Introducción a la correspondencia entre pruebas y programas:

El cálculo lambda puro

Alexandre Miquel

abril de 2021

Introducción

Alonzo **Church** (1903–1995)

1936: *An Unsolvable Problem of Elementary Number Theory*

1936: *A Note on the Entscheidungsproblem*

1941: *The Calculi of Lambda Conversions*

El problema de la decisión („*Entscheidungsproblem*“)

- En 1936, Church definió un problema de «aritmética elemental»... ... que no se puede resolver por «medios algorítmicos»
  - Medios algorítmicos = calculable por un término lambda
  - Problema sin solución algorítmica = problema de la parada
- El *Entscheidungsproblem* fue resuelto de modo independiente en 1936 por Alan Turing (1912–1954)
  - Medios algorítmicos = calculable por una máquina de Turing

Historia breve del cálculo lambda

Objetivo inicial: Definir un marco unificador similar a la teoría de conjuntos, donde los objetos primitivos son las funciones

- 1924: La lógica combinatoria, por Moses Schönfinkel (1889–1942)
- 1932: Church introduce un cálculo de funciones puras, con un sistema lógico en el estilo de Frege e Hilbert
- 1935: Kleene y Rosser (estudiantes de Church) demuestran que el sistema lógico definido por Church es inconsistente
- 1936: Church resuelve el problema de la decisión, usando la parte puramente computacional de su cálculo
- 1936: Turing resuelve el mismo problema, con sus máquinas
  $\Rightarrow$ Church lo invita en Princeton para dirigir su tesis de doctorado
- 1940: Church introduce el cálculo lambda simplemente tipado

¿Qué se necesita para trabajar con funciones?

- Variables (x, y, z, etc.) y operaciones básicas ($+$, $\times$, etc.) para formar expresiones, como por ejemplo: $3x+1$

- Un mecanismo para construir una función, abstrayendo una expresión con respecto a una variable. Notación:

$$\lambda x\,.\,3x+1 \qquad = \; (x \mapsto 3x+1)$$

- Un mecanismo para aplicar una función:

$$(\lambda x\,.\,3x+1)(4)$$

- Un mecanismo para evaluar una función: la β-reducción

$$(\lambda x\,.\,3x+1)(4) \;\to_\beta\; 3(4)+1 \;\to\; 12+1 \;\to\; 13$$

β-reducción = sustitución del argumento formal (variable) por el argumento real (expresión cualquiera)

- **Ideas de Church:** (1) Todo objeto matemático es una función (2) Toda computación es una sucesión de β-reducciones

El origen del λ del cálculo lambda

$(a_1, \ldots, \hat{a}_i, \ldots, a_n)$	la tupla $(a_1, \ldots, a_n)$ sin a_i
$\hat{x} \,.\, 3x + 1 \geq 0$	$= \{x : 3x + 1 \geq 0\}$ (Russell-Whitehead)
$\wedge x \,.\, 3x + 1, \qquad \lambda x \,.\, 3x + 1$	$= (x \mapsto 3x + 1)$ (Church)

La "biblia" del cálculo lambda:

H. Barendregt. *The Lambda Calculus: Its Syntax and Semantics.* Studies in Logic and the Foundations of Mathematics. North Holland, 1984

Plan

Plan

Sintaxis del cálculo lambda

Definición (Términos lambda)

Términos lambda $\quad M, N \quad ::= \quad x \quad | \quad \lambda x\,.\,M \quad | \quad MN$

Se escribe Λ al conjunto de todos los términos lambda

- **Abreviaturas:**

$$\begin{aligned} \lambda x_1 x_2 \cdots x_n\,.\,M &:\equiv \lambda x_1\,.\,\lambda x_2\,.\,\cdots \lambda x_n\,.\,M \\ MN_1 N_2 \cdots N_n &:\equiv (\cdots((MN_1)N_2)\cdots)M_n \end{aligned}$$

- Variables libres $FV(M)$ y variables ligadas $BV(M)$:

$$\begin{aligned} FV(x) &= \{x\} & BV(x) &= \varnothing \\ FV(\lambda x\,.\,M) &= FV(M) \setminus \{x\} & BV(\lambda x\,.\,M) &= BV(M) \cup \{x\} \\ FV(MN) &= FV(M) \cup FV(N) & BV(MN) &= BV(M) \cup BV(N) \end{aligned}$$

Sustitución y α-equivalencia

Ahora se trata de definir:

- la relación de α-equivalencia $M_1 \equiv_\alpha M_2$
- la operación de sustitución $M[x := N]$ (a menos de α-equivalencia)

con los cambios de nombres necesarios para evitar capturas:

$$\begin{array}{rcl} (\lambda x\,.\,x\,y)[y := x] & \not\equiv & \lambda x\,.\,x\,x \\ & \equiv_\alpha & \lambda z\,.\,z\,x \\ & \equiv_\alpha & \lambda y\,.\,y\,x \end{array}$$

Para ello:

1. Se define una primera operación parcial de sustitución $M\langle x := N\rangle$ sin cambios de nombres de variables
2. Se define (a partir de la operación anterior) la relación $M_1 \equiv_\alpha M_2$, y se demuestra que $M\langle x := N\rangle$ es compatible con $M_1 \equiv_\alpha M_2$
3. Se define la operación (total) de sustitución $M[x := N]$ a partir de la relación $M\langle x := N\rangle$, razonando sobre las clases de α-equivalencia

Sustitución sin cambios de variables ligadas (1/2)

Definición (Sustitución «ingenua», sin cambios de variables ligadas)

Se define la operación parcial $M\langle x := N\rangle$ por las siguientes cláusulas:

- $x\langle x := N\rangle \;:\equiv\; N$
- $y\langle x := N\rangle \;:\equiv\; y$ (suponiendo que $y \not\equiv x$)
- $(\lambda x \,.\, M)\langle x := N\rangle \;:\equiv\; \lambda x \,.\, M$
- $(\lambda y \,.\, M)\langle x := N\rangle \;:\equiv\; \lambda y \,.\, M\langle x := N\rangle$ si $x \notin FV(M)$ o $y \notin FV(N)$
- $(\lambda y \,.\, M)\langle x := N\rangle$ no está definido si $x \in FV(M)$ e $y \in FV(N)$
- $(M_1 M_2)\langle x := N\rangle \;:\equiv\; (M_1\langle x := N\rangle)(M_2\langle x := N\rangle)$

Ejemplos:

- $(\lambda y \,.\, y\, x)\langle x := z\, z\rangle \;\equiv\; \lambda y \,.\, y\,(z\, z)$
- $(\lambda z \,.\, z\, x)\langle x := z\, z\rangle$ no está definido

Sustitución sin cambios de variables ligadas (2/2)

Lema (Condiciones suficientes de definición)

(1) Si $x \notin FV(M)$, entonces $M\langle x := N\rangle \equiv M$ (siempre definido)

(2) Si $BV(M) \cap FV(N) = \varnothing$, entonces $M\langle x := N\rangle$ está definido

Demostración. (1) y (2) se demuestran por inducción sobre M (Ejercicio). □

Lema de sustitución

Para todos M, N, P, $x \not\equiv y$ tales que $x \notin FV(P)$, tenemos que:

$$M\langle x := N\rangle\langle y := P\rangle \;\equiv\; M\langle y := P\rangle\langle x := N\langle y := P\rangle\rangle$$

bajo la hipótesis que ambos lados estén definidos

Demostración. Por inducción sobre el término M (Ejercicio). □

α-equivalencia

Definición (α-equivalencia)

Se define inductivamente la relación de α-equivalencia $M_1 \equiv_\alpha M_2$ por las siguientes reglas:

$$\frac{}{x \equiv_\alpha x} \qquad \frac{M_1 \equiv_\alpha M_2 \quad N_1 \equiv_\alpha N_2}{M_1 N_1 \ \equiv_\alpha \ M_2 N_2}$$

$$\frac{M_1\langle x_1 := z\rangle \ \equiv_\alpha \ M_2\langle x_2 := z\rangle}{\lambda x_1 \,.\, M_1 \ \equiv_\alpha \ \lambda x_2 \,.\, M_2} \ (z \text{ fresca})$$

(z fresca $=$ $z \not\equiv x_1$, $z \not\equiv x_2$ y $z \notin FV(M_1) \cup BV(M_1) \cup FV(M_2) \cup BV(M_2)$)

Proposición (Propiedades de la α-equivalencia)

1. La α-equivalencia es una congruencia (= equivalencia compatible con λ y @)
2. Si $M \equiv_\alpha M'$, $N \equiv_\alpha N'$ y si $M\langle x := N\rangle$ y $M'\langle x := N'\rangle$ están definidos, entonces $M\langle x := N\rangle \equiv_\alpha M'\langle x := N'\rangle$

Demostración. Ejercicio

Sustitución con cambios de variables ligadas (1/2)

Definición (Sustitución)

Se define la operación (total) de sustitución $M[x := N]$ a menos de α-equivalencia por:

$$M[x := N] \ :\equiv_\alpha \ M'\langle x := N\rangle$$

tomando cualquier $M' \equiv_\alpha M$ tal que $M'\langle x := N\rangle$ esté definido

Proposición (Compatibilidad)

Si $M \equiv_\alpha M'$ y $N \equiv_\alpha N'$, entonces $M[x := N] \equiv_\alpha M'[x := N']$

Lema de sustitución

Para todos M, N, P, $x \not\equiv y$ tales que $x \notin FV(P)$, tenemos que:

$$M[x := N][y := P] \ \equiv_\alpha \ M[y := P][x := N[y := P]]$$

Sustitución con cambios de variables ligadas (2/2)

