Categorías cartesianas

Mario Román

March 9, 2018

Outline

Cálculo lambda simplemente tipado

Algunas categorías útiles

Estructura cartesiana cerrada

Argumentos diagonales

Extensiones

TODO Adjunciones

TODO Categorías cartesianas cerradas

TODO Cálculo lambda

Cálculo lambda simplemente tipado

Cálculo lambda: fragmento negativo

Tomamos un cálculo lambda muy simple con tres constructores de tipos. Un tipo con un solo elemento 1, un producto para cada dos tipos, y el tipo función entre dos tipos cualquiera.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Cálculo lambda: fragmento negativo

Tomamos un cálculo lambda muy simple con tres constructores de tipos. Un tipo con un solo elemento 1, un producto para cada dos tipos, y el tipo función entre dos tipos cualquiera.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Con ciertas constantes y reglas de eliminación (simplificación),

- proyecciones π_1, π_2 cumpliendo $\pi_1\langle a, b\rangle \equiv a$ y $\pi_2\langle a, b\rangle \equiv b$, y
- evaluación de funciones $(\lambda a.b)\ a_0 \equiv b[a_0/a]$.

Y ciertas reglas de unicidad.

Cálculo lambda: fragmento positivo

Además de los tipos anteriores, podemos añadir un tipo vacío 0 y un tipo unión $A + B$.

$$\frac{\Gamma \vdash z : 0}{\Gamma \vdash \mathrm{abort} : C} \qquad \frac{\Gamma, a : A \vdash c : C \qquad \Gamma, b : B \vdash c' : C}{\Gamma, u : A + B \vdash \mathrm{case}\ u\ \mathrm{of}\ c; c' : C}$$

Donde el tipo vacío no necesita reglas de evaluación porque no se puede construir de ninguna forma y el tipo unión tiene como reglas el análisis por casos usual.

Como lenguaje de programación

El cálculo lambda sin tipos es un modelo de computación; el cálculo lambda con tipos es la base de los lenguajes funcionales de programación.

Como lenguaje de programación

El cálculo lambda sin tipos es un modelo de computación; el cálculo lambda con tipos es la base de los lenguajes funcionales de programación.

```haskell
-- Haskell
map (\x -> x+2) [1,2,3] -- [3,4,5]
id = \x -> x            -- Define la identidad
```

Como lenguaje de programación

El cálculo lambda sin tipos es un modelo de computación; el cálculo lambda con tipos es la base de los lenguajes funcionales de programación.

```
-- Haskell
map (\x -> x+2) [1,2,3] -- [3,4,5]
id = \x -> x            -- Define la identidad
```

```
-- Mikrokosmos
swap = \pair.(snd pair, fst pair)
id = \x.x
ifelse false 4 3
```

Ejercicios

Encontrar términos lambda de tipo:

- $(A \times B \to C) \to (A \to B \to C)$
- $(((A \to B) \to A \to A) \to B) \to B$
- $(A + B \to C) \to (A \to C) \times (B \to C)$
- $A \to ((A \to \perp) \to \perp)$

Algunas categorías útiles

Conjuntos

Set, es la categoría de conjuntos y funciones entre conjuntos. Los objetos son conjuntos y los morfismos son funciones entre los conjuntos.

$$\begin{tikzcd} A \arrow[loop above, "\mathrm{id}"] \arrow[r, "f"] \arrow[dr, "g \circ f"'] & B \arrow[d, "g"] \\ & C \end{tikzcd}$$

Conjuntos en pares

Set $\times$ Set, la categoría de pares de conjuntos A, B con pares de funciones entre ellos. Nótese que los pares de funciones son completamente independientes y que la composición se hace por componentes.

$$\begin{tikzcd}
A, A' \arrow[loop above, "\mathrm{id},\mathrm{id}"] \arrow[r, "f{,}f'"] \arrow[dr, "g\circ f{,}g'\circ f'"'] & B, B' \arrow[d, "g{,}g'"] \\
& C, C'
\end{tikzcd}$$

Podría generalizarse a productos arbitrarios sobre Set.

Conjuntos en tiempo discreto

$\mathsf{Set}^{\mathbb{N}}$ es la categoría de los conjuntos bajo un parámetro discreto que puede interpretarse como un parámetro temporal. Los objetos son secuencias de funciones de avance temporal a_i de la forma

$$A_0 \xrightarrow{a_0} A_1 \xrightarrow{a_1} A_2 \xrightarrow{a_2} \cdots .$$

Hay una única función entre $A_n \to A_m$ para cada $n < m$.

Y un morfismo $f \colon A_n \to B_n$ es un diagrama conmutativo en cada instante de tiempo:

$$\begin{tikzcd}
A_0 \arrow[r, "a_0"] \arrow[d, "f_0"] & A_1 \arrow[r, "a_1"] \arrow[d, "f_1"] & A_2 \arrow[r, "a_2"] \arrow[d, "f_2"] & \cdots \\
B_0 \arrow[r, "b_0"] & B_1 \arrow[r, "b_1"] & B_2 \arrow[r, "b_2"] & \cdots
\end{tikzcd}$$

Conjuntos en tiempo continuo

$\mathrm{Set}^{\mathbb{R}}$ es la categoría de los conjuntos bajo un parámetro temporal continuo en $\mathbb{R}$. Los objetos son secuencias temporales $\{A_r\}$ donde hay una única función $A_r \to A_s$ cuando $r < s$.

