

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**
⇒ 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:

$$\lambda x . 3x + 1 \quad = \quad (x \mapsto 3x + 1)$$

- Un mecanismo para **aplicar una función**:

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

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

$$(\lambda x . 3x + 1)(4) \rightarrow_{\beta} 3(4) + 1 \rightarrow 12 + 1 \rightarrow 13$$

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

 $(a_1, \dots, \hat{a}_i, \dots, a_n)$

la tupla (a_1, \dots, a_n) sin a_i

 $\hat{x}. 3x + 1 \geq 0$

$= \{x : 3x + 1 \geq 0\}$ (Russell-Whitehead)

 $\wedge x. 3x + 1, \quad \lambda x. 3x + 1$

$= (x \mapsto 3x + 1)$ (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

- 1 Introducción
- 2 Sintaxis
- 3 β -Reducción
- 4 η -Reducción
- 5 Índices de De Bruijn
- 6 Computabilidad

Plan

1 Introducción

2 **Sintaxis**

3 β -Reducción

4 η -Reducción

5 Índices de De Bruijn

6 Computabilidad

Sintaxis del cálculo lambda

Definición (Términos lambda)

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

Se escribe Λ al conjunto de todos los términos lambda

- **Abreviaturas:**

$$\lambda x_1 x_2 \cdots x_n. M \quad \equiv \quad \lambda x_1. \lambda x_2. \cdots \lambda x_n. M$$

$$MN_1 N_2 \cdots N_n \quad \equiv \quad (\cdots ((MN_1)N_2) \cdots)M_n$$

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

$$FV(x) = \{x\}$$

$$BV(x) = \emptyset$$

$$FV(\lambda x. M) = FV(M) \setminus \{x\}$$

$$BV(\lambda x. M) = BV(M) \cup \{x\}$$

$$FV(MN) = FV(M) \cup FV(N)$$

$$BV(MN) = BV(M) \cup BV(N)$$

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:

$$\begin{aligned}
 (\lambda x . x y)[y := x] &\neq \lambda x . x x \\
 &\equiv_\alpha \lambda z . z x \\
 &\equiv_\alpha \lambda y . y x
 \end{aligned}$$

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 $y \neq x$)
- $(\lambda x . M)\langle x := N \rangle \equiv \lambda x . M$
- $(\lambda y . M)\langle x := N \rangle \equiv \lambda y . M\langle x := N \rangle$ 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 $BV(M) \cap FV(N) = \emptyset$, entonces $M\langle x := N \rangle$ está definido

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

Lema de sustitución

Para todos $M, N, P, x \neq y$ 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:

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

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

(z fresca = $z \neq x_1$, $z \neq x_2$ 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 $M, N, P, x \neq y$ 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 $y \neq x$)
- $(\lambda x. M)[x := N] \equiv \lambda x. M$
- $(\lambda y. M)[x := N] \equiv \lambda y. M[x := N]$ 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

1 Introducción

2 Sintaxis

3 β -Reducción

4 η -Reducción

5 Índices de De Bruijn

6 Computabilidad

Contextos con un agujero

(1/2)

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

Contextos $C ::= [] \mid \lambda x . C \mid CM \mid MC$

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 reemplazando en C la única ocurrencia de $[]$ por M .

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

| | | | |
|----|------------------------------|----------|----------------------------------|
| Si | $C \equiv []$, | entonces | $C[M] \equiv M$ |
| Si | $C \equiv \lambda x . C_0$, | entonces | $C[M] \equiv \lambda x . C_0[M]$ |
| Si | $C \equiv C_0 M_0$, | entonces | $C[M] \equiv C_0[M] M_0$ |
| Si | $C \equiv M_0 C_0$, | entonces | $C[M] \equiv M_0 C_0[M]$ |

