Lambda cálculo modulo isomorfismo de tipos

Alejandro Díaz-Caro
Departamento de Ciencia y Tecnología
Universidad Nacional de Quilmes

Trabajo en conjunto con **Gilles Dowek**
INRIA Paris-Rocquencourt

XII Jornadas de Ciencias de la Computación
15, 16 y 17 de Octubre de 2014 - Rosario, Santa Fe

Isomorfismos de tipos

Definición

$$A \equiv B \quad \Leftrightarrow \quad \exists \; \left\{ \begin{array}{l} \mathbf{prog}_1 : A \Rightarrow B \\ \mathbf{prog}_2 : B \Rightarrow A \end{array} \right\} \;\Big/\; \left\{ \begin{array}{l} \mathbf{prog}_2 \circ \mathbf{prog}_1 = Id_A \\ \mathbf{prof}_1 \circ \mathbf{prog}_2 = Id_B \end{array} \right\}$$

Isomorfismos de tipos

Definición

$$A \equiv B \quad \Leftrightarrow \quad \exists \quad \left\{ \begin{array}{l} \mathbf{prog}_1 : A \Rightarrow B \\ \mathbf{prog}_2 : B \Rightarrow A \end{array} \right\} \quad / \quad \left\{ \begin{array}{l} \mathbf{prog}_2 \circ \mathbf{prog}_1 = Id_A \\ \mathbf{prof}_1 \circ \mathbf{prog}_2 = Id_B \end{array} \right\}$$

Ejemplo

$$(A \wedge B) \equiv (B \wedge A)$$

$\mathbf{swap} : (A \wedge B) \Rightarrow (B \wedge A)$ $\mathbf{swap}\ \langle x, y \rangle = \langle y, x \rangle$	$\mathbf{swap}' : (B \wedge A) \Rightarrow (A \wedge B)$ $\mathbf{swap}'\ \langle y, x \rangle = \langle x, y \rangle$

$$\mathbf{swap}'\ \mathbf{swap}\ \langle \mathbf{a}, \mathbf{b} \rangle = \langle \mathbf{a}, \mathbf{b} \rangle \quad \text{y} \quad \mathbf{swap}\ \mathbf{swap}'\ \langle \mathbf{b}, \mathbf{a} \rangle = \langle \mathbf{b}, \mathbf{a} \rangle$$

Isomorfismos de tipos

Isomorfismos conocidos

Si sólo tenemos $\Rightarrow$ y $\wedge$ (funciones y pares)

- $(A \wedge B) \equiv (B \wedge A)$
- $((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$
- $(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$
- $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$

[Bruce, Di Cosmo, Longo
MSCS 2(2), 231–247, 1992]

Isomorfismos de tipos

Isomorfismos conocidos

Si sólo tenemos $\Rightarrow$ y $\wedge$ (funciones y pares)

[Bruce, Di Cosmo, Longo MSCS 2(2), 231–247, 1992]

- $(A \wedge B) \equiv (B \wedge A)$
- $((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$
- $(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$
- $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$

Isomorfismos de tipos

Isomorfismos conocidos

Si sólo tenemos $\Rightarrow$ y $\wedge$ (funciones y pares)

[Bruce, Di Cosmo, Longo MSCS 2(2), 231–247, 1992]

- $(A \wedge B) \equiv (B \wedge A)$
- $((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$
- $(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$
- $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$

$$\begin{array}{c} \textbf{assoc} : ((A \wedge B) \wedge C) \Rightarrow (A \wedge (B \wedge C)) \\ \textbf{assoc}\ \langle x, y \rangle = \langle \mathit{fst}\ x, \langle \mathit{snd}\ x, y \rangle\rangle \end{array}$$

$$\begin{array}{c} \textbf{assoc}' : (A \wedge (B \wedge C)) \Rightarrow ((A \wedge B) \wedge C) \\ \textbf{assoc}'\ \langle x, y \rangle = \langle\langle x, \mathit{fst}\ y \rangle, \mathit{snd}\ y \rangle \end{array}$$

$$\textbf{assoc}'\ \textbf{assoc}\ \langle\langle \mathbf{a}, \mathbf{b} \rangle, \mathbf{c} \rangle = \langle\langle \mathbf{a}, \mathbf{b} \rangle, \mathbf{c} \rangle$$

$$\textbf{assoc}\ \textbf{assoc}'\ \langle \mathbf{a}, \langle \mathbf{b}, \mathbf{c} \rangle\rangle = \langle \mathbf{a}, \langle \mathbf{b}, \mathbf{c} \rangle\rangle$$

Isomorfismos de tipos

Isomorfismos conocidos

Si sólo tenemos $\Rightarrow$ y $\wedge$ (funciones y pares)

[Bruce, Di Cosmo, Longo MSCS 2(2), 231–247, 1992]

- $(A \wedge B) \equiv (B \wedge A)$
- $((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$
- $(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$
- $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$

Isomorfismos de tipos

Isomorfismos conocidos

Si sólo tenemos $\Rightarrow$ y $\wedge$ (funciones y pares)

[Bruce, Di Cosmo, Longo MSCS 2(2), 231–247, 1992]

- $(A \wedge B) \equiv (B \wedge A)$
- $((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$
- $(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$
- $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$

$$\boxed{\begin{array}{c}\textbf{curry} : ((A \wedge B) \Rightarrow C) \Rightarrow A \Rightarrow B \Rightarrow C \\ \textbf{curry}\ f\ x\ y = f\ \langle x, y\rangle\end{array}}$$

$$\boxed{\begin{array}{c}\textbf{uncurry} : (A \Rightarrow B \Rightarrow C) \Rightarrow (A \wedge B) \Rightarrow C \\ \textbf{uncurry}\ g\ x = g\ (\mathit{fst}\ x)\ (\mathit{snd}\ x)\end{array}}$$

uncurry curry f = f y **curry uncurry g = g**

Isomorfismos de tipos

Isomorfismos conocidos

Si sólo tenemos $\Rightarrow$ y $\wedge$ (funciones y pares)

[Bruce, Di Cosmo, Longo MSCS 2(2), 231–247, 1992]

- $(A \wedge B) \equiv (B \wedge A)$
- $((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$
- $(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$
- $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$

Isomorfismos de tipos

Isomorfismos conocidos

Si sólo tenemos $\Rightarrow$ y $\wedge$ (funciones y pares)

[Bruce, Di Cosmo, Longo MSCS 2(2), 231–247, 1992]

- $(A \wedge B) \equiv (B \wedge A)$
- $((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$
- $(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$
- $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$

$$\begin{array}{l}\textbf{parf} : (A \Rightarrow (B \wedge C)) \Rightarrow ((A \Rightarrow B) \wedge (A \Rightarrow C)) \\ \quad \textbf{parf}\ f = \textbf{let}\ g\ x = \ \mathit{fst}\ (f\ x)\ \textbf{in} \\ \qquad\qquad\quad \textbf{let}\ h\ x = \ \mathit{snd}\ (f\ x)\ \textbf{in} \quad \langle g, h\rangle\end{array}$$

$$\begin{array}{c}\textbf{fpar} : ((A \Rightarrow B) \wedge (A \Rightarrow C)) \Rightarrow A \Rightarrow (B \wedge C) \\ \textbf{fpar}\ f\ x = \langle (\mathit{fst}\ f)\ x, (\mathit{snd}\ f)\ x\rangle\end{array}$$

$$\textbf{fpar parf f} = \textbf{f} \qquad \text{y} \qquad \textbf{parf fpar g} = \textbf{g}$$

Isomorfismos de tipos

Otros isomorfismos: composición

$$A \Rightarrow B \Rightarrow C \quad \equiv \quad B \Rightarrow A \Rightarrow C$$

porque

$$\begin{array}{l}
\mathbf{uncurry} : (A \Rightarrow B \Rightarrow C) \Rightarrow ((A \wedge B) \Rightarrow C) \\
\qquad\qquad \mathbf{conSwap} : ((A \wedge B) \Rightarrow C) \Rightarrow ((B \wedge A) \Rightarrow C) \\
\qquad\qquad\qquad\qquad \mathbf{curry} : ((B \wedge A) \Rightarrow C) \Rightarrow (B \Rightarrow A \Rightarrow C)
\end{array}$$

$$\mathbf{uncurry\ conSwap\ curry\ f} = \mathbf{f}$$

Isomorfismos de tipos

Otros isomorfismos: composición

$$A \Rightarrow B \Rightarrow C \quad \equiv \quad B \Rightarrow A \Rightarrow C$$

porque

$$\begin{array}{l}
\textbf{uncurry} : (A \Rightarrow B \Rightarrow C) \Rightarrow ((A \wedge B) \Rightarrow C) \\
\qquad\qquad \textbf{conSwap} : ((A \wedge B) \Rightarrow C) \Rightarrow ((B \wedge A) \Rightarrow C) \\
\qquad\qquad\qquad\qquad \textbf{curry} : ((B \wedge A) \Rightarrow C) \Rightarrow (B \Rightarrow A \Rightarrow C)
\end{array}$$

$$\textbf{uncurry conSwap curry f} = \textbf{f}$$

Nuestro objetivo:

Convertir isomorfismos en equivalencias

Queremos que, por ejemplo, $f\ \langle x, y \rangle = f\ x\ y$

Contenido de la charla

Una rapidísima introducción a λ-calculo

Historia e intuiciones

Introducido en 1936 por Alonzo Church (director de Alan Turing)

Motivación: Investigar los *fundamentos de la matemática*
(en particular, el concepto de recursión)

Una rapidísima introducción a λ-calculo

Historia e intuiciones

Introducido en 1936 por Alonzo Church (director de Alan Turing)

Motivación: Investigar los *fundamentos de la matemática*
(en particular, el concepto de recursión)

¿Porqué aún lo usamos?