Un morfismo entre conjuntos temporales debe cumplir

$$\begin{array}{ccc} A_r & \xrightarrow{a_{rs}} & A_s \\ \downarrow{\scriptstyle f_r} & & \downarrow{\scriptstyle f_s} \\ B_r & \xrightarrow{b_{rs}} & B_s \end{array}$$

para cualesquiera $r, s \in \mathbb{R}$. Generalizamos a conjuntos sobre cualquier preorden; estamos escribiendo generalizaciones de la teoría de conjuntos.

Grafos

La categoría de grafos puede derivarse desde la categoría de conjuntos tomando

$$\begin{tikzcd} E \arrow[r, bend left, "s"] \arrow[r, bend right, "t"'] & V \end{tikzcd}$$

Los morfismos son transformaciones naturales entre estructuras de este tipo.

$$\begin{tikzcd} E \arrow[r] \arrow[d, bend right] \arrow[d, bend left] & E' \arrow[d, bend right] \arrow[d, bend left] \\ V \arrow[r] & V' \end{tikzcd}$$

Álgebras de Heyting

Consideramos un conjunto de proposiciones como objetos y las implicaciones entre ellas como morfismos. Dadas dos proposiciones P, Q, tenemos las proposiciones $P \wedge Q$, $P \vee Q$, $P \to Q$, $\bot$, $\top$.

$$
\begin{tikzcd}
P \arrow[loop above] \arrow[r] \arrow[dr] & Q \arrow[d] \\
& R
\end{tikzcd}
$$

Sólo consideramos que existe como mucho un único morfismo entre dos proposiciones cualesquiera. *Otro ejemplo son los abiertos de un espacio topológico.*

Teoría de los naturales

Podemos tener una categoría con objetos $1, \mathbb{N}, \mathbb{N}^2, \mathbb{N}^3, \dots$ y un objeto de valores de verdad Ω (podemos suponer $\Omega = \{0, 1\}$). Consideramos también como objetos productos de estos objetos, como $\mathbb{N}^2 \times \Omega$.

Los morfismos $A_0 \times \cdots \times A_k \to B$ son expresiones en lógica de primer orden y aritméticas que toman variables de tipo $A_0, \dots, A_k$.

Espacios topológicos bajo ciertas condiciones

La categoría Top de todos los espacios topológicos no servirá como categoría cartesiana cerrada. Podemos considerar variantes que sí funcionan correctamente:

- CgTop, espacios topológicos Hausdorff compactamente generados.
- STop, espacios topológicos secuenciales.
- SSet, conjuntos simpliciales. También funcionan los complejos simpliciales y construcciones similares.

Así llegaríamos a una teoría sintética de la continuidad.

Otros ejemplos

- FinSet, la categoría de conjuntos finitos con todas las funciones entre ellos.
- GSet, conjuntos sobre los que actúa un grupo G.
- El *topos efectivo* es una categoría en la que los morfismos son funciones computables. Permite una teoría de la computabilidad sintética.

Estructura cartesiana cerrada

Unidad

Llamamos 1 al tipo unidad, un tipo con un solo término. Bajo cualquier contexto Γ podemos crear el único término de ese tipo, al que llamamos $*$.

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

Unidad

Llamamos 1 al tipo unidad, un tipo con un solo término. Bajo cualquier contexto Γ podemos crear el único término de ese tipo, al que llamamos $*$.

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

En algunas categorías existe un objeto 1. Su propiedad es que desde cualquier objeto Γ, existe un único morfismo hacia él, que llamamos $*$.

$$\frac{}{\Gamma \xrightarrow{*} 1}$$

Unidad: en categorías

- Set: dado cualquier conjunto A existe una única función $A \to \{*\}$, al *conjunto de un solo elemento*.

Unidad: en categorías

- Set: dado cualquier conjunto A existe una única función $A \to \{*\}$, al *conjunto de un solo elemento*.
- En lógica, cualquier proposición implica la proposición verdadera $A \to \top$.

Producto

Dados dos tipos cualesquiera, A y B; tenemos un tipo producto de ambos tipos, $A \times B$.

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

Producto

Dados dos tipos cualesquiera, A y B; tenemos un tipo producto de ambos tipos, $A \times B$.

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

En algunas categorías, dados dos objetos $A \times B$, existe un objeto producto cumpliendo la regla siguiente.

$$\frac{C \xrightarrow{f} A \qquad C \xrightarrow{g} B}{\overline{C \xrightarrow{\langle f,g \rangle} A \times B}}$$

Producto: en categorías

- Set: dados dos conjuntos, A y B, existe su producto cartesiano $A \times B$.
- Graph: puede construirse el grafo producto de dos grafos, teniendo como vértices pares de vértices y aristas entre los pares (a, b) y (a', b') cuando hay una arista entre ellos.
- En lógica, cualesquiera dos proposiciones tienen una conjunción $A \times B$.
- En nuestra teoría aritmética, considerábamos productos formales, $\mathbb{N} \times \Omega$, por ejemplo. Podríamos tener $\mathbb{N} \times \mathbb{N} \cong \mathbb{N}$.

Exponencial

Llamamos $A \to B$ al tipo de las funciones entre dos tipos A y B. Si podemos construir algo de tipo B desde algo de tipo A, tenemos una función lambda.

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

Exponencial

Llamamos $A \to B$ al tipo de las funciones entre dos tipos A y B. Si podemos construir algo de tipo B desde algo de tipo A, tenemos una función lambda.