- En las diapositivas anteriores, definimos la operación de sustitución $M[x := N]$ a menos de α-equivalencia
- Dicha operación también se puede definir directamente [Church 1941] sobre los términos lambda del modo siguiente:

- $x[x := N] \; :\equiv \; N$
- $y[x := N] \; :\equiv \; y$ (suponiendo que $y \not\equiv x$)
- $(\lambda x \,.\, M)[x := N] \; :\equiv \; \lambda x \,.\, M$
- $(\lambda y \,.\, M)[x := N] \; :\equiv \; \lambda y \,.\, M[x := N]$ si $x \notin FV(M)$ o $y \notin FV(N)$
- $(\lambda y \,.\, M)[x := N] \; :\equiv \; \lambda z \,.\, M[y := z][x := N]$
  si $x \in FV(M)$ e $y \in FV(N)$, tomando z fresca
- $(M_1 M_2)[x := N] \; :\equiv \; (M_1[x := N])(M_2[x := N])$

Obs.: Se necesita verificar que la definición está bien fundada (5ta cláusula)

- Esta definición es adecuada para una implementación

Plan

Contextos con un agujero (1/2)

- Los contextos (con un agujero) están definidos por la gramática:

Contextos $\quad C \quad ::= \quad [] \quad | \quad \lambda x\,.\,C \quad | \quad CM \quad | \quad MC$

Intuición: Contexto con un agujero $=$
término con una única ocurrencia de la construcción [] (el «agujero»)

- Dados un contexto C y un término M, se escribe $C[M]$ al término obtenido remplazando en C la única ocurrencia de [] por M.

  Se trata de un remplazo sintáctico, sin cambio de variable:

$$
\begin{array}{llll}
\text{Si} & C \equiv [], & \text{entonces} & C[M] :\equiv M \\
\text{Si} & C \equiv \lambda x\,.\,C_0, & \text{entonces} & C[M] :\equiv \lambda x\,.\,C_0[M] \\
\text{Si} & C \equiv C_0 M_0, & \text{entonces} & C[M] :\equiv C_0[M] M_0 \\
\text{Si} & C \equiv M_0 C_0, & \text{entonces} & C[M] :\equiv M_0 C_0[M]
\end{array}
$$

Contextos con un agujero (2/2)

- **Obs.:** Al contrario de la operación de sustitución, la operación $M \mapsto C[M]$ puede capturar variables libres de M, por ejemplo:

$$\text{Si} \quad C \equiv \lambda x\,.\,x\,[\,], \quad \text{entonces} \quad \begin{aligned} C[\mathbf{I}] &\equiv \lambda x\,.\,x\,\mathbf{I} \\ C[y] &\equiv \lambda x\,.\,x\,y \\ C[x] &\equiv \lambda x\,.\,x\,x \end{aligned}$$

  En particular, la α-conversión no tiene sentido sobre los contextos. Sin embargo, tenemos que:

$$\text{Si} \quad M \equiv_\alpha M', \quad \text{entonces} \quad C[M] \equiv_\alpha C[M']$$

- Los contextos también se componen entre sí mediante la operación $C \circ C' \;:\equiv\; C[C']$, de tal modo que $(C \circ C')[M] \;:\equiv\; C[C'[M]]$

- **Dicho de otro modo:** El conjunto de los contextos con un agujero es un monoide (no conmutativo) con elemento neutro $[\,]$, que actúa (por la izquierda) sobre el conjunto Λ mediante la operación $(C, M) \mapsto C[M]$

Relaciones compatibles y clausura contextual (1/2)

- Se dice que una relación binaria $\mathcal{R} \subseteq \Lambda \times \Lambda$ es:
  - compatible cuando $M\,\mathcal{R}\,M'$ implica $C[M]\,\mathcal{R}\,C[M']$ (para todo contexto C con un agujero)
  - sustitutiva cuando $M\,\mathcal{R}\,M'$ implica $M[x := N]\,\mathcal{R}\,M'[x := N]$ (para toda variable x y todo término N)

Definición (Clausura contextual)

Dada una relación binaria $\mathcal{R} \subseteq \Lambda \times \Lambda$, se llama clausura contextual de la relación $\mathcal{R}$ a la relación $\to_{\mathcal{R}}$ definida por las 4 reglas:

$$\frac{M\,\mathcal{R}\,M'}{M \to_{\mathcal{R}} M'}\ (\text{CBase}) \qquad \frac{M \to_{\mathcal{R}} M'}{\lambda x\,.\,M \to_{\mathcal{R}} \lambda x\,.\,M'}\ (\text{CLam})$$

$$\frac{M_1 \to_{\mathcal{R}} M_1'}{M_1 M_2 \to_{\mathcal{R}} M_1' M_2}\ (\text{CApp}_1) \qquad \frac{M_2 \to_{\mathcal{R}} M_2'}{M_1 M_2 \to_{\mathcal{R}} M_1 M_2'}\ (\text{CApp}_2)$$

- **Obs.:** $\to_{\mathcal{R}}$ es la mínima relación compatible que contiene $\mathcal{R}$

Relaciones compatibles y clausura contextual (2/2)

Proposición (Propiedades de la clausura contextual)

Dadas relaciones binarias $\mathcal{R}, \mathcal{S} \subseteq \Lambda \times \Lambda$:

(1) $\to_{\mathcal{R}}$ es la mínima relación compatible que contiene $\mathcal{R}$

(2) Si $\mathcal{R}$ es sustitutiva, entonces $\to_{\mathcal{R}}$ es también es sustitutiva

(3) Si $M \mathcal{R} M'$ implica $FV(M') \subseteq FV(M)$ (para todos M, M')
entonces $M \to_{\mathcal{R}} M'$ implica $FV(M') \subseteq FV(M)$ (para todos M, M')

(Misma propiedad con $=$ en lugar de $\subseteq$)

(4) Tenemos que: $\to_{(\mathcal{R} \cup \mathcal{S})} \; = \; \to_{\mathcal{R}} \cup \to_{\mathcal{S}}$

Relaciones inducidas

Dada una relación $\to \subseteq \Lambda \times \Lambda$, se definen las relaciones:

$\xrightarrow{0}$	$:=$	$\{(x,x) : x \in A\}$	**identidad**
$\xrightarrow{i+1}$	$:=$	$\xrightarrow{i} \circ \to$	**reducción en $i+1$ pasos**
$\xrightarrow{+}$	$:=$	$\bigcup_{i>0} \xrightarrow{i}$	**clausura transitiva**
$\xrightarrow{*}$	$:=$	$\xrightarrow{+} \cup \xrightarrow{0}$	**clausura reflexiva-transitiva**
$\xrightarrow{=}$	$:=$	$\to \cup \xrightarrow{0}$	**clausura reflexiva**
$\leftrightarrow$	$:=$	$\to \cup (\to)^{-1}$	**clausura simétrica**
$\cong$	$:=$	$(\leftrightarrow)^*$	**clausura reflexiva-simétrica-transitiva**

Obs.: Se nota que si $\to$ es compatible (resp. sustitutiva), entonces todas las relaciones inducidas son compatibles (resp. sustitutivas)

Definición de la β-reducción

Definición (β-reducción)

(1) La noción de β-reducción $\beta \subseteq \Lambda \times \Lambda$ está definida por la regla:

$$\frac{}{(\lambda x\,.\,M)\,N \;\; \beta \;\; M[x := N]}$$

(2) La relación $\to_\beta$ de β-reducción en un paso está definida como la clausura contextual de la relación β

Propiedades básicas de la β-reducción

(1) La relación β es sustitutiva, y por lo tanto:

(2) La relación $\to_\beta$ es compatible y sustitutiva

Además:

(3) Si $M \to_\beta M'$, entonces $FV(M') \subseteq FV(M)$

Obs.: Variables libres pueden desaparecer durante la β-reducción, por ejemplo:

$$(\lambda x\,.\,(\lambda y\,.\,y))\,z \;\to_\beta\; \lambda y\,.\,y$$

Ejemplos

Sea consideran los términos:

$$\mathbf{I} :\equiv \lambda x\,.\,x, \quad \mathbf{K} :\equiv \lambda xy\,.\,x, \quad \mathbf{B} :\equiv \lambda xyz\,.\,x\,(y\,z), \quad \Delta :\equiv \lambda x\,.\,x\,x$$

Para todos términos M, N, P, tenemos que:

$$\begin{aligned}
\mathbf{I}\,M &\equiv (\lambda x\,.\,x)\,M \;\to_\beta\; M \\
\mathbf{K}\,M\,N &\equiv (\lambda x\,.\,\lambda y\,.\,x)\,M\,N \;\to_\beta\; (\lambda y\,.\,M)\,N \;\to_\beta\; M \\
\mathbf{B}\,M\,N\,P &\equiv (\lambda x\,.\,\lambda y\,.\,\lambda z\,.\,x\,(y\,z))\,M\,N\,P \\
&\to_\beta (\lambda y\,.\,\lambda z\,.\,M\,(y\,z))\,N\,P \\
&\to_\beta (\lambda z\,.\,M\,(N\,z))\,P \;\to_\beta\; M\,(N\,P) \\
\Delta\,M &\equiv (\lambda x\,.\,x\,x)\,M \;\to_\beta\; M\,M
\end{aligned}$$

Y en particular:

$$\Delta\,\Delta \;\to_\beta\; \Delta\,\Delta$$

Normalización

Definiciones

Dados términos M, M', se dice que:

