

Teoría de Tipos Simples

Juan Carlos Agudelo-Agudelo

Instituto de Matemáticas
Facultad de Ciencias Exactas y Naturales
Universidad de Antioquia



“Type theory enables one to provide a ‘coded’ version – i.e. a full formalisation – of many mathematical topics. The formal system underlying type theory forces the user to work in a very precise manner. The real power of type theory is that well-formedness of the formalised expressions implies logical and mathematical correctness of the original content.” [Nederpelt and Geuvers, 2014, pág. XV]



- 1 Breve reseña histórica
- 2 Cálculo lambda sin tipos
 - Términos del cálculo lambda
 - α -conversión
 - β -reducción y β -conversión
 - Combinador de punto fijo
 - Expresividad del cálculo lambda
- 3 Cálculo lambda simplemente tipificado
 - Tipificado a la Curry
 - Tipificado a la Church
- 4 Lógica intuicionista
 - Interpretación BHK
 - Sistema de deducción natural
 - Semántica de Kripke
- 5 Correspondencia de Curry-Howard
- 6 Referencias bibliográficas



- 1 Breve reseña histórica
- 2 Cálculo lambda sin tipos
 - Términos del cálculo lambda
 - α -conversión
 - β -reducción y β -conversión
 - Combinador de punto fijo
 - Expresividad del cálculo lambda
- 3 Cálculo lambda simplemente tipificado
 - Tipificado a la Curry
 - Tipificado a la Church
- 4 Lógica intuicionista
 - Interpretación BHK
 - Sistema de deducción natural
 - Semántica de Kripke
- 5 Correspondencia de Curry-Howard
- 6 Referencias bibliográficas



Breve reseña histórica

- (1908) Bertran Russel propone la **teoría de tipos ramificada** (como solución a las paradojas).



Breve reseña histórica

- (1908) Bertran Russel propone la **teoría de tipos ramificada** (como solución a las paradojas).
- (1920s) Leon Chwistek y Frank P. Ramsey propusieron una teoría de tipos sin ramificación, ahora conocida como **teoría de tipos simples**.



Breve reseña histórica

- (1908) Bertran Russel propone la **teoría de tipos ramificada** (como solución a las paradojas).
- (1920s) Leon Chwistek y Frank P. Ramsey propusieron una teoría de tipos sin ramificación, ahora conocida como **teoría de tipos simples**.
- (1940) Alonso Church introdujo el **cálculo lambda simplemente tipificado**.



Breve reseña histórica

- (1908) Bertran Russel propone la **teoría de tipos ramificada** (como solución a las paradojas).
- (1920s) Leon Chwistek y Frank P. Ramsey propusieron una teoría de tipos sin ramificación, ahora conocida como **teoría de tipos simples**.
- (1940) Alonso Church introdujo el **cálculo lambda simplemente tipificado**.
- (1958) Haskell Curry estableció la **correspondencia entre al cálculo lambda simplemente tipificado y el fragmento implicativo de la lógica intuicionista**.



Breve reseña histórica

- (1908) Bertran Russel propone la **teoría de tipos ramificada** (como solución a las paradojas).
- (1920s) Leon Chwistek y Frank P. Ramsey propusieron una teoría de tipos sin ramificación, ahora conocida como **teoría de tipos simples**.
- (1940) Alonso Church introdujo el **cálculo lambda simplemente tipificado**.
- (1958) Haskell Curry estableció la **correspondencia entre al cálculo lambda simplemente tipificado y el fragmento implicativo de la lógica intuicionista**.
- (1969) William A. Howard y Nicolaas G. de Bruijn extendieron la correspondencia a la lógica de predicados de primer orden, lo que ahora es conocido como **correspondencia de Curry-Howard**.



Breve reseña histórica

- (1908) Bertran Russel propone la **teoría de tipos ramificada** (como solución a las paradojas).
- (1920s) Leon Chwistek y Frank P. Ramsey propusieron una teoría de tipos sin ramificación, ahora conocida como **teoría de tipos simples**.
- (1940) Alonso Church introdujo el **cálculo lambda simplemente tipificado**.
- (1958) Haskell Curry estableció la **correspondencia entre al cálculo lambda simplemente tipificado y el fragmento implicativo de la lógica intuicionista**.
- (1969) William A. Howard y Nicolaas G. de Bruijn extendieron la correspondencia a la lógica de predicados de primer orden, lo que ahora es conocido como **correspondencia de Curry-Howard**.
- (1970s) Per Martin-Löf propuso varias versiones diferentes de **teorías de tipos intuicionistas**.



Breve reseña histórica

- (1908) Bertran Russel propone la **teoría de tipos ramificada** (como solución a las paradojas).
- (1920s) Leon Chwistek y Frank P. Ramsey propusieron una teoría de tipos sin ramificación, ahora conocida como **teoría de tipos simples**.
- (1940) Alonso Church introdujo el **cálculo lambda simplemente tipificado**.
- (1958) Haskell Curry estableció la **correspondencia entre al cálculo lambda simplemente tipificado y el fragmento implicativo de la lógica intuicionista**.
- (1969) William A. Howard y Nicolaas G. de Bruijn extendieron la correspondencia a la lógica de predicados de primer orden, lo que ahora es conocido como **correspondencia de Curry-Howard**.
- (1970s) Per Martin-Löf propuso varias versiones diferentes de **teorías de tipos intuicionistas**.
- Hoy día existen varios **asistentes de pruebas** basados en teorías de tipos: Agda, Coq, Lean, etc.



Índice

- 1 Breve reseña histórica
- 2 Cálculo lambda sin tipos
 - Términos del cálculo lambda
 - α -conversión
 - β -reducción y β -conversión
 - Combinador de punto fijo
 - Expresividad del cálculo lambda
- 3 Cálculo lambda simplemente tipificado
 - Tipificado a la Curry
 - Tipificado a la Church
- 4 Lógica intuicionista
 - Interpretación BHK
 - Sistema de deducción natural
 - Semántica de Kripke
- 5 Correspondencia de Curry-Howard
- 6 Referencias bibliográficas



Cálculo lambda sin tipos

Referencias usadas en esta sección: [Nederpelt and Geuvers, 2014, Cap. 1], [Sørensen and Urzyczyn, 2006, Cap. 1], [Barendregt and Barendsen, 2000].



Cálculo lambda sin tipos

Referencias usadas en esta sección: [Nederpelt and Geuvers, 2014, Cap. 1], [Sørensen and Urzyczyn, 2006, Cap. 1], [Barendregt and Barendsen, 2000].

- El cálculo lambda fue propuesto por Alonso Church en la década de 1930, inicialmente como un sistema para formalizar la matemática.



Cálculo lambda sin tipos

Referencias usadas en esta sección: [Nederpelt and Geuvers, 2014, Cap. 1], [Sørensen and Urzyczyn, 2006, Cap. 1], [Barendregt and Barendsen, 2000].

- El cálculo lambda fue propuesto por Alonso Church en la década de 1930, inicialmente como un sistema para formalizar la matemática.
- En 1935, Stephen Kleene y J. Barkley Rosser probaron que el sistema original propuesto por Church era lógicamente inconsistente.



Cálculo lambda sin tipos

Referencias usadas en esta sección: [Nederpelt and Geuvers, 2014, Cap. 1], [Sørensen and Urzyczyn, 2006, Cap. 1], [Barendregt and Barendsen, 2000].

- El cálculo lambda fue propuesto por Alonso Church en la década de 1930, inicialmente como un sistema para formalizar la matemática.
- En 1935, Stephen Kleene y J. Barkley Rosser probaron que el sistema original propuesto por Church era lógicamente inconsistente.
- En 1936, Church redujo el sistema y dejó solo lo que tenía que ver con la formalización de la noción de computación, el sistema obtenido es conocido hoy como **cálculo lambda sin tipos**.



Cálculo lambda sin tipos

Referencias usadas en esta sección: [Nederpelt and Geuvers, 2014, Cap. 1], [Sørensen and Urzyczyn, 2006, Cap. 1], [Barendregt and Barendsen, 2000].

- El cálculo lambda fue propuesto por Alonso Church en la década de 1930, inicialmente como un sistema para formalizar la matemática.
- En 1935, Stephen Kleene y J. Barkley Rosser probaron que el sistema original propuesto por Church era lógicamente inconsistente.
- En 1936, Church redujo el sistema y dejó solo lo que tenía que ver con la formalización de la noción de computación, el sistema obtenido es conocido hoy como **cálculo lambda sin tipos**.

El cálculo lambda es un formalismo que permite expresar funciones y calcular funciones por medio de reglas de “reducción” de términos, simulando la manera como se hacen cálculos en diversas áreas de las matemáticas.



Ejemplo 1

$$\begin{aligned}(2 + 1) * (4 + 2) &\rightarrow 3 * (4 + 2) \\ &\rightarrow (3 * 4) + (3 * 2) \\ &\rightarrow 12 + (3 * 2) \\ &\rightarrow 12 + 6 \\ &\rightarrow 18\end{aligned}$$



Ejemplo 1

$$\begin{aligned}(2 + 1) * (4 + 2) &\rightarrow 3 * (4 + 2) \\ &\rightarrow (3 * 4) + (3 * 2) \\ &\rightarrow 12 + (3 * 2) \\ &\rightarrow 12 + 6 \\ &\rightarrow 18\end{aligned}$$

En el cálculo lambda, dado un término $t(x)$ (que puede depender de la variable x), se usa $\lambda x.t(x)$ para enfatizar el “rol abstracto” de la variable x en $t(x)$. Por ejemplo, $\lambda x.x + 5$ expresa la función que se obtiene al substituir la variable x por un valor específico en $x + 5$. De esta manera, $(\lambda x.x + 5)(2)$ expresa el resultado de aplicar dicha función al valor 2.



Términos del cálculo lambda

Se fija el **conjunto de variables** $V = \{x, y, z, \dots, x_1, y_1, z_1, \dots\}$.



Términos del cálculo lambda

Se fija el **conjunto de variables** $V = \{x, y, z, \dots, x_1, y_1, z_1, \dots\}$.

Definición 1

El **conjunto Λ de λ -términos** se define inductivamente por:

- 1 (Variable) Si $v \in V$, entonces $v \in \Lambda$.
- 2 (Abstracción) Si $v \in V$ y $M \in \Lambda$, entonces $(\lambda v.M) \in \Lambda$ (a v se le llama **variable vinculante**).
- 3 (Aplicación) Si $M, N \in \Lambda$, entonces $(MN) \in \Lambda$.



Términos del cálculo lambda

Se fija el **conjunto de variables** $V = \{x, y, z, \dots, x_1, y_1, z_1, \dots\}$.

Definición 1

El **conjunto Λ de λ -términos** se define inductivamente por:

- ❶ (Variable) Si $v \in V$, entonces $v \in \Lambda$.
- ❷ (Abstracción) Si $v \in V$ y $M \in \Lambda$, entonces $(\lambda v.M) \in \Lambda$ (a v se le llama **variable vinculante**).
- ❸ (Aplicación) Si $M, N \in \Lambda$, entonces $(MN) \in \Lambda$.

Observación: por medio de una “sintaxis abstracta”

$$\Lambda = V \mid (\lambda V.\Lambda) \mid (\Lambda\Lambda).$$



Términos del cálculo lambda

Se fija el **conjunto de variables** $V = \{x, y, z, \dots, x_1, y_1, z_1, \dots\}$.

Definición 1

El **conjunto Λ de λ -términos** se define inductivamente por:

- ❶ (Variable) Si $v \in V$, entonces $v \in \Lambda$.
- ❷ (Abstracción) Si $v \in V$ y $M \in \Lambda$, entonces $(\lambda v.M) \in \Lambda$ (a v se le llama **variable vinculante**).
- ❸ (Aplicación) Si $M, N \in \Lambda$, entonces $(MN) \in \Lambda$.

Observación: por medio de una “sintaxis abstracta”

$$\Lambda = V \mid (\lambda V.\Lambda) \mid (\Lambda\Lambda).$$

Ejemplo 2

Las siguientes expresiones son λ -términos: $x, y, z, (xx), (xy), (z(xy)), (\lambda x.(xx)), ((\lambda x.(xx))y), (y(\lambda x.(xx))), (\lambda x.(\lambda y.(y(xy))))$.

Términos del cálculo lambda

Notación:

- Se usará \equiv para denotar la **identidad sintáctica**.



Términos del cálculo lambda

Notación:

- Se usará \equiv para denotar la **identidad sintáctica**.
- Se usarán variantes de los elementos de V con $'$ para denotar variables.



Términos del cálculo lambda

Notación:

- Se usará \equiv para denotar la **identidad sintáctica**.
- Se usarán variantes de los elementos de V con $'$ para denotar variables.
- Se usarán letras mayúsculas L, M, N, \dots para denotar elementos de Λ .



Términos del cálculo lambda

Notación:

- Se usará \equiv para denotar la **identidad sintáctica**.
- Se usarán variantes de los elementos de V con $'$ para denotar variables.
- Se usarán letras mayúsculas L, M, N, \dots para denotar elementos de Λ .

Definición 2

El **conjunto de subtérminos** de un λ -término se define inductivamente por:

- 1 $\text{Sub}(v) = \{v\}$, si $v \in V$.
- 2 $\text{Sub}((\lambda v.M)) = \{(\lambda v.M)\} \cup \text{Sub}(M)$.
- 3 $\text{Sub}((MN)) = \{(MN)\} \cup \text{Sub}(M) \cup \text{Sub}(N)$.

Un **subtérmino propio de M** es un subtérmino de M que es diferente a M .



Términos del cálculo lambda

Notación: para disminuir el uso de paréntesis:

- Paréntesis externos en λ -términos serán omitidos.



Términos del cálculo lambda

Notación: para disminuir el uso de paréntesis:

- Paréntesis externos en λ -términos serán omitidos.
- Se asume que las aplicaciones son asociativas a izquierda: MNL abrevia $(MN)L$.