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

En algunas categorías, hay un objeto B^A que representa los morfismos $A \to B$ en esa categoría.

$$\frac{\Gamma \times A \xrightarrow{b_a} B}{\Gamma \xrightarrow{\lambda a.b_a} B^A}$$

Exponencial: en categorías

- Set: dados dos conjuntos, A y B, existe el conjunto de funciones entre ellos $\hom(A, B)$.
- En lógica, dadas dos proposiciones, existe la proposición implicación entre ellas ($P \to Q$), que es cierta cuando una implica a la otra.

Tenemos categorías cartesianas con tres adjunciones.

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

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

$$\frac{}{C \xrightarrow{\;*\;} 1} \qquad \frac{\Delta\; C \xrightarrow{f,g} A, B}{C \xrightarrow{\langle f,g \rangle} A \times B} \qquad \frac{C \times A \xrightarrow{\;f\;} B}{C \xrightarrow{\;\widetilde{f}\;} B^A}$$

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

$$\frac{\qquad}{C \xrightarrow{\;*\;} 1} \qquad \frac{C, C \xrightarrow{f,g} A, B}{C \xrightarrow{\langle f,g\rangle} A \times B} \qquad \frac{C \times A \xrightarrow{f} B}{C \xrightarrow{\widetilde{f}} B^A}$$

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b\rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

$$\frac{}{C \xrightarrow{\quad * \quad} 1} \qquad \frac{C \xrightarrow{f} A \qquad C \xrightarrow{g} B}{C \xrightarrow{\langle f,g \rangle} A \times B} \qquad \frac{C \times A \xrightarrow{\quad f \quad} B}{C \xrightarrow{\quad \widetilde{f} \quad} B^A}$$

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

$$\frac{}{C \xrightarrow{\ *\ } 1} \qquad \frac{C \xrightarrow{f} A \qquad C \xrightarrow{g} B}{C \xrightarrow{\langle f,g \rangle} A \times B} \qquad \frac{C \times A \xrightarrow{f_a} B}{C \xrightarrow{\lambda a.f_a} B^A}$$

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

$$\frac{}{C \xrightarrow{\ *\ } 1} \qquad \frac{C \xrightarrow{f} A \qquad C \xrightarrow{g} B}{C \xrightarrow{\langle f,g\rangle} A \times B} \qquad \frac{C \times A \xrightarrow{f_a} B}{C \xrightarrow{\lambda a.f_a} B^A}$$

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b\rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

$$\frac{}{\Gamma \xrightarrow{\;*\;} 1} \qquad \frac{\Gamma \xrightarrow{a} A \qquad \Gamma \xrightarrow{b} B}{\Gamma \xrightarrow{\langle a,b \rangle} A \times B} \qquad \frac{\Gamma \times A \xrightarrow{b_a} B}{\Gamma \xrightarrow{\lambda a.b_a} B^A}$$

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Resumen

Cada expresión en cálculo lambda puede interpretarse en cualquier categoría cartesiana cerrada.

Resumen

Cada expresión en cálculo lambda puede interpretarse en cualquier categoría cartesiana cerrada.

1. Los morfismos $f : \Gamma \to A$ son los términos de tipo A sobre el contexto Γ.

Resumen

Cada expresión en cálculo lambda puede interpretarse en cualquier categoría cartesiana cerrada.

1. Los morfismos $f : \Gamma \to A$ son los términos de tipo A sobre el contexto Γ.
2. $\Gamma \times A$ es el contexto Γ más un elemento de tipo A, y 1 es el contexto vacío.

Resumen

Cada expresión en cálculo lambda puede interpretarse en cualquier categoría cartesiana cerrada.

1. Los morfismos $f : \Gamma \to A$ son los términos de tipo A sobre el contexto Γ.
2. $\Gamma \times A$ es el contexto Γ más un elemento de tipo A, y 1 es el contexto vacío.
3. Los términos sobre un contexto vacío (programas) que devuelven algo de tipo A son morfismos $* \to A$ (elementos de A).

Resumen

Cada expresión en cálculo lambda puede interpretarse en cualquier categoría cartesiana cerrada.

1. Los morfismos $f : \Gamma \to A$ son los términos de tipo A sobre el contexto Γ.
2. $\Gamma \times A$ es el contexto Γ más un elemento de tipo A, y 1 es el contexto vacío.
3. Los términos sobre un contexto vacío (programas) que devuelven algo de tipo A son morfismos $* \to A$ (elementos de A).

Técnicamente, para cada teoría de cálculo lambda existe una categoría sintáctica y para cada categoría cartesiana existe un lenguaje de cálculo lambda interno. Salvo equivalencias, se comportan como inversos.

Ejemplo: swap

Un programa en cálculo lambda

```
swap :: A x B -> B x A
swap = \pair.(snd pair, fst pair)
```

se traduce a una construcción en categorías.

$$\begin{tikzcd}
 & & A \\
A \times B \arrow[r, dashed, "(\pi_2,\pi_1)"] \arrow[urr, bend left, "\pi_1"] \arrow[drr, bend right, "\pi_2"'] & B \times A \arrow[ur, "\pi_2"'] \arrow[dr, "\pi_1"] & \\
 & & B
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[d, "\lambda p.(\pi_2 p, \pi_1 p)"] \\
\hom(A \times B, B \times A)
\end{tikzcd}$$

Argumentos diagonales

Teorema del punto fijo (W. Lawvere)

En una categoría cartesiana cerrada, decimos que $f : A \to B$ es puntualmente sobreyectiva si, para cada $b : B$ existe un $a : A$ tal que $f(a) = b$.