- Las funciones recursivas son fundamentales en computación
- Es el modelo más simple para estudiar propiedades del cómputo

Una rapidísima introducción a λ-calculo

Historia e intuiciones

Introducido en 1936 por Alonzo Church (director de Alan Turing)

Motivación: Investigar los *fundamentos de la matemática*
(en particular, el concepto de recursión)

¿Porqué aún lo usamos?

- Las funciones recursivas son fundamentales en computación
- Es el modelo más simple para estudiar propiedades del cómputo

Dos simplificaciones fundamentales

- Anonimidad de funciones:
  Ejemplo: $\mathit{sumcuad}(x, y) = x^2 + y^2$
  se escribe anónimamente como $(x, y) \mapsto x^2 + y^2$

  $\boxed{\text{Los nombres no son necesarios}}$

Una rapidísima introducción a λ-calculo

Historia e intuiciones

Introducido en 1936 por Alonzo Church (director de Alan Turing)

Motivación: Investigar los *fundamentos de la matemática*
(en particular, el concepto de recursión)

¿Porqué aún lo usamos?

- Las funciones recursivas son fundamentales en computación
- Es el modelo más simple para estudiar propiedades del cómputo

Dos simplificaciones fundamentales

- Anonimidad de funciones:
  Ejemplo: $sumcuad(x, y) = x^2 + y^2$
  se escribe anónimamente como $(x, y) \mapsto x^2 + y^2$

  Los nombres no son necesarios

- Todas las funciones son de una sola variable:
  Ejemplo: $(x, y) \mapsto x^2 + y^2$
  se escribe como $x \mapsto (y \mapsto x^2 + y^2)$

  Una función de 2 variables es una función de 1 variable que retorna una función de 1 variable la cual hace el cálculo

Una rapidísima introducción a λ-calculo

Formalización

Lenguaje de términos (gramática)

$$\mathbf{t}, \mathbf{r} \quad ::= \quad x \quad | \quad \lambda x.\mathbf{t} \quad | \quad \mathbf{t}\ \mathbf{r}$$

- Una variable $x \in \mathit{Vars}$ es un λ-término
- Si $\mathbf{t}$ es un término, y x una variable, $\lambda x.\mathbf{t}$ es un término $\qquad (x \mapsto \mathbf{t})$
- Si $\mathbf{t}$ y $\mathbf{r}$ son términos, $\mathbf{t}\ \mathbf{r}$ es un término $\qquad$ (aplicación)

Estos son todos los términos posibles

Una rapidísima introducción a λ-calculo

Formalización

Lenguaje de términos (gramática)

$$\mathbf{t},\mathbf{r} \quad ::= \quad x \quad | \quad \lambda x.\mathbf{t} \quad | \quad \mathbf{t}\ \mathbf{r}$$

- Una variable $x \in \mathit{Vars}$ es un λ-término
- Si $\mathbf{t}$ es un término, y x una variable, $\lambda x.\mathbf{t}$ es un término ($x \mapsto \mathbf{t}$)
- Si $\mathbf{t}$ y $\mathbf{r}$ son términos, $\mathbf{t}\ \mathbf{r}$ es un término (aplicación)

Estos son todos los términos posibles

Una regla de reescritura (β-reducción)

$$(\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x]$$

Una rapidísima introducción a λ-calculo

Formalización

Lenguaje de términos (gramática)

$$\mathbf{t}, \mathbf{r} \quad ::= \quad x \quad | \quad \lambda x.\mathbf{t} \quad | \quad \mathbf{t}\ \mathbf{r}$$

- Una variable $x \in \mathit{Vars}$ es un λ-término
- Si $\mathbf{t}$ es un término, y x una variable, $\lambda x.\mathbf{t}$ es un término $\quad (x \mapsto \mathbf{t})$
- Si $\mathbf{t}$ y $\mathbf{r}$ son términos, $\mathbf{t}\ \mathbf{r}$ es un término $\quad$ (aplicación)

Estos son todos los términos posibles

Una regla de reescritura (β-reducción)

$$(\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x]$$

Ejemplo: Sea $x^2 + 1$ un λ-término (con alguna codificación)

$$f(x) = x^2 + 1 \qquad \text{se escribe} \qquad \lambda x.x^2 + 1$$

Una rapidísima introducción a λ-calculo

Lenguaje de términos (gramática)

$$\mathbf{t}, \mathbf{r} \quad ::= \quad x \quad | \quad \lambda x.\mathbf{t} \quad | \quad \mathbf{t}\ \mathbf{r}$$

- Una variable $x \in Vars$ es un λ-término
- Si $\mathbf{t}$ es un término, y x una variable, $\lambda x.\mathbf{t}$ es un término $(x \mapsto \mathbf{t})$
- Si $\mathbf{t}$ y $\mathbf{r}$ son términos, $\mathbf{t}\ \mathbf{r}$ es un término (aplicación)

Estos son todos los términos posibles

Una regla de reescritura (β-reducción)

$$(\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x]$$

Ejemplo: Sea $x^2 + 1$ un λ-término (con alguna codificación)

$$f(x) = x^2 + 1 \qquad \text{se escribe} \qquad \lambda x.x^2 + 1$$

$f(\mathbf{t})$ se escribe $(\lambda x.x^2 + 1)\ \mathbf{t}$ y β-reduce a

$$(x^2 + 1)[\mathbf{t}/x] \quad = \quad \mathbf{t}^2 + 1$$

Una rapidísima introducción al λ-calculo

Formas normales

No todo cómputo termina bien...

$$\text{Consideremos} \quad \lambda x.xx$$

(la función que toma un argumento y lo aplica a sí mismo)

Una rapidísima introducción al λ-calculo

Formas normales

No todo cómputo termina bien...

Consideremos $\lambda x.xx$
(la función que toma un argumento y lo aplica a sí mismo)

$$\Omega = (\lambda x.xx)(\lambda x.xx)$$

Una rapidísima introducción al λ-calculo

Formas normales

No todo cómputo termina bien...

$$\text{Consideremos} \quad \lambda x.xx$$

(la función que toma un argumento y lo aplica a sí mismo)

$$\Omega = (\lambda x.xx)(\lambda x.xx) \ \rightarrow \ xx[\lambda x.xx/x]$$

Una rapidísima introducción al λ-calculo

Formas normales

No todo cómputo termina bien...

Consideremos $\lambda x.xx$
(la función que toma un argumento y lo aplica a sí mismo)

$$\Omega = (\lambda x.xx)(\lambda x.xx) \;\to\; xx[\lambda x.xx/x] \;=\; (\lambda x.xx)(\lambda x.xx) = \Omega$$

Así que $\quad \Omega \to \Omega \to \Omega \to \cdots$

Una rapidísima introducción al λ-calculo

Formas normales

No todo cómputo termina bien...

Consideremos $\lambda x.xx$
(la función que toma un argumento y lo aplica a sí mismo)

$$\Omega = (\lambda x.xx)(\lambda x.xx) \ \to \ xx[\lambda x.xx/x] \ = \ (\lambda x.xx)(\lambda x.xx) = \Omega$$

Así que $\quad \Omega \to \Omega \to \Omega \to \cdots$

Normalización

t está en forma normal, si no reescribe a nada $\qquad$ ej. $\qquad \lambda x.x$

Una rapidísima introducción al λ-calculo

Formas normales

No todo cómputo termina bien...