Términos del cálculo lambda

Notación: para disminuir el uso de paréntesis:

- Paréntesis externos en λ -términos serán omitidos.
- Se asume que las aplicaciones son asociativas a izquierda: MNL abrevia $(MN)L$.
- Las aplicaciones tienen precedencia sobre las abstracciones: $\lambda x.MN$ abrevia $\lambda x.(MN)$.



Términos del cálculo lambda

Notación: para disminuir el uso de paréntesis:

- Paréntesis externos en λ -términos serán omitidos.
- Se asume que las aplicaciones son asociativas a izquierda: MNL abrevia $(MN)L$.
- Las aplicaciones tienen precedencia sobre las abstracciones: $\lambda x.MN$ abrevia $\lambda x.(MN)$.
- Abstracciones pueden ser combinadas usando un solo λ y son asociativas a derecha: $\lambda x_1 x_2 \dots x_n.M$ abrevia $\lambda x_1.(\lambda x_2.(\dots(\lambda x_n.M)))$.



Definición 3

El **conjunto de variables libres** de un λ -término se define inductivamente por:

- ① $FV(v) = \{v\}$, si $v \in V$.
- ② $FV(\lambda v.M) = FV(M) \setminus \{v\}$.
- ③ $FV(MN) = FV(M) \cup FV(N)$.

Los λ -términos sin variables libres son llamados **combinadores**.



Términos del cálculo lambda

Definición 3

El **conjunto de variables libres** de un λ -término se define inductivamente por:

- ① $FV(v) = \{v\}$, si $v \in V$.
- ② $FV(\lambda v.M) = FV(M) \setminus \{v\}$.
- ③ $FV(MN) = FV(M) \cup FV(N)$.

Los λ -términos sin variables libres son llamados **combinadores**.

Definición 4

El **renombramiento de la variable x por y en el λ -término M** , denotado $M^{x \rightarrow y}$, es el resultado de sustituir toda ocurrencia libre de x por y en M .



Definición 5

La relación $=_{\alpha}$ de α -conversión (o α -equivalencia) se define así:

- (Renombramiento) $\lambda x.M =_{\alpha} \lambda y.M^{x \rightarrow y}$ si $y \notin FV(M)$ y y no es una variable vinculante en M .
- (Compatibilidad) Si $M =_{\alpha} N$, entonces $ML =_{\alpha} NL$, $LM =_{\alpha} LN$ y $\lambda z.M =_{\alpha} \lambda z.N$ (para cualquier variable z).
- (Reflexividad) $M =_{\alpha} M$.
- (Simetría) Si $M =_{\alpha} N$, entonces $N =_{\alpha} M$.
- (Transitividad) Si $M =_{\alpha} N$ y $N =_{\alpha} L$, entonces $M =_{\alpha} L$.



Definición 5

La relación $=_\alpha$ de α -conversión (o α -equivalencia) se define así:

- (Renombramiento) $\lambda x.M =_\alpha \lambda y.M^{x \rightarrow y}$ si $y \notin \text{FV}(M)$ y y no es una variable vinculante en M .
- (Compatibilidad) Si $M =_\alpha N$, entonces $ML =_\alpha NL$, $LM =_\alpha LN$ y $\lambda z.M =_\alpha \lambda z.N$ (para cualquier variable z).
- (Reflexividad) $M =_\alpha M$.
- (Simetría) Si $M =_\alpha N$, entonces $N =_\alpha M$.
- (Transitividad) Si $M =_\alpha N$ y $N =_\alpha L$, entonces $M =_\alpha L$.

Ejemplo 3

$$\begin{aligned}\lambda x.x(\lambda y.xyz) &=_\alpha \lambda u.u(\lambda y.uyz), & \lambda x.x(\lambda y.xyz) &=_\alpha \lambda u.u(\lambda v.uvz), \\ \lambda x.x(\lambda y.xyz) &\neq_\alpha \lambda z.z(\lambda y.zyz), & \lambda x.x(\lambda y.xyz) &\neq_\alpha \lambda y.y(\lambda y.yyz).\end{aligned}$$

Definición 6

La **sustitución de la variable v por el término N** en λ -términos se define inductivamente por:

- $v[v := N] \equiv N$.
- $w[v := N] \equiv w$, si $w \not\equiv v$.
- $(PQ)[v := N] \equiv (P[v := N]Q[v := N])$.
- $(\lambda u.P)[v := N] \equiv \lambda w.(P^{u \rightarrow w}[v := N])$, si $\lambda w.P^{u \rightarrow w} =_{\alpha} \lambda u.P$ y $w \notin \text{FV}(N)$ (si $u \notin \text{FV}(N)$ o $v \notin \text{FV}(P)$ se puede tomar $w \equiv u$).



Definición 6

La **sustitución de la variable v por el término N** en λ -términos se define inductivamente por:

- $v[v := N] \equiv N$.
- $w[v := N] \equiv w$, si $w \not\equiv v$.
- $(PQ)[v := N] \equiv (P[v := N]Q[v := N])$.
- $(\lambda u.P)[v := N] \equiv \lambda w.(P^{u \rightarrow w}[v := N])$, si $\lambda w.P^{u \rightarrow w} =_{\alpha} \lambda u.P$ y $w \notin \text{FV}(N)$ (si $u \notin \text{FV}(N)$ o $v \notin \text{FV}(P)$ se puede tomar $w \equiv u$).

Ejemplo 4

$$\begin{aligned}(\lambda y.yx)[x := xy] &\equiv \lambda z.z(xy), & (\lambda xy.zzx)[z := y] &\equiv \lambda xu.yyx, \\(\lambda x.yx)[x := xy] &\equiv \lambda z.yz, & (\lambda x.yx)[x := xy] &\equiv \lambda x.yx.\end{aligned}$$

Lema 1

Si $M =_{\alpha} N$, entonces $FV(M) = FV(N)$.



Lema 1

Si $M =_{\alpha} N$, entonces $FV(M) = FV(N)$.

Lema 2

Si $M_1 =_{\alpha} M_2$ y $N_1 =_{\alpha} N_2$, entonces:

- ❶ $M_1 N_1 =_{\alpha} M_2 N_2$.
- ❷ $\lambda x. M_1 =_{\alpha} \lambda x. M_2$.
- ❸ $M_1[x := N_1] =_{\alpha} M_2[x := N_2]$.



Lema 1

Si $M =_{\alpha} N$, entonces $FV(M) = FV(N)$.

Lema 2

Si $M_1 =_{\alpha} M_2$ y $N_1 =_{\alpha} N_2$, entonces:

- ① $M_1 N_1 =_{\alpha} M_2 N_2$.
- ② $\lambda x. M_1 =_{\alpha} \lambda x. M_2$.
- ③ $M_1[x := N_1] =_{\alpha} M_2[x := N_2]$.

Observación: de aquí en adelante, términos α -equivalentes serán considerados como iguales, y abusando un poco de la notación se usará también \equiv para términos α -equivalentes.



Lema 1

Si $M =_{\alpha} N$, entonces $FV(M) = FV(N)$.

Lema 2

Si $M_1 =_{\alpha} M_2$ y $N_1 =_{\alpha} N_2$, entonces:

- ① $M_1 N_1 =_{\alpha} M_2 N_2$.
- ② $\lambda x. M_1 =_{\alpha} \lambda x. M_2$.
- ③ $M_1[x := N_1] =_{\alpha} M_2[x := N_2]$.

Observación: de aquí en adelante, términos α -equivalentes serán considerados como iguales, y abusando un poco de la notación se usará también \equiv para términos α -equivalentes.

Convención de Barendregt: se usarán variables diferentes para cada una de las lambda abstracciones en un λ -término, y dichas variables serán diferentes a las variables libres.



Definición 7

La relación \rightarrow_β de β -reducción (en un paso) se define así:

- ① (Base) $(\lambda x.M)N \rightarrow_\beta M[x := N]$.
- ② (Compatibilidad) Si $M \rightarrow_\beta N$, entonces $ML \rightarrow_\beta NL$, $LM \rightarrow_\beta LN$ y $\lambda x.M \rightarrow_\beta \lambda x.N$.

Una expresión de la forma $(\lambda x.M)N$ es llamada **redex** (o β -redex), y se dice que la expresión $M[x := N]$ es su **contracción**.



β -reducción y β -conversión

Definición 7

La relación \rightarrow_β de β -reducción (en un paso) se define así:

- 1 (Base) $(\lambda x.M)N \rightarrow_\beta M[x := N]$.
- 2 (Compatibilidad) Si $M \rightarrow_\beta N$, entonces $ML \rightarrow_\beta NL$, $LM \rightarrow_\beta LN$ y $\lambda x.M \rightarrow_\beta \lambda x.N$.

Una expresión de la forma $(\lambda x.M)N$ es llamada **redex** (o β -redex), y se dice que la expresión $M[x := N]$ es su **contracción**.

Ejemplo 5

$$\begin{array}{ll} (\lambda x.x(yx))z \rightarrow_\beta z(yz) & (\lambda y.(\lambda x.xy)z)v \rightarrow_\beta (\lambda x.xv)z \\ (\lambda y.(\lambda x.xy)z)v \rightarrow_\beta (\lambda y.zy)v & (\lambda x.xx)(\lambda x.xx) \rightarrow_\beta (\lambda x.xx)(\lambda x.xx) \end{array}$$



Definición 8

La relación \rightarrow_β de β -reducción (en zero o más pasos) se define así:
 $M \twoheadrightarrow_\beta N$ si existen términos M_0, M_1, \dots, M_n , para algún $n \geq 0$, tales que $M_0 \equiv M$, $M_n \equiv N$ y $M_i \rightarrow_\beta M_{i+1}$ para todo i tal que $0 \leq i < n$.



β -reducción y β -conversión

Definición 8

La relación \rightarrow_β de β -reducción (en zero o más pasos) se define así:
 $M \twoheadrightarrow_\beta N$ si existen términos M_0, M_1, \dots, M_n , para algún $n \geq 0$, tales que $M_0 \equiv M$, $M_n \equiv N$ y $M_i \rightarrow_\beta M_{i+1}$ para todo i tal que $0 \leq i < n$.

Ejemplo 6

$(\lambda y.(\lambda x.xy)z)v \twoheadrightarrow_\beta zv$, pues $(\lambda y.(\lambda x.xy)z)v \rightarrow_\beta (\lambda x.xv)z \rightarrow_\beta zv$.



Definición 9

Dos λ -términos M y N son β -convertibles (o β -iguales), denotado $M =_{\beta} N$, si existen términos M_0, M_1, \dots, M_n , para algún $n \geq 0$, tales que $M_0 \equiv M$, $M_n \equiv N$ y para todo i tal que $0 \leq i < n$ se tiene que $M_i \rightarrow_{\beta} M_{i+1}$ o $M_i \leftarrow_{\beta} M_{i+1}$ (donde \leftarrow_{β} es la relación inversa de \rightarrow_{β}).



Definición 9

Dos λ -términos M y N son β -convertibles (o β -iguales), denotado $M =_{\beta} N$, si existen términos M_0, M_1, \dots, M_n , para algún $n \geq 0$, tales que $M_0 \equiv M$, $M_n \equiv N$ y para todo i tal que $0 \leq i < n$ se tiene que $M_i \rightarrow_{\beta} M_{i+1}$ o $M_i \leftarrow_{\beta} M_{i+1}$ (donde \leftarrow_{β} es la relación inversa de \rightarrow_{β}).

Ejemplo 7

$(\lambda x.xv)z =_{\beta} (\lambda y.zy)v$, pues $(\lambda x.xv)z \rightarrow_{\beta} zv \leftarrow_{\beta} (\lambda y.zy)v$ (también se tiene que $(\lambda x.xv)z \leftarrow_{\beta} (\lambda y.(\lambda x.xy)z)v \rightarrow_{\beta} (\lambda y.zy)v$).



β -reducción y β -conversión

Definición 9

Dos λ -términos M y N son β -convertibles (o β -iguales), denotado $M =_{\beta} N$, si existen términos M_0, M_1, \dots, M_n , para algún $n \geq 0$, tales que $M_0 \equiv M$, $M_n \equiv N$ y para todo i tal que $0 \leq i < n$ se tiene que $M_i \rightarrow_{\beta} M_{i+1}$ o $M_i \leftarrow_{\beta} M_{i+1}$ (donde \leftarrow_{β} es la relación inversa de \rightarrow_{β}).

Ejemplo 7

$(\lambda x.xv)z =_{\beta} (\lambda y.zy)v$, pues $(\lambda x.xv)z \rightarrow_{\beta} zv \leftarrow_{\beta} (\lambda y.zy)v$ (también se tiene que $(\lambda x.xv)z \leftarrow_{\beta} (\lambda y.(\lambda x.xy)z)v \rightarrow_{\beta} (\lambda y.zy)v$).

Lema 3

- 1 Si $M \twoheadrightarrow_{\beta} N$ o $N \twoheadrightarrow_{\beta} M$, entonces $M =_{\beta} N$.
- 2 $=_{\beta}$ es una relación de equivalencia.

Definición 10

- 1 M está en forma β -normal (o es una β -nf) si no contiene ningún redex.
- 2 M tiene una forma β -normal (o es β -normalizable) si existe N tal que N es una β -nf y $M =_{\beta} N$.



Definición 10

- 1 M está en forma β -normal (o es una β -nf) si no contiene ningún redex.
- 2 M tiene una forma β -normal (o es β -normalizable) si existe N tal que N es una β -nf y $M =_{\beta} N$.

Ejemplo 8

- 1 $(\lambda y.(\lambda x.xy)z)v$ no es una β -nf, pero es β -normalizable.
- 2 $(\lambda x.xx)(\lambda x.xx)$ no es una β -nf ni es β -normalizable.
- 3 $(\lambda x.xxx)(\lambda x.xxx)$ no es una β -nf ni es β -normalizable.
- 4 $(\lambda u.v)((\lambda x.xx)(\lambda x.xx))$ no es una β -nf, pero es β -normalizable.