Teorema del punto fijo (W. Lawvere)

En una categoría cartesiana cerrada, decimos que $f : A \to B$ es puntualmente sobreyectiva si, para cada $b : B$ existe un $a : A$ tal que $f(a) = b$.

Teorema del punto fijo

Si existe $d : A \to B^A$ puntualmente sobreyectiva, entonces cada $f : B \to B$ tiene un punto fijo, un $b : B$ tal que $f(b) = b$.

Teorema del punto fijo (W. Lawvere)

En una categoría cartesiana cerrada, decimos que $f\colon A \to B$ es puntualmente sobreyectiva si, para cada $b : B$ existe un $a : A$ tal que $f(a) = b$.

Teorema del punto fijo

Si existe $d\colon A \to B^A$ puntualmente sobreyectiva, entonces cada $f\colon B \to B$ tiene un punto fijo, un $b : B$ tal que $f(b) = b$.

Demostración. Como d es puntualmente sobreyectiva, existe $x : A$ tal que $d(x) \equiv \lambda a.f(d(a,a))$; pero entonces se tiene el punto fijo

$$d(x,x) \equiv (\lambda a.f(d(a,a))) \equiv f(d(x,x)).$$

Corolario: Teorema de Cantor

Teorema del punto fijo

Si existe $d\colon A \to B^A$ puntualmente sobreyectiva, entonces cada $f\colon B \to B$ tiene un punto fijo, un $b : B$ tal que $f(b) = b$.

En Set, sea $f\colon 2 \to 2$ la permutación no trivial del conjunto de dos elementos. Nótese que f no puede tener puntos fijos, así que no puede existir $d\colon A \to 2^A$ puntualmente sobreyectiva.

Así, A no es isomorfo a $\mathcal{P}(A)$. En particular, $\mathcal{P}(\mathbb{N})$ no es numerable.

Corolario: Teorema de Russell

Teorema del punto fijo

Si existe $d\colon A \to B^A$ puntualmente sobreyectiva, entonces cada $f\colon B \to B$ tiene un punto fijo, un $b : B$ tal que $f(b) = b$.

Supongamos una clase de todos los conjuntos Sets sobre la que podemos expresar propiedades $P\colon \mathrm{Sets} \to 2$. Si asumimos que toda propiedad en la clase de los conjuntos corresponde a un conjunto por comprensión $\{y \in \mathrm{Sets} \mid P(y)\}$. Entonces habría una función sobreyectiva en puntos $\mathrm{Sets} \to 2^{\mathrm{Sets}}$ y toda función $f\colon 2 \to 2$ tendría un punto fijo.

Corolario: Teorema de Gödel

Teorema del punto fijo

Si existe $d\colon A \to B^A$ puntualmente sobreyectiva, entonces cada $f\colon B \to B$ tiene un punto fijo, un $b : B$ tal que $f(b) = b$.

> *The original aim of this article was to demystify the incompleteness theorem of Godel and the truth-definition theory of Tarski by showing that both are consequences of some very simple algebra in the cartesian-closed setting.*
> *– W. Lawvere, Diagonal arguments and cartesian closed categories (2006).*

Corolario: Teorema de Gödel

Teorema del punto fijo

Si existe $d \colon A \to B^A$ puntualmente sobreyectiva, entonces cada $f \colon B \to B$ tiene un punto fijo, un $b : B$ tal que $f(b) = b$.

Tenemos una categoría cartesiana cerrada generada por los objetos $1, \mathbb{N}, \mathbb{N}^2, \mathbb{N}^3 \dots$ y un objeto Ω que representa valores de verdad. Los morfismos $A \to B$ son expresiones sintácticas en una teoría que toman una variable de tipo A y devuelven algo de tipo B. Es

- consistente si hay un $\mathrm{not} \colon \Omega \to \Omega$ sin puntos fijos. Y tiene
- satisfacibilidad si existe $\mathrm{sat} \colon \mathbb{N} \times \mathbb{N} \to \Omega$, tal que cada predicado $\varphi \colon \mathbb{N} \to \Omega$ sea de la forma $\mathrm{sat}(n)$ para algún n, al que llamamos *número de Gödel* del predicado.

Cualquier teoría suficientemente fuerte como para que satisfacibilidad sea definible es inconsistente.

Extensiones

Bicartesianas cerradas

Incluimos sumas y un objeto final. Dicho de otra forma, incluimos un tipo unión y un tipo vacío.

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

System T

Incluimos los números naturales.

$$\begin{tikzcd} 1+\mathbb{N} \arrow[r] \arrow[d, "\langle 0,\mathrm{succ}\rangle"'] & 1+C \arrow[d, "\langle x,f\rangle"] \\ \mathbb{N} \arrow[r, "\mathrm{ind}\ x\ f"] & C \end{tikzcd}$$

$$\mathrm{ind}\colon C \to (C \to C) \to (\mathbb{N} \to C)$$
$$\mathrm{ind}\ x\ f\ 0 \equiv x$$
$$\mathrm{ind}\ x\ f\ (S\ n) \equiv f\ (\mathrm{ind}\ x\ f\ n)$$

Álgebras iniciales: la suma

Definiendo la suma.