Consideremos $\quad \lambda x.xx$
(la función que toma un argumento y lo aplica a sí mismo)

$$\Omega = (\lambda x.xx)(\lambda x.xx) \ \to \ xx[\lambda x.xx/x] \ = \ (\lambda x.xx)(\lambda x.xx) = \Omega$$

Así que $\quad \Omega \to \Omega \to \Omega \to \cdots$

Normalización

t está en forma normal, si no reescribe a nada $\quad$ ej. $\quad \lambda x.x$
t es normalizante si *puede* terminar $\quad$ ej. $\quad (\lambda x.\lambda y.y)\Omega$

Una rapidísima introducción al λ-calculo

Formas normales

No todo cómputo termina bien...

Consideremos $\lambda x.xx$
(la función que toma un argumento y lo aplica a sí mismo)

$$\Omega = (\lambda x.xx)(\lambda x.xx) \ \to \ xx[\lambda x.xx/x] \ = \ (\lambda x.xx)(\lambda x.xx) = \Omega$$

Así que $\quad \Omega \to \Omega \to \Omega \to \cdots$

Normalización

t está en forma normal, si no reescribe a nada	ej.	$\lambda x.x$
t es normalizante si *puede* terminar	ej.	$(\lambda x.\lambda y.y)\Omega$
t es fuertemente normalizante si *siempre* termina	ej.	$(\lambda x.x)(\lambda x.x)$

Una rapidísima introducción al λ-calculo

Formas normales

No todo cómputo termina bien...

Consideremos $\lambda x.xx$
(la función que toma un argumento y lo aplica a sí mismo)

$$\Omega = (\lambda x.xx)(\lambda x.xx) \;\to\; xx[\lambda x.xx/x] \;=\; (\lambda x.xx)(\lambda x.xx) = \Omega$$

Así que $\quad \Omega \to \Omega \to \Omega \to \cdots$

Normalización

t está en forma normal, si no reescribe a nada	ej.	$\lambda x.x$
t es normalizante si *puede* terminar	ej.	$(\lambda x.\lambda y.y)\Omega$
t es fuertemente normalizante si *siempre* termina	ej.	$(\lambda x.x)(\lambda x.x)$

¿Cómo saber si un término es (fuertemente) normalizante?

Clasificación estática de λ-términos (o sea, sin reducirlos)

Una rapidísima introducción al λ-calculo tipado

Tipos simples

Clasificación estática de λ-términos (o sea, sin reducirlos)

$$
\begin{array}{llcl}
\textbf{Términos} & \mathbf{t}, \mathbf{r}, \mathbf{s} & ::= & x^A \quad | \quad \lambda x^A.\mathbf{t} \quad | \quad \mathbf{t}\ \mathbf{r} \\
\textbf{Tipos} & A, B, C & ::= & \tau \quad | \quad A \Rightarrow B
\end{array}
$$

- τ es un *tipo de base*
- $A \Rightarrow B$ es el tipo función

Una rapidísima introducción al λ-calculo tipado

Tipos simples

Clasificación estática de λ-términos (o sea, sin reducirlos)

$$\begin{array}{lrcl} \textbf{Términos} & \mathbf{t}, \mathbf{r}, \mathbf{s} & ::= & x^A \quad | \quad \lambda x^A.\mathbf{t} \quad | \quad \mathbf{t}\ \mathbf{r} \\ \textbf{Tipos} & A, B, C & ::= & \tau \quad | \quad A \Rightarrow B \end{array}$$

- τ es un *tipo de base*
- $A \Rightarrow B$ es el tipo función

Contexto: conjunto de variables tipadas $\Gamma = x_1^{A_1}, \ldots, x_n^{A_n}$

$\Gamma \vdash \mathbf{t} : A$ "$\mathbf{t}$ tiene tipo A en el contexto Γ"

Clasificación estática de λ-términos (o sea, sin reducirlos)

$$\begin{array}{llcl} \textbf{Términos} & \mathbf{t}, \mathbf{r}, \mathbf{s} & ::= & x^A \mid \lambda x^A.\mathbf{t} \mid \mathbf{t}\ \mathbf{r} \\ \textbf{Tipos} & A, B, C & ::= & \tau \mid A \Rightarrow B \end{array}$$

- τ es un *tipo de base*
- $A \Rightarrow B$ es el tipo función

Contexto: conjunto de variables tipadas $\quad \Gamma = x_1^{A_1}, \ldots, x_n^{A_n}$

$\Gamma \vdash \mathbf{t} : A \quad$ "$\mathbf{t}$ tiene tipo A en el contexto Γ"

Reglas de tipado

Clasificación estática de λ-términos (o sea, sin reducirlos)

$$\begin{array}{lrcl} \textbf{Términos} & \mathbf{t}, \mathbf{r}, \mathbf{s} & ::= & x^A \quad | \quad \lambda x^A.\mathbf{t} \quad | \quad \mathbf{t}\ \mathbf{r} \\ \textbf{Tipos} & A, B, C & ::= & \tau \quad | \quad A \Rightarrow B \end{array}$$

- τ es un *tipo de base*
- $A \Rightarrow B$ es el tipo función

Contexto: conjunto de variables tipadas $\quad \Gamma = x_1^{A_1}, \dots, x_n^{A_n}$

$\Gamma \vdash \mathbf{t} : A \quad$ "$\mathbf{t}$ tiene tipo A en el contexto Γ"

Reglas de tipado

$$\frac{}{\Gamma, x^A \vdash x : A}\ ax$$

Una rapidísima introducción al λ-calculo tipado

Tipos simples

Clasificación estática de λ-términos (o sea, sin reducirlos)

$$\begin{array}{llcl} \textbf{Términos} & \mathbf{t}, \mathbf{r}, \mathbf{s} & ::= & x^A \mid \lambda x^A.\mathbf{t} \mid \mathbf{t}\ \mathbf{r} \\ \textbf{Tipos} & A, B, C & ::= & \tau \mid A \Rightarrow B \end{array}$$

- τ es un *tipo de base*
- $A \Rightarrow B$ es el tipo función

Contexto: conjunto de variables tipadas $\Gamma = x_1^{A_1}, \dots, x_n^{A_n}$

$\Gamma \vdash \mathbf{t} : A$ "**t** tiene tipo A en el contexto Γ"

Reglas de tipado

$$\frac{}{\Gamma, x^A \vdash x : A}\ ax \qquad \frac{\Gamma, x^A \vdash \mathbf{t} : B}{\Gamma \vdash \lambda x^A.\mathbf{t} : A \Rightarrow B}\ \Rightarrow_I$$

Clasificación estática de λ-términos (o sea, sin reducirlos)

$$
\begin{array}{llclclcl}
\textbf{Términos} & \mathbf{t}, \mathbf{r}, \mathbf{s} & ::= & x^A & \mid & \lambda x^A.\mathbf{t} & \mid & \mathbf{t}\ \mathbf{r} \\
\textbf{Tipos} & A, B, C & ::= & \tau & \mid & A \Rightarrow B & &
\end{array}
$$

- τ es un *tipo de base*
- $A \Rightarrow B$ es el tipo función

Contexto: conjunto de variables tipadas $\quad \Gamma = x_1^{A_1}, \dots, x_n^{A_n}$

$\Gamma \vdash \mathbf{t} : A \quad$ "$\mathbf{t}$ tiene tipo A en el contexto Γ"

Reglas de tipado

$$
\frac{}{\Gamma, x^A \vdash x : A}\ ax
\qquad
\frac{\Gamma, x^A \vdash \mathbf{t} : B}{\Gamma \vdash \lambda x^A.\mathbf{t} : A \Rightarrow B} \Rightarrow_I
\qquad
\frac{\Gamma \vdash \mathbf{t} : A \Rightarrow B \quad \Gamma \vdash \mathbf{r} : A}{\Gamma \vdash \mathbf{t}\ \mathbf{r} : B} \Rightarrow_E
$$

Clasificación estática de λ-términos (o sea, sin reducirlos)

Términos	$\mathbf{t},\mathbf{r},\mathbf{s}$	$::=$	$x^A \mid \lambda x^A.\mathbf{t} \mid \mathbf{t}\ \mathbf{r}$
Tipos	A,B,C	$::=$	$\tau \mid A \Rightarrow B$

- τ es un *tipo de base*
- $A \Rightarrow B$ es el tipo función

Contexto: conjunto de variables tipadas $\quad \Gamma = x_1^{A_1},\dots,x_n^{A_n}$

$\Gamma \vdash \mathbf{t} : A \qquad$ "$\mathbf{t}$ tiene tipo A en el contexto Γ"

