Lógica y tipos

Mario Román

January 18, 2018

Outline

Lógica intuicionista

Deducción natural

Cálculo lambda simplemente tipado

Categorías cartesianas

Tipos dependientes

Lógica intuicionista

Intuicionismo

La lógica intuicionista es la que se obtiene al no asumir la ley del tercio excluso, la matemática intuicionista es la que se construye sobre esta lógica.

Ley del tercio excluso
Para toda proposición A se tiene $A \vee \neg A$.

Nótese que tampoco negamos que pueda ser el caso, simplemente no la asumimos en nuestras demostraciones.

Doble negación

Asumiremos que $\neg P$ es por definición la implicación $P \to \bot$, donde $\bot$ es la proposición falsa.

$$\neg P \triangleq (P \to \bot)$$

El principio de eliminación de la doble negación es equivalente a la ley del tercio excluso, así que no podremos demostrarlo en lógica intuicionista.

Eliminación de la doble negación
Para toda proposición A se tiene $\neg\neg A \to A$.

Demostración por contradicción o por negación

Demostración por contradicción
Supongamos $\neg P$, *... llegamos a contradicción, luego* P.

Demostración de una negación
Supongamos P, *... llegamos a contradicción, luego* $\neg P$.

¡Nótese que no son lo mismo!, la segunda proviene directamente de la definición de una negación y es correcta en lógica intuicionista; pero la segunda asume implícitamente que $\neg\neg P \to P$ y no es correcta en general.

Constructivismo

El intuicionismo es una vertiente del constructivismo, una corriente en filosofía de las matemáticas que no admite demostraciones indirectas de existencia. Hace una distinción entre construir efectivamente un objeto o simplemente demostrar que si no existiera se llegaría a contradicción. Tiene varias ramas [1]:

[1]: Troelstra, Anne Sjerp. "History of constructivism in the twentieth century." (1991).

Constructivismo

El intuicionismo es una vertiente del constructivismo, una corriente en filosofía de las matemáticas que no admite demostraciones indirectas de existencia. Hace una distinción entre construir efectivamente un objeto o simplemente demostrar que si no existiera se llegaría a contradicción. Tiene varias ramas [1]:

1. intuicionismo, *evitar el tercio excluso*;

[1]: Troelstra, Anne Sjerp. "History of constructivism in the twentieth century." (1991).

Constructivismo

El intuicionismo es una vertiente del constructivismo, una corriente en filosofía de las matemáticas que no admite demostraciones indirectas de existencia. Hace una distinción entre construir efectivamente un objeto o simplemente demostrar que si no existiera se llegaría a contradicción. Tiene varias ramas [1]:

1. intuicionismo, *evitar el tercio excluso*;
2. predicativismo, *evitar definiciones autorreferentes*;

[1]: Troelstra, Anne Sjerp. "History of constructivism in the twentieth century." (1991).

Constructivismo

El intuicionismo es una vertiente del constructivismo, una corriente en filosofía de las matemáticas que no admite demostraciones indirectas de existencia. Hace una distinción entre construir efectivamente un objeto o simplemente demostrar que si no existiera se llegaría a contradicción. Tiene varias ramas [1]:

1. intuicionismo, *evitar el tercio excluso*;
2. predicativismo, *evitar definiciones autorreferentes*;
3. finitismo, *evitar el infinito explícito*;

[1]: Troelstra, Anne Sjerp. "History of constructivism in the twentieth century." (1991).

Constructivismo

El intuicionismo es una vertiente del constructivismo, una corriente en filosofía de las matemáticas que no admite demostraciones indirectas de existencia. Hace una distinción entre construir efectivamente un objeto o simplemente demostrar que si no existiera se llegaría a contradicción. Tiene varias ramas [1]:

1. intuicionismo, *evitar el tercio excluso*;
2. predicativismo, *evitar definiciones autorreferentes*;
3. finitismo, *evitar el infinito explícito*;
4. ultrafinitismo, *evitar números muy grandes* (???).

[1]: Troelstra, Anne Sjerp. "History of constructivism in the twentieth century." (1991).

Crítica de Hilbert al constructivismo

"after all, it is part of the task of science to liberate us from arbitrariness, sentiment and habit, and to protect us from the subjectivism that... finds its culmination in intuitionism."