$$\begin{tikzcd} 1+\mathbb{N} \arrow[r] \arrow[d, "\langle 0,succ\rangle"'] & 1+\mathrm{hom}(\mathbb{N},\mathbb{N}) \arrow[d, "\langle \mathrm{id},\mathrm{succ}\circ -\rangle"] \\ \mathbb{N} \arrow[r, "+"] & \mathrm{hom}(\mathbb{N},\mathbb{N}) \end{tikzcd}$$

Álgebras iniciales: listas

$$\begin{tikzcd} 1 + A \times \mathrm{List}(A) \arrow[r] \arrow[d, "\langle \mathrm{nil}, \mathrm{cons} \rangle"'] & 1 + A \times X \arrow[d, "\langle x, \bullet \rangle"] \\ \mathrm{List}(A) \arrow[r, "\varphi"] & X \end{tikzcd}$$

Martin-Löf

Incluimos funciones y pares dependientes. Podemos usar cuantificadores universales y existenciales.

$$\begin{tikzcd}
\Gamma \arrow[r, "b"] \arrow[rr, bend left, "\langle a,b \rangle"] \arrow[dr, equal] & \Gamma, y : B(a) \arrow[r] \arrow[d] & \Gamma, x : A, y : B \arrow[d, "\pi_B"] \arrow[dd, bend left, "\pi_\Sigma"] \\
& \Gamma \arrow[r, "a"] \arrow[dr, equal] & \Gamma, x : A \arrow[d, "\pi_A"] \\
& & \Gamma
\end{tikzcd}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B[a/x]}{\Gamma \vdash \langle a, b \rangle : \sum_{x:A} B}$$

La igualdad se puede obtener en este contexto relacionada a un morfismo diagonal para cada tipo.

$$\begin{tikzcd} \triangle^* C \arrow[r] \arrow[d] & C \arrow[d, "g"] \\ A \arrow[r, "\triangle"] \arrow[u, bend left, dashed, "\widetilde{k}"] & A \times A \end{tikzcd} \qquad \begin{tikzcd} A \arrow[rr, dashed, "k"] \arrow[dr, "\triangle"'] & & C \arrow[dl, "g"] \\ & A \times A & \end{tikzcd}$$

$$\frac{\begin{array}{cc} \Gamma \vdash a : A & \Gamma \vdash b : A \\ \Gamma, x : A \vdash c : C(x,x) & \Gamma \vdash p : a = b \end{array}}{\Gamma \vdash J_C(c,p) : C(a,b)}$$

TODO Adjunciones

Definición de adjunción

Sean dos funtores $F\colon \mathcal{C} \to \mathcal{D}$ y $G\colon \mathcal{D} \to \mathcal{C}$. Una adjunción entre ellos, $F \dashv G$, viene dada por una biyección

$$\varphi\colon \hom(FX, Y) \cong \hom(X, GY)$$

para cada $X \in \mathcal{C}$, $Y \in \mathcal{D}$.

Definición de adjunción

Sean dos funtores $F\colon \mathcal{C} \to \mathcal{D}$ y $G\colon \mathcal{D} \to \mathcal{C}$. Una adjunción entre ellos, $F \dashv G$, viene dada por una biyección

$$\varphi\colon \hom(FX, Y) \cong \hom(X, GY)$$

para cada $X \in \mathcal{C}$, $Y \in \mathcal{D}$. La dibujamos como.

$$\frac{FX \xrightarrow{f} Y}{X \xrightarrow{\varphi(f)} GY}$$

Definición de adjunción

Sean dos funtores $F\colon \mathcal{C} \to \mathcal{D}$ y $G\colon \mathcal{D} \to \mathcal{C}$. Una adjunción entre ellos, $F \dashv G$, viene dada por una biyección

$$\varphi\colon \hom(FX, Y) \cong \hom(X, GY)$$

para cada $X \in \mathcal{C}$, $Y \in \mathcal{D}$. La dibujamos como.

$$\frac{FX \xrightarrow{f} Y}{X \xrightarrow{\varphi(f)} GY}$$

Además se exigen ciertas condiciones de *naturalidad*.

Condiciones de naturalidad

Condiciones de naturalidad

$$\varphi(f \circ Fh) = \varphi(f) \circ h \qquad \varphi(k \circ f) = Gk \circ \varphi(f)$$

Condiciones de naturalidad

Condiciones de naturalidad

$$\varphi(f \circ Fh) = \varphi(f) \circ h \qquad \varphi(k \circ f) = Gk \circ \varphi(f)$$

Se dibujan como:

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

Condiciones de naturalidad

Condiciones de naturalidad

$$\varphi(f \circ Fh) = \varphi(f) \circ h \qquad \varphi(k \circ f) = Gk \circ \varphi(f)$$

Se dibujan como:

$$\frac{\begin{tikzcd} FX' \arrow[r, "Fh"] \arrow[rr, bend left, "f\circ Fh"] & FX \arrow[r, "f"] & Y \end{tikzcd}}{\begin{tikzcd} X' \arrow[r, "h"'] \arrow[rr, bend right, "\varphi(f)\circ h"'] & X \arrow[r, "\varphi(f)"'] & GY \end{tikzcd}} \qquad \frac{\begin{tikzcd} FX \arrow[r, "f"] \arrow[rr, bend left, "k\circ f"] & Y \arrow[r, "k"] & Y' \end{tikzcd}}{\begin{tikzcd} X \arrow[r, "\varphi(f)"'] \arrow[rr, bend right, "Gk\circ\varphi(f)"'] & GY \arrow[r, "Gk"'] & GY' \end{tikzcd}}$$

Unicidad: con esta definición, se demuestra que si $F \dashv G$ y $F' \dashv G$, entonces $F \cong F'$. Los adjuntos son esencialmente únicos.