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:

$$\text{Si } C \equiv \lambda x . x [], \text{ entonces } C[\mathbf{I}] \equiv \lambda x . x \mathbf{I}$$

$$C[y] \equiv \lambda x . x y$$

$$C[x] \equiv \lambda x . x x$$

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

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

- Los contextos también se componen entre sí mediante la operación $C \circ C' := C[C']$, de tal modo que $(C \circ C')[M] := C[C'[M]]$
- **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 $\rightarrow_{\mathcal{R}}$ definida por las 4 reglas:

$$\frac{M \mathcal{R} M'}{M \rightarrow_{\mathcal{R}} M'} \quad (\text{CBase})$$

$$\frac{M \rightarrow_{\mathcal{R}} M'}{\lambda x . M \rightarrow_{\mathcal{R}} \lambda x . M'} \quad (\text{CLam})$$

$$\frac{M_1 \rightarrow_{\mathcal{R}} M'_1}{M_1 M_2 \rightarrow_{\mathcal{R}} M'_1 M_2} \quad (\text{CApp}_1)$$

$$\frac{M_2 \rightarrow_{\mathcal{R}} M'_2}{M_1 M_2 \rightarrow_{\mathcal{R}} M_1 M'_2} \quad (\text{CApp}_2)$$

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

Proposición (Propiedades de la clausura contextual)

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

- (1) $\rightarrow_{\mathcal{R}}$ es la mínima relación compatible que contiene \mathcal{R}
- (2) Si \mathcal{R} es sustitutiva, entonces $\rightarrow_{\mathcal{R}}$ es también es sustitutiva
- (3) Si $M \mathcal{R} M'$ implica $FV(M') \subseteq FV(M)$ (para todos M, M')
 entonces $M \rightarrow_{\mathcal{R}} M'$ implica $FV(M') \subseteq FV(M)$ (para todos M, M')
 (Misma propiedad con $=$ en lugar de \subseteq)
- (4) Tenemos que: $\rightarrow_{(\mathcal{R} \cup \mathcal{S})} = \rightarrow_{\mathcal{R}} \cup \rightarrow_{\mathcal{S}}$

Relaciones inducidas

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

$\xrightarrow{0} := \{(x, x) : x \in A\}$ **identidad**

$\xrightarrow{i+1} := \xrightarrow{i} \circ \rightarrow$ **reducción en $i + 1$ pasos**

$\xrightarrow{+} := \bigcup_{i>0} \xrightarrow{i}$ **clausura transitiva**

$\xrightarrow{*} := \xrightarrow{+} \cup \xrightarrow{0}$ **clausura reflexiva-transitiva**

$\xrightarrow{=} := \rightarrow \cup \xrightarrow{0}$ **clausura reflexiva**

$\leftrightarrow := \rightarrow \cup (\rightarrow)^{-1}$ **clausura simétrica**

$\cong := (\leftrightarrow)^*$ **clausura reflexiva-simétrica-transitiva**

Obs.: Se nota que si \rightarrow 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:

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

- (2) La relación \rightarrow_{β} 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 \rightarrow_{β} es compatible y sustitutiva

Además:

- (3) Si $M \rightarrow_{\beta} M'$, entonces $FV(M') \subseteq FV(M)$

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

$$(\lambda x . (\lambda y . y)) z \rightarrow_{\beta} \lambda y . y$$

Ejemplos

Sea consideran los términos:

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

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

$$\mathbf{I} M \equiv (\lambda x . x) M \rightarrow_{\beta} M$$

$$\mathbf{K} M N \equiv (\lambda x . \lambda y . x) M N \rightarrow_{\beta} (\lambda y . M) N \rightarrow_{\beta} M$$

$$\begin{aligned} \mathbf{B} M N P &\equiv (\lambda x . \lambda y . \lambda z . x (y z)) M N P \\ &\rightarrow_{\beta} (\lambda y . \lambda z . M (y z)) N P \\ &\rightarrow_{\beta} (\lambda z . M (N z)) P \rightarrow_{\beta} M (N P) \end{aligned}$$

$$\Delta M \equiv (\lambda x . x x) M \rightarrow_{\beta} M M$$

Y en particular:

$$\Delta \Delta \rightarrow_{\beta} \Delta \Delta$$

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: $M \rightarrow_{\beta} M_1 \rightarrow_{\beta} M_2 \rightarrow_{\beta} M_3 \rightarrow_{\beta} \dots$

Ejemplos:

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

Formas normales

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

$$\begin{aligned} \mathbf{neu} & ::= x \mid \mathbf{neu\ nor} \\ \mathbf{nor} & ::= \mathbf{neu} \mid \lambda x. \mathbf{nor} \end{aligned}$$

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

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

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.