β -reducción y β -conversión

Definición 10

- 1 M está en forma β -normal (o es una β -nf) si no contiene ningún redex.
- 2 M tiene una forma β -normal (o es β -normalizable) si existe N tal que N es una β -nf y $M =_{\beta} N$.

Ejemplo 8

- 1 $(\lambda y.(\lambda x.xy)z)v$ no es una β -nf, pero es β -normalizable.
- 2 $(\lambda x.xx)(\lambda x.xx)$ no es una β -nf ni es β -normalizable.
- 3 $(\lambda x.xxx)(\lambda x.xxx)$ no es una β -nf ni es β -normalizable.
- 4 $(\lambda u.v)((\lambda x.xx)(\lambda x.xx))$ no es una β -nf, pero es β -normalizable.

Lema 4

Si M es una β -nf, entonces $M \rightarrow_{\beta} N$ implica $M \equiv N$.

Definición 11

- 1 Un **camino de reducción finito desde M** es una secuencia de términos N_0, N_1, \dots, N_n tales que $N_0 \equiv M$ y $N_i \rightarrow_{\beta} N_{i+1}$ para todo i tal que $0 \leq i < n$.
- 2 Un **camino de reducción infinito desde M** es una secuencia de términos N_0, N_1, \dots tales que $N_0 \equiv M$ y $N_i \rightarrow_{\beta} N_{i+1}$ para todo $i \in \mathbb{N}$.



Definición 11

- 1 Un **camino de reducción finito desde M** es una secuencia de términos N_0, N_1, \dots, N_n tales que $N_0 \equiv M$ y $N_i \rightarrow_{\beta} N_{i+1}$ para todo i tal que $0 \leq i < n$.
- 2 Un **camino de reducción infinito desde M** es una secuencia de términos N_0, N_1, \dots tales que $N_0 \equiv M$ y $N_i \rightarrow_{\beta} N_{i+1}$ para todo $i \in \mathbb{N}$.

Definición 12

- 1 M es **débilmente normalizable** si existe N en β -nf tal que $M \rightarrow_{\beta} N$.
- 2 M es **fuertemente normalizable** si no existe un camino de reducción infinito desde M .



Ejemplo 9

- 1 $(\lambda y.(\lambda x.xy)z)v$ es fuertemente normalizable.
- 2 $(\lambda x.xx)(\lambda x.xx)$ no es (débil ni fuertemente) normalizable.
- 3 $(\lambda u.v)((\lambda x.xx)(\lambda x.xx))$ es débilmente normalizable, pero no es fuertemente normalizable.



Ejemplo 9

- 1 $(\lambda y.(\lambda x.xy)z)v$ es fuertemente normalizable.
- 2 $(\lambda x.xx)(\lambda x.xx)$ no es (débil ni fuertemente) normalizable.
- 3 $(\lambda u.v)((\lambda x.xx)(\lambda x.xx))$ es débilmente normalizable, pero no es fuertemente normalizable.

Teorema 1 (Propiedad de Church-Rosser o de confluencia)

Si $M \rightarrow_{\beta} N_1$ y $M \rightarrow_{\beta} N_2$, entonces existe N_3 tal que $N_1 \rightarrow_{\beta} N_3$ y $N_2 \rightarrow_{\beta} N_3$.

Demostración.

Ver en [Barendregt and Barendsen, 2000, págs. 25-29].



Corolario 1

Si $M =_{\beta} N$, entonces existe L tal que $M \rightarrow_{\beta} L$ y $N \rightarrow_{\beta} L$.



Corolario 1

Si $M =_{\beta} N$, entonces existe L tal que $M \rightarrow_{\beta} L$ y $N \rightarrow_{\beta} L$.

Lema 5

- 1 *Si N es una forma β -normal de M , entonces $M \rightarrow_{\beta} N$.*
- 2 *Todo λ -término tiene máximo una forma β -normal.*



Combinador de punto fijo

Teorema 2 (de punto fijo)

Para todo $L \in \Lambda$ existe $M \in \Lambda$ tal que $LM =_{\beta} M$.

Demostración.

Tome $M := (\lambda x.L(xx))(\lambda x.L(xx))$.



Combinador de punto fijo

Teorema 2 (de punto fijo)

Para todo $L \in \Lambda$ existe $M \in \Lambda$ tal que $LM =_{\beta} M$.

Demostración.

Tome $M := (\lambda x.L(xx))(\lambda x.L(xx))$.



Observación: abstrayendo sobre M , se obtiene el **combinador de punto fijo de Curry** $\mathbf{Y} := \lambda y.(\lambda x.y(xx))(\lambda x.y(xx))$ (se tiene que $L(\mathbf{Y}L) =_{\beta} \mathbf{Y}L$).
Existen otros combinadores de punto fijo.



Definición 13

Combinadores usuales:

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

$$\mathbf{K} \equiv \lambda xy.x \ (\mathbf{K}MN \rightarrow_{\beta} M),$$

$$\mathbf{K}_* \equiv \lambda xy.y \ (\mathbf{K}_*MN \rightarrow_{\beta} N),$$

$$\mathbf{S} \equiv \lambda xyz.xz(yz) \ (\mathbf{S}MNL \rightarrow_{\beta} ML(NL)).$$



Definición 13

Combinadores usuales:

$$\begin{aligned}\mathbf{I} &\equiv \lambda x.x \ (\mathbf{I}M \rightarrow_{\beta} M), \\ \mathbf{K} &\equiv \lambda xy.x \ (\mathbf{K}MN \rightarrow_{\beta} M), \\ \mathbf{K}_* &\equiv \lambda xy.y \ (\mathbf{K}_*MN \rightarrow_{\beta} N), \\ \mathbf{S} &\equiv \lambda xyz.xz(yz) \ (\mathbf{S}MNL \rightarrow_{\beta} ML(NL)).\end{aligned}$$

Definición 14

Representación de valores booleanos:

$$\begin{aligned}\mathbf{True} &\equiv \mathbf{K}, \\ \mathbf{False} &\equiv \mathbf{K}_*, \\ \text{If } B \text{ then } P \text{ else } Q &\equiv BPQ.\end{aligned}$$

Definición 15

Representación de pares ordenados:

$$\langle M, N \rangle \equiv \lambda z. zMN \quad (\lambda z. \text{if } z \text{ then } M \text{ else } N),$$

$$\pi_1 \equiv \lambda p. p(\mathbf{True}) \quad (\pi_1(\langle M, N \rangle) \rightarrow_{\beta} M),$$

$$\pi_2 \equiv \lambda p. p(\mathbf{False}) \quad (\pi_2(\langle M, N \rangle) \rightarrow_{\beta} N).$$



Definición 15

Representación de pares ordenados:

$$\begin{aligned}\langle M, N \rangle &\equiv \lambda z. z M N \quad (\lambda z. \text{if } z \text{ then } M \text{ else } N), \\ \pi_1 &\equiv \lambda p. p(\mathbf{True}) \quad (\pi_1(\langle M, N \rangle) \rightarrow_\beta M), \\ \pi_2 &\equiv \lambda p. p(\mathbf{False}) \quad (\pi_2(\langle M, N \rangle) \rightarrow_\beta N).\end{aligned}$$

Definición 16

Representación de números naturales (codificación de Barendregt):

$$\begin{aligned}\ulcorner 0 \urcorner &\equiv \mathbf{I}, \\ \ulcorner n + 1 \urcorner &\equiv \langle \mathbf{False}, \ulcorner n \urcorner \rangle.\end{aligned}$$



Lema 6

Existen combinadores **S**⁺, **P**⁻ y **Zero** tales que:

$$\begin{aligned}\mathbf{S}^+ \ulcorner n \urcorner &=_{\beta} \ulcorner n + 1 \urcorner, \\ \mathbf{P}^- \ulcorner n + 1 \urcorner &=_{\beta} \ulcorner n \urcorner, \\ \mathbf{Zero} \ulcorner 0 \urcorner &=_{\beta} \mathbf{True}, \\ \mathbf{Zero} \ulcorner n + 1 \urcorner &=_{\beta} \mathbf{False}.\end{aligned}$$

Lema 6

Existen combinadores **S**⁺, **P**⁻ y **Zero** tales que:

$$\begin{aligned}\mathbf{S}^+ \ulcorner n \urcorner &=_{\beta} \ulcorner n + 1 \urcorner, \\ \mathbf{P}^- \ulcorner n + 1 \urcorner &=_{\beta} \ulcorner n \urcorner, \\ \mathbf{Zero} \ulcorner 0 \urcorner &=_{\beta} \mathbf{True}, \\ \mathbf{Zero} \ulcorner n + 1 \urcorner &=_{\beta} \mathbf{False}.\end{aligned}$$

Demostración.

Tome:

$$\begin{aligned}\mathbf{S}^+ &\equiv \lambda x. \langle \mathbf{False}, x \rangle, \\ \mathbf{P}^- &\equiv \lambda x. x \mathbf{False}, \\ \mathbf{Zero} &\equiv \lambda x. x \mathbf{True}.\end{aligned}$$

Definición 17

Una **función numérico teórica** $f: \mathbb{N}^k \rightarrow \mathbb{N}$ es **λ -definible** si existe un combinador $F \in \Lambda$ tal que:

$$F^{\ulcorner n_1 \urcorner} \dots^{\ulcorner n_k \urcorner} =_{\beta} f(n_1, \dots, n_k)^{\ulcorner \urcorner},$$

para todo $n_1, \dots, n_m \in \mathbb{N}$. En tal caso, se dice que f es **λ -definida** por F .



Expresividad del cálculo lambda

Definición 17

Una **función numérico teórica** $f: \mathbb{N}^k \rightarrow \mathbb{N}$ es **λ -definible** si existe un combinador $F \in \Lambda$ tal que:

$$F^{\ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner} =_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner,$$

para todo $n_1, \dots, n_m \in \mathbb{N}$. En tal caso, se dice que f es **λ -definida** por F .

Teorema 3

Una función $f: \mathbb{N}^k \rightarrow \mathbb{N}$ es λ -definible si y solo si es una función recursiva.

Demostración.

Ver en [Barendregt and Barendsen, 2000, págs. 19–20].



Observación: Una codificación diferente de los números naturales propuesta por Alonso Church, y la demostración del Teorema 3 bajo dicha codificación (y para funciones recursivas parciales en general), se encuentra en [Sørensen and Urzyczyn, 2006, págs. 20–21].



Observación: Una codificación diferente de los números naturales propuesta por Alonso Church, y la demostración del Teorema 3 bajo dicha codificación (y para funciones recursivas parciales en general), se encuentra en [Sørensen and Urzyczyn, 2006, págs. 20–21].

Corolario 2

Los siguientes problemas son “indecidibles” (o “no computables”):

- ❶ *Dado $M \in \Lambda$, ¿ M es (débilmente) normalizable?*
- ❷ *Dado $M \in \Lambda$, ¿ M es fuertemente normalizable?*

Demostración.

Ver en [Sørensen and Urzyczyn, 2006, Corolario 1.7.7]



Expresividad del cálculo lambda

Algunas propiedades contraintuitivas o “no deseables” del λ -cálculo son:

- 1 Permite la auto-aplicación de términos (MM).



Expresividad del cálculo lambda

Algunas propiedades contraintuitivas o “no deseables” del λ -cálculo son:

- 1 Permite la auto-aplicación de términos (MM).
- 2 No garantiza la existencia de formas normales (posibilidad de “cálculos infinitos”).



Expresividad del cálculo lambda

Algunas propiedades contraintuitivas o “no deseables” del λ -cálculo son:

- 1 Permite la auto-aplicación de términos (MM).
- 2 No garantiza la existencia de formas normales (posibilidad de “cálculos infinitos”).
- 3 Todo λ -término tiene un punto fijo.



Expresividad del cálculo lambda

Algunas propiedades contraintuitivas o “no deseables” del λ -cálculo son:

- 1 Permite la auto-aplicación de términos (MM).
- 2 No garantiza la existencia de formas normales (posibilidad de “cálculos infinitos”).
- 3 Todo λ -término tiene un punto fijo.

Observación: Como la operación de sucesor es λ -definible, el λ -término que la define tiene un punto fijo, pero dicho punto fijo no es un numeral. Lo mismo pasa con muchas otras operaciones sobre números naturales.



Índice

- 1 Breve reseña histórica
- 2 Cálculo lambda sin tipos
 - Términos del cálculo lambda
 - α -conversión
 - β -reducción y β -conversión
 - Combinador de punto fijo
 - Expresividad del cálculo lambda
- 3 Cálculo lambda simplemente tipificado
 - Tipificado a la Curry
 - Tipificado a la Church
- 4 Lógica intuicionista
 - Interpretación BHK
 - Sistema de deducción natural
 - Semántica de Kripke
- 5 Correspondencia de Curry-Howard
- 6 Referencias bibliográficas



Cálculo lambda simplemente tipificado

Referencias usadas en esta sección: [Nederpelt and Geuvers, 2014, Cap. 2], [Sørensen and Urzyczyn, 2006, Cap. 3], [Barendregt and Barendsen, 2000].



Cálculo lambda simplemente tipificado

Referencias usadas en esta sección: [Nederpelt and Geuvers, 2014, Cap. 2], [Sørensen and Urzyczyn, 2006, Cap. 3], [Barendregt and Barendsen, 2000].

El **cálculo lambda simplemente tipificado** fue propuesto por Alonso Church en:

Church, A., 1940: A formulation of the simple theory of types, Journal of Symbolic Logic, 5, pp. 56–68.



Cálculo lambda simplemente tipificado

Referencias usadas en esta sección: [Nederpelt and Geuvers, 2014, Cap. 2], [Sørensen and Urzyczyn, 2006, Cap. 3], [Barendregt and Barendsen, 2000].

El **cálculo lambda simplemente tipificado** fue propuesto por Alonso Church en:

Church, A., 1940: A formulation of the simple theory of types, Journal of Symbolic Logic, 5, pp. 56–68.

Aunque Haskell Curry ya había propuesto la inclusión de tipos en su sistema de **lógica combinatoria** (una variante del cálculo lambda) en:

Curry, H.B. (1934). Functionality in combinatory logic, Proceedings of the National Academy of Science USA 20, pp. 584–590.