Reglas de tipado

$$\frac{}{\Gamma, x^A \vdash x : A}\,ax \qquad \frac{\Gamma, x^A \vdash \mathbf{t} : B}{\Gamma \vdash \lambda x^A.\mathbf{t} : A \Rightarrow B}\Rightarrow_I \qquad \frac{\Gamma \vdash \mathbf{t} : A \Rightarrow B \quad \Gamma \vdash \mathbf{r} : A}{\Gamma \vdash \mathbf{t}\ \mathbf{r} : B}\Rightarrow_E$$

Ejemplo de derivación de tipo

$$\frac{}{x^A \vdash x : A}\,ax$$

Una rapidísima introducción al λ-calculo tipado

Tipos simples

Clasificación estática de λ-términos (o sea, sin reducirlos)

$$
\begin{array}{llcl}
\textbf{Términos} & \mathbf{t}, \mathbf{r}, \mathbf{s} & ::= & x^A \mid \lambda x^A.\mathbf{t} \mid \mathbf{t}\ \mathbf{r} \\
\textbf{Tipos} & A, B, C & ::= & \tau \mid A \Rightarrow B
\end{array}
$$

- τ es un *tipo de base*
- $A \Rightarrow B$ es el tipo función

Contexto: conjunto de variables tipadas $\Gamma = x_1^{A_1}, \dots, x_n^{A_n}$

$\Gamma \vdash \mathbf{t} : A$ "**t** tiene tipo A en el contexto Γ"

Reglas de tipado

$$
\frac{}{\Gamma, x^A \vdash x : A}\ ax \qquad \frac{\Gamma, x^A \vdash \mathbf{t} : B}{\Gamma \vdash \lambda x^A.\mathbf{t} : A \Rightarrow B}\ \Rightarrow_I \qquad \frac{\Gamma \vdash \mathbf{t} : A \Rightarrow B \quad \Gamma \vdash \mathbf{r} : A}{\Gamma \vdash \mathbf{t}\ \mathbf{r} : B}\ \Rightarrow_E
$$

Ejemplo de derivación de tipo

$$
\dfrac{\dfrac{}{x^A \vdash x : A}\ ax}{\vdash \lambda x^A.x : A \Rightarrow A}\ \Rightarrow_I
$$

Una rapidísima introducción al λ-calculo tipado

Tipos simples

Clasificación estática de λ-términos (o sea, sin reducirlos)

Términos	$\mathbf{t}, \mathbf{r}, \mathbf{s}$	$::=$	$x^A \mid \lambda x^A.\mathbf{t} \mid \mathbf{t}\ \mathbf{r}$
Tipos	A, B, C	$::=$	$\tau \mid A \Rightarrow B$

- τ es un *tipo de base*
- $A \Rightarrow B$ es el tipo función

Contexto: conjunto de variables tipadas $\qquad \Gamma = x_1^{A_1}, \dots, x_n^{A_n}$

$\Gamma \vdash \mathbf{t} : A \qquad$ "$\mathbf{t}$ tiene tipo A en el contexto Γ"

Reglas de tipado

$$\frac{}{\Gamma, x^A \vdash x : A}\,ax \qquad \frac{\Gamma, x^A \vdash \mathbf{t} : B}{\Gamma \vdash \lambda x^A.\mathbf{t} : A \Rightarrow B}\Rightarrow_I \qquad \frac{\Gamma \vdash \mathbf{t} : A \Rightarrow B \quad \Gamma \vdash \mathbf{r} : A}{\Gamma \vdash \mathbf{t}\ \mathbf{r} : B}\Rightarrow_E$$

Ejemplo de derivación de tipo

$$\frac{\dfrac{}{y^{A\Rightarrow A} \vdash y : A \Rightarrow A}\,ax}{\vdash \lambda y^{A\Rightarrow A}.y : (A \Rightarrow A) \Rightarrow (A \Rightarrow A)}\Rightarrow_I \qquad \frac{\dfrac{}{x^A \vdash x : A}\,ax}{\vdash \lambda x^A.x : A \Rightarrow A}\Rightarrow_I$$

Clasificación estática de λ-términos (o sea, sin reducirlos)

$$\begin{array}{lrcl} \textbf{Términos} & \mathbf{t}, \mathbf{r}, \mathbf{s} & ::= & x^A \mid \lambda x^A.\mathbf{t} \mid \mathbf{t}\ \mathbf{r} \\ \textbf{Tipos} & A, B, C & ::= & \tau \mid A \Rightarrow B \end{array}$$

- τ es un *tipo de base*
- $A \Rightarrow B$ es el tipo función

Contexto: conjunto de variables tipadas $\quad \Gamma = x_1^{A_1}, \dots, x_n^{A_n}$

$\Gamma \vdash \mathbf{t} : A \qquad$ "$\mathbf{t}$ tiene tipo A en el contexto Γ"

Reglas de tipado

$$\frac{}{\Gamma, x^A \vdash x : A}\ ax \qquad \frac{\Gamma, x^A \vdash \mathbf{t} : B}{\Gamma \vdash \lambda x^A.\mathbf{t} : A \Rightarrow B}\ \Rightarrow_I \qquad \frac{\Gamma \vdash \mathbf{t} : A \Rightarrow B \quad \Gamma \vdash \mathbf{r} : A}{\Gamma \vdash \mathbf{t}\ \mathbf{r} : B}\ \Rightarrow_E$$

Ejemplo de derivación de tipo

$$\dfrac{\dfrac{\dfrac{}{y^{A\Rightarrow A} \vdash y : A \Rightarrow A}\ ax}{\vdash \lambda y^{A\Rightarrow A}.y : (A \Rightarrow A) \Rightarrow (A \Rightarrow A)}\ \Rightarrow_I \qquad \dfrac{\dfrac{}{x^A \vdash x : A}\ ax}{\vdash \lambda x^A.x : A \Rightarrow A}\ \Rightarrow_I}{\vdash (\lambda y^{A\Rightarrow A}.y)\ (\lambda x^A.x) : A \Rightarrow A}\ \Rightarrow_E$$

Una rapidísima introducción al λ-calculo tipado

Tipos simples

Clasificación estática de λ-términos (o sea, sin reducirlos)

$$
\begin{array}{llcl}
\textbf{Términos} & \mathbf{t}, \mathbf{r}, \mathbf{s} & ::= & x^A \mid \lambda x^A.\mathbf{t} \mid \mathbf{t}\ \mathbf{r} \\
\textbf{Tipos} & A, B, C & ::= & \tau \mid A \Rightarrow B
\end{array}
$$

- τ es un *tipo de base*
- $A \Rightarrow B$ es el tipo función

Contexto: conjunto de variables tipadas $\quad \Gamma = x_1^{A_1}, \ldots, x_n^{A_n}$

$\Gamma \vdash \mathbf{t} : A \qquad$ "$\mathbf{t}$ tiene tipo A en el contexto Γ"

Reglas de tipado

$$
\frac{}{\Gamma, x^A \vdash x : A}\ ax
\qquad
\frac{\Gamma, x^A \vdash \mathbf{t} : B}{\Gamma \vdash \lambda x^A.\mathbf{t} : A \Rightarrow B} \Rightarrow_I
\qquad
\frac{\Gamma \vdash \mathbf{t} : A \Rightarrow B \quad \Gamma \vdash \mathbf{r} : A}{\Gamma \vdash \mathbf{t}\ \mathbf{r} : B} \Rightarrow_E
$$

Ejemplo de derivación de tipo

$$
\dfrac{\dfrac{\dfrac{}{y^{A\Rightarrow A} \vdash y : A \Rightarrow A}\ ax}{\vdash \lambda y^{A\Rightarrow A}.y : (A \Rightarrow A) \Rightarrow (A \Rightarrow A)} \Rightarrow_I \qquad \dfrac{\dfrac{}{x^A \vdash x : A}\ ax}{\vdash \lambda x^A.x : A \Rightarrow A} \Rightarrow_I}{\vdash (\lambda y^{A\Rightarrow A}.y)\ (\lambda x^A.x) : A \Rightarrow A} \Rightarrow_E
$$

Verificación: $(\lambda y^{A\Rightarrow A}.y)\ (\lambda x^A.x)$ reescribe a $\lambda x^A.x$ (de tipo $A \Rightarrow A$)

Una rapidísima introducción al λ-calculo tipado

Normalización

Ω no tiene tipo en esta teoría

Ω no tiene tipo en esta teoría

Más aún. . .

Theorem (Normalización fuerte)

Si $\mathbf{t}$ *tiene tipo,* $\mathbf{t}$ *es fuertemente normalizante.*

Ω no tiene tipo en esta teoría

Más aún...

Theorem (Normalización fuerte)

Si $\mathbf{t}$ tiene tipo, $\mathbf{t}$ es fuertemente normalizante.