"Taking the principle of excluded middle from the mathematician would be the same, say, as proscribing the telescope to the astronomer or to the boxer the use of his fists. To prohibit existence statements and the principle of excluded middle is tantamount to relinquishing the science of mathematics altogether." [2]

[2]: Martin-Löf, Per. "The Hilbert-Brouwer controversy resolved?." One hundred years of intuitionism (1907–2007) (2008): 243-256.

Deducción natural

El sistema de la deducción natural

La deducción natural es un sistema lógico intuicionista creado por Gerhard Gentzen. Usará las conectivas lógicas $\to, \wedge, \vee$, y dos proposiciones $\top, \bot$, representando las proposiciones verdadera y falsa.

$$\dfrac{\dfrac{\dfrac{A, B \vdash A \qquad A, B \vdash B}{A, B \vdash A \wedge B}}{A \vdash B \to A \wedge B}}{\vdash A \to B \to A \wedge B}$$

Las reglas del sistema vienen en pares de introducción/eliminación para cada constructor. Las demostraciones son árboles de derivación y existen reglas de simplificación de demostraciones.

La conjunción

La conjunción se forma con una regla de introducción y dos de eliminación.

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

Bajo un contexto (conjunto de hipótesis) Γ cualquiera, si podemos deducir $\Gamma \vdash A$ y $\Gamma \vdash B$, deducimos $\Gamma \vdash A \wedge B$. La regla de eliminación es inversa a la introducción.

La disyunción

La disyunción se forma con dos reglas de introducción y una de eliminación

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

Nótese cómo la regla mezcla inferencias con contextos distintos. Sólo podemos introducir una disyunción cuando conocemos cuál estamos introduciendo. La regla de eliminación es la demostración por casos.

La implicación

La implicación se forma cada vez que podemos deducir algo desde una hipótesis.

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

Proposiciones verdadera y falsa

La proposición verdadera se sigue de cualquier contexto

$$\frac{}{\Gamma \vdash \top}$$

mientras que de la proposición falsa se sigue cualquier cosa

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash C}$$

Ejemplos

Se cumplen

- $A \wedge B \rightarrow B \wedge A$
- $(\neg A \vee \neg B) \rightarrow \neg(A \wedge B)$
- $A \rightarrow \neg\neg A$,
- $(\neg A \wedge \neg B) \rightarrow \neg(A \vee B)$
- $\neg(A \vee B) \rightarrow (\neg A \wedge \neg B)$

y no se cumplen en general

- $A \vee \neg A$
- $\neg(A \wedge B) \rightarrow (\neg A \vee \neg B)$
- $\neg\neg A \rightarrow A$,

¿cómo demostrar que algo no se cumple en general?

Ejemplo: demostración

Demostramos que $A \to \neg\neg A$,

$$\dfrac{\dfrac{\dfrac{A, A \to \bot \vdash A \to \bot \qquad A, A \to \bot \vdash A}{A, A \to \bot \vdash \bot}}{A \vdash (A \to \bot) \to \bot}}{\vdash A \to ((A \to \bot) \to \bot)}$$

Ejemplo: demostrar que algo no se cumple en general

Para demostrar que algo no se puede demostrar desde la teoría puede hacerse buscando modelos de la teoría en los que no se cumpla necesariamente.

Un modelo de lógica booleana son los subconjuntos. Nuestro modelo de lógica intuicionista serán los abiertos de un espacio topógico, donde cada proposición es un abierto y demostrar $A_1, \dots, A_n \vdash B$ es demostrar $A_1 \cap \dots \cap A_n \subseteq B$.

Ejemplo: demostrar que algo no se cumple en general

Para demostrar que algo no se puede demostrar desde la teoría puede hacerse buscando modelos de la teoría en los que no se cumpla necesariamente.