Ejemplo: grupos libres

Entre la categoría de los conjuntos y la categoría de los grupos, el funtor grupo libre $\langle - \rangle \colon \mathsf{Set} \to \mathsf{Grp}$ y el funtor que olvida la estructura de grupo $\mathcal{U} \colon \mathsf{Grp} \to \mathsf{Set}$.

Ejemplo: grupos libres

Entre la categoría de los conjuntos y la categoría de los grupos, el funtor grupo libre $\langle - \rangle \colon \mathsf{Set} \to \mathsf{Grp}$ y el funtor que olvida la estructura de grupo $\mathcal{U} \colon \mathsf{Grp} \to \mathsf{Set}$.

$$\frac{\langle X \rangle \xrightarrow{f} Y}{\overline{\quad X \xrightarrow{\varphi(f)} \mathcal{U}Y \quad}}$$

Ejemplo: grupos libres

Entre la categoría de los conjuntos y la categoría de los grupos, el funtor grupo libre $\langle - \rangle \colon \mathsf{Set} \to \mathsf{Grp}$ y el funtor que olvida la estructura de grupo $\mathcal{U} \colon \mathsf{Grp} \to \mathsf{Set}$.

$$\frac{\langle X \rangle \xrightarrow{\ f\ } Y}{X \xrightarrow{\varphi(f)} \mathcal{U}Y}$$

Los homomorfismos de grupos del grupo libre a otro grupo están en biyección con las funciones arbitrarias entre los conjuntos subyacentes.

Ejemplo: grupos libres

Entre la categoría de los conjuntos y la categoría de los grupos, el funtor grupo libre $\langle - \rangle \colon \mathsf{Set} \to \mathsf{Grp}$ y el funtor que olvida la estructura de grupo $\mathcal{U} \colon \mathsf{Grp} \to \mathsf{Set}$.

$$\frac{\langle X \rangle \xrightarrow{f} Y}{X \xrightarrow{\varphi(f)} \mathcal{U}Y}$$

Los homomorfismos de grupos del grupo libre a otro grupo están en biyección con las funciones arbitrarias entre los conjuntos subyacentes.

Condiciones de naturalidad

$$\varphi(f \circ Fh) = \varphi(f) \circ h \qquad \varphi(k \circ f) = Gk \circ \varphi(f)$$

Ejemplo: topología discreta y topología trivial

Sea el funtor $\mathcal{U}\colon \mathsf{Top} \to \mathsf{Set}$ que envía cada espacio topológico a su conjunto subyacente. Tiene un adjunto a la derecha y otro a la izquierda.

Ejemplo: topología discreta y topología trivial

Sea el funtor $\mathcal{U}\colon \mathrm{Top} \to \mathrm{Set}$ que envía cada espacio topológico a su conjunto subyacente. Tiene un adjunto a la derecha y otro a la izquierda.

$$\frac{\tau_d X \xrightarrow{\quad f \quad} Y}{X \xrightarrow{\varphi(f)} \mathcal{U} Y} \qquad \frac{\mathcal{U} X \xrightarrow{\quad f \quad} Y}{X \xrightarrow{\varphi(f)} \tau_t Y}$$

Que son la topología discreta y la topología trivial. Hay una cadena $\tau_d \dashv \mathcal{U} \dashv \tau_t$.

Ejemplo: topología discreta y topología trivial

Sea el funtor $\mathcal{U}\colon \mathsf{Top} \to \mathsf{Set}$ que envía cada espacio topológico a su conjunto subyacente. Tiene un adjunto a la derecha y otro a la izquierda.

$$\frac{\tau_d X \xrightarrow{\ f\ } Y}{\overline{X \xrightarrow{\varphi(f)} \mathcal{U}Y}} \qquad \frac{\mathcal{U}X \xrightarrow{\ f\ } Y}{\overline{X \xrightarrow{\varphi(f)} \tau_t Y}}$$

Que son la topología discreta y la topología trivial. Hay una cadena $\tau_d \dashv \mathcal{U} \dashv \tau_t$.

Condiciones de naturalidad

$$\varphi(f \circ Fh) = \varphi(f) \circ h \qquad \varphi(k \circ f) = Gk \circ \varphi(f)$$

Ejemplo: producto

Consideramos el funtor diagonal $\Delta \colon \mathcal{C} \to \mathcal{C} \times \mathcal{C}$ que lleva $f \colon A \to B$ a $\langle f, f \rangle \colon \langle A, A \rangle \to \langle B, B \rangle$. El funtor tiene una adjunción a la derecha.

$$\frac{\Delta\ C \xrightarrow{f,g} \langle A, B \rangle}{C \xrightarrow{\langle f,g \rangle} A \times B}$$

Condiciones de naturalidad

$$\varphi(f \circ Fh) = \varphi(f) \circ h \qquad \varphi(k \circ f) = Gk \circ \varphi(f)$$

Ejemplo: producto

Consideramos el funtor diagonal $\Delta \colon \mathcal{C} \to \mathcal{C} \times \mathcal{C}$ que lleva $f \colon A \to B$ a $\langle f, f \rangle \colon \langle A, A \rangle \to \langle B, B \rangle$. El funtor tiene una adjunción a la derecha.