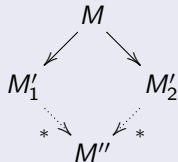
Confluencia local de la β -reducción

Se puede demostrar sin dificultad que:

Proposición (Confluencia local)

La β -reducción es localmente confluyente:

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



Problema: No se puede deducir fácilmente la propiedad de confluencia
(En efecto, no se puede usar el lema de Newman, ya que \rightarrow_{β} 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:

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

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

Observaciones:

- La relación \Rightarrow es compatible (obvio por (PRefI), (PLam) y (PApp))
- Tenemos que: $(\rightarrow_{\beta}) \subset (\Rightarrow) \subset (\overset{*}{\rightarrow}_{\beta})$ (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:

- (PRef1). 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 $y \notin FV(M_2M'_2NN')$.
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 $M \equiv M_1M_2$ y $M' \equiv M'_1M'_2$, 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 $M[x := N] \equiv M_1[x := N]M_2[x := N] \Rightarrow M'_1[x := N']M'_2[x := N'] \equiv M'[x := N']$. □

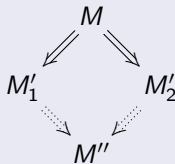
Reducción paralela

(3/7)

Proposición (Propiedad del diamante)

La reducción paralela cumple la propiedad del diamante:

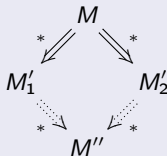
Si $M \Rightarrow M'_1$ y $M \Rightarrow M'_2$,
entonces existe M'' tal que
 $M'_1 \Rightarrow M''$ y $M'_2 \Rightarrow M''$



Corolario (Confluencia de la reducción paralela)

La reducción paralela es confluyente:

Si $M \Rightarrow^* M'_1$ y $M \Rightarrow^* M'_2$,
entonces existe M'' tal que
 $M'_1 \Rightarrow^* M''$ y $M'_2 \Rightarrow^* M''$



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

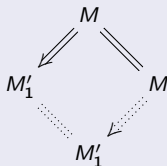
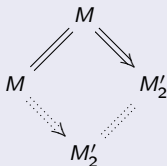
Reducción paralela

(4/7)

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:

| | | $M \Rightarrow M'_2$ | | | |
|----------------------|---------|----------------------|---------|--------|--------|
| | | (PRefI) | (PBeta) | (PLam) | (PApp) |
| $M \Rightarrow M'_1$ | (PRefI) | X | X | X | X |
| | (PBeta) | X | X | | X |
| | (PLam) | X | | X | |
| | (PApp) | X | X | | X |

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



(...)

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 $P \Rightarrow P'_2$

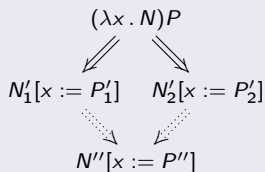
Por HI, sabemos que

- 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.



(...)

Reducción paralela

(6/7)

Demostración de la propiedad del diamante (continuación).

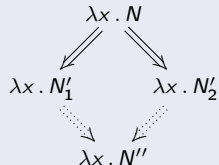
- (PLam)/(PLam). En este caso, sabemos que

- $M \equiv \lambda x . N$
- $M'_1 \equiv \lambda x . N'_1$, 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 $M'' := \lambda x . N''$.



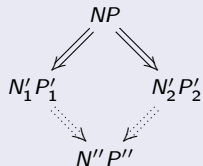
- (PApp)/(PApp). En este caso, sabemos que

- $M \equiv NP$
- $M'_1 \equiv N'_1 P'_1$, 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 $M'' := N'' P''$.