Un modelo de lógica booleana son los subconjuntos. Nuestro modelo de lógica intuicionista serán los abiertos de un espacio topógico, donde cada proposición es un abierto y demostrar $A_1, \dots, A_n \vdash B$ es demostrar $A_1 \cap \dots \cap A_n \subseteq B$.

lógica	abiertos de Ω
$P \wedge Q$	$P \cap Q$
$P \vee Q$	$P \cup Q$
$P \to Q$	$\mathrm{int}(\overline{P} \cup Q)$
$\top$	Ω
$\bot$	$\varnothing$
$\neg P$	$\mathrm{int}\left(\overline{P}\right)$

Ejemplo: demostrar que algo no se cumple en general

Comprobamos que las reglas de la deducción natural se cumplen y podemos usar abiertos para buscar contraejemplos de

- $A \vee \neg A$
- $\neg\neg A \to A$
- $\neg(A \wedge B) \to (\neg A \vee \neg B)$
- $\neg(A \vee B) \to (\neg A \wedge \neg B)$

Cálculo lambda simplemente tipado

Interpretación de Brower-Heyting-Kolmogorov

La idea de Heyting-Kolmogorov es leer las conectivas como

- una demostración de $A \wedge B$ es una demostración de A y una demostración de B;
- una demostración de $A \vee B$ es una demostración de A o una demostración de B (indicando cuál);
- una demostración de $A \to B$ es un razonamiento que transforma una demostración de A en una demostración de B;
- $\bot$ no tiene ninguna demostración.

Esto abre la posibilidad de interpretar $A \to B$ como funciones (en el sentido de algoritmos o procesos de transformación) entre demostraciones.

El cálculo lambda tipado

El cálculo lambda es un modelo de computación, basado en términos lambda (abstracción y aplicación de funciones). Por ejemplo, $\lambda x.x^2$ sería la función que eleva al cuadrado y $(\lambda x.x^2)\ 4 \equiv 16$.

Expresaremos el cálculo lambda con varias reglas, algunas de creación y eliminación de términos y otras sobre cómo calcular con los términos.

Tipo función

Los términos lambda provienen de abstraer una expresión simbólica en una función. Las reglas para crearlos son

$$\frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash \lambda a.b : A \to B} \qquad \frac{\Gamma \vdash f : A \to B \qquad \Gamma \vdash a : A}{\Gamma \vdash f\ a : B}$$

Tipo función

Los términos lambda provienen de abstraer una expresión simbólica en una función. Las reglas para crearlos son

$$\frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash \lambda a.b : A \to B} \qquad \frac{\Gamma \vdash f : A \to B \qquad \Gamma \vdash a : A}{\Gamma \vdash f\ a : B}$$

Se calcula sabiendo que

- aplicar una función lambda sobre un x es substituir las ocurrencias de la variable por las del argumento $(\lambda a.b)\ x \equiv b[x/a]$.

Tipo función

Los términos lambda provienen de abstraer una expresión simbólica en una función. Las reglas para crearlos son

$$\frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash \lambda a.b : A \to B} \qquad \frac{\Gamma \vdash f : A \to B \qquad \Gamma \vdash a : A}{\Gamma \vdash f\ a : B}$$

Se calcula sabiendo que

- aplicar una función lambda sobre un x es substituir las ocurrencias de la variable por las del argumento $(\lambda a.b)\ x \equiv b[x/a]$.

¡Sigue la misma regla que la implicación!

Tipo par

Los términos de tipo par, $A \times B$, se crean con un término de tipo A y otro de tipo B una función. Las reglas para crearlos son

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma \vdash m : A \times B}{\Gamma \vdash \pi_1\ m : A} \qquad \frac{\Gamma \vdash m : A \times B}{\Gamma \vdash \pi_2\ m : B}$$

Tipo par

Los términos de tipo par, $A \times B$, se crean con un término de tipo A y otro de tipo B una función. Las reglas para crearlos son

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma \vdash m : A \times B}{\Gamma \vdash \pi_1\ m : A} \qquad \frac{\Gamma \vdash m : A \times B}{\Gamma \vdash \pi_2\ m : B}$$

Se calcula sabiendo que

- $\pi_1 \langle x, y \rangle \equiv x$, $\pi_2 \langle x, y \rangle \equiv y$, siendo las proyecciones;
- $\langle \pi_1 m, \pi_2 m \rangle \equiv m$, como regla de unicidad.