Cálculo lambda simplemente tipificado

Referencias usadas en esta sección: [Nederpelt and Geuvers, 2014, Cap. 2], [Sørensen and Urzyczyn, 2006, Cap. 3], [Barendregt and Barendsen, 2000].

El **cálculo lambda simplemente tipificado** fue propuesto por Alonso Church en:

Church, A., 1940: A formulation of the simple theory of types, Journal of Symbolic Logic, 5, pp. 56–68.

Aunque Haskell Curry ya había propuesto la inclusión de tipos en su sistema de **lógica combinatoria** (una variante del cálculo lambda) en:

Curry, H.B. (1934). Functionality in combinatory logic, Proceedings of the National Academy of Science USA 20, pp. 584–590.

Estos dos trabajos dan origen a dos variantes deferentes de tipificado: **a la Church** y **a la Curry**.



Cálculo lambda simplemente tipificado

Las funciones se aplican a cierto tipo de objetos y producen cierto tipo de objetos. Esta idea se formaliza mediante la adición de tipos al cálculo lambda, lo que evita algunas propiedades contraintuitivas de dicho formalismo.



Cálculo lambda simplemente tipificado

Las funciones se aplican a cierto tipo de objetos y producen cierto tipo de objetos. Esta idea se formaliza mediante la adición de tipos al cálculo lambda, lo que evita algunas propiedades contraintuitivas de dicho formalismo.

Se fija el **conjunto de variables de tipos** $\mathbb{V} = \{\alpha, \beta, \gamma, \dots, \alpha_1, \beta_1, \gamma_1, \dots\}$.



Cálculo lambda simplemente tipificado

Las funciones se aplican a cierto tipo de objetos y producen cierto tipo de objetos. Esta idea se formaliza mediante la adición de tipos al cálculo lambda, lo que evita algunas propiedades contraintuitivas de dicho formalismo.

Se fija el **conjunto de variables de tipos** $\mathbb{V} = \{\alpha, \beta, \gamma, \dots, \alpha_1, \beta_1, \gamma_1, \dots\}$.

Definición 18

El **conjunto de tipos simples** \mathbb{T} se define inductivamente por:

- ① (Variable de tipo) Si $\sigma \in \mathbb{V}$, entonces $\sigma \in \mathbb{T}$.
- ② (Tipo flecha o tipo función) Si $\sigma, \tau \in \mathbb{T}$, entonces $(\sigma \rightarrow \tau) \in \mathbb{T}$.



Cálculo lambda simplemente tipificado

Las funciones se aplican a cierto tipo de objetos y producen cierto tipo de objetos. Esta idea se formaliza mediante la adición de tipos al cálculo lambda, lo que evita algunas propiedades contraintuitivas de dicho formalismo.

Se fija el **conjunto de variables de tipos** $\mathbb{V} = \{\alpha, \beta, \gamma, \dots, \alpha_1, \beta_1, \gamma_1, \dots\}$.

Definición 18

El **conjunto de tipos simples** \mathbb{T} se define inductivamente por:

- ① (Variable de tipo) Si $\sigma \in \mathbb{V}$, entonces $\sigma \in \mathbb{T}$.
- ② (Tipo flecha o tipo función) Si $\sigma, \tau \in \mathbb{T}$, entonces $(\sigma \rightarrow \tau) \in \mathbb{T}$.

Observación: por medio de una “sintaxis abstracta”: $\mathbb{T} = \mathbb{V} \mid (\mathbb{T} \rightarrow \mathbb{T})$



Cálculo lambda simplemente tipificado

Las funciones se aplican a cierto tipo de objetos y producen cierto tipo de objetos. Esta idea se formaliza mediante la adición de tipos al cálculo lambda, lo que evita algunas propiedades contraintuitivas de dicho formalismo.

Se fija el **conjunto de variables de tipos** $\mathbb{V} = \{\alpha, \beta, \gamma, \dots, \alpha_1, \beta_1, \gamma_1, \dots\}$.

Definición 18

El **conjunto de tipos simples** \mathbb{T} se define inductivamente por:

- ① (Variable de tipo) Si $\sigma \in \mathbb{V}$, entonces $\sigma \in \mathbb{T}$.
- ② (Tipo flecha o tipo función) Si $\sigma, \tau \in \mathbb{T}$, entonces $(\sigma \rightarrow \tau) \in \mathbb{T}$.

Observación: por medio de una “sintaxis abstracta”: $\mathbb{T} = \mathbb{V} \mid (\mathbb{T} \rightarrow \mathbb{T})$

Ejemplo 10

Las siguientes expresiones son tipos simples: α , $(\alpha \rightarrow \beta)$,
 $((\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow (\alpha \rightarrow \beta)))$.

Cálculo lambda simplemente tipificado

Notación:

- Se pueden omitir los paréntesis externos.



Cálculo lambda simplemente tipificado

Notación:

- Se pueden omitir los paréntesis externos.
- El tipo flecha es asociativo a derecha.



Cálculo lambda simplemente tipificado

Notación:

- Se pueden omitir los paréntesis externos.
- El tipo flecha es asociativo a derecha.
- Se usarán variantes de los elementos de \mathbb{V} con ' para denotar variables de tipos.



Cálculo lambda simplemente tipificado

Notación:

- Se pueden omitir los paréntesis externos.
- El tipo flecha es asociativo a derecha.
- Se usarán variantes de los elementos de \mathbb{V} con ' para denotar variables de tipos.

La interpretación de los tipos simples es:

- Las variables de tipos representan **tipos básicos** (por ejemplo *nat*: el conjunto de los números naturales).



Cálculo lambda simplemente tipificado

Notación:

- Se pueden omitir los paréntesis externos.
- El tipo flecha es asociativo a derecha.
- Se usarán variantes de los elementos de \mathbb{V} con ' para denotar variables de tipos.

La interpretación de los tipos simples es:

- Las variables de tipos representan **tipos básicos** (por ejemplo *nat*: el conjunto de los números naturales).
- Un tipo flecha $\alpha \rightarrow \beta$ representa un **tipo de función** (o el conjunto de todas las funciones) con entrada de tipo α y salida de tipo β (por ejemplo $\text{nat} \rightarrow \text{real}$: el conjunto de funciones de *nat* en *real*).



Tipificado a la Curry

El **tipificado a la Curry** es también conocido como **tipificado implícito**.



Tipificado a la Curry

El **tipificado a la Curry** es también conocido como **tipificado implícito**.

Definición 19

- 1 Una **sentencia** es de la forma $M : \sigma$, donde $M \in \Lambda$ y $\sigma \in \mathbb{T}$. En dicha sentencia, M es llamado el **sujeto** y σ el **tipo**.
- 2 Una **declaración** es una sentencia donde el sujeto es una variable.
- 3 Un **contexto** (o una **base**) es un conjunto (finito) de declaraciones con diferentes sujetos.
- 4 Un **juicio** tiene la forma $\Gamma \vdash M : \sigma$, donde Γ es un contexto y $M : \sigma$ es una sentencia.



Tipificado a la Curry

El **tipificado a la Curry** es también conocido como **tipificado implícito**.

Definición 19

- 1 Una **sentencia** es de la forma $M : \sigma$, donde $M \in \Lambda$ y $\sigma \in \mathbb{T}$. En dicha sentencia, M es llamado el **sujeto** y σ el **tipo**.
- 2 Una **declaración** es una sentencia donde el sujeto es una variable.
- 3 Un **contexto** (o una **base**) es un conjunto (finito) de declaraciones con diferentes sujetos.
- 4 Un **juicio** tiene la forma $\Gamma \vdash M : \sigma$, donde Γ es un contexto y $M : \sigma$ es una sentencia.

Observación: En algunos textos, los contextos se definen como listas en lugar de conjuntos. En sistemas de tipos más complejos, es necesario usar listas en lugar de conjuntos.



Tipificado a la Curry

Definición 20

El **sistema de asignación de tipos a la Curry**, denotado $\lambda \rightarrow$, consiste en las siguientes reglas:

Si $x : \sigma \in \Gamma$, entonces $\Gamma \vdash x : \sigma$ (Var)

$$\frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} \text{ (Apl)}$$

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau} \text{ (Abst)}$$



Tipificado a la Curry

Definición 20

El **sistema de asignación de tipos a la Curry**, denotado $\lambda \rightarrow$, consiste en las siguientes reglas:

Si $x : \sigma \in \Gamma$, entonces $\Gamma \vdash x : \sigma$ (Var)

$$\frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} \text{ (Apl)} \quad \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau} \text{ (Abst)}$$

Definición 21

Una **derivación** de $\Gamma \vdash N : \varphi$ es un árbol construido mediante aplicación de las reglas en la definición anterior, cuyas hojas son aplicación de la regla (Var) y cuya raíz es $\Gamma \vdash N : \varphi$.



Ejemplo 11

Los siguientes juicios pueden ser derivados en $\lambda \rightarrow$:

- ① $\vdash \mathbf{I} : \alpha \rightarrow \alpha.$
- ② $\vdash \mathbf{K} : \alpha \rightarrow \beta \rightarrow \alpha.$
- ③ $x : \alpha \rightarrow \beta, y : \beta \rightarrow \delta \vdash \lambda z. y(xz) : \alpha \rightarrow \delta.$



Ejemplo 11

Los siguientes juicios pueden ser derivados en $\lambda \rightarrow$:

- ① $\vdash \mathbf{I} : \alpha \rightarrow \alpha$.
- ② $\vdash \mathbf{K} : \alpha \rightarrow \beta \rightarrow \alpha$.
- ③ $x : \alpha \rightarrow \beta, y : \beta \rightarrow \delta \vdash \lambda z. y(xz) : \alpha \rightarrow \delta$.

Observación: Para derivaciones más complicadas, el formato de árbol se vuelve inmanejable. En [Nederpelt and Geuvers, 2014, p. 45-46], se propone un formato simplificado para presentar derivaciones, al cual llaman **formato bandera**. En dicho formato, cada declaración que se toma como hipótesis se despliega en forma de bandera (encerrada en un rectángulo y con una línea vertical que indica dónde la hipótesis es descargada). En el **formato bandera reducido** se elimina además el uso de la regla (Var).



Definición 22

Dado un contexto $\Gamma = \{x_1 : \sigma_1, \dots, x_n : \sigma_n\}$:

- 1 El **dominio de Γ** es $\text{dom}(\Gamma) = \{x_1, \dots, x_n\}$.
- 2 El **rango de Γ** es $\text{ran}(\Gamma) = \{\sigma_1, \dots, \sigma_n\}$.
- 3 $\Gamma(x_i) = \sigma_i$ (para $1 \leq i \leq n$).
- 4 Si $A \subseteq V$, entonces $\Gamma \upharpoonright A = \{x : \sigma \mid x \in A \text{ y } \Gamma(x) = \sigma\}$.



Definición 22

Dado un contexto $\Gamma = \{x_1 : \sigma_1, \dots, x_n : \sigma_n\}$:

- 1 El **dominio de Γ** es $\text{dom}(\Gamma) = \{x_1, \dots, x_n\}$.
- 2 El **rango de Γ** es $\text{ran}(\Gamma) = \{\sigma_1, \dots, \sigma_n\}$.
- 3 $\Gamma(x_i) = \sigma_i$ (para $1 \leq i \leq n$).
- 4 Si $A \subseteq V$, entonces $\Gamma \upharpoonright A = \{x : \sigma \mid x \in A \text{ y } \Gamma(x) = \sigma\}$.

Observación: Un contexto puede ser visto como una función parcial de V en \mathbb{T} .



Definición 22

Dado un contexto $\Gamma = \{x_1 : \sigma_1, \dots, x_n : \sigma_n\}$:

- 1 El **dominio de Γ** es $\text{dom}(\Gamma) = \{x_1, \dots, x_n\}$.
- 2 El **rango de Γ** es $\text{ran}(\Gamma) = \{\sigma_1, \dots, \sigma_n\}$.
- 3 $\Gamma(x_i) = \sigma_i$ (para $1 \leq i \leq n$).
- 4 Si $A \subseteq V$, entonces $\Gamma \upharpoonright A = \{x : \sigma \mid x \in A \text{ y } \Gamma(x) = \sigma\}$.

Observación: Un contexto puede ser visto como una función parcial de V en \mathbb{T} .

Lema 7 (de Debilitamiento)

Si $\Gamma \vdash M : \sigma$ y Δ es un contexto tal que $\Gamma \subseteq \Delta$, entonces $\Delta \vdash M : \sigma$.



Lema 8 (de Generación)

- ① Si $\Gamma \vdash x : \sigma$, entonces $(x : \sigma) \in \Gamma$.
- ② Si $\Gamma \vdash MN : \tau$, entonces $\Gamma \vdash M : \sigma \rightarrow \tau$ y $\Gamma \vdash N : \sigma$, para algún $\sigma \in \mathbb{T}$.
- ③ Si $\Gamma \vdash \lambda x.M : \rho$, entonces existen $\sigma, \tau \in \mathbb{T}$ tales que $\Gamma, x : \sigma \vdash M : \tau$ y $\rho \equiv \sigma \rightarrow \tau$.



Lema 8 (de Generación)

- 1 Si $\Gamma \vdash x : \sigma$, entonces $(x : \sigma) \in \Gamma$.
- 2 Si $\Gamma \vdash MN : \tau$, entonces $\Gamma \vdash M : \sigma \rightarrow \tau$ y $\Gamma \vdash N : \sigma$, para algún $\sigma \in \mathbb{T}$.
- 3 Si $\Gamma \vdash \lambda x.M : \rho$, entonces existen $\sigma, \tau \in \mathbb{T}$ tales que $\Gamma, x : \sigma \vdash M : \tau$ y $\rho \equiv \sigma \rightarrow \tau$.