- M' es en forma (β-)normal cuando $M' \not\rightarrow_\beta$
- M' es una forma (β-)normal de M cuando $M \xrightarrow{*}_\beta M' \not\rightarrow_\beta$
- M es normalizante cuando tiene forma normal
- M es fuertemente normalizante cuando no existe ninguna sucesión infinita de β-reducciones: $M \rightarrow_\beta M_1 \rightarrow_\beta M_2 \rightarrow_\beta M_3 \rightarrow_\beta \cdots$

Ejemplos:

- Los términos $\mathbf{I}$, $\mathbf{K}$, $\mathbf{B}$, Δ son en forma normal
- El término $\Delta\,\mathbf{I} \rightarrow_\beta \mathbf{I}\,\mathbf{I} \rightarrow_\beta \mathbf{I}$ es fuertemente normalizante
- El término $\Delta\,\Delta \rightarrow_\beta \Delta\,\Delta$ no tiene forma normal
- El término $\mathbf{K}\,\mathbf{I}\,(\Delta\,\Delta) \xrightarrow{2}_\beta \mathbf{I}$ es normalizante, pero no es fuertemente normalizante, pues $\mathbf{K}\,\mathbf{I}\,(\Delta\,\Delta) \rightarrow_\beta \mathbf{K}\,\mathbf{I}\,(\Delta\,\Delta)$

Formas normales

Se consideran las dos formas de términos **neu** ("neutros") y **nor** ("normales") definidas por las gramáticas:

$$\begin{array}{rcl} \textbf{neu} & ::= & x \mid \textbf{neu nor} \\ \textbf{nor} & ::= & \textbf{neu} \mid \lambda x \,.\, \textbf{nor} \end{array}$$

Dicho de otro modo, los "normales" están dados por la gramática:

$$\textbf{nor} \quad ::= \quad \lambda x_1 \cdots x_n \,.\, y\ \textbf{nor}_1 \cdots \textbf{nor}_k \qquad (n, k \geq 0)$$

Proposición (Caracterización de las formas normales)

Un término M es en forma β-normal si y sólo si es de la forma **nor**

Demostración: Ejercicio.

Confluencia local de la β-reducción

Se puede demostrar sin dificultad que:

Proposición (Confluencia local)

La β-reducción es localmente confluente:

Si $M \to_\beta M_1'$ y $M \to_\beta M_2'$,
entonces existe M'' tal que
$M_1' \xrightarrow{*}_\beta M''$ y $M_2' \xrightarrow{*}_\beta M''$

$$\begin{tikzcd}
 & M \arrow[dl] \arrow[dr] & \\
M_1' \arrow[dr, dotted, "*"'] & & M_2' \arrow[dl, dotted, "*"] \\
 & M'' &
\end{tikzcd}$$

Problema: No se puede deducir fácilmente la propiedad de confluencia
(En efecto, no se puede usar el lema de Newman, ya que $\to_\beta$ no es normalizante)

Solución: Razonar sobre una relación auxiliar: la reducción paralela
[Martin-Löf 1971]

Reducción paralela (1/7)

Definición (β-reducción paralela)

La relación $\Rightarrow$ de β-reducción paralela está definida inductivamente por las siguientes reglas:

$$\frac{}{M \Rightarrow M}\ \text{(PRefl)} \qquad \frac{M \Rightarrow M' \quad N \Rightarrow N'}{(\lambda x\,.\,M)N \ \Rightarrow\ M'[x := N']}\ \text{(PBeta)}$$

$$\frac{M \Rightarrow M'}{\lambda x\,.\,M \ \Rightarrow\ \lambda x\,.\,M'}\ \text{(PLam)} \qquad \frac{M \Rightarrow M' \quad N \Rightarrow N'}{MN \ \Rightarrow\ M'N'}\ \text{(PApp)}$$

Observaciones:

- La relación $\Rightarrow$ es compatible (obvio por (PRefl), (PLam) y (PApp))
- Tenemos que: $(\rightarrow_\beta) \subset (\Rightarrow) \subset (\xrightarrow{*}_\beta)$ (Demostración: ejercicio)

Lema (Sustitución paralela)

Si $M \Rightarrow M'$ y $N \Rightarrow N'$, entonces $M[x := N] \Rightarrow M'[x := N']$

Reducción paralela (2/7)

Demostración del lema de sustitución paralela. Por inducción sobre la derivación de $M \Rightarrow M'$, distinguiendo los casos en función de la última regla aplicada:

- (PRefl). Tenemos que $M \equiv M'$.
  Se demuestra por inducción sobre M que $M[x := N] \Rightarrow M[x := N']$, distinguiendo los tres casos posibles (variable, abstracción, aplicación).

- (PBeta). Tenemos que $M \equiv (\lambda y \,.\, M_1)M_2$ y $M' \equiv M_1'[y := M_2']$, con $M_1 \Rightarrow M_1'$ y $M_2 \Rightarrow M_2'$. S.p.d.g, se puede suponer que $y \notin FV(M_2 M_2' N N')$.
  Por HI, sabemos que $M_1[x := N] \Rightarrow M_1'[x := N']$ y $M_2[x := N] \Rightarrow M_2'[x := N']$.
  Por (PBeta), se deduce que $M[x := N] \equiv (\lambda y \,.\, M_1[x := N])(M_2[x := N])$
  $\Rightarrow M_1'[x := N'][y := M_2'[x := N']] \equiv (M_1'[y := M_2'])[x := N'] \equiv M'[x := N']$.

- (PLam). Tenemos que $M \equiv \lambda y \,.\, M_1$ y $M' \equiv \lambda y \,.\, M_1'$, con $M_1 \Rightarrow M_1'$.
  S.p.d.g, se puede suponer que $y \notin FV(NN')$.
  Por HI, sabemos que $M_1[x := N] \Rightarrow M_1'[x := N']$. Por (PLam), se deduce que $M[x := N] \equiv \lambda y \,.\, M_1[x := N] \Rightarrow \lambda y \,.\, M_1'[x := N'] \equiv M'[x := N']$.

- (PApp). Tenemos que $M \equiv M_1 M_2$ y $M' \equiv M_1' M_2'$, con $M_1 \Rightarrow M_1'$ y $M_2 \Rightarrow M_2'$.
  Por HI, sabemos que $M_1[x := N] \Rightarrow M_1'[x := N']$ y $M_2[x := N] \Rightarrow M_2'[x := N']$.
  Por (PApp), se deduce que $M[x := N] \equiv M_1[x := N]M_2[x := N]$
  $\Rightarrow M_1'[x := N']M_2'[x := N'] \equiv M'[x := N']$. $\square$

Reducción paralela

Proposición (Propiedad del diamante)

La reducción paralela cumple la propiedad del diamante:

Si $M \Rightarrow M_1'$ y $M \Rightarrow M_2'$,
entonces existe M'' tal que
$M_1' \Rightarrow M''$ y $M_2' \Rightarrow M''$

$$\begin{tikzcd}[column sep=small]
 & M \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\
M_1' \arrow[dr, Rightarrow, dotted] & & M_2' \arrow[dl, Rightarrow, dotted] \\
 & M'' &
\end{tikzcd}$$

Corolario (Confluencia de la reducción paralela)

La reducción paralela es confluente:

Si $M \Rightarrow^* M_1'$ y $M \Rightarrow^* M_2'$,
entonces existe M'' tal que
$M_1' \Rightarrow^* M''$ y $M_2' \Rightarrow^* M''$

$$\begin{tikzcd}[column sep=small]
 & M \arrow[dl, Rightarrow, "*"'] \arrow[dr, Rightarrow, "*"] & \\
M_1' \arrow[dr, Rightarrow, dotted, "*"'] & & M_2' \arrow[dl, Rightarrow, dotted, "*"] \\
 & M'' &
\end{tikzcd}$$

Demostración. Sigue del resultado demostrado en el Práctico 2, Ejercicio 3 (1). □

Reducción paralela

Demostración de la propiedad del diamante. Por inducción sobre las derivaciones de $M \Rightarrow M'_1$ y $M \Rightarrow M'_2$, distinguiendo los casos en función de los pares de reglas que acaban éstas. Se observa que los casos posibles son los siguientes:

$M \Rightarrow M'_2$; $M \Rightarrow M'_1$

	(PRefl)	(PBeta)	(PLam)	(PApp)
(PRefl)	X	X	X	X
(PBeta)	X	X		X
(PLam)	X		X	
(PApp)	X	X		X

- (PRefl)/_. En este caso, tenemos que $M'_1 \equiv M$; basta con tomar $M'' :\equiv M'_2$.
- _/(PRefl). En este caso, tenemos que $M'_2 \equiv M$; basta con tomar $M'' :\equiv M'_1$.

$$\begin{tikzcd}
& M \arrow[dl, equal] \arrow[dr, Rightarrow] & \\
M \arrow[dr, Rightarrow, dotted] & & M'_2 \arrow[dl, equal, dotted] \\
& M'_2 &
\end{tikzcd}
\qquad
\begin{tikzcd}
& M \arrow[dl, Rightarrow] \arrow[dr, equal] & \\
M'_1 \arrow[dr, equal, dotted] & & M \arrow[dl, Rightarrow, dotted] \\
& M'_1 &
\end{tikzcd}$$

(...)

Reducción paralela (5/7)

Demostración de la propiedad del diamante (continuación).