(...)

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 $N \Rightarrow N'_2$.

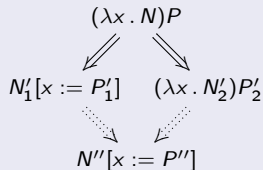
Por HI, sabemos que

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

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.



Confluencia de la β -reducción

(1/2)

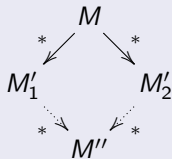
- Vimos que la reducción paralela \Rightarrow es confluente
- Por otro lado, sabemos que $(\rightarrow_{\beta}) \subset (\Rightarrow) \subset (\overset{*}{\rightarrow}_{\beta})$.
Esto implica que: $(\overset{*}{\rightarrow}_{\beta}) \subset (\overset{*}{\Rightarrow}) \subset (\overset{*}{\rightarrow}_{\beta})^*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

Teorema (Confluencia de la β -reducción)

La β -reducción es confluente:

Si $M \overset{*}{\rightarrow}_{\beta} M'_1$ y $M \overset{*}{\rightarrow}_{\beta} M'_2$,
entonces existe M'' tal que

$M'_1 \overset{*}{\rightarrow}_{\beta} M''$ y $M'_2 \overset{*}{\rightarrow}_{\beta} M''$



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 $M_2 \xrightarrow{*}_{\beta} M'$



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

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

Plan

- 1 Introducción
- 2 Sintaxis
- 3 β -Reducción
- 4 η -Reducción**
- 5 Índices de De Bruijn
- 6 Computabilidad

El problema de la extensionalidad

(1/2)

- Dos términos M, M' son **extensionalmente equivalentes** cuando

$$(1) \quad M N \cong_{\beta} M' N \quad (\text{para todo } N)$$

Pregunta: ¿Son tales términos β -convertibles?

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

$$(1') \quad M x \cong_{\beta} M' x \quad (x \text{ variable fresca})$$

- **Contraejemplo:** Tenemos que

$$(\lambda z . y z) x \rightarrow_{\beta} y x$$

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

$$\frac{Mx \cong M'x}{M \cong M'} \quad (x \text{ fresca})$$

Proposición

Las dos condiciones son equivalentes:

(1) \cong es extensional

(2) \cong es tal que $\lambda x. Mx \cong M$ 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 $(\lambda x. Mx)z \rightarrow_{\beta} Mz$, entonces $(\lambda x. Mx)z \cong Mz$, y luego $\lambda x. Mx \cong M$ por (1).

(2) \Rightarrow (1) Supongamos que $Mx \cong M'x$, con $x \notin FV(M)$ y $x \notin FV(M')$. Entonces $\lambda x. Mx \cong \lambda x. M'x$ por compatibilidad, y luego $M \cong M'$ por (2). \square

- Motiva la nueva regla: $\lambda x. Mx \rightarrow_{\eta} M \quad (x \notin FV(M))$

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 \rightarrow_{η} 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 \rightarrow_{η} es compatible y sustitutiva

Además:

- (3) Si $M \rightarrow_{\eta} M'$, 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 $M \rightarrow_{\eta} M'$, tenemos que $|M| > |M'|$

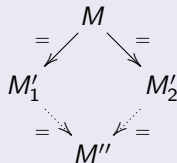
Y por lo tanto:

(2) La relación \rightarrow_{η} es **fuertemente normalizante**

Proposición (Propiedad del diamante)

La relación $\overrightarrow{\rightarrow}_{\eta}$ cumple la propiedad del diamante:

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



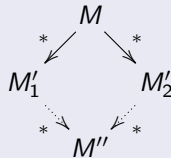
Demostración. Ejercicio.

