

¿Por qué tu computadora no entiende ZFC?

Una introducción a Lean y la Teoría de Tipos.

Luis M. Reyes

Instituto de Matemáticas, UNAM

June 3, 2026



Tabla de contenido

Verdad vs Algoritmos.

Cálculo Lambda y Teoría de Tipos.

Matemáticas en Lean.

Verdad vs Algoritmos.

¿A quién va dirigido?

Proposición 1

Sea $f : \alpha \rightarrow \beta$ una función suprayectiva y $Y \subseteq \beta$. Entonces la imagen de la preimagen de Y es el propio Y . Formalmente: $f[f^{-1}[Y]] = Y$.

¿A quién va dirigido?

Proposición 1

Sea $f : \alpha \rightarrow \beta$ una función suprayectiva y $Y \subseteq \beta$. Entonces la imagen de la preimagen de Y es el propio Y . Formalmente: $f[f^{-1}[Y]] = Y$.

Proof.

Sea $x \in f(f^{-1}(B))$, entonces $\exists y \in f^{-1}(B)$ tal que

$$x = f(y) \tag{1}$$

Pero por definición de imagen inversa, se tiene $x = f(y) \in B$, obteniendo $x \in B$.

Conversamente, sea $x \in B$, como f es sobreyectiva, entonces $\exists y \in A$ tal que $f(y) = x$, pero esto implica que $y \in f^{-1}(B)$, por definición de imagen inversa, implica que $x = f(y) \in f(f^{-1}(B))$. Demostrando que

$$x \in B \Leftrightarrow x \in f(f^{-1}(B))$$

concluyendo por definición de igualdad

Lógica Matemática.

¿Cómo se razona y deduce correctamente?,

Lógica Sintáctica. $\leftarrow$ Lógica $\rightarrow$ Lógica Semántica.

Sintaxis: Lógica Proposicional.

1. **Alfabeto:** $V \cup \{\neg, \rightarrow, (,)\}$
2. **Expresiones bien formadas:** Satisfacen las siguientes reglas:
 - 2.1 $p \in V$ entonces es bien formada.
 - 2.2 ϕ, ψ son bien formada entonces $(\neg\phi)$ y $(\phi \rightarrow \psi)$ son bien formadas.
 - 2.3 Una expresión bien formada es una combinación finita de las dos anteriores.
3. **Axiomas:**
 - 3.1

$$\varphi \rightarrow (\psi \rightarrow \varphi)$$

3.2

$$(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$$

3.3

$$(\neg\psi \rightarrow \neg\varphi) \rightarrow (\varphi \rightarrow \psi)$$

4. **Reglas de Inferencia:**

4.1

$$\frac{\varphi, \quad \varphi \rightarrow \psi}{\psi}$$

Sintaxis: Lógica Proposicional

Define

$$\varphi \vee \psi := ((\neg \varphi) \rightarrow \psi).$$

■ Principio del tercero excluido: $\models P \vee \neg P$.

Sintaxis: Lógica Proposicional

Define

$$\varphi \vee \psi := ((\neg\varphi) \rightarrow \psi).$$

■ **Principio del tercero excluido:** $\models P \vee \neg P$.

1. De (3.1) : $\varphi \rightarrow ((\varphi \rightarrow \varphi) \rightarrow \varphi)$
2. De (3.2) : $(\varphi \rightarrow ((\varphi \rightarrow \varphi) \rightarrow \varphi)) \rightarrow ((\varphi \rightarrow (\varphi \rightarrow \varphi)) \rightarrow (\varphi \rightarrow \varphi))$
3. Inferencia : $(\varphi \rightarrow (\varphi \rightarrow \varphi)) \rightarrow (\varphi \rightarrow \varphi)$
4. De (3.1) : $\varphi \rightarrow (\varphi \rightarrow \varphi)$
5. Inferencia : $\varphi \rightarrow \varphi$, sustituyendo $\varphi = \neg P$, $\neg P \rightarrow \neg P$.

Semántica: Lógica Proposicional.

1. La verdad se mide como una función $A^n \rightarrow \{0, 1\}$.

Verificación de la tautología:

$$\models P \vee \neg P$$

P	$\neg P$	$P \vee \neg P$
V	F	V
F	V	V

Sintaxis vs Semantica: Lógica de Primer Orden.

$$\varphi \equiv \exists x(x \cdot x + 1 = 0)$$

Sintaxis vs Semantica: Lógica de Primer Orden.

$$\varphi \equiv \exists x(x \cdot x + 1 = 0)$$

Usando verdad de Tarski:

1. **Modelo 1:** Se tiene

$$\mathbb{R} \not\models \varphi$$

2. **Modelo 2:** Se tiene

$$\mathbb{C} \models \varphi$$

Demostracion y Computabilidad.