Tipo par

Los términos de tipo par, $A \times B$, se crean con un término de tipo A y otro de tipo B una función. Las reglas para crearlos son

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma \vdash m : A \times B}{\Gamma \vdash \pi_1 \ m : A} \qquad \frac{\Gamma \vdash m : A \times B}{\Gamma \vdash \pi_2 \ m : B}$$

Se calcula sabiendo que

- $\pi_1 \langle x, y \rangle \equiv x$, $\pi_2 \langle x, y \rangle \equiv y$, siendo las proyecciones;
- $\langle \pi_1 m, \pi_2 m \rangle \equiv m$, como regla de unicidad.

¡Sigue las mismas reglas que la conjunción!

Tipo unión

Los términos de tipo unión, $A+B$ se crean con un término de tipo A o uno de tipo B indicando cuál. Las reglas para crearlos son

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash \mathrm{inl}\ a : A+B} \qquad \frac{\Gamma \vdash b : B}{\Gamma \vdash \mathrm{inr}\ b : A+B}$$

$$\frac{\Gamma \vdash m : A+B \qquad \Gamma, a : A \vdash n : C \qquad \Gamma, b : B \vdash p : C}{\Gamma \vdash (\mathrm{case}\ m\ \mathrm{of}\ n;\ p) : C}$$

Se calcula por casos sabiendo que

- $(\mathrm{case}\ m\ \mathrm{of}\ n;\ p) \equiv n[a'/a]$ cuando $m \equiv \mathrm{inl}\ a'$,
- $(\mathrm{case}\ m\ \mathrm{of}\ n;\ p) \equiv p[a'/a]$ cuando $m \equiv \mathrm{inr}\ b'$,
- $(\mathrm{case}\ m\ \mathrm{of}\ n[\mathrm{inl}\ a/c];\ n[\mathrm{inr}\ b/c]) \equiv n[m/c]$.

Tipo unión

Los términos de tipo unión, $A + B$ se crean con un término de tipo A o uno de tipo B indicando cuál. Las reglas para crearlos son

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash \mathrm{inl}\ a : A + B} \qquad \frac{\Gamma \vdash b : B}{\Gamma \vdash \mathrm{inr}\ b : A + B}$$

$$\frac{\Gamma \vdash m : A + B \qquad \Gamma, a : A \vdash n : C \qquad \Gamma, b : B \vdash p : C}{\Gamma \vdash (\mathrm{case}\ m\ \mathrm{of}\ n;\ p) : C}$$

Se calcula por casos sabiendo que

- $(\mathrm{case}\ m\ \mathrm{of}\ n;\ p) \equiv n[a'/a]$ cuando $m \equiv \mathrm{inl}\ a'$,
- $(\mathrm{case}\ m\ \mathrm{of}\ n;\ p) \equiv p[a'/a]$ cuando $m \equiv \mathrm{inr}\ b'$,
- $(\mathrm{case}\ m\ \mathrm{of}\ n[\mathrm{inl}\ a/c];\ n[\mathrm{inr}\ b/c]) \equiv n[m/c]$.

¡Sigue las mismas reglas que la disyunción!

Tipo 1 y 0

Hay un sólo término de tipo 1, dado por

$$\frac{}{\Gamma \vdash * : 1}$$

y no existe ningún término de tipo 0, pero desde uno de ese tipo se podría crear un término de cualquier tipo

$$\frac{\Gamma \vdash z : 0}{\Gamma \vdash \Box\, z : A}$$

Curry-Howard

Cada término lambda es entonces una codificación de las reglas que se han usado en el árbol de derivación de una fórmula en lógica intuicionista.

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash a \colon A \qquad \Gamma \vdash b \colon B}{\Gamma \vdash \langle a, b \rangle \colon A \times B}}{\Gamma' \vdash \lambda b.\langle a, b \rangle \colon B \to A \times B}}{\vdash \lambda a.\lambda b.\langle a, b \rangle \colon A \to B \to A \times B}$$

donde $\Gamma = \{a : A, b : B\}$ y $\Gamma' = \{a : A\}$.

Curry-Howard II

- Cada tipo es una expresión de lógica proposicional.
- Cada demostración es así un término lambda.
- Las reglas de computación que simplifican los términos lambda se corresponden con reglas de simplificación para los árboles de derivación. Cuando ejecutamos estamos simplificando una demostración.