Eslogan: *"Well-typed programs cannot go wrong"* — `[R. Milner'78]`

¿Cómo se relaciona esto con lógica?

Una palabra sobre la correspondencia de Curry-Howard

Lógica clásica: una proposición se asume verdadera o falsa

¿Cómo se relaciona esto con lógica?

Una palabra sobre la correspondencia de Curry-Howard

Lógica clásica: una proposición se asume verdadera o falsa

Lógica intuicionista: una proposición es verdadera si hay una prueba constructiva que muestre que lo es.

¡La ley del tercero excluido no es un axioma!
(y tampoco se puede demostrar) en lógica intuicionista $A \vee \neg A$

¿Cómo se relaciona esto con lógica?

Una palabra sobre la correspondencia de Curry-Howard

Lógica clásica: una proposición se asume verdadera o falsa

Lógica intuicionista: una proposición es verdadera si hay una prueba constructiva que muestre que lo es.

¡La ley del tercero excluido no es un axioma! (y tampoco se puede demostrar) en lógica intuicionista	$A \vee \neg A$

Lógica intuicionista mínima

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

¿Cómo se relaciona esto con lógica?

Una palabra sobre la correspondencia de Curry-Howard

Lógica clásica: una proposición se asume verdadera o falsa

Lógica intuicionista: una proposición es verdadera si hay una prueba constructiva que muestre que lo es.

¡La ley del tercero excluido no es un axioma! (y tampoco se puede demostrar) en lógica intuicionista	$A \vee \neg A$

Lógica intuicionista mínima

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

Reglas de tipado

$$\frac{}{\Gamma, x^A \vdash x : A}\, ax \qquad \frac{\Gamma, x^A \vdash \mathbf{t} : B}{\Gamma \vdash \lambda x^A.\mathbf{t} : A \Rightarrow B} \Rightarrow_I \qquad \frac{\Gamma \vdash \mathbf{t} : A \Rightarrow B \quad \Gamma \vdash \mathbf{r} : A}{\Gamma \vdash \mathbf{t}\ \mathbf{r} : B} \Rightarrow_E$$

¿Cómo se relaciona esto con lógica?

Una palabra sobre la correspondencia de Curry-Howard

Lógica clásica: una proposición se asume verdadera o falsa

Lógica intuicionista: una proposición es verdadera si hay una prueba constructiva que muestre que lo es.

¡La ley del tercero excluido no es un axioma! (y tampoco se puede demostrar) en lógica intuicionista	$A \vee \neg A$

Lógica intuicionista mínima

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

Reglas de tipado

$$\frac{}{\Gamma, x^A \vdash x : A}\ ax \qquad \frac{\Gamma, x^A \vdash \mathbf{t} : B}{\Gamma \vdash \lambda x^A.\mathbf{t} : A \Rightarrow B} \Rightarrow_I \qquad \frac{\Gamma \vdash \mathbf{t} : A \Rightarrow B \quad \Gamma \vdash \mathbf{r} : A}{\Gamma \vdash \mathbf{t}\ \mathbf{r} : B} \Rightarrow_E$$

El λ-término es la prueba de la propocisión

$\boxed{\text{¡Las pruebas... son programas!}}$

Haskell Curry y William Howard
entre 1934 y 1969

¿Cómo se relaciona esto con lógica?

Una palabra sobre la correspondencia de Curry-Howard

Lógica clásica: una proposición se asume verdadera o falsa

Lógica intuicionista: una proposición es verdadera si hay una prueba constructiva que muestre que lo es.

¡La ley del tercero excluido no es un axioma! (y tampoco se puede demostrar) en lógica intuicionista	$A \vee \neg A$

Lógica intuicionista mínima

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

Reglas de tipado

$$\frac{}{\Gamma, x^A \vdash x : A}\ ax \qquad \frac{\Gamma, x^A \vdash \mathbf{t} : B}{\Gamma \vdash \lambda x^A.\mathbf{t} : A \Rightarrow B} \Rightarrow_I \qquad \frac{\Gamma \vdash \mathbf{t} : A \Rightarrow B \quad \Gamma \vdash \mathbf{r} : A}{\Gamma \vdash \mathbf{t}\ \mathbf{r} : B} \Rightarrow_E$$

El λ-término es la prueba de la proposición

¡Las pruebas. . . son programas!

Haskell Curry y William Howard
entre 1934 y 1969

Lógicas más complejas se corresponden a sistemas más complejos

Contenido de la charla

1. Una rapidísima introducción al λ del $\lambda c.c$
   - Una rapidísima introducción a λ-calculo
   - Una rapidísima introducción al λ-calculo tipado
2. Convirtiendo isomorfismos en equivalencias
   - λ-calculo modulo isomorfismos
   - Normalizando
   - Computando
3. Conclusiones

Equivalencia proposicional vs definicional

$\forall x.\forall y.(x \;=\; y) \Rightarrow (\mathsf{par}(x) = \mathsf{par}(y))$

Equivalencia proposicional vs definicional

$\forall x.\forall y.(x \;=\; y) \Rightarrow (\mathrm{par}(x) = \mathrm{par}(y))$	$(2+2 \;=\; 4) \Rightarrow (\mathrm{par}(2+2) = \mathrm{par}(4))$
	$(2+2 \;=\; 5) \Rightarrow (\mathrm{par}(2+2) = \mathrm{par}(5))$

Equivalencia proposicional vs definicional

$$\forall x.\forall y.(x \underbrace{=}_{\text{Proposición}} y) \Rightarrow (\mathrm{par}(x) = \mathrm{par}(y))$$

$$(2+2 = 4) \Rightarrow (\mathrm{par}(2+2) = \mathrm{par}(4))$$

$$(2+2 \underbrace{=}_{\text{Computación}} 5) \Rightarrow (\mathrm{par}(2+2) = \mathrm{par}(5))$$

Equivalencia proposicional vs definicional

$$\forall x.\forall y.(x = y) \Rightarrow (\mathrm{par}(x) = \mathrm{par}(y))$$

Proposición

$$(2+2 = 4) \Rightarrow (\mathrm{par}(2+2) = \mathrm{par}(4))$$
$$(2+2 = 5) \Rightarrow (\mathrm{par}(2+2) = \mathrm{par}(5))$$

Computación

$$\forall X.\forall Y.(X \equiv Y) \Rightarrow (X \Leftrightarrow Y)$$

Proposicional

$$(A \equiv B) \Rightarrow (A \Leftrightarrow B)$$

¡Decidible! `[DiCosmo y otros]`

(en algunos casos)

Equivalencia proposicional vs definicional

$\forall x.\forall y.(x \;=\; y) \Rightarrow (\mathrm{par}(x) = \mathrm{par}(y))$

Proposición

$(2+2 \;=\; 4) \Rightarrow (\mathrm{par}(2+2) = \mathrm{par}(4))$

$(2+2 \;=\; 5) \Rightarrow (\mathrm{par}(2+2) = \mathrm{par}(5))$

Computación

$\forall X.\forall Y.(X \;\equiv\; Y) \Rightarrow (X \Leftrightarrow Y)$

Proposicional

$(A \;\equiv\; B) \Rightarrow (A \Leftrightarrow B)$

¡Decidible! `[DiCosmo y otros]`

(en algunos casos)

Nosotros queremos ir más lejos:

$$(A \;\equiv\; B) \Rightarrow (\mathbf{t} : A \Leftrightarrow \mathbf{t} : B)$$

El objetivo es igualar tipos isomorfos

La configuración inicial

- Tipos simples, con conjunción e implicación

$$A, B, C ::= \tau \mid A \Rightarrow B \mid A \wedge B$$

- Una relación de equivalencia de tipos (basada en los isomorfismos[1])
  1. $A \wedge B \equiv B \wedge A$ (conm)
  2. $A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$ (aso)
  3. $(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$ (curry)
  4. $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$ (distrib)

[1] Bruce, Di Cosmo, Longo, MSCS 2(2), 231–247, 1992

La configuración inicial

- Tipos simples, con conjunción e implicación

$$A, B, C ::= \tau \mid A \Rightarrow B \mid A \wedge B$$

- Una relación de equivalencia de tipos (basada en los isomorfismos[1])
  1. $A \wedge B \equiv B \wedge A$ (conm)
  2. $A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$ (aso)
  3. $(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$ (curry)
  4. $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$ (distrib)

Queremos

$$\boxed{[A{\equiv}B]\,\frac{\Gamma \vdash \mathbf{r} : A}{\Gamma \vdash \mathbf{r} : B}}$$