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 $M'_2 \xrightarrow{*}_{\eta} M''$



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

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

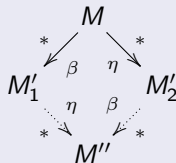
- (1) La forma η -normal de cualquier término es única
- (2) La η -reducción es Church-Rosser
- (3) Dos términos son η -convertibles sii tienen misma forma η -normal

Commutación β/η

Proposición (Commutación β/η)

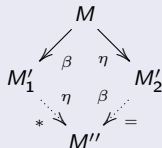
La relación \rightarrow_η conmuta con \rightarrow_β :

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



Demostración. En primer lugar, se demuestra (Ejercicio) que las relaciones \rightarrow_β y \rightarrow_η 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 $M'_2 \xrightarrow{*}_\eta M''$



Demostración: ejercicio.

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

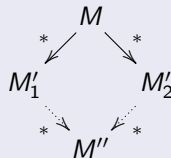
Confluencia de la $\beta\eta$ -reducción

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

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 $M'_2 \xrightarrow{*}_{\beta\eta} M''$



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

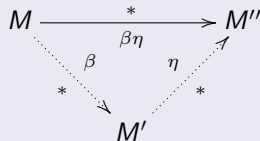
Corolario (Church-Rosser + unicidad de las formas $\beta\eta$ -normales)

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

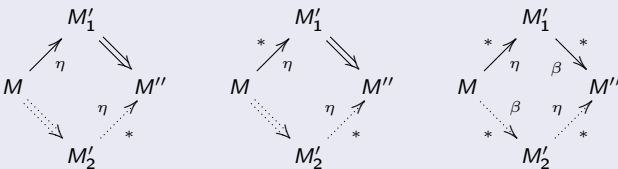
Postergación de la regla η

Proposición (Postergación de la regla η)

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



Demostración. Sigue de los siguientes diagramas (ejercicio):



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



Plan

- 1 Introducción
- 2 Sintaxis
- 3 β -Reducción
- 4 η -Reducción
- 5 Índices de De Bruijn**
- 6 Computabilidad

¿Cómo evitar los problemas de α -conversión?

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

1 = último λ , 2 = penúltimo λ , etc.

- **Ejemplos:**

$$\lambda x . x \rightsquigarrow \lambda . 1$$

$$\lambda y . y \rightsquigarrow \lambda . 1$$

$$\lambda x . \lambda y . x \rightsquigarrow \lambda . \lambda . 2$$

$$\lambda x . \lambda y . y \rightsquigarrow \lambda . \lambda . 1$$

$$\lambda x . \lambda y . \lambda z . x z (y z) \rightsquigarrow \lambda . \lambda . \lambda . 3 1 (2 1)$$

$$\lambda x . x (\lambda y . y x) \rightsquigarrow \lambda . 1 (\lambda . 1 2)$$

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

Definición (Términos lambda con índices de De Bruijn)

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

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

$$FI(i) := \{i\}$$

$$\begin{aligned} FI(\lambda.M) &:= (FI(M) \setminus \{1\}) - 1 \\ &:= \{i \geq 1 : i + 1 \in FI(M)\} \end{aligned}$$

$$FI(MN) := FI(M) \cup FI(N)$$

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

Términos con índices de De Bruijn

(2/2)

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

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

$$(x_i)_{x_1, \dots, x_n}^{DB} \quad ::= \quad i$$

$$(\lambda y . M)_{x_1, \dots, x_n}^{DB} \quad ::= \quad \lambda . (M)_{y, x_1, \dots, x_n}^{DB}$$

$$(MN)_{x_1, \dots, x_n}^{DB} \quad ::= \quad (M)_{x_1, \dots, x_n}^{DB} (N)_{x_1, \dots, x_n}^{DB}$$

Proposición (Criterio de α -equivalencia)

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

$$M \equiv_{\alpha} M' \quad \text{sii} \quad (M)_{x_1, \dots, x_n}^{DB} \equiv (M')_{x_1, \dots, x_n}^{DB}$$

Demostración. Ejercicio.