Lema 9 (de Variables Libres)

- 1 Si $\Gamma \vdash M : \sigma$ y Γ' es un contexto tal que $\Gamma(x) = \Gamma'(x)$ para todo $x \in \text{FV}(M)$, entonces $\Gamma' \vdash M : \sigma$.
- 2 Si $\Gamma \vdash M : \sigma$, entonces $\text{FV}(M) \subseteq \text{dom}(\Gamma)$.
- 3 Si $\Gamma \vdash M : \sigma$, entonces $\Gamma \upharpoonright \text{FV}(M) \vdash M : \sigma$.



Lema 10 (de Subtérminos)

Si $\Gamma \vdash M : \sigma$ y M' es un subtérmino de M , entonces $\Gamma' \vdash M' : \sigma'$ para algún Γ' y σ' .



Lema 10 (de Subtérminos)

Si $\Gamma \vdash M : \sigma$ y M' es un subtérmino de M , entonces $\Gamma' \vdash M' : \sigma'$ para algún Γ' y σ' .

Notación:

- La sustitución de una variable de tipo α por un tipo τ en un tipo σ se denotará por $\sigma[\alpha := \tau]$.



Lema 10 (de Subtérminos)

Si $\Gamma \vdash M : \sigma$ y M' es un subtérmino de M , entonces $\Gamma' \vdash M' : \sigma'$ para algún Γ' y σ' .

Notación:

- La sustitución de una variable de tipo α por un tipo τ en un tipo σ se denotará por $\sigma[\alpha := \tau]$.
- La sustitución de una variable de tipo α por un tipo τ en cada uno de los tipos de un contexto Γ se denotará por $\Gamma[\alpha : \tau]$.



Lema 10 (de Subtérminos)

Si $\Gamma \vdash M : \sigma$ y M' es un subtérmino de M , entonces $\Gamma' \vdash M' : \sigma'$ para algún Γ' y σ' .

Notación:

- La sustitución de una variable de tipo α por un tipo τ en un tipo σ se denotará por $\sigma[\alpha := \tau]$.
- La sustitución de una variable de tipo α por un tipo τ en cada uno de los tipos de un contexto Γ se denotará por $\Gamma[\alpha : \tau]$.

Lema 11 (de Sustitución)

- 1 Si $\Gamma \vdash M : \sigma$, entonces $\Gamma[\alpha := \tau] \vdash M : \sigma[\alpha := \tau]$.
- 2 Si $\Gamma, x : \sigma \vdash M : \tau$ y $\Gamma \vdash N : \sigma$, entonces $\Gamma \vdash M[x := N] : \tau$.

Lema 12 (de Reducción del Sujeto)

Si $\Gamma \vdash M : \sigma$ y $M \rightarrow_{\beta} M'$, entonces $\Gamma \vdash M' : \sigma$.



Tipificado a la Curry

Lema 12 (de Reducción del Sujeto)

Si $\Gamma \vdash M : \sigma$ y $M \rightarrow_{\beta} M'$, entonces $\Gamma \vdash M' : \sigma$.

Definición 23

Un término $M \in \Lambda$ es **tipificable** (o **lícito**) si existe un contexto Γ y un tipo σ tales que $\Gamma \vdash M : \sigma$.



Tipificado a la Curry

Lema 12 (de Reducción del Sujeto)

Si $\Gamma \vdash M : \sigma$ y $M \rightarrow_{\beta} M'$, entonces $\Gamma \vdash M' : \sigma$.

Definición 23

Un término $M \in \Lambda$ es **tipificable** (o **lícito**) si existe un contexto Γ y un tipo σ tales que $\Gamma \vdash M : \sigma$.

Se tiene los siguientes problemas asociados a la asignación de tipos:

- **Tipificabilidad**: determinar si un término M es tipificable (una variante consiste en determinar la tipificabilidad en un contexto determinado).



Tipificado a la Curry

Lema 12 (de Reducción del Sujeto)

Si $\Gamma \vdash M : \sigma$ y $M \rightarrow_{\beta} M'$, entonces $\Gamma \vdash M' : \sigma$.

Definición 23

Un término $M \in \Lambda$ es **tipificable** (o **lícito**) si existe un contexto Γ y un tipo σ tales que $\Gamma \vdash M : \sigma$.

Se tiene los siguientes problemas asociados a la asignación de tipos:

- **Tipificabilidad**: determinar si un término M es tipificable (una variante consiste en determinar la tipificabilidad en un contexto determinado).
- **Chequeo de tipos**: dado un contexto Γ , un término M y un tipo σ , determinar si $\Gamma \vdash M : \sigma$.



Tipificado a la Curry

Lema 12 (de Reducción del Sujeto)

Si $\Gamma \vdash M : \sigma$ y $M \rightarrow_{\beta} M'$, entonces $\Gamma \vdash M' : \sigma$.

Definición 23

Un término $M \in \Lambda$ es **tipificable** (o **lícito**) si existe un contexto Γ y un tipo σ tales que $\Gamma \vdash M : \sigma$.

Se tiene los siguientes problemas asociados a la asignación de tipos:

- **Tipificabilidad**: determinar si un término M es tipificable (una variante consiste en determinar la tipificabilidad en un contexto determinado).
- **Chequeo de tipos**: dado un contexto Γ , un término M y un tipo σ , determinar si $\Gamma \vdash M : \sigma$.
- **Habitabilidad de un tipo**: dado un contexto Γ y un tipo σ , determinar si existe un término M tal que $\Gamma \vdash M : \sigma$.



Tipificado a la Curry

Observación: Los problemas de tipificabilidad, chequeo de tipos y habitabilidad en $\lambda \rightarrow$ son decidibles (ver [Sørensen and Urzyczyn, 2006, Sección 3.2] o [Barendregt, 1992, Sección 4.4]). En sistemas de tipos más expresivos algunos de estos problemas no son decidibles.

Observación: En el tipificado a la Curry no hay unicidad de tipos.



Tipificado a la Church

El **tipificado a la Church** es también conocido como **tipificado explícito**. En este tipificado, se debe especificar el tipo de las variables en las λ -abstracciones.



Tipificado a la Church

El **tipificado a la Church** es también conocido como **tipificado explícito**. En este tipificado, se debe especificar el tipo de las variables en las λ -abstracciones.

Definición 24

El conjunto de **λ -términos pre-tipificados** (o **λ -términos a la Church**) es:

$$\Lambda_{\mathbb{T}} = V \mid (\lambda V : \mathbb{T}. \Lambda_{\mathbb{T}}) \mid (\Lambda_{\mathbb{T}} \Lambda_{\mathbb{T}}).$$



Tipificado a la Church

El **tipificado a la Church** es también conocido como **tipificado explícito**. En este tipificado, se debe especificar el tipo de las variables en las λ -abstracciones.

Definición 24

El conjunto de **λ -términos pre-tipificados** (o **λ -términos a la Church**) es:

$$\Lambda_{\mathbb{T}} = V \mid (\lambda V : \mathbb{T}. \Lambda_{\mathbb{T}}) \mid (\Lambda_{\mathbb{T}} \Lambda_{\mathbb{T}}).$$

Observaciones:

- Las definiciones para λ -términos se adecuan para λ -términos pre-tipificados de manera natural (es solo agregar los tipos de las variables en las abstracciones).



Tipificado a la Church

El **tipificado a la Church** es también conocido como **tipificado explícito**. En este tipificado, se debe especificar el tipo de las variables en las λ -abstracciones.

Definición 24

El conjunto de **λ -términos pre-tipificados** (o **λ -términos a la Church**) es:

$$\Lambda_{\mathbb{T}} = V \mid (\lambda V : \mathbb{T}. \Lambda_{\mathbb{T}}) \mid (\Lambda_{\mathbb{T}} \Lambda_{\mathbb{T}}).$$

Observaciones:

- Las definiciones para λ -términos se adecuan para λ -términos pre-tipificados de manera natural (es solo agregar los tipos de las variables en las abstracciones).
- Originalmente, Church adicionaba toda la información de los tipos a los términos y definía inductivamente el conjunto de los términos tipificables.



Tipificado a la Church

Definición 25

El **sistema de asignación de tipos a la Church**, denotado también como $\lambda \rightarrow$, consiste en las siguientes reglas:

Si $x : \sigma \in \Gamma$, entonces $\Gamma \vdash x : \sigma$ (Var)

$$\frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} \text{ (Apl)} \quad \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x : \sigma. M : \sigma \rightarrow \tau} \text{ (Abst)}$$



Tipificado a la Church

Definición 25

El **sistema de asignación de tipos a la Church**, denotado también como $\lambda \rightarrow$, consiste en las siguientes reglas:

Si $x : \sigma \in \Gamma$, entonces $\Gamma \vdash x : \sigma$ (Var)

$$\frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} \text{ (Apl)} \quad \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x : \sigma. M : \sigma \rightarrow \tau} \text{ (Abst)}$$

Observación: Se usará el mismo símbolo para denotar los sistemas de asignación de tipos simples a la Curry y a la Church y se especificará en cuál sistema se está trabajando cuando sea necesario.



Ejemplo 12

Los siguientes juicios pueden ser derivados en $\lambda \rightarrow$ (a la Church):

- ① $\vdash \lambda x : \alpha. x : \alpha \rightarrow \alpha.$
- ② $\vdash \lambda x : \alpha. \lambda y : \beta. x : \alpha \rightarrow \beta \rightarrow \alpha.$
- ③ $x : \alpha \rightarrow \beta, y : \beta \rightarrow \delta \vdash \lambda z : \alpha. y(xz) : \alpha \rightarrow \delta.$



Tipificado a la Church

Ejemplo 12

Los siguientes juicios pueden ser derivados en $\lambda \rightarrow$ (a la Church):

- ① $\vdash \lambda x : \alpha. x : \alpha \rightarrow \alpha.$
- ② $\vdash \lambda x : \alpha. \lambda y : \beta. x : \alpha \rightarrow \beta \rightarrow \alpha.$
- ③ $x : \alpha \rightarrow \beta, y : \beta \rightarrow \delta \vdash \lambda z : \alpha. y(xz) : \alpha \rightarrow \delta.$

Observación: Cambiando λ -términos por λ -términos pre-tipificados, las definiciones y propiedades que se presentaron para $\lambda \rightarrow$ a la Curry también aplican para $\lambda \rightarrow$ a la Church.



Tipificado a la Church

Ejemplo 12

Los siguientes juicios pueden ser derivados en $\lambda \rightarrow$ (a la Church):

- ① $\vdash \lambda x : \alpha. x : \alpha \rightarrow \alpha.$
- ② $\vdash \lambda x : \alpha. \lambda y : \beta. x : \alpha \rightarrow \beta \rightarrow \alpha.$
- ③ $x : \alpha \rightarrow \beta, y : \beta \rightarrow \delta \vdash \lambda z : \alpha. y(xz) : \alpha \rightarrow \delta.$

Observación: Cambiando λ -términos por λ -términos pre-tipificados, las definiciones y propiedades que se presentaron para $\lambda \rightarrow$ a la Curry también aplican para $\lambda \rightarrow$ a la Church.

Lema 13 (Unicidad de Tipos)

Si $\Gamma \vdash M : \alpha$ y $\Gamma \vdash M : \beta$ en $\lambda \rightarrow$ a la Church, entonces $\alpha \equiv \beta$.



Definición 26

La **función de borrado** de λ -términos a la Church a λ -términos a la Curry, denotada $|\cdot|$, se define inductivamente por:

$$\begin{aligned}|x| &= x, \\ |MN| &= MN, \\ |\lambda x : \alpha.M| &= \lambda x.M.\end{aligned}$$



Tipificado a la Church

Definición 26

La **función de borrado** de λ -términos a la Church a λ -términos a la Curry, denotada $|\cdot|$, se define inductivamente por:

$$\begin{aligned}|x| &= x, \\ |MN| &= MN, \\ |\lambda x : \alpha. M| &= \lambda x. M.\end{aligned}$$

Lema 14

- 1 Si $\Gamma \vdash M : \alpha$ a la Church, entonces $\Gamma \vdash |M| : \alpha$ a la Curry.
- 2 Si $\Gamma \vdash M : \alpha$ a la Curry, entonces existe un término a la Church M' tal que $|M'| = M$ y $\Gamma \vdash M' : \alpha$ a la Church.



Teorema 4 (de Normalización Fuerte)

Si $\Gamma \vdash M : \alpha$, entonces M es fuertemente normalizable.



Tipificado a la Church

Teorema 4 (de Normalización Fuerte)

Si $\Gamma \vdash M : \alpha$, entonces M es fuertemente normalizable.

Demostración.

Una demostración para $\lambda \rightarrow$ a la Church (en su versión ortodoxa) está en [Sørensen and Urzyczyn, 2006, Sección 3.5] (siguiendo el método de Turing para probar la normalización débil y siguiendo ideas de Klop y de Gandy para mostrar la normalización fuerte a partir de la débil) y una demostración para $\lambda \rightarrow$ a la Curry está en [Barendregt, 1992, Sección 4.3] (siguiendo el método de Tait)



Tipificado a la Church

Teorema 4 (de Normalización Fuerte)

Si $\Gamma \vdash M : \alpha$, entonces M es fuertemente normalizable.

Demostración.

Una demostración para $\lambda \rightarrow$ a la Church (en su versión ortodoxa) está en [Sørensen and Urzyczyn, 2006, Sección 3.5] (siguiendo el método de Turing para probar la normalización débil y siguiendo ideas de Klop y de Gandy para mostrar la normalización fuerte a partir de la débil) y una demostración para $\lambda \rightarrow$ a la Curry está en [Barendregt, 1992, Sección 4.3] (siguiendo el método de Tait)

Observación: Se dice que un sistema de tipos es fuertemente normalizable si todo término en dicho sistema es fuertemente normalizable. El teorema anterior muestra que $\lambda \rightarrow$ (a la Curry y a la Church) es fuertemente normalizable.