[1]Bruce, Di Cosmo, Longo, MSCS 2(2), 231–247, 1992

Conjunción asociativa y conmutativa

$$\frac{\Gamma \vdash \mathbf{r} : A \quad \Gamma \vdash \mathbf{s} : B}{\Gamma \vdash \ \langle \mathbf{r}, \mathbf{s} \rangle \ : A \wedge B} \ (\wedge_i)$$

Conjunción asociativa y conmutativa

$$\frac{\Gamma \vdash \mathbf{r} : A \quad \Gamma \vdash \mathbf{s} : B}{\Gamma \vdash \ \langle \mathbf{r}, \mathbf{s} \rangle \ : A \wedge B} \ (\wedge_i)$$

$$\begin{aligned} A \wedge B &\equiv B \wedge A \\ A \wedge (B \wedge C) &\equiv (A \wedge B) \wedge C \end{aligned}$$

Entonces

$$\begin{aligned} \langle \mathbf{r}, \mathbf{s} \rangle &\leftrightarrows \ \langle \mathbf{s}, \mathbf{r} \rangle \\ \langle \mathbf{r}, \ \langle \mathbf{s}, \mathbf{t} \rangle \ \rangle &\leftrightarrows \ \langle \ \langle \mathbf{r}, \mathbf{s} \rangle \ , \mathbf{t} \rangle \end{aligned}$$

Conjunción asociativa y conmutativa

$$\frac{\Gamma \vdash \mathbf{r} : A \quad \Gamma \vdash \mathbf{s} : B}{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : A \wedge B}\ (\wedge_i)$$

$$\begin{aligned} A \wedge B &\equiv B \wedge A \\ A \wedge (B \wedge C) &\equiv (A \wedge B) \wedge C \end{aligned}$$

Entonces
$$\begin{aligned} \langle \mathbf{r}, \mathbf{s} \rangle &\leftrightarrows \langle \mathbf{s}, \mathbf{r} \rangle \\ \langle \mathbf{r}, \langle \mathbf{s}, \mathbf{t} \rangle \rangle &\leftrightarrows \langle \langle \mathbf{r}, \mathbf{s} \rangle, \mathbf{t} \rangle \end{aligned}$$

¿Y la eliminación?

$$\frac{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : A \wedge B}{\Gamma \vdash \pi_1 \langle \mathbf{r}, \mathbf{s} \rangle : A}\ (\wedge_e)$$

Conjunción asociativa y conmutativa

$$\frac{\Gamma \vdash \mathbf{r} : A \quad \Gamma \vdash \mathbf{s} : B}{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : A \wedge B} \ (\wedge_i)$$

Entonces

$$\begin{aligned} A \wedge B &\equiv B \wedge A \\ A \wedge (B \wedge C) &\equiv (A \wedge B) \wedge C \\ \langle \mathbf{r}, \mathbf{s} \rangle &\leftrightarrows \langle \mathbf{s}, \mathbf{r} \rangle \\ \langle \mathbf{r}, \langle \mathbf{s}, \mathbf{t} \rangle \rangle &\leftrightarrows \langle \langle \mathbf{r}, \mathbf{s} \rangle, \mathbf{t} \rangle \end{aligned}$$

¿Y la eliminación?

$$\frac{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : A \wedge B}{\Gamma \vdash \pi_1 \langle \mathbf{r}, \mathbf{s} \rangle : A} \ (\wedge_e)$$

¡Pero $A \wedge B = B \wedge A$!

$$\frac{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : B \wedge A}{\Gamma \vdash \pi_1 \langle \mathbf{r}, \mathbf{s} \rangle : B} \ (\wedge_e)$$

Más aún $\langle \mathbf{r}, \mathbf{s} \rangle = \langle \mathbf{s}, \mathbf{r} \rangle$ así que $\pi_1 \langle \mathbf{r}, \mathbf{s} \rangle = \pi_1 \langle \mathbf{s}, \mathbf{r} \rangle$!!

Conjunción asociativa y conmutativa

$$\frac{\Gamma \vdash \mathbf{r} : A \quad \Gamma \vdash \mathbf{s} : B}{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : A \wedge B} \ (\wedge_i)$$

Entonces

$$\begin{array}{rcl} A \wedge B & \equiv & B \wedge A \\ A \wedge (B \wedge C) & \equiv & (A \wedge B) \wedge C \\ \langle \mathbf{r}, \mathbf{s} \rangle & \leftrightarrows & \langle \mathbf{s}, \mathbf{r} \rangle \\ \langle \mathbf{r}, \langle \mathbf{s}, \mathbf{t} \rangle \rangle & \leftrightarrows & \langle \langle \mathbf{r}, \mathbf{s} \rangle, \mathbf{t} \rangle \end{array}$$

¿Y la eliminación?

$$\frac{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : A \wedge B}{\Gamma \vdash \pi_1 \langle \mathbf{r}, \mathbf{s} \rangle : A} \ (\wedge_e)$$

¡Pero $A \wedge B = B \wedge A$!

$$\frac{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : B \wedge A}{\Gamma \vdash \pi_1 \langle \mathbf{r}, \mathbf{s} \rangle : B} \ (\wedge_e)$$

Más aún $\quad \langle \mathbf{r}, \mathbf{s} \rangle = \langle \mathbf{s}, \mathbf{r} \rangle \quad$ así que $\pi_1 \langle \mathbf{r}, \mathbf{s} \rangle = \pi_1 \langle \mathbf{s}, \mathbf{r} \rangle$!!

Solución: Church-style – Proyección con respecto al tipo

Si $\mathbf{r} : A$ entonces $\pi_A \langle \mathbf{r}, \mathbf{s} \rangle \to \mathbf{r}$

Conjunción asociativa y conmutativa

$$\frac{\Gamma \vdash \mathbf{r} : A \quad \Gamma \vdash \mathbf{s} : B}{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : A \wedge B} \ (\wedge_i)$$

$$A \wedge B \equiv B \wedge A$$
$$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$$

Entonces

$$\langle \mathbf{r}, \mathbf{s} \rangle \leftrightarrows \langle \mathbf{s}, \mathbf{r} \rangle$$
$$\langle \mathbf{r}, \langle \mathbf{s}, \mathbf{t} \rangle \rangle \leftrightarrows \langle \langle \mathbf{r}, \mathbf{s} \rangle, \mathbf{t} \rangle$$

¿Y la eliminación?

$$\frac{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : A \wedge B}{\Gamma \vdash \pi_1 \langle \mathbf{r}, \mathbf{s} \rangle : A} \ (\wedge_e)$$

¡Pero $A \wedge B = B \wedge A$!

$$\frac{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : B \wedge A}{\Gamma \vdash \pi_1 \langle \mathbf{r}, \mathbf{s} \rangle : B} \ (\wedge_e)$$

Más aún $\langle \mathbf{r}, \mathbf{s} \rangle = \langle \mathbf{s}, \mathbf{r} \rangle$ así que $\pi_1 \langle \mathbf{r}, \mathbf{s} \rangle = \pi_1 \langle \mathbf{s}, \mathbf{r} \rangle$!!

Solución: Church-style – Proyección con respecto al tipo

Si $\mathbf{r} : A$ entonces $\pi_A \langle \mathbf{r}, \mathbf{s} \rangle \to \mathbf{r}$

Esto induce no determinismo

Si $\begin{array}{l} \mathbf{r} : A \\ \mathbf{s} : A \end{array}$ entonces $\begin{array}{l} \pi_A \langle \mathbf{r}, \mathbf{s} \rangle \to \mathbf{r} \\ \pi_A \langle \mathbf{r}, \mathbf{s} \rangle \to \mathbf{s} \end{array}$

Conjunción asociativa y conmutativa

$$\frac{\Gamma \vdash \mathbf{r} : A \quad \Gamma \vdash \mathbf{s} : B}{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : A \wedge B} (\wedge_i)$$

Entonces

$$\begin{array}{rcl} A \wedge B & \equiv & B \wedge A \\ A \wedge (B \wedge C) & \equiv & (A \wedge B) \wedge C \\ \langle \mathbf{r}, \mathbf{s} \rangle & \leftrightarrows & \langle \mathbf{s}, \mathbf{r} \rangle \\ \langle \mathbf{r}, \langle \mathbf{s}, \mathbf{t} \rangle \rangle & \leftrightarrows & \langle \langle \mathbf{r}, \mathbf{s} \rangle, \mathbf{t} \rangle \end{array}$$

¿Y la eliminación?

$$\frac{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : A \wedge B}{\Gamma \vdash \pi_1 \langle \mathbf{r}, \mathbf{s} \rangle : A} (\wedge_e)$$

¡Pero $A \wedge B = B \wedge A$!

$$\frac{\Gamma \vdash \langle \mathbf{r}, \mathbf{s} \rangle : B \wedge A}{\Gamma \vdash \pi_1 \langle \mathbf{r}, \mathbf{s} \rangle : B} (\wedge_e)$$

Más aún $\langle \mathbf{r}, \mathbf{s} \rangle = \langle \mathbf{s}, \mathbf{r} \rangle$ así que $\pi_1 \langle \mathbf{r}, \mathbf{s} \rangle = \pi_1 \langle \mathbf{s}, \mathbf{r} \rangle$!!