Definición (Sustitución con índices de De Bruijn)

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

$$i[p := P] \equiv \begin{cases} i & \text{si } i < p \\ \uparrow_1^p P & \text{si } i = p \\ i - 1 & \text{si } i > p \end{cases}$$

$$(\lambda.M)[p := P] \equiv \lambda.M[p + 1 := P]$$

$$(MN)[p := P] \equiv M[p := P]N[p := P]$$

Lema

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

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

Demostración. Ejercicio.

Proposición (Propiedades del lifting y de la sustitución)

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

$$(1) \uparrow_k^1 M \equiv M$$

$$(2) \uparrow_j^p(\uparrow_k^n M) \equiv \uparrow_k^{p+n-1} M \quad (\text{si } k \leq j < k+n)$$

$$(3) \uparrow_j^p(\uparrow_k^n M) \equiv \uparrow_{k+p-1}^n(\uparrow_j^p M) \quad (\text{si } j \leq k)$$

$$(4) (\uparrow_k^{n+1} M)[p := N] \equiv \uparrow_k^n M \quad (\text{si } k \leq p \leq k+n-1)$$

$$(5) \uparrow_k^n(M[p := N]) \equiv (\uparrow_k^n M)[p+n-1 := N] \quad (\text{si } k \leq p)$$

$$(6) \uparrow_{k+p-1}^n(M[p := N]) \equiv (\uparrow_{k+p}^n M)[p+n := \uparrow_k^n N]$$

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

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:

$$(\lambda.M)N \rightarrow_{\beta} M[1 := N]$$

+ 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

$$M, N ::= x \mid i \mid \lambda_x.M \mid MN$$

Plan

- 1 Introducción
- 2 Sintaxis
- 3 β -Reducción
- 4 η -Reducción
- 5 Índices de De Bruijn
- 6 **Computabilidad**

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

$$\begin{aligned} \text{true} &::= \lambda xy. x \\ \text{false} &::= \lambda xy. y \\ \text{if} &::= \lambda bxy. b x y \end{aligned}$$

- Tenemos que:

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

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

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

Representación de los pares

- Se definen:

$$\langle M_1, M_2 \rangle \equiv \lambda z . z M_1 M_2 \quad (z \text{ fresca})$$

$$\text{pair} \equiv \lambda xyz . z x y$$

$$\text{fst} \equiv \lambda z . z (\lambda xy . x)$$

$$\text{snd} \equiv \lambda z . z (\lambda xy . y)$$

- Tenemos que:

$$\text{pair } M_1 M_2 \xrightarrow{*}_{\beta} \langle M_1, M_2 \rangle$$

$$\text{fst } \langle M_1, M_2 \rangle \xrightarrow{*}_{\beta} M_1$$

$$\text{snd } \langle M_1, M_2 \rangle \xrightarrow{*}_{\beta} M_2$$

Representación de las n -uplas

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

$$\langle M_1, \dots, M_n \rangle \equiv \lambda z. z M_1 \cdots M_n \quad (z \text{ fresca})$$

$$\text{upla}_n \equiv \lambda x_1 \cdots x_n z. z x_1 \cdots x_n$$

$$\text{proj}_i^n \equiv \lambda z. z (\lambda x_1 \cdots x_n. x_i) \quad (1 \leq i \leq n)$$

- Tenemos que:

$$\text{upla}_n M_1 \cdots M_n \xrightarrow{*}_{\beta} \langle M_1, \dots, M_n \rangle$$

$$\text{proj}_i^n \langle M_1, \dots, M_n \rangle \xrightarrow{*}_{\beta} M_i$$

Representación de los enteros naturales

(1/2)

- Se definen los **enteros de Church** por:

$$\begin{aligned} \bar{n} &::= \lambda f . \lambda x . \underbrace{f(\dots(f x)\dots)}_n \\ &\equiv \lambda f . \underbrace{f \circ \dots \circ f}_n \end{aligned} \quad (n \in \mathbb{N})$$

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

$$\bar{n} F M \xrightarrow{*}_{\beta} \underbrace{F(\dots(F M)\dots)}_n$$

- = recursión **acotada**
- = bucle «**for**»

Representación de los enteros naturales

(2/2)