Índice

- 1 Breve reseña histórica
- 2 Cálculo lambda sin tipos
 - Términos del cálculo lambda
 - α -conversión
 - β -reducción y β -conversión
 - Combinador de punto fijo
 - Expresividad del cálculo lambda
- 3 Cálculo lambda simplemente tipificado
 - Tipificado a la Curry
 - Tipificado a la Church
- 4 **Lógica intuicionista**
 - Interpretación BHK
 - Sistema de deducción natural
 - Semántica de Kripke
- 5 Correspondencia de Curry-Howard
- 6 Referencias bibliográficas



Lógica intuicionista

Referencia usada en esta sección: [Sørensen and Urzyczyn, 2006] (con algunos cambios de notación).



Referencia usada en esta sección: [Sørensen and Urzyczyn, 2006] (con algunos cambios de notación).

“Classical logic is based on the fundamental notion of truth. The truth of a statement is an “absolute” property of this statement, in that it is independent of any reasoning, understanding, or action. A well-formed and unambiguous declarative statement is either true or false, whether or not we (or anybody else) know it, prove it, or verify it in any possible way.” [Sørensen and Urzyczyn, 2006, pág. 27]



Referencia usada en esta sección: [Sørensen and Urzyczyn, 2006] (con algunos cambios de notación).

“Classical logic is based on the fundamental notion of truth. The truth of a statement is an “absolute” property of this statement, in that it is independent of any reasoning, understanding, or action. A well-formed and unambiguous declarative statement is either true or false, whether or not we (or anybody else) know it, prove it, or verify it in any possible way.” [Sørensen and Urzyczyn, 2006, pág. 27]

En la Lógica Intuicionista (IL) “There is no absolute truth, there is only the knowledge and intuitive construction of the idealized mathematician, the creative subject. A logical judgement is only considered “true” if the creative subject can verify its correctness.” [Sørensen and Urzyczyn, 2006, pág. 28]



Lógica intuicionista

En la lógica clásica es válido el principio del **tercer excluido** (o **tertium non datur**): $P \vee \neg P$ es válido para cualquier proposición P . Mientras que en la lógica intuicionista dicho principio no es válido.



Lógica intuicionista

En la lógica clásica es válido el principio del **tercer excluido** (o **tertium non datur**): $P \vee \neg P$ es válido para cualquier proposición P . Mientras que en la lógica intuicionista dicho principio no es válido.

Considere la siguiente sentencia: *Existen números irracionales x y y tales que x^y es racional.*



Lógica intuicionista

En la lógica clásica es válido el principio del **tercer excluido** (o **tertium non datur**): $P \vee \neg P$ es válido para cualquier proposición P . Mientras que en la lógica intuicionista dicho principio no es válido.

Considere la siguiente sentencia: *Existen números irracionales x y y tales que x^y es racional.*

Demostración usando tercer exclusivo: Si $\sqrt{2}^{\sqrt{2}}$ es racional, tome $x = y = \sqrt{2}$; en caso contrario tome $x = \sqrt{2}^{\sqrt{2}}$ y $y = \sqrt{2}$.



Lógica intuicionista

En la lógica clásica es válido el principio del **tercer excluido** (o **tertium non datur**): $P \vee \neg P$ es válido para cualquier proposición P . Mientras que en la lógica intuicionista dicho principio no es válido.

Considere la siguiente sentencia: *Existen números irracionales x y y tales que x^y es racional.*

Demostración usando tercer exclusivo: Si $\sqrt{2}^{\sqrt{2}}$ es racional, tome $x = y = \sqrt{2}$; en caso contrario tome $x = \sqrt{2}^{\sqrt{2}}$ y $y = \sqrt{2}$.

Demostración constructiva: Para $x = \sqrt{2}$ y $y = 2\log_2 3$ se tiene que x y y son irracionales y $x^y = 3 \in \mathbb{Q}$.



Lógica intuicionista

En la lógica clásica es válido el principio del **tercer excluido** (o **tertium non datur**): $P \vee \neg P$ es válido para cualquier proposición P . Mientras que en la lógica intuicionista dicho principio no es válido.

Considere la siguiente sentencia: *Existen números irracionales x y y tales que x^y es racional.*

Demostración usando tercer exclusivo: Si $\sqrt{2}^{\sqrt{2}}$ es racional, tome $x = y = \sqrt{2}$; en caso contrario tome $x = \sqrt{2}^{\sqrt{2}}$ y $y = \sqrt{2}$.

Demostración constructiva: Para $x = \sqrt{2}$ y $y = 2\log_2 3$ se tiene que x y y son irracionales y $x^y = 3 \in \mathbb{Q}$.

En la lógica intuicionista se acepta solo razonamiento “constructivo”.



Interpretación BHK

Los conectivos de la lógica proposicional intuicionista son \wedge , \vee , \supset y la constante \perp .



Interpretación BHK

Los conectivos de la lógica proposicional intuicionista son \wedge, \vee, \supset y la constante \perp .

Bajo la **interpretación de Brouwer-Heyting-Kolmokorov (BHK)**:

- Una construcción de $\varphi_1 \wedge \varphi_2$ consiste en una construcción de φ_1 y una construcción de φ_2 .
- Una construcción de $\varphi_1 \vee \varphi_2$ consiste en un indicador $i \in \{1, 2\}$ y una construcción de φ_i .
- Una construcción de $\varphi_1 \supset \varphi_2$ consiste en un método (función) que transforma cualquier construcción de φ_1 en una construcción de φ_2 .
- No existe construcción de \perp .



Interpretación BHK

Los conectivos de la lógica proposicional intuicionista son \wedge, \vee, \supset y la constante \perp .

Bajo la **interpretación de Brouwer-Heyting-Kolmogorov (BHK)**:

- Una construcción de $\varphi_1 \wedge \varphi_2$ consiste en una construcción de φ_1 y una construcción de φ_2 .
- Una construcción de $\varphi_1 \vee \varphi_2$ consiste en un indicador $i \in \{1, 2\}$ y una construcción de φ_i .
- Una construcción de $\varphi_1 \supset \varphi_2$ consiste en un método (función) que transforma cualquier construcción de φ_1 en una construcción de φ_2 .
- No existe construcción de \perp .

La **negación intuicionista** está definida por $\neg\varphi \stackrel{\text{def}}{=} \varphi \supset \perp$.

- Una construcción de $\neg\varphi$ consiste en un método (función) que transforma cualquier construcción de φ en una construcción inexistente.



Interpretación BHK

Para la interpretación de cuantificadores se supone que variables individuales y términos algebraicos son interpretados sobre un dominio de “objetos” D .



Interpretación BHK

Para la interpretación de cuantificadores se supone que variables individuales y términos algebraicos son interpretados sobre un dominio de “objetos” D .

- Una construcción de $\forall x\varphi(x)$ es un método (función) que transforma todo objeto $a \in D$ en una construcción de $\varphi(a)$.
- Una construcción de $\exists x\varphi(x)$ consiste en un objeto $a \in D$ y una construcción de $\varphi(a)$.



Sistema de deducción natural

Un **juicio** es una expresión de la forma $\Gamma \vdash \varphi$, donde $\Gamma \cup \{\varphi\}$ es un conjunto finito de fórmulas, y se lee “ Γ prueba a φ ”.



Sistema de deducción natural

Un **juicio** es una expresión de la forma $\Gamma \vdash \varphi$, donde $\Gamma \cup \{\varphi\}$ es un conjunto finito de fórmulas, y se lee “ Γ prueba a φ ”.

Notación: en la parte izquierda de los juicios, se escribe:

- $\varphi_1, \dots, \varphi_n$ en lugar de $\{\varphi_1, \dots, \varphi_n\}$.
- Γ, φ en lugar de $\Gamma \cup \{\varphi\}$.
- Γ, Δ en lugar de $\Gamma \cup \Delta$.
- Si $\Gamma = \emptyset$, se omite Γ .



Sistema de deducción natural

Un **juicio** es una expresión de la forma $\Gamma \vdash \varphi$, donde $\Gamma \cup \{\varphi\}$ es un conjunto finito de fórmulas, y se lee “ Γ prueba a φ ”.

Notación: en la parte izquierda de los juicios, se escribe:

- $\varphi_1, \dots, \varphi_n$ en lugar de $\{\varphi_1, \dots, \varphi_n\}$.
- Γ, φ en lugar de $\Gamma \cup \{\varphi\}$.
- Γ, Δ en lugar de $\Gamma \cup \Delta$.
- Si $\Gamma = \emptyset$, se omite Γ .

En un sistema de deducción natural (con hipótesis, o con “estilo de cálculo de secuentes”), los **axiomas** son juicios y las **reglas** permiten derivar juicios a partir de otros juicios.



Sistema de deducción natural

Axiomas y reglas para la lógica proposicional intuicionista (sistema NJ(\supset , \perp , \wedge , \vee) o simplemente NJ):

$$\Gamma, \varphi \vdash \varphi \text{ (Ax)}$$

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \supset \psi} (\supset I)$$

$$\frac{\Gamma \vdash \varphi \supset \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi} (\supset E)$$

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi} (\wedge I)$$

$$\frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi} (\wedge E_1) \quad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi} (\wedge E_2)$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi} (\vee I_1) \quad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi} (\vee I_2) \quad \frac{\Gamma \vdash \varphi \vee \psi \quad \Gamma, \varphi \vdash \theta \quad \Gamma, \psi \vdash \theta}{\Gamma \vdash \theta} (\vee E)$$

$$\frac{\Gamma \vdash \perp}{\Gamma \vdash \varphi} (\perp E)$$



Sistema de deducción natural

Reglas para cuantificadores (la extensión de NJ con estas reglas será denotada NJQ):

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall x \varphi} (\forall I)$$

$$\frac{\Gamma \vdash \forall x \varphi}{\Gamma \vdash \varphi[x := t]} (\forall E)$$

$$\frac{\Gamma \vdash \varphi[x := t]}{\Gamma \vdash \exists x \varphi} (\exists I)$$

$$\frac{\Gamma \vdash \exists x \varphi \quad \Gamma, \varphi \vdash \psi}{\Gamma \vdash \psi} (\exists E)$$

Condiciones:

- En $(\forall I)$: $x \notin FV(\Gamma)$.
- En $(\forall E)$ y $(\exists I)$: t es libre para x en φ .
- En $(\exists E)$: $x \notin FV(\Gamma \cup \{\psi\})$.

Donde $FV(\star)$ denota el conjunto de las variables libres en \star :



Definición 27

Una **prueba** (o **derivación**) formal de un juicio $\Gamma \vdash \varphi$, en un sistema de deducción natural N , es un árbol finito de juicios que satisface las siguientes condiciones:

- La raíz del árbol es $\Gamma \vdash \varphi$.
- Todas las ramas del árbol son axiomas de N .
- La etiqueta de cada nodo madre se obtiene de las etiquetas de los nodos hijos usando una de las reglas de N .

Si existe una prueba de $\Gamma \vdash \varphi$ en el sistema N se escribe $\Gamma \vdash_N \varphi$. Para un conjunto infinito Γ , se escribe $\Gamma \vdash_N \varphi$ para expresar que existe $\Gamma_0 \subseteq \Gamma$ finito tal que $\Gamma_0 \vdash_N \varphi$.



Definición 27

Una **prueba** (o **derivación**) formal de un juicio $\Gamma \vdash \varphi$, en un sistema de deducción natural N , es un árbol finito de juicios que satisface las siguientes condiciones:

- La raíz del árbol es $\Gamma \vdash \varphi$.
- Todas las ramas del árbol son axiomas de N .
- La etiqueta de cada nodo madre se obtiene de las etiquetas de los nodos hijos usando una de las reglas de N .

Si existe una prueba de $\Gamma \vdash \varphi$ en el sistema N se escribe $\Gamma \vdash_N \varphi$. Para un conjunto infinito Γ , se escribe $\Gamma \vdash_N \varphi$ para expresar que existe $\Gamma_0 \subseteq \Gamma$ finito tal que $\Gamma_0 \vdash_N \varphi$.

Observación: Es usual omitir el subíndice en \vdash_N , para no sobrecargar la notación.



Proposición 1

- ① $\vdash (\alpha \supset (\beta \supset \delta)) \supset ((\alpha \supset \beta) \supset (\alpha \supset \delta)).$
- ② $\alpha \supset \beta, \beta \supset \delta \vdash \alpha \supset \delta.$
- ③ $\alpha \wedge (\beta \vee \delta) \vdash (\alpha \wedge \beta) \vee (\alpha \wedge \delta).$
- ④ $\vdash \forall x(\alpha \supset \beta) \supset (\exists x(\alpha) \supset \beta),$ *si x no aparece libre en β .*



Semántica de Kripke

Se presenta primero la semántica de Kripke para la lógica proposicional intuicionista, donde se supone que V es el conjunto de variables proposicionales.



Semántica de Kripke

Se presenta primero la semántica de Kripke para la lógica proposicional intuicionista, donde se supone que V es el conjunto de variables proposicionales.

Definición 28

Un **modelo de Kripke** es una estructura $\mathcal{C} = \langle C, \leq, \Vdash \rangle$, donde:

- C es un conjunto no vacío, cuyos elementos son llamados **estados** o **mundos posibles**.
- \leq es un orden parcial en C y es llamada **relación de accesibilidad**.
- \Vdash es una relación binaria de C en V que satisface la siguiente condición de **monotonidad**:

Si $c \leq c'$ y $c \Vdash p$ entonces $c' \Vdash p$.

\Vdash es llamada **relación de forzamiento**, y si $c \Vdash p$ se dice que c fuerza a p .

La relación de forzamiento se extiende a todas las fórmulas así:

Definición 29

Si $\mathcal{C} = \langle C, \leq, \Vdash \rangle$ es un modelo de Kripke, entonces:

- $c \Vdash \varphi \wedge \psi$ si y solo si $c \Vdash \varphi$ y $c \Vdash \psi$.
- $c \Vdash \varphi \vee \psi$ si y solo si $c \Vdash \varphi$ o $c \Vdash \psi$.
- $c \Vdash \varphi \supset \psi$ si y solo si $c' \nVdash \varphi$ o $c' \Vdash \psi$ para todo $c' \geq c$.
- $c \nVdash \perp$.