Nuestra demostración anterior de $A \to \neg\neg A$ es simplemente el término lambda $\lambda a.(\lambda f.f\ a)$, donde $a : A$ y $f : \neg A$.

Ejercicios

- Demostrar conmutatividad de la conjunción definiendo una función lambda que intercambia los dos elementos de un par.
- Demostrar $\neg(A \vee B) \to (\neg A \wedge \neg B)$ con un término lambda.
- Demostrar $\neg\neg(A \vee \neg A)$ con un término lambda.

Mikrokosmos permite definir términos lambda en este sistema y dibujar los árboles de derivación.

Tipos inductivos

Podemos añadir tipos inductivos al cálculo lambda. Los tipos inductivos vienen dados por una lista de generadores. El ejemplo usual son los naturales, tienen generadores

- $0 : \mathbb{N}$,
- $S : \mathbb{N} \to \mathbb{N}$.

Tipos inductivos

Podemos añadir tipos inductivos al cálculo lambda. Los tipos inductivos vienen dados por una lista de generadores. El ejemplo usual son los naturales, tienen generadores

- $0 : \mathbb{N}$,
- $S : \mathbb{N} \to \mathbb{N}$.

Y podemos construir términos $0 : \mathbb{N}$, $S\ 0 : \mathbb{N}$, $S\ (S\ 0) : \mathbb{N}, \dots$

Tipos inductivos

Podemos añadir tipos inductivos al cálculo lambda. Los tipos inductivos vienen dados por una lista de generadores. El ejemplo usual son los naturales, tienen generadores

- $0 : \mathbb{N}$,
- $S : \mathbb{N} \to \mathbb{N}$.

Y podemos construir términos $0 : \mathbb{N}$, $S\ 0 : \mathbb{N}$, $S\ (S\ 0) : \mathbb{N}, \dots$

Sus reglas de eliminación corresponderían a una inducción estructural. Por ejemplo, $\mathrm{ind} : A \to (A \to A) \to \mathbb{N} \to A$ con

- $\mathrm{ind}\ a\ f\ 0 \equiv a$
- $\mathrm{ind}\ a\ f\ (S\ n) \equiv f\ (\mathrm{ind}\ a\ f\ n)$

Categorías cartesianas

Adjuntos

Dos funtores $F\colon \mathcal{C} \to \mathcal{D}$ y $G\colon \mathcal{D} \to \mathcal{C}$ son adjuntos cuando hay una biyección *natural* $\varphi\colon\ \hom(Fx, y) \cong \hom(x, Gy)$. Se escribe $F \dashv G$.

Adjuntos

Dos funtores $F\colon \mathcal{C} \to \mathcal{D}$ y $G\colon \mathcal{D} \to \mathcal{C}$ son adjuntos cuando hay una biyección *natural* $\varphi\colon\ \hom(Fx, y) \cong \hom(x, Gy)$. Se escribe $F \dashv G$.

Que es una biyección se puede traducir a

$$\frac{Fx \xrightarrow{f} y}{x \xrightarrow{\varphi(f)} Gy}$$

y un ejemplo son los grupos libres o los espacios topológicos discretos.

Adjuntos

Dos funtores $F\colon \mathcal{C} \to \mathcal{D}$ y $G\colon \mathcal{D} \to \mathcal{C}$ son adjuntos cuando hay una biyección *natural* $\varphi\colon\ \hom(Fx, y) \cong \hom(x, Gy)$. Se escribe $F \dashv G$.

Que es una biyección se puede traducir a

$$\frac{Fx \xrightarrow{f} y}{x \xrightarrow{\varphi(f)} Gy}$$

y un ejemplo son los grupos libres o los espacios topológicos discretos.

Pero además hay una condición de naturalidad.