- Se definen:

$$\text{succ} \quad \equiv \quad \lambda n f x . f (n f x)$$

$$\text{null} \quad \equiv \quad \lambda n . n (\lambda x . \text{false}) \text{true}$$

$$\text{plus} \quad \equiv \quad \lambda n m f x . n f (m f x)$$

$$\text{mult} \quad \equiv \quad \lambda n m . n (\text{plus } m) \bar{0}$$

$$\text{pow} \quad \equiv \quad \lambda p n . n (\text{mult } p) \bar{1}$$

$$\text{pred} \quad \equiv \quad \lambda n . \text{fst} (n (\lambda z . z (\lambda x y . \langle y, \text{succ } y \rangle)) (\bar{0}, \bar{0}))$$

$$\text{minus} \quad \equiv \quad \lambda n m . m \text{ pred } n$$

- Tenemos que: $\text{succ } \bar{n} \xrightarrow{*}_{\beta} \overline{n+1}$

$$\text{null } \bar{n} \xrightarrow{*}_{\beta} \begin{cases} \text{true} & \text{si } n = 0 \\ \text{false} & \text{si } n \geq 1 \end{cases}$$

$$\text{plus } \bar{n} \bar{m} \xrightarrow{*}_{\beta} \overline{n+m} \quad (\text{etc.})$$

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

$$\mathbf{Y} ::= \lambda f. (\lambda x. f (x x)) (\lambda x. f (x x))$$

Proposición (Combinador de punto fijo)

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

Demostración. Tenemos que

$$\begin{aligned} \mathbf{Y} M &\equiv (\lambda f. (\lambda x. f (x x)) (\lambda x. f (x x))) M \\ &\rightarrow_{\beta} (\lambda x. M (x x)) (\lambda x. M (x x)) \\ &\rightarrow_{\beta} M ((\lambda x. M (x x)) (\lambda x. M (x x))) \quad \beta \leftarrow M(\mathbf{Y} M) \quad \square \end{aligned}$$

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 $F_M ::= \mathbf{Y} M$. □

- 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

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

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

$$\begin{aligned} \mathbf{Y} M &\rightarrow_{\beta} (\lambda x. M(x x)) (\lambda x. M(x x)) \\ &\rightarrow_{\beta} M((\lambda x. M(x x)) (\lambda x. M(x x))) \\ &\rightarrow_{\beta} M(M((\lambda x. M(x x)) (\lambda x. M(x x)))) \\ &\quad \vdots \\ &\rightarrow_{\beta} M(M(M(M(M(M(M \dots))))))) \end{aligned}$$

En particular: $\mathbf{Y} \mathbf{I} \xrightarrow{3}_{\beta} \Delta \Delta \rightarrow_{\beta} \Delta \Delta \rightarrow_{\beta} \dots$ ($\Delta ::= \lambda x. x x$)

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

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

Puntos fijos

(3/5)

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

$$\begin{aligned} \text{fact} &::= \mathbf{Y} M, & \text{con} \\ M &::= \lambda fx. \text{if } (\text{null } x) \bar{1} (\text{mult } x (f (\text{pred } x))) \end{aligned}$$

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

$$\begin{aligned} \text{fact } \bar{0} &\cong_{\beta} M \text{ fact } \bar{0} \\ &\cong_{\beta} \text{if } (\text{null } \bar{0}) \bar{1} (\text{mult } \bar{0} (\text{fact } (\text{pred } \bar{0}))) \cong_{\beta} \bar{1} \\ \text{fact } (\overline{n+1}) &\cong_{\beta} M \text{ fact } (\overline{n+1}) \\ &\cong_{\beta} \text{if } (\text{null } (\overline{n+1})) \bar{1} (\text{mult } (\overline{n+1}) (\text{fact } (\text{pred } (\overline{n+1})))) \\ &\cong_{\beta} \text{mult } (\overline{n+1}) (\text{fact } \bar{n}) \end{aligned}$$