Semántica de Kripke

La relación de forzamiento se extiende a todas las fórmulas así:

Definición 29

Si $\mathcal{C} = \langle C, \leq, \Vdash \rangle$ es un modelo de Kripke, entonces:

- $c \Vdash \varphi \wedge \psi$ si y solo si $c \Vdash \varphi$ y $c \Vdash \psi$.
- $c \Vdash \varphi \vee \psi$ si y solo si $c \Vdash \varphi$ o $c \Vdash \psi$.
- $c \Vdash \varphi \supset \psi$ si y solo si $c' \nVdash \varphi$ o $c' \Vdash \psi$ para todo $c' \geq c$.
- $c \nVdash \perp$.

Observación: Por la definición de la negación intuicionista, $c \Vdash \neg \varphi$ si y solo si para todo $c' \geq c$ se tiene que $c' \nVdash \varphi$.



Semántica de Kripke

La relación de forzamiento se extiende a todas las fórmulas así:

Definición 29

Si $\mathcal{C} = \langle C, \leq, \Vdash \rangle$ es un modelo de Kripke, entonces:

- $c \Vdash \varphi \wedge \psi$ si y solo si $c \Vdash \varphi$ y $c \Vdash \psi$.
- $c \Vdash \varphi \vee \psi$ si y solo si $c \Vdash \varphi$ o $c \Vdash \psi$.
- $c \Vdash \varphi \supset \psi$ si y solo si $c' \nVdash \varphi$ o $c' \Vdash \psi$ para todo $c' \geq c$.
- $c \nVdash \perp$.

Observación: Por la definición de la negación intuicionista, $c \Vdash \neg \varphi$ si y solo si para todo $c' \geq c$ se tiene que $c' \nVdash \varphi$.

Lema 15

Si $c \leq c'$ y $c \Vdash \varphi$ entonces $c' \Vdash \varphi$.



Semántica de Kripke

Las ideas intuitivas en la semántica de Kripke para IL son:

- Los estados representan estados de conocimiento.



Semántica de Kripke

Las ideas intuitivas en la semántica de Kripke para IL son:

- Los estados representan estados de conocimiento.
- Las proposiciones forzadas en un estado son las proposiciones cuya verdad es conocida en dicho estado de conocimiento.



Semántica de Kripke

Las ideas intuitivas en la semántica de Kripke para IL son:

- Los estados representan estados de conocimiento.
- Las proposiciones forzadas en un estado son las proposiciones cuya verdad es conocida en dicho estado de conocimiento.
- La relación de accesibilidad establece las posibles variaciones (o progreso) en los estados de conocimiento.



Semántica de Kripke

Las ideas intuitivas en la semántica de Kripke para IL son:

- Los estados representan estados de conocimiento.
- Las proposiciones forzadas en un estado son las proposiciones cuya verdad es conocida en dicho estado de conocimiento.
- La relación de accesibilidad establece las posibles variaciones (o progreso) en los estados de conocimiento.
- La condición de monotonicidad establece que las verdades conocidas nunca se olvidan.



Notación:

- Se escribe $\mathcal{C}, c \Vdash \varphi$ para indicar que $c \Vdash \varphi$ en el modelo \mathcal{C} .
- Se escribe $\mathcal{C}, c \Vdash \Gamma$ para indicar que $\mathcal{C}, c \Vdash \gamma$ para todo $\gamma \in \Gamma$.



Notación:

- Se escribe $\mathcal{C}, c \Vdash \varphi$ para indicar que $c \Vdash \varphi$ en el modelo \mathcal{C} .
- Se escribe $\mathcal{C}, c \Vdash \Gamma$ para indicar que $\mathcal{C}, c \Vdash \gamma$ para todo $\gamma \in \Gamma$.

Definición 30 (Relación de consecuencia semántica)

Una fórmula φ se sigue semántica de un conjunto de fórmulas Γ , lo que se denota $\Gamma \Vdash \varphi$, si para todo modelo de Kripke \mathcal{C} y todo $c \in C$ se tiene que si $\mathcal{C}, c \Vdash \Gamma$ entonces $\mathcal{C}, c \Vdash \varphi$.



Notación:

- Se escribe $\mathcal{C}, c \Vdash \varphi$ para indicar que $c \Vdash \varphi$ en el modelo \mathcal{C} .
- Se escribe $\mathcal{C}, c \Vdash \Gamma$ para indicar que $\mathcal{C}, c \Vdash \gamma$ para todo $\gamma \in \Gamma$.

Definición 30 (Relación de consecuencia semántica)

Una fórmula φ se sigue semántica de un conjunto de fórmulas Γ , lo que se denota $\Gamma \Vdash \varphi$, si para todo modelo de Kripke \mathcal{C} y todo $c \in \mathcal{C}$ se tiene que si $\mathcal{C}, c \Vdash \Gamma$ entonces $\mathcal{C}, c \Vdash \varphi$.

Teorema 5

$\Gamma \vdash \varphi$ si y solo si $\Gamma \Vdash \varphi$.

Demostración.

Ver [Sørensen and Urzyczyn, 2006, Teorema 2.5.9].



Proposición 2

- $\not\models p \vee \neg p.$
- $\not\models \neg\neg p \supset p.$
- $\vdash p \supset \neg\neg p.$
- $\not\models ((p \supset q) \supset p) \supset p.$



Proposición 2

- $\not\vdash p \vee \neg p.$
- $\not\vdash \neg\neg p \supset p.$
- $\vdash p \supset \neg\neg p.$
- $\not\vdash ((p \supset q) \supset p) \supset p.$

Teorema 6 (Propiedad de la disyunción)

Si $\vdash \varphi \vee \psi$, entonces $\vdash \varphi$ o $\vdash \psi$.



Definición 31

Tomando fórmulas que contienen solo el conectivo \supset , el **fragmento implicativo** de NJ, denotado $NJ(\supset)$, es el sistema de deducción natural solo con (Ax) , $(\supset I)$ y $(\supset E)$.



Definición 31

Tomando fórmulas que contienen solo el conectivo \supset , el **fragmento implicativo** de NJ, denotado $NJ(\supset)$, es el sistema de deducción natural solo con (Ax) , $(\supset I)$ y $(\supset E)$.

Teorema 7

Para el fragmento implicativo de NJ, se tiene que $\Gamma \vdash \varphi$ si y solo si $\Gamma \Vdash \varphi$.

Demostración.

Ver [Sørensen and Urzyczyn, 2006, Teorema 2.6.1].



Semántica de Kripke

En la semántica de Kripke para la lógica intuicionista de primer orden, en cada estado las fórmulas se interpretan bajo la noción clásica de modelo. Los modelos pueden cambiar de un estado a otro, adicionando nuevos elementos al dominio o aceptando nuevas proposiciones atómicas, lo que refleja el progreso en el conocimiento.



En la semántica de Kripke para la lógica intuicionista de primer orden, en cada estado las fórmulas se interpretan bajo la noción clásica de modelo. Los modelos pueden cambiar de un estado a otro, adicionando nuevos elementos al dominio o aceptando nuevas proposiciones atómicas, lo que refleja el progreso en el conocimiento.

Definición 32

Una estructura $\mathcal{B} = \langle B, f_1^{\mathcal{B}}, \dots, f_n^{\mathcal{B}}, r_1^{\mathcal{B}}, \dots, r_m^{\mathcal{B}}, c_1^{\mathcal{B}}, \dots, c_l^{\mathcal{B}} \rangle$ es una **extensión** de $\mathcal{A} = \langle A, f_1^{\mathcal{A}}, \dots, f_n^{\mathcal{A}}, r_1^{\mathcal{A}}, \dots, r_m^{\mathcal{A}}, c_1^{\mathcal{A}}, \dots, c_l^{\mathcal{A}} \rangle$, lo que se denota $\mathcal{A} \subseteq \mathcal{B}$, si:

- $A \subseteq B$.
- $f_i^{\mathcal{A}} \subseteq f_i^{\mathcal{B}}$, para todo $i \in \{1, \dots, n\}$.
- $r_i^{\mathcal{A}} \subseteq r_i^{\mathcal{B}}$, para todo $i \in \{1, \dots, m\}$.
- $c_i^{\mathcal{A}} = c_i^{\mathcal{B}}$, para todo $i \in \{1, \dots, l\}$.

Definición 33

Un **modelo de Kripke** (para un lenguaje de primer orden) es una estructura $\mathcal{C} = \langle C, \leq, \{\mathcal{A}_c : c \in C\} \rangle$, donde:

- C es un conjunto no vacío de **estados**.
- \leq es un orden parcial en C y es llamada **relación de accesibilidad**.
- Los \mathcal{A}_c son modelos clásicos que satisfacen la siguiente condición de **monotonicidad**:

si $c \leq c'$ entonces $\mathcal{A}_c \subseteq \mathcal{A}_{c'}$.



Definición 34

Dado un modelo de Kripke $\mathcal{C} = \langle C, \leq, \{\mathcal{A}_c : c \in C\} \rangle$, la **relación de forzamiento** en un estado $c \in C$ y bajo una evaluación e (que asigna objetos del dominio de \mathcal{A}_c a variables) se define así:

- $c, e \Vdash rt_1 \dots t_n$ si y solo $\mathcal{A}_c, e \Vdash rt_1 \dots t_n$ (clásicamente).
- $c, e \nVdash \perp$.
- $c, e \Vdash \varphi \wedge \psi$ si y solo si $c, e \Vdash \varphi$ y $c, e \Vdash \psi$.
- $c, e \Vdash \varphi \vee \psi$ si y solo si $c, e \Vdash \varphi$ o $c, e \Vdash \psi$.
- $c, e \Vdash \varphi \supset \psi$ si y solo si $c', e \nVdash \varphi$ o $c', e \Vdash \psi$ para todo $c' \geq c$.
- $c, e \Vdash \exists x \varphi$ si y solo si $c, e(x \mapsto a) \Vdash \varphi$ para algún a en el dominio de \mathcal{A}_c .
- $c, e \Vdash \forall x \varphi$ si y solo si $c', e(x \mapsto a) \Vdash \varphi$ para todo $c' \geq c$ y todo a en el dominio de $\mathcal{A}_{c'}$.

Definición 35 (Relación de consecuencia semántica)

Una fórmula φ se sigue semántica de un conjunto de fórmulas Γ , lo que se denota $\Gamma \Vdash \varphi$, si para todo modelo de Kripke \mathcal{C} , todo $c \in C$ y toda evaluación e se tiene que si $\mathcal{C}, c, e \Vdash \Gamma$ entonces $\mathcal{C}, c, e \Vdash \varphi$.



Definición 35 (Relación de consecuencia semántica)

Una fórmula φ se sigue semántica de un conjunto de fórmulas Γ , lo que se denota $\Gamma \Vdash \varphi$, si para todo modelo de Kripke \mathcal{C} , todo $c \in C$ y toda evaluación e se tiene que si $\mathcal{C}, c, e \Vdash \Gamma$ entonces $\mathcal{C}, c, e \Vdash \varphi$.

Teorema 8

$\Gamma \vdash \varphi$ si y solo si $\Gamma \Vdash \varphi$.

Demostración.

Ver [Sørensen and Urzyczyn, 2006, Teorema 8.6.7].



Semántica de Kripke

Definición 35 (Relación de consecuencia semántica)

Una fórmula φ se sigue semántica de un conjunto de fórmulas Γ , lo que se denota $\Gamma \Vdash \varphi$, si para todo modelo de Kripke \mathcal{C} , todo $c \in C$ y toda evaluación e se tiene que si $\mathcal{C}, c, e \Vdash \Gamma$ entonces $\mathcal{C}, c, e \Vdash \varphi$.

Teorema 8

$\Gamma \vdash \varphi$ si y solo si $\Gamma \Vdash \varphi$.

Demostración.

Ver [Sørensen and Urzyczyn, 2006, Teorema 8.6.7]. □

Proposición 3

$\neg(\neg\forall x\neg\varphi) \supset (\exists x\varphi)$.

Índice

- 1 Breve reseña histórica
- 2 Cálculo lambda sin tipos
 - Términos del cálculo lambda
 - α -conversión
 - β -reducción y β -conversión
 - Combinador de punto fijo
 - Expresividad del cálculo lambda
- 3 Cálculo lambda simplemente tipificado
 - Tipificado a la Curry
 - Tipificado a la Church
- 4 Lógica intuicionista
 - Interpretación BHK
 - Sistema de deducción natural
 - Semántica de Kripke
- 5 Correspondencia de Curry-Howard
- 6 Referencias bibliográficas



Correspondencia de Curry-Howard

Referencia usada en esta sección: [Sørensen and Urzyczyn, 2006] (con algunos cambios de notación).



Correspondencia de Curry-Howard

Referencia usada en esta sección: [Sørensen and Urzyczyn, 2006] (con algunos cambios de notación).

Se usan las expresiones **isomorfismo de Curry-Howard**, **correspondencia Curry-Howard**, **correspondencia de proposiciones como tipos**, entre otras, para hacer referencia a teoremas que muestran equivalencias entre derivaciones en un sistema de tipos y derivaciones en un sistema lógico, bajo una cierta interpretación de fórmulas lógicas como siendo tipos.



Correspondencia de Curry-Howard

Referencia usada en esta sección: [Sørensen and Urzyczyn, 2006] (con algunos cambios de notación).

Se usan las expresiones **isomorfismo de Curry-Howard**, **correspondencia Curry-Howard**, **correspondencia de proposiciones como tipos**, entre otras, para hacer referencia a teoremas que muestran equivalencias entre derivaciones en un sistema de tipos y derivaciones en un sistema lógico, bajo una cierta interpretación de fórmulas lógicas como siendo tipos.

En 1958, Haskell Curry estableció la **correspondencia entre el cálculo lambda simplemente tipificado y el fragmento implicativo de la lógica intuicionista**. William A. Howard y Nicolaas G. de Bruijn extendieron dicha correspondencia a la lógica de predicados de primer orden.



Correspondencia de Curry-Howard

En las siguientes definiciones, se supone que el conjunto de variables y el conjunto de variables de tipos es el mismo conjunto V .