Naturalidad

La condición de naturalidad no la trataremos en detalle, pero la podemos dibujar como

$$\begin{array}{c} \begin{tikzcd} Fx' \arrow[r, "Fh"] \arrow[rr, bend left, "f \circ Fh"] & Fx \arrow[r, "f"] & y \end{tikzcd} \\ \hline \begin{tikzcd} x' \arrow[r, "h"'] \arrow[rr, bend right, "\varphi(f) \circ h"'] & x \arrow[r, "\varphi(f)"'] & Gy \end{tikzcd} \end{array} \qquad \begin{array}{c} \begin{tikzcd} Fx \arrow[r, "f"] \arrow[rr, bend left, "k \circ f"] & y \arrow[r, "k"] & y' \end{tikzcd} \\ \hline \begin{tikzcd} x \arrow[r, "\varphi(f)"'] \arrow[rr, bend right, "Gk \circ \varphi(f)"'] & Gy \arrow[r, "Gk"'] & Gy' \end{tikzcd} \end{array}$$

Cálculo lambda como adjuntos I

Supongamos una categoría donde existen los siguientes adjuntos

$$\frac{* \longrightarrow *}{C \xrightarrow{!} 1} \qquad \frac{C, C \xrightarrow{f,g} A, B}{C \xrightarrow{f,g} A \times B} \qquad \frac{C \times A \xrightarrow{f} B}{C \xrightarrow{\widetilde{f}} B^A}$$

Son

1. adjunto derecho al funtor a la categoría de un solo objeto,

Cálculo lambda como adjuntos I

Supongamos una categoría donde existen los siguientes adjuntos

$$\frac{* \longrightarrow *}{C \xrightarrow{!} 1} \qquad \frac{C, C \xrightarrow{f,g} A, B}{C \xrightarrow{f,g} A \times B} \qquad \frac{C \times A \xrightarrow{f} B}{C \xrightarrow{\widetilde{f}} B^A}$$

Son

1. adjunto derecho al funtor a la categoría de un solo objeto,
2. adjunto derecho al funtor diagonal,

Cálculo lambda como adjuntos I

Supongamos una categoría donde existen los siguientes adjuntos

$$\frac{* \longrightarrow *}{C \xrightarrow{!} 1} \qquad \frac{C, C \xrightarrow{f,g} A, B}{C \xrightarrow{f,g} A \times B} \qquad \frac{C \times A \xrightarrow{f} B}{C \xrightarrow{\widetilde{f}} B^A}$$

Son

1. adjunto derecho al funtor a la categoría de un solo objeto,
2. adjunto derecho al funtor diagonal,
3. adjunto derecho al al funtor producto $- \times A$.

Cálculo lambda como adjuntos I

Supongamos una categoría donde existen los siguientes adjuntos

$$\frac{* \longrightarrow *}{C \xrightarrow{!} 1} \qquad \frac{C, C \xrightarrow{f,g} A, B}{C \xrightarrow{f,g} A \times B} \qquad \frac{C \times A \xrightarrow{f} B}{C \xrightarrow{\widetilde{f}} B^A}$$

Son

1. adjunto derecho al funtor a la categoría de un solo objeto,
2. adjunto derecho al funtor diagonal,
3. adjunto derecho al al funtor producto $- \times A$.

Una categoría de este tipo se llama categoría cartesiana cerrada.

Cálculo lambda como adjuntos I

Supongamos una categoría donde existen los siguientes adjuntos

$$\frac{* \longrightarrow *}{C \xrightarrow{!} 1} \qquad \frac{C, C \xrightarrow{f,g} A, B}{C \xrightarrow{f,g} A \times B} \qquad \frac{C \times A \xrightarrow{f} B}{C \xrightarrow{\widetilde{f}} B^A}$$

Son

1. adjunto derecho al funtor a la categoría de un solo objeto,
2. adjunto derecho al funtor diagonal,
3. adjunto derecho al al funtor producto $- \times A$.

Una categoría de este tipo se llama categoría cartesiana cerrada. Son de nuevo las reglas del cálculo lambda para $1, \times, \rightarrow$ cuando leemos cada morfismo como $\vdash$.

Cálculo lambda como adjuntos II

Con los siguientes adjuntos se tiene el cálculo lambda completo

$$\frac{* \longrightarrow *}{0 \xrightarrow{!} C} \qquad \frac{A, B \xrightarrow{f,g} C, C}{A + B \xrightarrow{f+g} C}$$

cuando existen, la categoría se llama bicartesiana cerrada.