Solución: Church-style – Proyección con respecto al tipo

Si $\mathbf{r} : A$ entonces $\pi_A \langle \mathbf{r}, \mathbf{s} \rangle \to \mathbf{r}$

Esto induce no determinismo

Si $\begin{array}{l} \mathbf{r} : A \\ \mathbf{s} : A \end{array}$ entonces $\begin{array}{l} \pi_A \langle \mathbf{r}, \mathbf{s} \rangle \to \mathbf{r} \\ \pi_A \langle \mathbf{r}, \mathbf{s} \rangle \to \mathbf{s} \end{array}$

Si nos interesa la teoría de la demostración tanto **r** como **s** son pruebas válidas de A

Conjunción asociativa y conmutativa

$$\frac{\Gamma\vdash \mathbf{r}:A \quad \Gamma\vdash \mathbf{s}:B}{\Gamma\vdash \mathbf{r}+\mathbf{s} : A\wedge B}\ (\wedge_i)$$

Entonces

$$\begin{aligned} A\wedge B &\equiv B\wedge A\\ A\wedge(B\wedge C) &\equiv (A\wedge B)\wedge C\\ \mathbf{r}+\mathbf{s} &\leftrightarrows (\mathbf{s}+\mathbf{r})\\ \mathbf{r}+(\mathbf{s}+\mathbf{t}) &\leftrightarrows (\mathbf{r}+\mathbf{s})+\mathbf{t}\end{aligned}$$

¿Y la eliminación?

$$\frac{\Gamma\vdash \mathbf{r}+\mathbf{s} : A\wedge B}{\Gamma\vdash \pi_1(\mathbf{r}+\mathbf{s}):A}\ (\wedge_e)$$

¡Pero $A\wedge B = B\wedge A$!

$$\frac{\Gamma\vdash \mathbf{r}+\mathbf{s} : B\wedge A}{\Gamma\vdash \pi_1(\mathbf{r}+\mathbf{s}):B}\ (\wedge_e)$$

Más aún $\quad \mathbf{r}+\mathbf{s} = \mathbf{s}+\mathbf{r}\quad$ así que $\pi_1(\mathbf{r}+\mathbf{s}) = \pi_1(\mathbf{s}+\mathbf{r})$!!

Solución: Church-style – Proyección con respecto al tipo

Si $\mathbf{r}:A$ entonces $\pi_A(\mathbf{r}+\mathbf{s})\to\mathbf{r}$

Esto induce no determinismo

Si $\begin{array}{l}\mathbf{r}:A\\ \mathbf{s}:A\end{array}$ entonces $\begin{array}{l}\pi_A(\mathbf{r}+\mathbf{s})\to\mathbf{r}\\ \pi_A(\mathbf{r}+\mathbf{s})\to\mathbf{s}\end{array}$

Si nos interesa la teoría de la demostración
tanto **r** como **s** son pruebas válidas de A

Curryficación

$$(A \wedge B) \Rightarrow C \quad \equiv \quad A \Rightarrow B \Rightarrow C$$

induce

$$\mathbf{r}(\mathbf{s} + \mathbf{t}) \quad \leftrightarrows \quad \mathbf{rst}$$

Curryficación

$$(A \wedge B) \Rightarrow C \quad \equiv \quad A \Rightarrow B \Rightarrow C$$

induce

$$\mathbf{r}(\mathbf{s} + \mathbf{t}) \quad \leftrightarrows \quad \mathbf{rst}$$

$$(\lambda x^A.\mathbf{r})\ \mathbf{s} \quad \to \quad \mathbf{r}[\mathbf{s}/x]$$

Curryficación

$$(A \wedge B) \Rightarrow C \quad \equiv \quad A \Rightarrow B \Rightarrow C$$

induce

$$\mathbf{r}(\mathbf{s} + \mathbf{t}) \quad \leftrightarrows \quad \mathbf{rst}$$

Si $\mathbf{s} : A$, $\quad (\lambda x^{A}.\mathbf{r})\ \mathbf{s} \quad \to \quad \mathbf{r}[\mathbf{s}/x]$

Curryficación

$$(A \wedge B) \Rightarrow C \quad \equiv \quad A \Rightarrow B \Rightarrow C$$

induce

$$\mathbf{r}(\mathbf{s}+\mathbf{t}) \quad \leftrightarrows \quad \mathbf{rst}$$

Si $\mathbf{s} : A$, $\quad (\lambda x^A.\mathbf{r})\ \mathbf{s} \quad \to \quad \mathbf{r}[\mathbf{s}/x]$

Otras opciones:

$$\begin{aligned} \lambda x^{A\wedge B}.\mathbf{t} \quad &\leftrightarrows \quad \lambda y^A.\lambda z^B.\mathbf{t}[y+z/x] \\ \lambda x^A.\lambda y^B.\mathbf{t} \quad &\leftrightarrows \quad \lambda z^{A\wedge B}.\mathbf{t}[\pi_A(z)/x, \pi_B(z)/y] \end{aligned}$$

Distributividad de la implicación sobre la conjunción

$$A \Rightarrow (B \wedge C) \quad \equiv \quad (A \Rightarrow B) \wedge (A \Rightarrow C)$$

induce

$\lambda x^A.(\mathbf{r} + \mathbf{s}) \leftrightarrows \lambda x^A.\mathbf{r} + \lambda x^A.\mathbf{s}$

Distributividad de la implicación sobre la conjunción

$$A \Rightarrow (B \wedge C) \quad \equiv \quad (A \Rightarrow B) \wedge (A \Rightarrow C)$$

induce

$$\lambda x^A.(\mathbf{r} + \mathbf{s}) \leftrightarrows \lambda x^A.\mathbf{r} + \lambda x^A.\mathbf{s} \qquad \text{y} \qquad \lambda x^A.\pi_B(\mathbf{r}) \leftrightarrows \pi_{A \Rightarrow B}(\lambda x^A.\mathbf{r})$$

Distributividad de la implicación sobre la conjunción

$$A \Rightarrow (B \wedge C) \quad \equiv \quad (A \Rightarrow B) \wedge (A \Rightarrow C)$$

induce

$$\lambda x^A.(\mathbf{r} + \mathbf{s}) \leftrightarrows \lambda x^A.\mathbf{r} + \lambda x^A.\mathbf{s} \qquad \text{y} \qquad \lambda x^A.\pi_B(\mathbf{r}) \leftrightarrows \pi_{A \Rightarrow B}(\lambda x^A.\mathbf{r})$$

Ejemplo

$$\dfrac{\dfrac{\vdash \lambda x^{A \wedge B}.x : (A \wedge B) \Rightarrow (A \wedge B)}{\vdash \lambda x^{A \wedge B}.x : ((A \wedge B) \Rightarrow A) \wedge ((A \wedge B) \Rightarrow B)}\ (\equiv)}{\vdash \pi_{(A \wedge B) \Rightarrow A}(\lambda x^{A \wedge B}.x) : (A \wedge B) \Rightarrow A}\ (\wedge_e)$$

$$\pi_{(A \wedge B) \Rightarrow A}(\lambda x^{A \wedge B}.x) \leftrightarrows \lambda x^{A \wedge B}.\pi_A(x)$$

Distributividad de la implicación sobre la conjunción

Otras opciones

$$A \Rightarrow (B \wedge C) \quad \equiv \quad (A \Rightarrow B) \wedge (A \Rightarrow C)$$

$$\begin{array}{rcl}
\lambda x^A.(\mathbf{r}+\mathbf{s}) & \leftrightarrows & \lambda x^A.\mathbf{r} + \lambda x^A.\mathbf{s} \\
\lambda x^A.\pi_B(\mathbf{r}) & \leftrightarrows & \pi_{A\Rightarrow B}(\lambda x^A.\mathbf{r})
\end{array}
\qquad
\begin{array}{rcl}
\Rightarrow_i, \wedge_i & \leftrightarrows & \wedge_i, \Rightarrow_i \\
\Rightarrow_i, \wedge_e & \leftrightarrows & \wedge_e, \Rightarrow_i
\end{array}$$