- (PBeta)/(PBeta). En este caso, sabemos que
  - $M \equiv (\lambda x\,.\,N)P$
  - $M'_1 \equiv N'_1[x := P'_1]$, con $N \Rightarrow N'_1$ y $P \Rightarrow P'_1$
  - $M'_2 \equiv N'_2[x := P'_2]$, con $N \Rightarrow N'_2$ y $P \Rightarrow P'_2$

  Por HI, sabemos que

  - existe N'' tal que $N'_1 \Rightarrow N''$ y $N'_2 \Rightarrow N''$
  - existe P'' tal que $P'_1 \Rightarrow P''$ y $P'_2 \Rightarrow P''$

  Tomando $M'' :\equiv N''[x := P'']$, se observa que

  - $M'_1 \equiv N'_1[x := P'_1] \Rightarrow N''[x := P''] \equiv M''$
  - $M'_2 \equiv N'_2[x := P'_2] \Rightarrow N''[x := P''] \equiv M''$

  aplicando 2 veces el lema de sustitución paralela.

$$
\begin{tikzcd}
 & (\lambda x\,.\,N)P \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\
N'_1[x := P'_1] \arrow[dr, Rightarrow, dotted] & & N'_2[x := P'_2] \arrow[dl, Rightarrow, dotted] \\
 & N''[x := P''] &
\end{tikzcd}
$$

(...)

Reducción paralela (6/7)

Demostración de la propiedad del diamante (continuación).

- (PLam)/(PLam). En este caso, sabemos que
  - $M \equiv \lambda x\,.\,N$
  - $M'_1 \equiv \lambda x\,.\,N'_1$, con $N \Rightarrow N'_1$
  - $M'_2 \equiv \lambda x\,.\,N'_2$, con $N \Rightarrow N'_2$

  Por HI, sabemos que
  - existe N'' tal que $N'_1 \Rightarrow N''$ y $N'_2 \Rightarrow N''$

  Basta con tomar $M'' :\equiv \lambda x\,.\,N''$.

$$\begin{tikzcd}
 & \lambda x\,.\,N \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\
\lambda x\,.\,N'_1 \arrow[dr, Rightarrow, dotted] & & \lambda x\,.\,N'_2 \arrow[dl, Rightarrow, dotted] \\
 & \lambda x\,.\,N'' &
\end{tikzcd}$$

- (PApp)/(PApp). En este caso, sabemos que
  - $M \equiv NP$
  - $M'_1 \equiv N'_1 P'_1$, con $N \Rightarrow N'_1$ y $P \Rightarrow P'_1$
  - $M'_2 \equiv N'_2 P'_2$, con $N \Rightarrow N'_2$ y $P \Rightarrow P'_2$

  Por HI, sabemos que
  - existe N'' tal que $N'_1 \Rightarrow N''$ y $N'_2 \Rightarrow N''$
  - existe P'' tal que $P'_1 \Rightarrow P''$ y $P'_2 \Rightarrow P''$

  Basta con tomar $M'' :\equiv N'' P''$.

$$\begin{tikzcd}
 & NP \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\
N'_1 P'_1 \arrow[dr, Rightarrow, dotted] & & N'_2 P'_2 \arrow[dl, Rightarrow, dotted] \\
 & N'' P'' &
\end{tikzcd}$$

(...)

Reducción paralela

Demostración de la propiedad del diamante (fin).

- (PBeta)/(PApp). En este caso, sabemos que
  - $M \equiv (\lambda x\,.\,N)P$
  - $M'_1 \equiv N'_1[x := P'_1]$, con $N \Rightarrow N'_1$ y $P \Rightarrow P'_1$
  - $M'_2 \equiv R'_2 P'_2$, con $\lambda x\,.\,N \Rightarrow R'_2$ y $P \Rightarrow P'_2$

  Además, analizando las derivaciones posibles de $\lambda x\,.\,N \Rightarrow R'_2$, se deduce que $R'_2 \equiv \lambda x\,.\,N'_2$, con $N \Rightarrow N'_2$.

  Por HI, sabemos que
  - existe N'' tal que $N'_1 \Rightarrow N''$ y $N'_2 \Rightarrow N''$
  - existe P'' tal que $P'_1 \Rightarrow P''$ y $P'_2 \Rightarrow P''$

$$\begin{tikzcd}[column sep=tiny] & (\lambda x\,.\,N)P \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\ N'_1[x := P'_1] \arrow[dr, Rightarrow, dotted] & & (\lambda x\,.\,N'_2)P'_2 \arrow[dl, Rightarrow, dotted] \\ & N''[x := P''] & \end{tikzcd}$$

  Tomando $M'' :\equiv N''[x := P'']$, se concluye que
  - $M'_1 \equiv N'_1[x := P'_1] \Rightarrow N''[x := P''] \equiv M''$ por el lema de sustitución paralela
  - $M'_2 \equiv (\lambda x\,.\,N'_2)P'_2 \Rightarrow N''[x := P''] \equiv M''$ por (PBeta)

- (PApp)/(PBeta). Caso simétrico del caso anterior. $\square$

Confluencia de la β-reducción (1/2)

- Vimos que la reducción paralela $\Rightarrow$ es confluente
- Por otro lado, sabemos que $(\rightarrow_\beta) \subset (\Rightarrow) \subset (\xrightarrow{*}_\beta)$.
  Esto implica que: $(\xrightarrow{*}_\beta) \subset (\overset{*}{\Rightarrow}) \subset (\xrightarrow{*}_\beta)^* = (\xrightarrow{*}_\beta)$
- Por lo tanto, tenemos que $(\xrightarrow{*}_\beta) = (\overset{*}{\Rightarrow})$, y luego:

Teorema (Confluencia de la β-reducción)

La β-reducción es confluente:

Si $M \xrightarrow{*}_\beta M_1'$ y $M \xrightarrow{*}_\beta M_2'$,
entonces existe M'' tal que
$M_1' \xrightarrow{*}_\beta M''$ y $M_2' \xrightarrow{*}_\beta M''$

$$\begin{tikzcd}
 & M \arrow[dl, "*"'] \arrow[dr, "*"] & \\
M_1' \arrow[dr, dotted, "*"'] & & M_2' \arrow[dl, dotted, "*"] \\
 & M'' &
\end{tikzcd}$$

Confluencia de la β-reducción (2/2)

Corolario (Church-Rosser)

La β-reducción es Church-Rosser:

Si $M_1 \cong_\beta M_2$
entonces existe M' tal que
$M_1 \xrightarrow{*}_\beta M'$ y $M_2 \xrightarrow{*}_\beta M'$

$$\begin{tikzcd} M_1 \arrow[rr, "\sim"] \arrow[dr, dotted, "*"'] & & M_2 \arrow[dl, dotted, "*"] \\ & M' & \end{tikzcd}$$

Corolario (formas normales)

(1) La forma normal de un término, cuando existe, es única

(2) Dos formas normales son β-convertibles si y sólo si son iguales

Obs.: (2) = consistencia computacional del cálculo lambda:
la β-conversión nunca identifica dos formas normales distintas

$$x \not\cong_\beta y, \qquad \mathbf{I} \not\cong_\beta \mathbf{K}, \qquad \mathbf{K} \not\cong_\beta \Delta \qquad \text{(etc.)}$$

Plan

El problema de la extensionalidad (1/2)

- Dos términos M, M' son extensionalmente equivalentes cuando

(1) $$M\,N \;\cong_\beta\; M'\,N \qquad \text{(para todo } N\text{)}$$

Pregunta: ¿Son tales términos β-convertibles?

- Por sustitutividad, la condición (1) es equivalente a

(1') $$M\,x \;\cong_\beta\; M'\,x \qquad (x \text{ variable fresca})$$

- **Contraejemplo:** Tenemos que

$$(\lambda z\,.\,y\,z)\,x \;\to_\beta\; y\,x$$

mientras

$$\lambda z\,.\,y\,z \;\not\cong_\beta\; y$$

- ¿Cómo extender la relación de β-conversión de tal modo que dos términos extensionalmente equivalentes sean convertibles?

El problema de la extensionalidad (2/2)

- Sea $\cong \;\subseteq\; \Lambda \times \Lambda$ una relación de equivalencia compatible y sustitutiva que contiene la β-conversión. Se dice que la relación $\cong$ es extensional cuando cumple la regla

$$\frac{M\,x \cong M'\,x}{M \cong M'} \;(x \text{ fresca})$$

Proposición

Las dos condiciones son equivalentes:

(1) $\cong$ es extensional

(2) $\cong$ es tal que $\lambda x\,.\,M\,x \;\cong\; M$ para todos M, x con $x \notin FV(M)$

Demostración. (1) $\Rightarrow$ (2) Dados M, x, z tales que $x, z \notin FV(M)$, se observa que $(\lambda x\,.\,M\,x)\,z \to_\beta M\,z$, entonces $(\lambda x\,.\,M\,x)\,z \cong M\,z$, y luego $\lambda x\,.\,M\,x \cong M$ por (1).
(2) $\Rightarrow$ (1) Supongamos que $M\,x \cong M'\,x$, con $x \notin FV(M)$ y $x \notin FV(M')$. Entonces $\lambda x\,.\,M\,x \cong \lambda x\,.\,M'\,x$ por compatibilidad, y luego $M \cong M'$ por (2). $\square$

- Motiva la nueva regla: $\lambda x\,.\,M\,x \;\to_\eta\; M \qquad (x \notin FV(M))$

Definición de la η-reducción

Definición (η-reducción)

(1) La noción de η-reducción $\eta \subseteq \Lambda \times \Lambda$ está definida por la regla:

$$\frac{}{\lambda x\,.\,M\,x \;\; \eta \;\; M}\ \text{si } x \notin FV(M)$$

(2) La relación $\to_\eta$ de η-reducción en un paso está definida como la clausura contextual de la relación η

Propiedades básicas de la η-reducción

(1) La relación η es sustitutiva, y por lo tanto:

(2) La relación $\to_\eta$ es compatible y sustitutiva

Además:

(3) Si $M \to_\eta M'$, entonces $FV(M') = FV(M)$

Obs.: Variables libres nunca desaparecen (ni aparecen) durante la η-reducción

Propiedades de la η-reducción

Proposición (Normalización fuerte)

(1) Para todos $M \to_\eta M'$, tenemos que $|M| > |M'|$

Y por lo tanto:

(2) La relación $\to_\eta$ es fuertemente normalizante

Proposición (Propiedad del diamante)

La relación $\overset{=}{\to}_\eta$ cumple la propiedad del diamante:

Si $M \overset{=}{\to}_\eta M_1'$ y $M \overset{=}{\to}_\eta M_2'$,
entonces existe M'' tal que
$M_1' \overset{=}{\to}_\eta M''$ y $M_2' \overset{=}{\to}_\eta M''$

$$\begin{tikzcd} & M \arrow[dl, "="'] \arrow[dr, "="] & \\ M_1' \arrow[dr, dotted, "="'] & & M_2' \arrow[dl, dotted, "="] \\ & M'' & \end{tikzcd}$$

Demostración. Ejercicio.

