

$$\left(\dfrac{\begin{array}{c}\vdots\ d\\ \Gamma \vdash A\end{array}}{\Gamma \vdash \forall X\, A}\ (\forall^2\text{-in})\right)^* \;:\equiv\; \lambda\alpha_X \,.\, d^*$$

- **Eliminación de $\forall$ de 2$^{\text{do}}$ orden:** (con $A' \cong A[x := P]$)

$$\left(\dfrac{\begin{array}{c}\vdots\ d\\ \Gamma \vdash \forall x\, A\end{array}}{\Gamma \vdash A'}\ (\forall^2\text{-el})\right)^* \;:\equiv\; d^*\ B^*$$

donde $P \equiv \hat{x}_1 \cdots \hat{x}_k\ B$

Obs.: Los cortes de $\forall$ de 2$^{\text{do}}$ orden se traducen en redexes de 2$^{\text{da}}$ forma

Eliminación de los cortes en el sistema $HA2^{-}$

Proposición (Propiedades de la traducción $d \mapsto d^*$)

1. **Invariante de tipado**

   Para toda derivación $d : (A_1, \ldots, A_n \vdash B)$ (sistema $HA2^{-}$)
   tenemos que $\xi_1 : A_1^*, \ldots, \xi_n : A_n^* \vdash d^* : B^*$ (sistema F)

   donde $\xi_1, \ldots, \xi_n$ son las variables asociadas a las hipótesis $A_1, \ldots, A_n$

2. **Invariante de reducción**

   Toda reducción de corte $d \rightsquigarrow d'$ en el sistema $HA2^{-}$ se traduce en un paso de reducción $d^* \succ d'^*$ en el sistema F

Demostración. Ejercicio.

Combinando el resultado anterior con el teorema de normalización fuerte para el sistema F, se concluye inmediatamente que la reducción de los cortes es fuertemente normalizante en el sistema $HA2^{-}$ $\square$

Traducción de los enteros naturales

- **Problema:** La traducción borra todos los términos de primer orden
  $\Rightarrow$ *¿Adónde se fueron los enteros naturales?*

- **Respuesta:** Para utilizar el principio de inducción, se necesita relativizar las cuantificaciones de 1^er^ orden con el predicado

$$\mathbb{N}(x) \;:\equiv\; \forall Z\,(0 \in Z \Rightarrow \forall y\,(y \in Z \Rightarrow s(y) \in Z) \Rightarrow x \in Z)$$

cuya traducción en el sistema F es el tipo

$$(\mathbb{N}(x))^* \quad\equiv\quad \forall\alpha_Z\,.\,(\alpha_Z \to (\alpha_Z \to \alpha_Z) \to \alpha_Z) \quad\equiv\quad \mathrm{Nat}$$

Lema (Traducción de los enteros naturales)

Para cada término de la forma $s^n(0)$ (entero concreto)

1. La fórmula $s^n(0) \in \mathbb{N}$ tiene una única derivación sin cortes en HA2$^-$...
2. ... cuya traducción en el sistema F es el entero de Church $\overline{n}$

Extracción de programas

Proposición (Extracción de programas en el sistema F)

Cada función cuya existencia es derivable en HA2$^-$ es definible en el sistema F por un término de tipo $\mathsf{Nat} \to \mathsf{Nat}$

Demostración. Sea una derivación d (en HA2$^-$) de una fórmula de la forma

$$\forall x \ (x \in \mathbb{N} \Rightarrow \exists y \ (y \in \mathbb{N} \land A(x, y)))$$

Traduciendo la derivación d en el sistema F, se obtiene un término

$$d^* \quad : \quad \mathsf{Nat} \to \forall \alpha \,.\, (\mathsf{Nat} \times A^* \to \alpha) \to \alpha$$

(usando la codificación de $\exists$ al 2$^{\text{do}}$ orden), de tal modo que el término

$$\lambda x : \mathsf{Nat} \,.\, d^* x \ \mathsf{Nat} \ \mathsf{fst} \quad : \quad \mathsf{Nat} \to \mathsf{Nat}$$

(donde $\mathsf{fst} : \mathsf{Nat} \times A^* \to \mathsf{Nat}$ es la primera proyección) calcule la función deseada □

Obs.: Estamos trampeando un poco, pues el cálculo del término d^* podría ser bloqueado por la constante inerte Ω. Dos opciones para arreglar el argumento:

1. Demostrar que Ω nunca bloquea el cálculo de d^*
2. Definir una traducción modificada que no necesita Ω [cf Proofs and Types]

Teorema de representación

Más generalmente:

Proposición (Extracción de programas en el sistema F)

Si d es una derivación de $\vdash (\forall \vec{x} \in \mathbb{N})(\exists y \in \mathbb{N})\, A(\vec{x}, y)$ (en el sistema HA2$^-$), entonces el término

$$F \;:\equiv\; \lambda x_1, \ldots, x_k : \mathrm{Nat}\,.\, d^* x_1 \cdots x_k \;\mathrm{Nat}\;\mathrm{fst} \;:\; \mathrm{Nat}^k \to \mathrm{Nat} \qquad \text{(sistema F)}$$

calcula una función tal que $\vdash A(\vec{n}, F(\vec{n}))$ para todo $\vec{n} = (n_1, \ldots, n_k) \in \mathbb{N}^k$

Teorema de representación

Las funciones cuya existencia es demostrable en HA2$^-$ son exactamente las funciones definibles en el sistema F

Demostración. Parte directa (existencia en HA2$^-$ $\Rightarrow$ definible en el sistema F): cf proposición anterior.

Parte recíproca (definible en el sistema F $\Rightarrow$ existencia en HA2$^-$): codificación de los términos del sistema F y de la reducción en HA2$^-$ (ejercicio muy técnico). $\square$

Conclusión: Sistema F $=$ lenguaje de programación de HA2