1. **Problema 10 de Hilbert:** Sea P un polinomio con coeficientes enteros:

$$\psi := \exists x_1 \dots \exists x_n [P(x_1, \dots, x_n) = 0]$$

Demostracion y Computabilidad.

1. **Problema 10 de Hilbert:** Sea P un polinomio con coeficientes enteros:

$$\Psi := \exists x_1 \dots \exists x_n [P(x_1, \dots, x_n) = 0]$$

- Idear un proceso de acuerdo con el cual pueda determinarse, en un número finito de operaciones, si la ecuación es resoluble en números racionales enteros.

Demostracion y Computabilidad.

1. **Problema 10 de Hilbert:** Sea P un polinomio con coeficientes enteros:

$$\Psi := \exists x_1 \dots \exists x_n [P(x_1, \dots, x_n) = 0]$$

- Idear un proceso de acuerdo con el cual pueda determinarse, en un número finito de operaciones, si la ecuación es resoluble en números racionales enteros.
- Objetivo:

$$\mathbb{N} \models \Psi$$

Cálculo Lambda y Teoría de Tipos.

Sintaxis del Algoritmo: Cálculo Lambda.

1. **Alfabeto:** V , λ , punto, $($, $)$.
2. **Expresiones bien formada:**
 - 2.1 **Variable** (Referencia): Si $x \in V$, entonces $x \in \Lambda$.
 - 2.2 **Abstracción** (Definición de Función): Si $x \in V$ y $M \in \Lambda$, entonces $(\lambda x.M) \in \Lambda$.
 - 2.3 **Aplicación** (Llamada a Función): Si $M, N \in \Lambda$, entonces $(M N) \in \Lambda$.
3. **Axiomas**

3.1 α -conversión:

$$\lambda x.M \equiv_{\alpha} \lambda y.M[x := y]$$

3.2 β -reducción:

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

Cálculo Lambda.

■ Inferencia:

1. **Reflexividad:** $M \rightarrow M$.
2. **Transitividad:** Si $M \rightarrow N$ y $N \rightarrow P$, entonces $M \rightarrow P$.
3. **Congruencia (Contexto):**
 - 3.1 Si $M \rightarrow M'$, entonces $(M\ N) \rightarrow (M'\ N)$.
 - 3.2 Si $N \rightarrow N'$, entonces $(M\ N) \rightarrow (M\ N')$.
 - 3.3 Si $M \rightarrow M'$, entonces $(\lambda x.M) \rightarrow (\lambda x.M')$.

Semántica: Igualdad Definicional

Definición 2 (Igualdad Definicional)

Sean t y s dos términos. Decimos que son **definicionalmente iguales**, denotado $t \equiv s$, si ambos reducen a la misma **forma normal** (salvo renombramiento de variables ligadas).

Semántica: Igualdad Definicional

Definición 2 (Igualdad Definicional)

Sean t y s dos términos. Decimos que son **definicionalmente iguales**, denotado $t \equiv s$, si ambos reducen a la misma **forma normal** (salvo renombramiento de variables ligadas).

$$2 + 2 = 4$$

Sobre Teoría de Tipos.

Definición 3 (El Tipo Función)

Dados dos tipos A y B :

- **Formación:** $A \rightarrow B$ es un tipo.
- **Introducción (λ -abstracción):** Si asumiendo $x : A$ podemos construir un término $\Phi : B$, entonces la abstracción $\lambda x. \Phi$ tiene tipo $A \rightarrow B$.
- **Eliminación (Aplicación):** Si $f : A \rightarrow B$ y $a : A$, entonces $f(a) : B$.
- **Computación:** $(\lambda x. \Phi)(a) \equiv \Phi[x := a]$.

Definición 4 (Contexto)

Un contexto Γ es una lista finita de declaraciones de variables con sus respectivos tipos $(x_1 : A_1, \dots, x_n : A_n)$. El juicio $\Gamma \vdash t : B$ indica que el término t tiene tipo B bajo las suposiciones listadas en Γ .

Diccionario Curry-Howard.

Lógica: Implicación	Tipos: Funciones
$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}$ <i>Introducción ($\Rightarrow$)</i>	$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x. t : A \rightarrow B}$ <i>Abstracción (λ)</i>
$\frac{\Gamma \vdash A \rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}$ <i>Eliminación (Modus Ponens)</i>	$\frac{\Gamma \vdash f : A \rightarrow B \quad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B}$ <i>Aplicación</i>

Tipos Productos y Sumas.

Definición 5 (Tipos Π o Productos Dependientes)

Dada una familia de tipos P indexada por A , el **Tipo** Π , denotado $\Pi_{(x:A)} P(x)$, generaliza las funciones.