Confluencia de la η-reducción

Teorema (Confluencia de la η-reducción)

La η-reducción es confluente:

Si $M \xrightarrow{*}_\eta M'_1$ y $M \xrightarrow{*}_\eta M'_2$,
entonces existe M'' tal que
$M'_1 \xrightarrow{*}_\eta M''$ y $M'_2 \xrightarrow{*}\ \eta M''$

$$
\begin{tikzcd}
 & M \arrow[dl, "*"'] \arrow[dr, "*"] & \\
M'_1 \arrow[dr, dotted, "*"'] & & M'_2 \arrow[dl, dotted, "*"] \\
 & M'' &
\end{tikzcd}
$$

Demostración. Sigue del resultado demostrado en el Práctico 2, Ejercicio 3 (2). $\square$

Corolario (Church-Rosser + unicidad de las formas η-normales)

(1) La forma η-normal de cualquier término es única
(2) La η-reducción es Church-Rosser
(3) Dos términos son η-convertibles sii tienen misma forma η-normal

Conmutation β/η

Proposición (Conmutación β/η)

La relación $\to_\eta$ conmuta con $\to_\beta$:

Si $M \xrightarrow{*}_\beta M_1'$ y $M \xrightarrow{*}_\eta M_2'$,
entonces existe M'' tal que
$M_1' \xrightarrow{*}_\beta M''$ y $M_2' \xrightarrow{*}_\eta M''$

$$\begin{tikzcd}
& M \arrow[dl, "*"', "\beta"] \arrow[dr, "*", "\eta"'] & \\
M_1' \arrow[dr, dotted, "*"', "\eta"] & & M_2' \arrow[dl, dotted, "*", "\beta"'] \\
& M'' &
\end{tikzcd}$$

Demostración. En primer lugar, se demuestra (Ejercicio) que las relaciones $\to_\beta$ y $\to_\eta$ tienen la siguiente propiedad de conmutación fuerte:

Si $M \xrightarrow{*}_\beta M_1'$ y $M \xrightarrow{*}_\eta M_2'$,
entonces existe M'' tal que
$M_1' \xrightarrow{*}_\beta M''$ y $M_2' \xrightarrow{*}_\eta M''$

$$\begin{tikzcd}
& M \arrow[dl, "\beta"] \arrow[dr, "\eta"'] & \\
M_1' \arrow[dr, dotted, "*"', "\eta"] & & M_2' \arrow[dl, dotted, "="', "\beta"] \\
& M'' &
\end{tikzcd}$$

Demostración: ejercicio.

Luego se concluye usando el resultado demostrado en el Práctico 2, Ejercicio 4 (3). □

Confluencia de la $\beta\eta$-reducción

Se define la $\beta\eta$-reducción por $\quad \to_{\beta\eta} \;:=\; \to_{(\beta\cup\eta)} \;=\; \to_\beta \cup \to_\eta$

Teorema (Confluencia de la $\beta\eta$-reducción)

La $\beta\eta$-reducción es confluente:

Si $M \xrightarrow{*}_{\beta\eta} M'_1$ y $M \xrightarrow{*}_{\beta\eta} M'_2$,
entonces existe M'' tal que
$M'_1 \xrightarrow{*}_{\beta\eta} M''$ y $M'_2 \xrightarrow{*} \; \beta\eta M''$

$$\begin{tikzcd}
& M \arrow[dl, "*"'] \arrow[dr, "*"] & \\
M'_1 \arrow[dr, dotted, "*"'] & & M'_2 \arrow[dl, dotted, "*"] \\
& M'' &
\end{tikzcd}$$

Demostración. Sigue del resultado demostrado en el Práctico 2, Ejercicio 4 (2). □

Corolario (Church-Rosser + unicidad de las formas $\beta\eta$-normales)

(1) La $\beta\eta$-reducción es Church-Rosser

(2) La forma $\beta\eta$-normal de un término, cuando existe, es única

Postergación de la regla η

Proposición (Postergación de la regla η)

Si $M \xrightarrow{*}_{\beta\eta} M''$, entonces existe M' tal que $M \xrightarrow{*}_{\beta} M'$ y $M' \xrightarrow{*}_{\eta} M''$

$$\begin{tikzcd}
M \arrow[rr, "*", "\beta\eta"'] \arrow[dr, dotted, "\beta", "*"'] & & M'' \\
& M' \arrow[ur, dotted, "\eta", "*"'] &
\end{tikzcd}$$

Demostración. Sigue de los siguientes diagramas (ejercicio):

$$\begin{tikzcd}
& M_1' \arrow[dr, Rightarrow] & \\
M \arrow[ur, "\eta"'] \arrow[dr, Rightarrow, dotted] & & M'' \\
& M_2' \arrow[ur, dotted, "\eta", "*"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& M_1' \arrow[dr, Rightarrow] & \\
M \arrow[ur, "*", "\eta"'] \arrow[dr, Rightarrow, dotted] & & M'' \\
& M_2' \arrow[ur, dotted, "\eta", "*"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& M_1' \arrow[dr, "*", "\beta"'] & \\
M \arrow[ur, "*", "\eta"'] \arrow[dr, dotted, "\beta", "*"'] & & M'' \\
& M_2' \arrow[ur, dotted, "\eta", "*"'] &
\end{tikzcd}$$

(donde $\Rightarrow$ indica la β-reducción paralela, y observando que $\overset{*}{\Rightarrow} \; = \; \xrightarrow{*}_{\beta}$) □

Plan

¿Cómo evitar los problemas de α-conversión?

- **Idea:** Remplazar cada variable ligada por un entero natural $i \geq 1$ (índice de De Bruijn) que indica la posición del correspondiente λ:

$$1 = \text{último } \lambda, \quad 2 = \text{penúltimo } \lambda, \quad \text{etc.}$$

- **Ejemplos:**

$$\begin{array}{rcl}
\lambda x \,.\, x & \leadsto & \lambda.1 \\
\lambda y \,.\, y & \leadsto & \lambda.1 \\
\lambda x \,.\, \lambda y \,.\, x & \leadsto & \lambda.\lambda.2 \\
\lambda x \,.\, \lambda y \,.\, y & \leadsto & \lambda.\lambda.1 \\
\lambda x \,.\, \lambda y \,.\, \lambda z \,.\, x\, z\, (y\, z) & \leadsto & \lambda.\lambda.\lambda.3\,1\,(2\,1) \\
\lambda x \,.\, x\, (\lambda y \,.\, y\, x) & \leadsto & \lambda.1\,(\lambda.1\,2)
\end{array}$$

- **Obs.:** Siguiendo [Barendregt 1984], usamos aquí índices $n \geq 1$. Sin embargo en muchas implementaciones concretas, se usan índices $n \geq 0$

Términos con índices de De Bruijn (1/2)

Definición (Términos lambda con índices de De Bruijn)

Términos con índices $\quad M, N \quad ::= \quad i \quad | \quad \lambda.M \quad | \quad MN \qquad (i \geq 1)$

- El conjunto $FI(M)$ de los índices libres de un término M con índices de De Bruijn está definido por:

$$
\begin{array}{rcl}
FI(i) & := & \{i\} \\
FI(\lambda.M) & := & (FI(M) \setminus \{1\}) - 1 \\
 & := & \{i \geq 1 \ : \ i + 1 \in FI(M)\} \\
FI(MN) & := & FI(M) \cup FI(N)
\end{array}
$$

- **Intuición:** Cada índice libre $i \in FV(M)$ representa la i-ésima variable x_i de algún "contexto de variables" $\{x_1, \ldots, x_n\}$

Términos con índices de De Bruijn (2/2)

Definición (Traducción $M \mapsto (M)^{\mathrm{DB}}_{x_1,\ldots,x_n}$)

Cada término lambda M con variables libres $x_1, \ldots, x_n$ se traduce en un término $(M)^{\mathrm{DB}}_{x_1,\ldots,x_n}$ con índices de De Bruijn libres $1..n$ definido por:

$$
\begin{array}{rcl}
(x_i)^{\mathrm{DB}}_{x_1,\ldots,x_n} & :\equiv & i \\
(\lambda y\,.\,M)^{\mathrm{DB}}_{x_1,\ldots,x_n} & :\equiv & \lambda.(M)^{\mathrm{DB}}_{y,x_1,\ldots,x_n} \\
(MN)^{\mathrm{DB}}_{x_1,\ldots,x_n} & :\equiv & (M)^{\mathrm{DB}}_{x_1,\ldots,x_n}(N)^{\mathrm{DB}}_{x_1,\ldots,x_n}
\end{array}
$$

Proposición (Criterio de α-equivalencia)

Para todos $M, M' \in \Lambda$ tales que $FV(MM') \subseteq \{x_1, \ldots, x_n\}$, tenemos que:

$$M \equiv_\alpha M' \qquad \text{sii} \qquad (M)^{\mathrm{DB}}_{x_1,\ldots,x_n} \equiv (M')^{\mathrm{DB}}_{x_1,\ldots,x_n}$$

Demostración. Ejercicio.