$$\frac{\langle C, C \rangle \xrightarrow{f,g} \langle A, B \rangle}{C \xrightarrow{\langle f,g \rangle} A \times B}$$

Condiciones de naturalidad

$$\varphi(f \circ Fh) = \varphi(f) \circ h \qquad \varphi(k \circ f) = Gk \circ \varphi(f)$$

Ejemplo: producto

Consideramos el funtor diagonal $\Delta\colon \mathcal{C} \to \mathcal{C} \times \mathcal{C}$ que lleva $f\colon A \to B$ a $\langle f, f\rangle\colon \langle A, A\rangle \to \langle B, B\rangle$. El funtor tiene una adjunción a la derecha.

$$\frac{C \xrightarrow{f} A \qquad\qquad C \xrightarrow{g} B}{C \xrightarrow{\langle f,g\rangle} A \times B}$$

Condiciones de naturalidad

$$\varphi(f \circ Fh) = \varphi(f) \circ h \qquad \varphi(k \circ f) = Gk \circ \varphi(f)$$

Ejemplo: suma

El mismo funtor diagonal $\Delta \colon \mathcal{C} \to \mathcal{C} \times \mathcal{C}$ tiene adjunción a la izquierda.

$$\frac{A + B \xrightarrow{f+g} C}{\langle A, B \rangle \xrightarrow{f,g} \Delta\, C}$$

Hay una cadena $+ \dashv \Delta \dashv \times$.

Condiciones de naturalidad

$$\varphi(f \circ Fh) = \varphi(f) \circ h \qquad \varphi(k \circ f) = Gk \circ \varphi(f)$$

Ejemplo: suma

El mismo funtor diagonal $\Delta\colon \mathcal{C} \to \mathcal{C} \times \mathcal{C}$ tiene adjunción a la izquierda.

$$\frac{A + B \xrightarrow{f+g} C}{\langle A, B\rangle \xrightarrow{f,g} \langle C, C\rangle}$$

Hay una cadena $+ \dashv \Delta \dashv \times$.

Condiciones de naturalidad

$$\varphi(f \circ Fh) = \varphi(f) \circ h \qquad \varphi(k \circ f) = Gk \circ \varphi(f)$$

El mismo funtor diagonal $\Delta \colon \mathcal{C} \to \mathcal{C} \times \mathcal{C}$ tiene adjunción a la izquierda.

$$\frac{A + B \xrightarrow{f+g} C}{\overline{A \xrightarrow{f} C \qquad B \xrightarrow{g} C}}$$

Hay una cadena $+ \dashv \Delta \dashv \times$.

Condiciones de naturalidad

$$\varphi(f \circ Fh) = \varphi(f) \circ h \qquad \varphi(k \circ f) = Gk \circ \varphi(f)$$

TODO Categorías cartesianas cerradas

Categorías cartesianas cerradas

Vamos a ver tres adjuntos que pueden existir en una categoría cualquiera. Las categorías cartesianas cerradas son las que tienen esos tres adjuntos.

Unidad

El objeto 1 de una categoría es un objeto tal que desde cualquier otro objeto sólo existe una única flecha hacia él.

$$\frac{* \longrightarrow *}{C \xrightarrow{!} 1}$$

La categoría superior es una categoría trivial con un solo objeto. Hacia ella, tenemos un funtor trivial $*$. Desde ella, tenemos un funtor que elige un objeto 1 en la categoría.

Productos

Las flechas al producto de dos objetos están unívocamente determinadas por flechas a cada uno de los objetos

$$\frac{\Delta\ C \xrightarrow{f,g} A, B}{C \xrightarrow{\langle f,g\rangle} A \times B}$$

La categoría superior tiene como objetos pares de objetos y como morfismos pares de morfismos. Hacia ella, tenemos el funtor diagonal. Desde ella, tenemos el funtor producto de dos objetos.

Productos

Las flechas al producto de dos objetos están unívocamente determinadas por flechas a cada uno de los objetos

$$\frac{C, C \xrightarrow{f,g} A, B}{C \xrightarrow{\langle f,g \rangle} A \times B}$$

La categoría superior tiene como objetos pares de objetos y como morfismos pares de morfismos. Hacia ella, tenemos el funtor diagonal. Desde ella, tenemos el funtor producto de dos objetos.

Productos

Las flechas al producto de dos objetos están unívocamente determinadas por flechas a cada uno de los objetos

$$\frac{C \xrightarrow{f} A \qquad C \xrightarrow{g} B}{C \xrightarrow{\langle f, g \rangle} A \times B}$$

La categoría superior tiene como objetos pares de objetos y como morfismos pares de morfismos. Hacia ella, tenemos el funtor diagonal. Desde ella, tenemos el funtor producto de dos objetos.

Exponenciales

Tener una flecha desde el producto por un objeto A hacia otro B es tener una flecha a la exponencial B^A.

$$\frac{C \times A \xrightarrow{f} B}{C \xrightarrow{\widetilde{f}} B^{A}}$$

Ambas categorías son la misma y al morfismo $\widetilde{f}$ se le llama *traspuesta* de f.