Tipos dependientes

Añadir cuantificadores

Si añadimos cuantificadores a la lógica proposicional intuicionista, obtenemos tipos dependientes, tenemos familias de tipos B que pueden depender de un término $a : A$; cada $B(a)$ es un tipo distinto.

- El cuantificador universal viene dado por funciones dependientes escritas como $\prod_{a:A} B(a)$, un elemento de ese tipo es una función que lleva cada $a : A$ a un elemento de $B(a)$.
- El cuantificador existencial viene dado por pares dependientes escritos como $\sum_{a:A} B(a)$, un elemento de ese tipo es un par dado por un $a : A$ y por un $b : B(a)$.

Añadir cuantificadores

Si añadimos cuantificadores a la lógica proposicional intuicionista, obtenemos tipos dependientes, tenemos familias de tipos B que pueden depender de un término $a : A$; cada $B(a)$ es un tipo distinto.

- El cuantificador universal viene dado por funciones dependientes escritas como $\prod_{a:A} B(a)$, un elemento de ese tipo es una función que lleva cada $a : A$ a un elemento de $B(a)$.
- El cuantificador existencial viene dado por pares dependientes escritos como $\sum_{a:A} B(a)$, un elemento de ese tipo es un par dado por un $a : A$ y por un $b : B(a)$.

Pueden interpretarse lógicamente, pero hay que tener en cuenta que el cuantificador existencial requiere saber exactamente qué termino cumple la propiedad.

Ejemplos de cuantificador

Supongamos una familia de tipos $\mathrm{Vect}(A, n)$, un vector con n entradas de tipo A; lo podemos ver como $\mathrm{Vect}(A, n) \equiv A \times \overset{n}{\ldots} \times A$.

Ejemplos de cuantificador

Supongamos una familia de tipos $\mathrm{Vect}(A, n)$, un vector con n entradas de tipo A; lo podemos ver como $\mathrm{Vect}(A, n) \equiv A \times \overset{n}{\ldots} \times A$.

- El tipo $\prod_{n:\mathbb{N}} \mathrm{Vect}(A, n)$ es el tipo de las funciones que a cada natural le asignan un vector de longitud n. Por ejemplo, una función que a n le asigne el vector $(a, \overset{n}{\ldots}, a)$, donde $a : A$.

Ejemplos de cuantificador

Supongamos una familia de tipos $\mathrm{Vect}(A, n)$, un vector con n entradas de tipo A; lo podemos ver como $\mathrm{Vect}(A, n) \equiv A \times \overset{n}{\dots} \times A$.

- El tipo $\prod_{n:\mathbb{N}} \mathrm{Vect}(A, n)$ es el tipo de las funciones que a cada natural le asignan un vector de longitud n. Por ejemplo, una función que a n le asigne el vector $(a, \overset{n}{\dots}, a)$, donde $a : A$.
- El tipo $\sum_{n:\mathbb{N}} \mathrm{Vect}(A, n)$ es el tipo de los vectores de cualquier longitud, etiquetados con su longitud. Por ejemplo, $(3, (a, b, c))$, donde $a, b, c : A$.

Ejemplos de cuantificador

Supongamos una familia de tipos $\mathrm{Vect}(A, n)$, un vector con n entradas de tipo A; lo podemos ver como $\mathrm{Vect}(A, n) \equiv A \times \overset{n}{\ldots} \times A$.

- El tipo $\prod_{n:\mathbb{N}} \mathrm{Vect}(A, n)$ es el tipo de las funciones que a cada natural le asignan un vector de longitud n. Por ejemplo, una función que a n le asigne el vector $(a, \overset{n}{\ldots}, a)$, donde $a : A$.

- El tipo $\sum_{n:\mathbb{N}} \mathrm{Vect}(A, n)$ es el tipo de los vectores de cualquier longitud, etiquetados con su longitud. Por ejemplo, $(3, (a, b, c))$, donde $a, b, c : A$.

Si aceptamos la existencia de un "tipo de todos los tipos", $\mathcal{U}$, tenemos que

$$\mathrm{Vect} : \prod_{A:\mathcal{U}} \prod_{n:\mathbb{N}} \mathcal{U}, \qquad \mathrm{Vect} :\equiv \lambda A.\lambda n.\mathrm{ind}\ A\ (\lambda X.X \times A)\ n$$