Lifting y sustitución

Definición (Lifting)

Se define la operación de lifting $\uparrow^p_k M \quad (k, p \geq 1)$ por:

$$
\begin{aligned}
\uparrow^p_k i \quad &:\equiv \quad \begin{cases} i & \text{si } i < k \\ i + p - 1 & \text{si } i \geq k \end{cases} \\
\uparrow^p_k (\lambda.M) \quad &:\equiv \quad \lambda.\uparrow^p_{k+1} M \\
\uparrow^p_k (MN) \quad &:\equiv \quad \uparrow^p_k M \uparrow^p_k N
\end{aligned}
$$

Obs.: Tenemos que $\uparrow^1_k M \equiv M \quad (k \geq 1)$

Lema

Para todo $M \in \Lambda$ tal que $FV(M) \subseteq \{x_1, \ldots, x_n\}$, tenemos que:

$$(M)^{\mathrm{DB}}_{x_1,\ldots,x_{k-1},y_1,\ldots,y_{p-1},x_k,\ldots,x_n} \equiv \uparrow^p_k (M)^{\mathrm{DB}}_{x_1,\ldots,x_n}$$

Demostración. Ejercicio.

Lifting y sustitución (2/3)

Definición (Sustitución con índices de De Bruijn)

Se define la operación de sustitución $M[p := P]$ $(p \geq 1)$ por:

$$
\begin{aligned}
i[p := P] \quad &:\equiv \quad \begin{cases} i & \text{si } i < p \\ \uparrow_1^p P & \text{si } i = p \\ i - 1 & \text{si } i > p \end{cases} \\
(\lambda.M)[p := P] \quad &:\equiv \quad \lambda.M[p+1 := P] \\
(MN)[p := P] \quad &:\equiv \quad M[p := P]N[p := P]
\end{aligned}
$$

Lema

Para todos $M, P \in \Lambda$ tales que $FV(M) \subseteq \{x_1, \ldots, x_p, y_1, \ldots, y_n\}$ y $FV(P) \subseteq \{y_1, \ldots, y_n\}$, tenemos que:

$$(M[x_p := P])^{\text{DB}}_{x_1,\ldots,x_{p-1},y_1,\ldots,y_n} \equiv (M)^{\text{DB}}_{x_1,\ldots,x_p,y_1,\ldots,y_n}[p := (P)^{\text{DB}}_{y_1,\ldots,y_n}]$$

Demostración. Ejercicio.

Lifting y sustitución (3/3)

Proposición (Propiedades del lifting y de la sustitución)

Para todos términos M, N, P con índices de De Bruijn,
y para todos $n, p, j, k \geq 1$:

(1) $\uparrow^1_k M \;\equiv\; M$

(2) $\uparrow^p_j(\uparrow^n_k M) \;\equiv\; \uparrow^{p+n-1}_k M$ (si $k \leq j < k+n$)

(3) $\uparrow^p_j(\uparrow^n_k M) \;\equiv\; \uparrow^n_{k+p-1}(\uparrow^p_j M)$ (si $j \leq k$)

(4) $(\uparrow^{n+1}_k M)[p := N] \;\equiv\; \uparrow^n_k M$ (si $k \leq p \leq k+n-1$)

(5) $\uparrow^n_k(M[p := N]) \;\equiv\; (\uparrow^n_k M)[p+n-1 := N]$ (si $k \leq p$)

(6) $\uparrow^n_{k+p-1}(M[p := N]) \;\equiv\; (\uparrow^n_{k+p} M)[p+n := \uparrow^n_k N]$

(7) $(M[p := N])[p+n-1 := P] \;\equiv\; M[p+n := P][p := N[n := P]]$

Obs.: (7) es el lema de sustitución para los términos con índices de De Bruijn

Demostración. Ejercicio.

Cómo utilizar los términos con índices de De Bruijn

Con las notaciones anteriores, se define la β-reducción por:

$$(\lambda.M)N \quad \to_\beta \quad M[1 := N]$$

$+$ clausura contextual

En una implementación concreta:

- Se pueden usar índices $i \geq 0$
- Se pueden usar nombres para las variables libres, reservando los índices para las variables ligadas (convención de Coquand)
- Cada λ (o símbolo ligador) puede ser acompañado con una sugerencia de nombre de variable x (para desplegar el término)

Términos $\quad M, N \quad ::= \quad x \quad | \quad i \quad | \quad \lambda_x.M \quad | \quad MN$

Plan

Computabilidad

¿Cómo programar en el cálculo lambda?

- Booleanos
- Pares y *n*-uplas
- Enteros naturales
- Puntos fijos (i.e. recursión no acotada)

Expresividad del cálculo lambda

- ¿Cuáles son las funciones definibles en el cálculo lambda?

Problema de la decisión (*„Entscheidungsproblem“*)

Representación de los booleanos

- Se definen

$$\begin{aligned} \texttt{true} &:\equiv \lambda xy\,.\,x \\ \texttt{false} &:\equiv \lambda xy\,.\,y \\ \texttt{if} &:\equiv \lambda bxy\,.\,b\,x\,y \end{aligned}$$

- Tenemos que:

$$\begin{aligned} \texttt{if true } N\ N' &\xrightarrow{*}_\beta N \\ \texttt{if false } N\ N' &\xrightarrow{*}_\beta N' \end{aligned}$$

- A partir de las construcciones anteriores, se pueden definir las operaciones booleanas usuales:

$$\begin{aligned} \texttt{not} &:\equiv \lambda x\,.\,\texttt{if}\ x\ \texttt{false}\ \texttt{true} \\ \texttt{and} &:\equiv \lambda xy\,.\,\texttt{if}\ x\ y\ \texttt{false} \\ \texttt{or} &:\equiv \lambda xy\,.\,\texttt{if}\ x\ \texttt{true}\ y \end{aligned}$$

Representación de los pares

- Se definen:

$$
\begin{array}{rcll}
\langle M_1, M_2 \rangle & :\equiv & \lambda z\,.\,z\,M_1\,M_2 & (z \text{ fresca}) \\
\texttt{pair} & :\equiv & \lambda xyz\,.\,z\,x\,y & \\
\texttt{fst} & :\equiv & \lambda z\,.\,z\,(\lambda xy\,.\,x) & \\
\texttt{snd} & :\equiv & \lambda z\,.\,z\,(\lambda xy\,.\,y) &
\end{array}
$$

- Tenemos que:

$$
\begin{array}{rcl}
\texttt{pair}\,M_1\,M_2 & \stackrel{*}{\rightarrow}_\beta & \langle M_1, M_2 \rangle \\
\texttt{fst}\,\langle M_1, M_2 \rangle & \stackrel{*}{\rightarrow}_\beta & M_1 \\
\texttt{snd}\,\langle M_1, M_2 \rangle & \stackrel{*}{\rightarrow}_\beta & M_2
\end{array}
$$

Representación de las n-uplas

- Más generalmente ($n \geq 0$) se definen:

$$\begin{array}{rcll} \langle M_1, \ldots, M_n \rangle & :\equiv & \lambda z\,.\,z\,M_1 \cdots M_n & (z \text{ fresca}) \\ \texttt{upla}_n & :\equiv & \lambda x_1 \cdots x_n z\,.\,z\,x_1 \cdots x_n & \\ \texttt{proj}_i^n & :\equiv & \lambda z\,.\,z\,(\lambda x_1 \cdots x_n\,.\,x_i) & (1 \leq i \leq n) \end{array}$$

- Tenemos que:

$$\begin{array}{rcl} \texttt{upla}_n\,M_1 \cdots M_n & \xrightarrow{*}_\beta & \langle M_1, \ldots, M_n \rangle \\ \texttt{proj}_i^n\,\langle M_1, \ldots, M_n \rangle & \xrightarrow{*}_\beta & M_i \end{array}$$

Representación de los enteros naturales (1/2)

- Se definen los enteros de Church por:

$$\begin{aligned} \overline{n} \quad &:\equiv \quad \lambda f\,.\,\lambda x\,.\,\underbrace{f(\cdots(f}_{n}\ x)\cdots) \\ &\equiv \quad \lambda f\,.\,\underbrace{f\circ\cdots\circ f}_{n} \qquad\qquad (n\in\mathbb{N}) \end{aligned}$$

- **Intuición:** Entero de Church $\overline{n}$
  $=$ operador que permite iterar n veces una función:

$$\overline{n}\ \ F\ M \quad \xrightarrow{*}_{\beta} \quad \underbrace{F\,(\cdots(F}_{n}\ \ M)\cdots)$$

  $=$ recursión acotada
  $=$ bucle «`for`»

Representación de los enteros naturales (2/2)