Correspondencia de Curry-Howard

En las siguientes definiciones, se supone que el conjunto de variables y el conjunto de variables de tipos es el mismo conjunto V .

Definición 36

La **función de interpretación** $|\cdot|$, de fórmulas del fragmento implicativo de IL en tipos de $\lambda \rightarrow$, se define inductivamente por:

$$\begin{aligned} |p| &= p, \text{ para } p \in V; \\ |(\alpha \supset \beta)| &= (|\alpha| \rightarrow |\beta|). \end{aligned}$$



Correspondencia de Curry-Howard

En las siguientes definiciones, se supone que el conjunto de variables y el conjunto de variables de tipos es el mismo conjunto V .

Definición 36

La **función de interpretación** $|\cdot|$, de fórmulas del fragmento implicativo de IL en tipos de $\lambda \rightarrow$, se define inductivamente por:

$$\begin{aligned} |p| &= p, \text{ para } p \in V; \\ |(\alpha \supset \beta)| &= (|\alpha| \rightarrow |\beta|). \end{aligned}$$

Definición 37

La **función inversa** a $|\cdot|$, denotada $|\cdot|^{-1}$, se define inductivamente por:

$$\begin{aligned} |p|^{-1} &= p, \text{ para } p \in V; \\ |(\alpha \rightarrow \beta)|^{-1} &= (|\alpha|^{-1} \supset |\beta|^{-1}). \end{aligned}$$

Correspondencia de Curry-Howard

Notación:

- Para un conjunto de fórmulas $\Gamma = \{\varphi_1, \dots, \varphi_n\}$, se usará $|\Gamma|$ para denotar el contexto $\{x_1 : |\varphi_1|, \dots, x_n : |\varphi_n|\}$.
- Para un contexto $\Gamma = \{x_1 : \varphi_1, \dots, x_n : \varphi_n\}$, se usará $|\Gamma|^{-1}$ para denotar el conjunto de fórmulas $\{|\varphi_1|^{-1}, \dots, |\varphi_n|^{-1}\}$.



Correspondencia de Curry-Howard

Notación:

- Para un conjunto de fórmulas $\Gamma = \{\varphi_1, \dots, \varphi_n\}$, se usará $|\Gamma|$ para denotar el contexto $\{x_1 : |\varphi_1|, \dots, x_n : |\varphi_n|\}$.
- Para un contexto $\Gamma = \{x_1 : \varphi_1, \dots, x_n : \varphi_n\}$, se usará $|\Gamma|^{-1}$ para denotar el conjunto de fórmulas $\{|\varphi_1|^{-1}, \dots, |\varphi_n|^{-1}\}$.

Teorema 9 (Correspondencia Curry-Howard para el fragmento implicativo de IL)

- 1 Si $\Gamma \vdash \varphi$ en $NJ(\supset)$, entonces $|\Gamma| \vdash M : |\varphi|$ en $\lambda \rightarrow$, para algún término M .
- 2 Si $\Gamma \vdash M : \varphi$ en $\lambda \rightarrow$, entonces $|\Gamma|^{-1} \vdash |\varphi|^{-1}$ en $NJ(\supset)$.



Correspondencia de Curry-Howard

Notación:

- Para un conjunto de fórmulas $\Gamma = \{\varphi_1, \dots, \varphi_n\}$, se usará $|\Gamma|$ para denotar el contexto $\{x_1 : |\varphi_1|, \dots, x_n : |\varphi_n|\}$.
- Para un contexto $\Gamma = \{x_1 : \varphi_1, \dots, x_n : \varphi_n\}$, se usará $|\Gamma|^{-1}$ para denotar el conjunto de fórmulas $\{|\varphi_1|^{-1}, \dots, |\varphi_n|^{-1}\}$.

Teorema 9 (Correspondencia Curry-Howard para el fragmento implicativo de IL)

- 1 Si $\Gamma \vdash \varphi$ en $NJ(\supset)$, entonces $|\Gamma| \vdash M : |\varphi|$ en $\lambda \rightarrow$, para algún término M .
- 2 Si $\Gamma \vdash M : \varphi$ en $\lambda \rightarrow$, entonces $|\Gamma|^{-1} \vdash |\varphi|^{-1}$ en $NJ(\supset)$.

Observación: Bajo la correspondencia Curry-Howard los términos del cálculo lambda codifican demostraciones.



Correspondencia de Curry-Howard

En teoría de la demostración, un **rodeo** consiste en una regla de introducción seguida de una regla de eliminación aplicada a la misma fórmula introducida. La **normalización** de una demostración consiste en eliminar los rodeos.



Correspondencia de Curry-Howard

En teoría de la demostración, un **rodeo** consiste en una regla de introducción seguida de una regla de eliminación aplicada a la misma fórmula introducida. La **normalización** de una demostración consiste en eliminar los rodeos.

El formato general de un **rodeo en NJ(\supset)** es:

$$\frac{\Gamma \vdash \varphi \quad \frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \supset \psi} \supset I}{\Gamma \vdash \psi} \supset E$$



Correspondencia de Curry-Howard

En formato estándar de deducción natural:

$$\begin{array}{c}
 \begin{array}{c}
 (*) \\
 \vdots \\
 \varphi
 \end{array}
 \quad
 \frac{
 \begin{array}{c}
 [\varphi] \\
 \vdots \\
 \psi
 \end{array}
 }{
 \varphi \supset \psi
 }
 \supset I \\
 \hline
 \psi
 \quad
 \supset E
 \end{array}
 \Longrightarrow
 \begin{array}{c}
 (*) \\
 \vdots \\
 \varphi \\
 \vdots \\
 \psi
 \end{array}$$



Correspondencia de Curry-Howard

En formato estándar de deducción natural:

$$\begin{array}{c}
 \begin{array}{c}
 (*) \\
 \vdots \\
 \varphi
 \end{array}
 \quad
 \frac{\begin{array}{c}
 \vdots \\
 \psi
 \end{array} \supset I}{\varphi \supset \psi} \supset E \\
 \hline
 \psi
 \end{array}
 \Rightarrow
 \begin{array}{c}
 (*) \\
 \vdots \\
 \varphi \\
 \vdots \\
 \psi
 \end{array}$$

En el sistema de tipos simples (β -reducción corresponde a eliminación de rodeos):

$$\begin{array}{c}
 \begin{array}{c}
 (*) \\
 \vdots \\
 M : \varphi
 \end{array}
 \quad
 \frac{\begin{array}{c}
 [x : \varphi] \\
 \vdots \\
 N : \psi
 \end{array} \supset I}{\lambda x : \varphi. N : \varphi \rightarrow \psi} \supset E \\
 \hline
 (\lambda x : \varphi. N) M : \psi
 \end{array}
 \Rightarrow
 \begin{array}{c}
 (*) \\
 \vdots \\
 M : \varphi \\
 \vdots \\
 N[x := M] : \psi
 \end{array}$$



Correspondencia de Curry-Howard

Para ampliar la correspondencia de Curry-Howard a la lógica proposicional intuicionista, se extiende el cálculo lambda simplemente tipificado con **productos**, **sumas** (o **uniones disyuntas**) y el **tipo vacío**:

$$\frac{\Gamma \vdash M : \varphi \quad \Gamma \vdash N : \psi}{\Gamma \vdash \langle M, N \rangle : \varphi \times \psi} \text{ (Prod)}$$

$$\frac{\Gamma \vdash M : \varphi \times \psi}{\Gamma \vdash \pi_1(M) : \varphi} \text{ (Proj}_1\text{)} \quad \frac{\Gamma \vdash M : \varphi \times \psi}{\Gamma \vdash \pi_2(M) : \psi} \text{ (Proj}_2\text{)}$$

$$\frac{\Gamma \vdash M : \varphi}{\Gamma \vdash \text{in}_1^{\varphi+\psi}(M) : \varphi + \psi} \text{ (Inj}_1\text{)} \quad \frac{\Gamma \vdash M : \psi}{\Gamma \vdash \text{in}_2^{\varphi+\psi}(M) : \varphi + \psi} \text{ (Inj}_2\text{)}$$

$$\frac{\Gamma \vdash L : \varphi + \psi \quad \Gamma, x : \varphi \vdash M : \theta \quad \Gamma, y : \psi \vdash N : \theta}{\Gamma \vdash (\text{case } L \text{ of } [x]M \text{ or } [y]N) : \theta} \text{ (Case)}$$



Correspondencia de Curry-Howard

$$\frac{\Gamma \vdash M : \emptyset}{\Gamma \vdash \epsilon_{\varphi}(M) : \varphi} \text{ (Emp)}$$



Correspondencia de Curry-Howard

$$\frac{\Gamma \vdash M : \emptyset}{\Gamma \vdash \epsilon_{\varphi}(M) : \varphi} \text{ (Emp)}$$

Para los nuevos λ -términos, se adicionan las siguientes β -reducciones:

$$\pi_1(\langle M, N \rangle) \rightarrow_{\beta} M,$$

$$\pi_2(\langle M, N \rangle) \rightarrow_{\beta} N,$$

$$\text{case in}_1^{\varphi+\psi}(M) \text{ of } [x]P \text{ or } [y]Q \rightarrow_{\beta} P[x := M],$$

$$\text{case in}_2^{\varphi+\psi}(M) \text{ of } [x]P \text{ or } [y]Q \rightarrow_{\beta} Q[y := M].$$



$$\frac{\Gamma \vdash M : \emptyset}{\Gamma \vdash \epsilon_{\varphi}(M) : \varphi} \text{ (Emp)}$$

Para los nuevos λ -términos, se adicionan las siguientes β -reducciones:

$$\pi_1(\langle M, N \rangle) \rightarrow_{\beta} M,$$

$$\pi_2(\langle M, N \rangle) \rightarrow_{\beta} N,$$

$$\text{case in}_1^{\varphi+\psi}(M) \text{ of } [x]P \text{ or } [y]Q \rightarrow_{\beta} P[x := M],$$

$$\text{case in}_2^{\varphi+\psi}(M) \text{ of } [x]P \text{ or } [y]Q \rightarrow_{\beta} Q[y := M].$$

Observación: Los superíndices en los términos $\text{in}_i^{\varphi+\psi}$ son usualmente omitidos, aunque son necesarios para asegurar la unicidad de tipos en el tipificado a la Church.



Correspondencia de Curry-Howard

Las funciones para interpretar fórmulas como tipos y viceversa se extienden así:

Definición 38

La **función de interpretación** $|\cdot|$, de fórmulas de la lógica proposicional intuicionista en el sistema de tipos simples extendido, se define inductivamente por:

$$\begin{aligned} |p| &= p, \text{ para } p \in V; \\ |(\alpha \supset \beta)| &= (|\alpha| \rightarrow |\beta|); \\ |(\alpha \wedge \beta)| &= (|\alpha| \times |\beta|); \\ |(\alpha \vee \beta)| &= (|\alpha| + |\beta|); \\ |\perp| &= \emptyset. \end{aligned}$$



Definición 39

La **función inversa** a $|\cdot|$, denotada $|\cdot|^{-1}$, se define inductivamente por:

$$\begin{aligned} |p|^{-1} &= p, \text{ para } p \in V; \\ |(\alpha \rightarrow \beta)|^{-1} &= (|\alpha|^{-1} \supset |\beta|^{-1}); \\ |(\alpha \times \beta)|^{-1} &= (|\alpha|^{-1} \wedge |\beta|^{-1}); \\ |(\alpha + \beta)|^{-1} &= (|\alpha|^{-1} \vee |\beta|^{-1}); \\ |\emptyset|^{-1} &= \perp. \end{aligned}$$



Correspondencia de Curry-Howard

Definición 39

La **función inversa** a $|\cdot|$, denotada $|\cdot|^{-1}$, se define inductivamente por:

$$\begin{aligned} |p|^{-1} &= p, \text{ para } p \in V; \\ |(\alpha \rightarrow \beta)|^{-1} &= (|\alpha|^{-1} \supset |\beta|^{-1}); \\ |(\alpha \times \beta)|^{-1} &= (|\alpha|^{-1} \wedge |\beta|^{-1}); \\ |(\alpha + \beta)|^{-1} &= (|\alpha|^{-1} \vee |\beta|^{-1}); \\ |\emptyset|^{-1} &= \perp. \end{aligned}$$

Teorema 10 (Correspondencia Curry-Howard para la lógica proposicional intuicionista)

- ① Si $\Gamma \vdash \varphi$ en NJ, entonces $|\Gamma| \vdash M : |\varphi|$ en $\lambda \rightarrow$ extendido, para algún término M .
- ② Si $\Gamma \vdash M : \varphi$ en $\lambda \rightarrow$ extendido, entonces $|\Gamma|^{-1} \vdash |\varphi|^{-1}$ en NJ.

Índice

- 1 Breve reseña histórica
- 2 Cálculo lambda sin tipos
 - Términos del cálculo lambda
 - α -conversión
 - β -reducción y β -conversión
 - Combinador de punto fijo
 - Expresividad del cálculo lambda
- 3 Cálculo lambda simplemente tipificado
 - Tipificado a la Curry
 - Tipificado a la Church
- 4 Lógica intuicionista
 - Interpretación BHK
 - Sistema de deducción natural
 - Semántica de Kripke
- 5 Correspondencia de Curry-Howard
- 6 Referencias bibliográficas



Referencias bibliográficas



Barendregt, H. (1992).

Lambda calculi with types.

In Abramsky, S., Gabbay, D. M., and Maibaum, T. S. E., editors, *Handbook of Logic in Computer Science*, volume 2, pages 117–309. Clarendon Press.



Barendregt, H. and Barendsen, E. (Revised edition, march 2000).

Introduction to lambda calculus.



Nederpelt, R. and Geuvers, H. (2014).

Type Theory and Formal Proof. An Introduction.

Cambridge University Press.



Sørensen, M. H. and Urzyczyn, P. (2006).

Lectures on the Curry-Howard Isomorphism, volume 149 of *Studies in Logic and the Foundations of Mathematics*.

Elsevier.