TODO Cálculo lambda

Cálculo lambda simplemente tipado

Tomamos un cálculo lambda muy simple con tres constructores de tipos. Un tipo con un solo elemento 1, un producto para cada dos tipos, y el tipo función entre dos tipos cualquiera.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Cálculo lambda simplemente tipado

Tomamos un cálculo lambda muy simple con tres constructores de tipos. Un tipo con un solo elemento 1, un producto para cada dos tipos, y el tipo función entre dos tipos cualquiera.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Con ciertas constantes y reglas de eliminación (simplificación),

- proyecciones π_1, π_2 cumpliendo $\pi_1\langle a, b \rangle \equiv a$ y $\pi_2\langle a, b \rangle \equiv b$, y
- evaluación de funciones $(\lambda a.b)\ a_0 \equiv b[a_0/a]$.

Y ciertas reglas de unicidad.

Como lenguaje de programación

El cálculo lambda sin tipos es un modelo de computación; el cálculo lambda con tipos es la base de los lenguajes funcionales de programación.

Como lenguaje de programación

El cálculo lambda sin tipos es un modelo de computación; el cálculo lambda con tipos es la base de los lenguajes funcionales de programación.

```haskell
-- Haskell
map (\x -> x+2) [1,2,3] -- [3,4,5]
id = \x -> x            -- Define la identidad
```

Como lenguaje de programación

El cálculo lambda sin tipos es un modelo de computación; el cálculo lambda con tipos es la base de los lenguajes funcionales de programación.

```
-- Haskell
map (\x -> x+2) [1,2,3] -- [3,4,5]
id = \x -> x            -- Define la identidad
```

```
-- Mikrokosmos
swap = \pair.(snd pair, fst pair)
id = \x.x
ifelse false 4 3
```

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

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

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

$$\frac{}{C \xrightarrow{\ *\ } 1} \qquad \frac{\Delta\, C \xrightarrow{f,g} A, B}{C \xrightarrow{\langle f,g \rangle} A \times B} \qquad \frac{C \times A \xrightarrow{f} B}{C \xrightarrow{\widetilde{f}} B^A}$$

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

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

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

$$\frac{}{\overline{C \xrightarrow{\ *\ } 1}} \qquad \frac{C \xrightarrow{f} A \qquad C \xrightarrow{g} B}{\overline{C \xrightarrow{\langle f,g\rangle} A \times B}} \qquad \frac{C \times A \xrightarrow{f} B}{\overline{C \xrightarrow{\widetilde{f}} B^A}}$$

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b\rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

$$\frac{}{C \xrightarrow{\ *\ } 1} \qquad \frac{C \xrightarrow{f} A \qquad C \xrightarrow{g} B}{C \xrightarrow{\langle f,g \rangle} A \times B} \qquad \frac{C \times A \xrightarrow{f_a} B}{C \xrightarrow{\lambda a. f_a} B^A}$$

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

$$\frac{}{C \xrightarrow{\quad * \quad} 1} \qquad \frac{C \xrightarrow{f} A \qquad C \xrightarrow{g} B}{C \xrightarrow{\langle f,g \rangle} A \times B} \qquad \frac{C \times A \xrightarrow{\quad f_a \quad} B}{C \xrightarrow{\lambda a.f_a} B^A}$$

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \times B} \qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Traducción entre categorías y cálculo lambda

Tenemos categorías cartesianas con tres adjunciones.

$$\frac{}{\Gamma \xrightarrow{\;*\;} 1}\qquad \frac{\Gamma \xrightarrow{a} A \qquad \Gamma \xrightarrow{b} B}{\Gamma \xrightarrow{\langle a,b\rangle} A \times B}\qquad \frac{\Gamma \times A \xrightarrow{b_a} B}{\Gamma \xrightarrow{\lambda a.b_a} B^A}$$

Y un cálculo lambda con tres reglas.

$$\frac{}{\Gamma \vdash * : 1}\qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b\rangle : A \times B}\qquad \frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B}$$

Reescribimos para comprobar que son similares.

Resumen

Cada expresión en cálculo lambda puede interpretarse en cualquier categoría cartesiana cerrada.

Resumen

Cada expresión en cálculo lambda puede interpretarse en cualquier categoría cartesiana cerrada.

1. Los morfismos $f : \Gamma \to A$ son los términos de tipo A sobre el contexto Γ.

Resumen

Cada expresión en cálculo lambda puede interpretarse en cualquier categoría cartesiana cerrada.

1. Los morfismos $f : \Gamma \to A$ son los términos de tipo A sobre el contexto Γ.
2. $\Gamma \times A$ es el contexto Γ más un elemento de tipo A, y 1 es el contexto vacío.

Resumen

Cada expresión en cálculo lambda puede interpretarse en cualquier categoría cartesiana cerrada.

1. Los morfismos $f : \Gamma \to A$ son los términos de tipo A sobre el contexto Γ.
2. $\Gamma \times A$ es el contexto Γ más un elemento de tipo A, y 1 es el contexto vacío.
3. Los términos sobre un contexto vacío (programas) que devuelven algo de tipo A son morfismos $* \to A$ (elementos de A).

Ejemplo: swap

Un programa en cálculo lambda

```
swap :: A x B -> B x A
swap = \pair.(snd pair, fst pair)
```

se traduce a una construcción en categorías.

$$
\begin{tikzcd}
 & & A \\
A \times B \arrow[r, dashed, "(\pi_2, \pi_1)"] \arrow[urr, bend left, "\pi_1"] \arrow[drr, bend right, "\pi_2"'] & B \times A \arrow[ur, "\pi_2"'] \arrow[dr, "\pi_1"] & \\
 & & B
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[d, "\lambda p.(\pi_2 p, \pi_1 p)"] \\
\hom(A \times B, B \times A)
\end{tikzcd}
$$