- Se definen:

$$
\begin{aligned}
\texttt{succ} &:\equiv \lambda nfx\,.\,f\,(n\,f\,x)\\
\texttt{null} &:\equiv \lambda n\,.\,n\,(\lambda x\,.\,\texttt{false})\,\texttt{true}\\
\texttt{plus} &:\equiv \lambda nmfx\,.\,n\,f\,(m\,f\,x)\\
\texttt{mult} &:\equiv \lambda nm\,.\,n\,(\texttt{plus}\,m)\,\overline{0}\\
\texttt{pow} &:\equiv \lambda pn\,.\,n\,(\texttt{mult}\,p)\,\overline{1}\\
\texttt{pred} &:\equiv \lambda n\,.\,\texttt{fst}\,(n\,(\lambda z\,.\,z\,(\lambda xy\,.\,\langle y,\,\texttt{succ}\,y\rangle))\,\langle\overline{0},\,\overline{0}\rangle)\\
\texttt{minus} &:\equiv \lambda nm\,.\,m\,\texttt{pred}\,n
\end{aligned}
$$

- Tenemos que:

$$
\begin{aligned}
\texttt{succ}\,\overline{n} &\xrightarrow{*}_\beta \overline{n+1}\\
\texttt{null}\,\overline{n} &\xrightarrow{*}_\beta \begin{cases}\texttt{true} & \text{si } n=0\\ \texttt{false} & \text{si } n\geq 1\end{cases}\\
\texttt{plus}\,\overline{n}\,\overline{m} &\xrightarrow{*}_\beta \overline{n+m} \qquad\qquad \text{(etc.)}
\end{aligned}
$$

Puntos fijos (1/5)

- El combinador de punto fijo de Church es el término:

$$\mathbf{Y} \;:\equiv\; \lambda f\,.\,(\lambda x\,.\,f\,(x\,x))\,(\lambda x\,.\,f\,(x\,x))$$

Proposición (Combinador de punto fijo)

Para todo término M, tenemos que: $\mathbf{Y}\,M \;\cong_\beta\; M\,(\mathbf{Y}\,M)$

Demostración. Tenemos que

$$\begin{array}{rcll} \mathbf{Y}\,M & \equiv & (\lambda f\,.\,(\lambda x\,.\,f\,(x\,x))\,(\lambda x\,.\,f\,(x\,x)))\,M & \\ & \to_\beta & (\lambda x\,.\,M\,(x\,x))\,(\lambda x\,.\,M\,(x\,x)) & \\ & \to_\beta & M\,((\lambda x\,.\,M\,(x\,x))\,(\lambda x\,.\,M\,(x\,x))) \quad {}_\beta\!\leftarrow & M\,(\mathbf{Y}\,M) \end{array}$$

$\square$

Corolario (Existencia de puntos fijos)

Todo término M tiene un punto fijo F_M, tal que $M\,F_M \;\cong_\beta\; F_M$

Demostración. Basta con tomar $F_M \;:\equiv\; \mathbf{Y}\,M$. $\square$

Puntos fijos (2/5)

- La existencia de puntos fijos explica por qué el cálculo lambda es inconsistente como sistema lógico (al menos de modo ingenuo)

  En efecto, en tal sistema se podría formar la fórmula

$$\phi \quad :\equiv \quad \mathbf{Y}\,(\neg) \quad \cong_\beta \quad \neg\,\phi$$

- Se observa que $\mathbf{Y}\,M$ nunca es fuertemente normalizante:

$$\begin{array}{lll} \mathbf{Y}\,M & \to_\beta & (\lambda x\,.\,M\,(x\,x))\,(\lambda x\,.\,M\,(x\,x)) \\ & \to_\beta & M\,((\lambda x\,.\,M\,(x\,x))\,(\lambda x\,.\,M\,(x\,x))) \\ & \to_\beta & M\,\big(M\,((\lambda x\,.\,M\,(x\,x))\,(\lambda x\,.\,M\,(x\,x)))\big) \\ & \vdots & \\ & \to_\beta & M\,(M\,(M\,(M\,(M\,(M\,(M\,\cdots)))))) \end{array}$$

  En particular: $\mathbf{Y\,I} \xrightarrow{3}_\beta \Delta\,\Delta \to_\beta \Delta\,\Delta \to_\beta \cdots \qquad (\Delta :\equiv \lambda x\,.\,xx)$

- Pero a veces, $\mathbf{Y}\,M$ tiene forma normal, por ejemplo:

$$\mathbf{Y}\,(\mathbf{K\,I}) \quad \cong_\beta \quad \mathbf{K\,I}\,\big(\mathbf{Y}\,(\mathbf{K\,I})\big) \quad \xrightarrow{*}_\beta \quad \mathbf{I}$$

Puntos fijos (3/5)

- Puntos fijos son muy útiles en programación, por ejemplo:

$$\begin{array}{rcl} \mathtt{fact} & :\equiv & \mathbf{Y}\,M, \qquad \text{con} \\ M & :\equiv & \lambda fx\,.\,\mathtt{if}\ (\mathtt{null}\,x)\ \overline{1}\ (\mathtt{mult}\ x\ (f\ (\mathtt{pred}\ x))) \end{array}$$

- Tenemos que $\mathtt{fact} \cong_\beta M\ \mathtt{fact}$, y por lo tanto:

$$\begin{array}{rcl} \mathtt{fact}\ \overline{0} & \cong_\beta & M\ \mathtt{fact}\ \overline{0} \\ & \cong_\beta & \mathtt{if}\ (\mathtt{null}\,\overline{0})\ \overline{1}\ (\mathtt{mult}\ \overline{0}\ (\mathtt{fact}\ (\mathtt{pred}\ \overline{0}))) \quad \cong_\beta \quad \overline{1} \\ \mathtt{fact}\ (\overline{n+1}) & \cong_\beta & M\ \mathtt{fact}\ (\overline{n+1}) \\ & \cong_\beta & \mathtt{if}\ (\mathtt{null}\,(\overline{n+1}))\ \overline{1}\ (\mathtt{mult}\ (\overline{n+1})\ (\mathtt{fact}\ (\mathtt{pred}\ (\overline{n+1})))) \\ & \cong_\beta & \mathtt{mult}\ (\overline{n+1})\ (\mathtt{fact}\ \overline{n}) \end{array}$$

Proposición

Para todo $n \in \mathbb{N}$: $\mathtt{fact}\ \overline{n} \xrightarrow{*}_\beta \overline{n!}$

- **Intuición:** puntos fijos = recursión no acotada
  = bucle «while»

Puntos fijos

- **Observación:** El combinador de punto fijo de Church tiene un pequeño defecto sintáctico:

$$\begin{array}{lccc} & \mathbf{Y}\,M & \cong_\beta & M\,(\mathbf{Y}\,M), \\ \text{pero} & & \not\xrightarrow{*}_\beta & M\,(\mathbf{Y}\,M) \end{array}$$

- Se puede corregir usando el combinador de punto fijo de Turing:

$$\mathbf{T} \;:\equiv\; (\lambda yf\,.\,f\,(y\,y\,f))(\lambda yf\,.\,f\,(y\,y\,f))$$

Proposición (Combinador de punto fijo)

Para todo término M, tenemos que: $\mathbf{T}\,M \xrightarrow{*}_\beta M\,(\mathbf{T}\,M)$

Demostración. Ejercicio

Puntos fijos (5/5)

- **Otro ejemplo de combinador de punto fijo:**

$$\mathbf{U} \;:\equiv\; \lambda abc \ldots xyz\,.\,z\,(e\,s\,e\;c\,o\,m\,b\,i\,n\,a\,d\,o\,r\;e\,s\;u\,n\,a\;e\,s\,t\,u\,p\,i\,d\,e\,z)$$

$$\mathbf{V} \;:\equiv\; \underbrace{\mathbf{U}\;\cdots\;\mathbf{U}}_{26}$$

- **Ejercicio:** Verificar que $\mathbf{V}\,M \;\xrightarrow{*}_{\beta}\; M\,(\mathbf{V}\,M)$ (para todo M)

λ-definibilidad

- **Notaciones:** Dado $k \geq 1$, se escriben:

$$\mathbb{N}^k \rightharpoonup \mathbb{N} \quad = \quad \text{conjunto de la funciones parciales de } \mathbb{N}^k \text{ a } \mathbb{N}$$
$$\mathbb{N}^k \to \mathbb{N} \quad = \quad \text{conjunto de la funciones totales de } \mathbb{N}^k \text{ a } \mathbb{N}$$

Tenemos que $(\mathbb{N}^k \to \mathbb{N}) \subseteq (\mathbb{N}^k \rightharpoonup \mathbb{N})$

Definición (Funciones λ-definibles)

Se dice que una función parcial $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ es λ-definible cuando existe un término lambda M tal que para todos $n_1, \ldots, n_k \in \mathbb{N}$:

$$\begin{cases} M\,\overline{n}_1 \cdots \overline{n}_k \xrightarrow{*}_\beta \overline{f(n_1, \ldots, n_k)} & \text{si } (n_1, \ldots, n_k) \in \mathrm{dom}(f) \\ M\,\overline{n}_1 \ldots \overline{n}_k \text{ diverge} & \text{si } (n_1, \ldots, n_k) \notin \mathrm{dom}(f) \end{cases}$$

En este caso, se dice que M es una λ-definición de f

- **Pregunta:** ¿Cuáles son las funciones λ-definibles?