Distributividad de la implicación sobre la conjunción

Otras opciones

$$A \Rightarrow (B \wedge C) \quad \equiv \quad (A \Rightarrow B) \wedge (A \Rightarrow C)$$

$$\begin{array}{rcl@{\qquad\qquad}rcl}
\lambda x^A.(\mathbf{r}+\mathbf{s}) & \leftrightarrows & \lambda x^A.\mathbf{r} + \lambda x^A.\mathbf{s} & \Rightarrow_i, \wedge_i & \leftrightarrows & \wedge_i, \Rightarrow_i \\
\lambda x^A.\pi_B(\mathbf{r}) & \leftrightarrows & \pi_{A\Rightarrow B}(\lambda x^A.\mathbf{r}) & \Rightarrow_i, \wedge_e & \leftrightarrows & \wedge_e, \Rightarrow_i \\
\\
(\mathbf{r}+\mathbf{s})\mathbf{t} & \leftrightarrows & \mathbf{rt} + \mathbf{st} & \Rightarrow_e, \wedge_i & \leftrightarrows & \wedge_i, \Rightarrow_e \\
\pi_{A\Rightarrow B}(\mathbf{r})\mathbf{s} & \leftrightarrows & \pi_B(\mathbf{rs})\,^* & \Rightarrow_e, \wedge_e & \leftrightarrows & \wedge_e, \Rightarrow_e
\end{array}$$

* si $\mathbf{r} : A \Rightarrow (B \wedge C)$

α-equivalencia y substitución-$\equiv$

Reglas

- Si $A \equiv B$, $\mathbf{r} \leftrightarrows \mathbf{r}[A/B]$
- Si $\mathbf{r} =_\alpha \mathbf{r}'$, $\mathbf{r} \leftrightarrows \mathbf{r}'$

Ejemplo

$$\text{Sea } A \equiv B \qquad \overbrace{\lambda x^A.\mathbf{r} + \lambda y^B.\mathbf{s}}^{A \Rightarrow (C_1 \wedge C_2)}$$

α-equivalencia y substitución-$\equiv$

Reglas

- Si $A \equiv B$, $\mathbf{r} \leftrightarrows \mathbf{r}[A/B]$
- Si $\mathbf{r} =_\alpha \mathbf{r}'$, $\mathbf{r} \leftrightarrows \mathbf{r}'$

Ejemplo

$$\text{Sea } A \equiv B \qquad \overbrace{\lambda x^A.\mathbf{r} + \lambda y^B.\mathbf{s}}^{A \Rightarrow (C_1 \wedge C_2)} \leftrightarrows^* \lambda x^A.\mathbf{r} + \lambda x^A.\mathbf{s}[x/y][A/B]$$
$$\leftrightarrows \lambda x^A.(\mathbf{r} + \mathbf{s}[x/y][A/B])$$

Normalización

$\boxed{\textbf{r} \text{ está en forma normal si sólo puede seguir “reduciendo” con } \leftrightarrows}$

Forma normal

$$\mathrm{Red}(\mathbf{r}) = \{\mathbf{s} \mid \mathbf{r} \leftrightarrows^* \mathbf{r}' \hookrightarrow \mathbf{s}' \leftrightarrows^* \mathbf{s}\}$$

r está en forma normal si $\mathrm{Red}(\mathbf{r}) = \emptyset$

Normalización

$\boxed{\textbf{r} \text{ está en forma normal si sólo puede seguir “reduciendo” con } \leftrightarrows}$

Forma normal

$$\mathrm{Red}(\mathbf{r}) = \{\mathbf{s} \mid \mathbf{r} \leftrightarrows^* \mathbf{r}' \hookrightarrow \mathbf{s}' \leftrightarrows^* \mathbf{s}\}$$

r está en forma normal si $\mathrm{Red}(\mathbf{r}) = \emptyset$

Theorem (Normalización fuerte)

Si $\Gamma \vdash \mathbf{r} : A$ *entonces* **r** *es fuertemente normalizante*

Prueba. Método de reducibilidad

Computando con este cálculo

Pares

$$\pi_{Nat}(3,4) \to 3 \quad \text{y} \quad \pi_{Nat}(3,4) \to 4$$

Pero es posible encodear pares que se comporten de manera estándar

Computando con este cálculo

Pares

$$\pi_{Nat}(3,4) \to 3 \quad \text{y} \quad \pi_{Nat}(3,4) \to 4$$

Pero es posible encodear pares que se comporten de manera estándar

Estándar	Encodeado
$\langle \mathbf{r}, \mathbf{s} \rangle : A \wedge A$	$\lambda x^{\mathbb{1}}.\mathbf{r} + \lambda x^{\mathbb{2}}.\mathbf{s} : \mathbb{1} \Rightarrow A \wedge \mathbb{2} \Rightarrow A$
$\pi_1 \langle \mathbf{r}, \mathbf{s} \rangle$	$\pi_{\mathbb{1} \Rightarrow A}(\lambda x^{\mathbb{1}}.\mathbf{r} + \lambda x^{\mathbb{2}}.\mathbf{s}) y^{\mathbb{1}}$

$$\{\pi_{\mathbb{1} \Rightarrow Nat}([3]_{\mathbb{1}}, [4]_{\mathbb{2}})\}_{\mathbb{1}} \to 3 \quad \text{y} \quad \{\pi_{\mathbb{2} \Rightarrow Nat}([3]_{\mathbb{1}}, [4]_{\mathbb{2}})\}_{\mathbb{2}} \to 4$$

con $$[\mathbf{r}]_A = \lambda x^A.\mathbf{r}$$ $$\{\mathbf{r}\}_A = \mathbf{r} y^A$$

Computando con este cálculo

Booleanos

$$
\begin{aligned}
(\lambda x^A.\lambda y^A.x)\mathbf{rs} &\rightleftarrows (\lambda x^A.\lambda y^A.x)(\mathbf{r},\mathbf{s}) \\
&\rightleftarrows (\lambda x^A.\lambda y^A.x)(\mathbf{s},\mathbf{r}) \\
&\rightleftarrows (\lambda x^A.\lambda y^A.x)\mathbf{sr} \to^* \mathbf{s}
\end{aligned}
$$

Entonces

$\mathbf{true} \sim \mathbf{false}$

Computando con este cálculo

Booleanos

$$
\begin{aligned}
(\lambda x^A.\lambda y^A.x)\mathbf{rs} &\rightleftarrows (\lambda x^A.\lambda y^A.x)(\mathbf{r},\mathbf{s}) \\
&\rightleftarrows (\lambda x^A.\lambda y^A.x)(\mathbf{s},\mathbf{r}) \\
&\rightleftarrows (\lambda x^A.\lambda y^A.x)\mathbf{sr} \to^* \mathbf{s}
\end{aligned}
$$

Entonces
true $\sim$ **false**

$A \Rightarrow A \Rightarrow A$
es un singletón!

Computando con este cálculo

Booleanos

$$
\begin{aligned}
(\lambda x^A.\lambda y^A.x)\mathbf{rs} &\rightleftarrows (\lambda x^A.\lambda y^A.x)(\mathbf{r},\mathbf{s}) \\
&\rightleftarrows (\lambda x^A.\lambda y^A.x)(\mathbf{s},\mathbf{r}) \\
&\rightleftarrows (\lambda x^A.\lambda y^A.x)\mathbf{sr} \to^* \mathbf{s}
\end{aligned}
$$

Entonces
true $\sim$ **false**

$A \Rightarrow A \Rightarrow A$
es un singletón!

Pero $(A \Rightarrow B) \Rightarrow B \Rightarrow B$ no lo es

$\boxed{\text{Así que también es posible encodear los booleanos}}$

Contenido de la charla

Resumen

¿Qué hicimos?

Definimos un nuevo cálculo donde proposiciones isomorfas tienen las mismas pruebas

Resumen

¿Qué hicimos?

Definimos un nuevo cálculo donde proposiciones isomorfas tienen las mismas pruebas

¿Porqué?

Resumen

¿Qué hicimos?

Definimos un nuevo cálculo donde proposiciones isomorfas tienen las mismas pruebas

¿Porqué?

Si $A \equiv B$, una **prueba** de A en una biblioteca no debería ser distinguible de una prueba de B

$\dfrac{A \quad B}{A \wedge B}$ y $\dfrac{B \quad A}{B \wedge A}$ son lo mismo!

Si $A \equiv B$, y una **función** está definida sobre A debería poder usarse como B

Si $\mathbf{f}\ \langle \mathbf{a}, \mathbf{b} \rangle$ es válido, también debería serlo $\mathbf{f}\ \mathbf{a}\ \mathbf{b}$, o incluso $\mathbf{f}\ \mathbf{a}$.