Proposición

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

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

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

$$\mathbf{Y} M \cong_{\beta} M(\mathbf{Y} M),$$

pero

$$\not\rightarrow_{\beta}^* M(\mathbf{Y} M)$$

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

$$\mathbf{T} \equiv (\lambda y f . f (y y f))(\lambda y f . f (y y f))$$

Proposición (Combinador de punto fijo)

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

Demostración. Ejercicio

- Otro ejemplo de combinador de punto fijo:

$\mathbf{U} ::= \lambda abc \dots xyz.z(\text{ese combinador es una estupidez})$

$\mathbf{V} ::= \underbrace{\mathbf{U} \dots \mathbf{U}}_{26}$

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

λ -definibilidad

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

$\text{IN}^k \rightarrow \text{IN}$ = conjunto de las **funciones parciales** de IN^k a IN

$\text{IN}^k \rightarrow \text{IN}$ = conjunto de las **funciones totales** de IN^k a IN

Tenemos que $(\text{IN}^k \rightarrow \text{IN}) \subseteq (\text{IN}^k \rightarrow \text{IN})$

Definición (Funciones λ -definibles)

Se dice que una función parcial $f : \text{IN}^k \rightarrow \text{IN}$ es **λ -definible** cuando existe un término lambda M tal que para todos $n_1, \dots, n_k \in \text{IN}$:

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

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** $z : \mathbb{IN} \rightarrow \mathbb{IN}$, definida por

$$z(n) := 0$$

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

- La **función sucesor** $s : \mathbb{IN} \rightarrow \mathbb{IN}$, definida por

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

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

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

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

(para todo $(n_1, \dots, n_k) \in \mathbb{IN}^k$)

Esquemas de composición y de recursión primitiva

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

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

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

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

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

$$h(0, n_1, \dots, n_k) := f(n_1, \dots, n_k)$$

$$h(n+1, n_1, \dots, n_k) := g(n, h(n, n_1, \dots, n_k), n_1, \dots, n_k)$$

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

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

Esquema de minimización

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

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

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

Se nota $h = \mu(f)$

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

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

En particular: $h = \mu(f)$ 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 $\subseteq \bigcup_{k \geq 1} (\mathbb{N}^k \rightarrow \mathbb{N})$ que contiene todas las funciones iniciales

$$z, \quad s, \quad \pi_i^k \quad (k \geq i \geq 1)$$

y está cerrado por composición, recursión primitiva y minimización:

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

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 $f : \mathbb{N}^k \rightarrow \mathbb{N}$ recursiva, la condición $(n_1, \dots, n_k) \notin \text{dom}(f)$ expresa que el cálculo de $f(n_1, \dots, n_k)$ **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 $f : \mathbb{N}^k \rightarrow \mathbb{N}$, las siguientes condiciones son equivalentes:

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

¹Con memoria y tiempo ilimitados

Otro teorema del punto fijo

Ahora se supone dada una codificación efectiva² $M \mapsto \ulcorner M \urcorner$ 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 $M \ulcorner F_M \urcorner \cong_{\beta} F_M$

Demostración. Como la codificación es efectiva, existen términos **A** y **N** tales que:

- $\mathbf{A} \ulcorner M \urcorner \ulcorner N \urcorner \cong_{\beta} \ulcorner MN \urcorner$ (para todos $M, N \in \Lambda$)
- $\mathbf{N} \bar{n} \cong_{\beta} \overline{\lambda f x. \underbrace{f(\dots(f x)\dots)}_n}$ (para todo $n \in \mathbb{N}$)

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

$$\begin{aligned} F_M &\equiv W_M \ulcorner W_M \urcorner \\ &\cong_{\beta} M(\mathbf{A} \ulcorner W_M \urcorner (\mathbf{N} \ulcorner W_M \urcorner)) \\ &\cong_{\beta} M \ulcorner W_M \ulcorner W_M \urcorner \urcorner \equiv M \ulcorner F_M \urcorner \end{aligned}$$

□

²Es 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

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

para todo término M

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

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

Ambos casos son absurdos, y por lo tanto, el término H no existe. □