Funciones iniciales (recordatorio)

Se llaman funciones iniciales a las siguientes funciones:

- La **función nula** $z : \mathbb{N} \to \mathbb{N}$, definida por

$$z(n) := 0$$

(para todo $n \in \mathbb{N}$)

- La **función sucesor** $s : \mathbb{N} \to \mathbb{N}$, definida por

$$s(n) := n + 1$$

(para todo $n \in \mathbb{N}$)

- Las **proyecciones** $\pi_i^k : \mathbb{N}^k \to \mathbb{N}$ $(k \geq i \geq 1)$, definidas por

$$\pi_i^k(n_1, \ldots, n_k) := n_i$$

(para todo $(n_1, \ldots, n_k) \in \mathbb{N}^k$)

Esquemas de composición y de recursión primitiva

- **Esquema de composición** A partir de $f_1, \ldots, f_p : \mathbb{N}^k \rightharpoonup \mathbb{N}$ y $g : \mathbb{N}^p \rightharpoonup \mathbb{N}$, definir la función $h : \mathbb{N}^k \rightharpoonup \mathbb{N}$ por

$$h(n_1, \ldots, n_k) := g(f_1(n_1, \ldots, n_k), \ldots, f_p(n_1, \ldots, n_k))$$

(para todo $(n_1, \ldots, n_k) \in \mathbb{N}^k$ donde el lado derecho está definido)

Se nota $h = g \circ (f_1, \ldots, f_p)$

- **Esquema de recursión primitiva** A partir de $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ y $g : \mathbb{N}^{k+2} \rightharpoonup \mathbb{N}$, definir la función $h : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ por:

$$\begin{aligned} h(0, n_1, \ldots, n_k) &:= f(n_1, \ldots, n_k) \\ h(n+1, n_1, \ldots, n_k) &:= g(n, h(n, n_1, \ldots, n_k), n_1, \ldots, n_k) \end{aligned}$$

(para todos $n \in \mathbb{N}$ y $(n_1, \ldots, n_k) \in \mathbb{N}^k$ donde el lado derecho está definido)

Se nota $h = \mathbf{rec}(f, g)$

Esquema de minimización

- **Esquema de minimización** A partir de $f : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$, definir la función $h : \mathbb{N}^k \rightharpoonup \mathbb{N}$ por

$$
\begin{array}{rcl}
h(n_1, \ldots, n_k) & := & \mu n \, . \, f(n, n_1, \ldots, n_k) > 0 \\
& = & \text{el único } n \in \mathbb{N} \text{ (cuando existe) tal que} \\
& & (\forall m < n) \, f(m, n_1, \ldots, n_k) > 0 \quad \text{y} \\
& & f(n, n_1, \ldots, n_k) = 0
\end{array}
$$

(para todo $(n_1, \ldots, n_k) \in \mathbb{N}^k$ donde el lado derecho está definido)

Se nota $h = \boldsymbol{\mu}(f)$

Obs.: El valor $h(n_1, \ldots, n_k)$ no está definido cuando:

- o bien existe $n \in \mathbb{N}$ tal que $f(m, n_1, \ldots, n_k) > 0$ para todo $m < n$ mientras $f(n, n_1, \ldots, n_k)$ no está definido
- o bien $f(m, n_1, \ldots, n_k) > 0$ para todo $m \in \mathbb{N}$

En particular: $h = \boldsymbol{\mu}(f)$ puede ser parcial aunque f sea total

Funciones recursivas generales (1/2)

Definición (Funciones recursivas)

El conjunto de las funciones recursivas es el mínimo conjunto $\subseteq \bigcup_{k\geq 1}(\mathbb{N}^k \rightharpoonup \mathbb{N})$ que contiene todas las funciones iniciales

$$\boldsymbol{z}, \qquad \boldsymbol{s}, \qquad \boldsymbol{\pi}_i^k \quad (k \geq i \geq 1)$$

y está cerrado por composición, recursión primitiva y minimización:

$$(f_1, \ldots, f_p, g) \mapsto g \circ (f_1, \ldots, f_p) \qquad (f, g) \mapsto \mathbf{rec}(f, g) \qquad f \mapsto \boldsymbol{\mu}(f)$$

Intuiciones:

Esquema de recursión primitiva	=	recursión acotada	
	=	bucle «`for`»	(Pascal)
Esquema de minimización	=	recursión no acotada	
	=	bucle «`while`»	(Pascal, C, etc.)

Funciones recursivas generales (2/2)

Observaciones:

- Las funciones recursivas generales (que pueden ser parciales) contienen todas las funciones recursivas primitivas (todas totales)
- Existen funciones recursivas totales que no son recursivas primitivas, por ejemplo la función de Ackermann
- Funciones parciales sólo pueden ser construidas por minimización (= único esquema que puede introducir la parcialidad)
- Dada $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ recursiva, la condición $(n_1, \ldots, n_k) \notin \mathrm{dom}(f)$ expresa que el cálculo de $f(n_1, \ldots, n_k)$ no termina (= diverge)
- Asimetría fundamental de la computación (y de la lógica):

terminación	$\#$	no terminación
(computación finita)		(computación infinita)

- Para saber que una función recursiva es total, en general se necesita una prueba... ¿pero en cuál sistema formal?

Todos los caminos conducen a ~~Roma~~ la computabilidad

Teorema [Church-Turing-Kleene]

Para toda función parcial $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$, las siguientes condiciones son equivalentes:

1. f es recursiva
2. f es λ-definible
3. f es computable por máquina de Turing

$\Big\}$ f es computable

Y de modo equivalente: f es programable en Fortran, Pascal, C, C++, Lisp, Java, Python, Ruby, OCaml, Haskell, etc.[1]

Tesis de Church-Turing

Toda función que puede ser "calculada por un algoritmo" (noción intuitiva)
es una función computable (noción formal)

[1] Con memoria y tiempo ilimitados

Otro teorema del punto fijo

Ahora se supone dada una codificación efectiva[2] $M \mapsto \ulcorner M \urcorner$ de los términos lambda por los enteros naturales: $\ulcorner M \urcorner \in \mathbb{N}$

Teorema (Punto fijo a través de la codificación)

Para todo término M, existe un término F_M tal que $M\,\overline{\ulcorner F_M \urcorner} \cong_\beta F_M$

Demostración. Como la codificación es efectiva, existen términos $\mathbf{A}$ y $\mathbf{N}$ tales que:

- $\mathbf{A}\,\overline{\ulcorner M \urcorner}\,\overline{\ulcorner N \urcorner} \cong_\beta \overline{\ulcorner M\,N \urcorner}$ (para todos $M, N \in \Lambda$)
- $\mathbf{N}\,\overline{n} \cong_\beta \overline{\ulcorner \lambda fx\,.\,\underbrace{f\,(\cdots(f}_{n}\;x)\cdots) \urcorner} \equiv \overline{\ulcorner \overline{n} \urcorner}$ (para todo $n \in \mathbb{N}$)

Sea $F_M :\equiv W_M\,\overline{\ulcorner W_M \urcorner}$, con $W_M :\equiv \lambda x\,.\,M\,(\mathbf{A}\,x\,(\mathbf{N}\,x))$. Se observa que:

$$\begin{array}{rcl} F_M & \equiv & W_M\,\overline{\ulcorner W_M \urcorner} \\ & \cong_\beta & M\,(\mathbf{A}\,\overline{\ulcorner W_M \urcorner}\,(\mathbf{N}\,\overline{\ulcorner W_M \urcorner})) \\ & \cong_\beta & M\,\overline{\ulcorner W_M\,\overline{\ulcorner W_M \urcorner} \urcorner} \;\equiv\; M\,\overline{\ulcorner F_M \urcorner} \end{array}$$

$\square$

[2] Es decir: todas las operaciones sintácticas sobre los términos (abstracción, aplicación, sustitución, etc.) son calculables a través de la codificación $M \mapsto \ulcorner M \urcorner$

El problema de la parada

Teorema (Problema de la parada)

No existe ningún término lambda H que decida si un término tiene forma normal o no, es decir tal que

$$H\,\overline{\ulcorner M \urcorner} \xrightarrow{*}_{\beta} \begin{cases} \texttt{true} & \text{si } M \text{ tiene forma normal} \\ \texttt{false} & \text{si no} \end{cases}$$

para todo término M

Demostración. Supongamos que H existe. Se define $G :\equiv \lambda x\,.\,\texttt{if}\,(H\,x)\,(\Delta\,\Delta)\,\mathbf{I}$ y se considera un punto fijo $F \cong_\beta G\,\overline{\ulcorner F \urcorner} \cong_\beta \texttt{if}\,(H\,\overline{\ulcorner F \urcorner})\,(\Delta\,\Delta)\,\mathbf{I}$.

- Si F tiene forma normal, entonces $F \cong_\beta \texttt{if true}\,(\Delta\,\Delta)\,\mathbf{I} \cong_\beta \Delta\,\Delta$. Entonces F no tiene forma normal: contradicción.
- Si F no tiene forma normal, entonces $F \cong_\beta \texttt{if false}\,(\Delta\,\Delta)\,\mathbf{I} \cong_\beta \mathbf{I}$. Entonces F tiene forma normal: contradicción.

Ambos casos son absurdos, y por lo tanto, el término H no existe. $\square$

El problema de la decisión („*Entscheidungsproblem*“)

Problema de la decisión [Hilbert & Ackermann, 1928]

¿Existe un "método efectivo" que permita decidir si una fórmula ϕ de la lógica de primer orden es universalmente válida ($\models \phi$) o no?

Antes de todo, se necesita definir lo que es un "método efectivo"

- En su artículo de 1936, Church:
  1. Introduce el cálculo lambda, con la tesis que éste constituye una modelación adecuada de la noción intuitiva de "método efectivo"
  2. Muestra que el problema de la parada es indecidible
  3. Muestra cómo asociar (de modo efectivo) a cada término lambda M una fórmula ϕ_M (en un lenguaje de primer orden adecuado) tal que

     $$M \text{ tiene forma normal} \qquad \text{sii} \qquad \models \phi_M$$

  4. El punto anterior constituye una reducción del problema de la parada al problema de la decisión, lo que implica que éste es indecidible.