- **Introducción:** Un término es una función $\lambda x.t$ tal que para cada entrada $a : A$, el resultado tiene tipo $P(a)$.
- **Eliminación:** La aplicación $f(a)$ devuelve un valor de tipo $P(a)$.

Interpretación Lógica: Corresponde al cuantificador universal $\forall x \in A, P(x)$.

Definición 6 (Tipos Σ o Sumas Dependientes)

Dada una familia P sobre A , el **Tipo** Σ , denotado $\Sigma_{(x:A)} P(x)$, generaliza el producto cartesiano.

- **Introducción:** Un elemento es un par dependiente $\langle a, b \rangle$ donde $a : A$ y $b : P(a)$.
- **Eliminación:** Proyecciones para recuperar el testigo a y la prueba b .

Interpretación Lógica: Corresponde al cuantificador existencial $\exists x \in A. P(x)$

Matemáticas en Lean.

Escribiendo en Lean.

1. **Datos** (`def`): Se utiliza para definir objetos cuyo contenido computacional es relevante, tales como funciones, estructuras algebraicas o números.
2. **Proposiciones** (`theorem` / `lemma`): Se reserva para afirmar verdades lógicas.

Concepto Teórico	Notación Matemática	Sintaxis en Lean 4
Juicio de Tipado	$a \in A$ (conjuntos) ó $a : A$	<code>a : A</code>
λ -Abstracción	$\lambda x. t$	<code>fun x => t</code>
Tipo Función	$A \rightarrow B$	<code>A -> B</code>
Cuantificador Universal	$\forall x \in A, P(x)$	<code>forall x : A, P x</code>
Implicación	$P \implies Q$	<code>P -> Q</code>
Aplicación	$f(x)$	<code>f x</code>

1. **El Contexto Local** (Γ): La lista de objetos y hechos disponibles.
2. **El Objetivo** ($\vdash$ **Target**): La proposición que falta demostrar dado el contexto actual.

Traducir Demostraciones.

1. El inicio: Configuración del Contexto
2. El desarrollo: Razonamiento hacia atrás
3. Manejo de Igualdades y Reescritura
4. Descomposición de Hipótesis
5. El cierre: Exactitud

El ejemplo

Se define el tipo

```
theorem image_preimage_of_surjective
  {alpha beta : Type} (f : alpha -> beta)
  (h_surj : Surjective f) (Y : Set beta) :
  f '' (f ⁻¹ Y) = Y := by
```

La prueba

Razonamiento Matemático	Código en Lean (Tácticas)
1. Para demostrar la igualdad de conjuntos, tomamos un y arbitrario y probamos la doble implicación ($y \in A \iff y \in B$).	<code>ext y</code> <code>constructor</code>
<i>blue!10</i> Dirección 1 ($\rightarrow$): Sea $y \in f[f^{-1}[Y]]$. Ver $y \in Y$.	
2. Asumimos la hipótesis. Por def. de imagen, existe un x tal que $x \in f^{-1}[Y]$ y $f(x) = y$.	<code>intro h</code> <code>rcases h with ⟨x, hx, heq⟩</code>
3. Sustituimos $f(x)$ por y . Como $x \in f^{-1}[Y]$, entonces $f(x) \in Y$. Al sustituir, $y \in Y$.	<code>rw [← heq]</code> <code>exact hx</code>

La prueba

Razonamiento Matemático	Código en Lean (Tácticas)
<i>blue!10</i> Dirección 2 ($\leftarrow$): Sea $y \in Y$. Ver $y \in f[f^{-1}[Y]]$.	
4. Asumimos que $y \in Y$.	<code>intro hy</code>
5. Como f es suprayectiva, por definición existe un x tal que $f(x) = y$.	<code>rcases h_surj y with ⟨x, hx⟩</code>
6. Para demostrar que y está en la imagen, debemos exhibir un testigo. Usamos este x .	<code>rw [← hx]</code> <code>use x</code>

La prueba

Razonamiento Matemático	Código en Lean (Tácticas)
<i>(Continuación Dirección 2...)</i>	
7. Debemos probar dos cosas: a) Que x está en la preimagen ($f(x) \in Y$). Cierto pues $y \in Y$ y $f(x) = y$. b) Que la imagen de x es $f(x)$. Trivial.	<code>constructor</code> · <code>rw [hx]; exact hy</code> · <code>rfl</code>

Bibliografía I

- [1] Jeremy Avigad and Patrick Massot. Mathematics in lean, 2020.
- [2] Andrej Bauer, Jason Gross, Peter LeFanu Lumsdaine, Michael Shulman, Matthieu Sozeau, and Bas Spitters. The HoTT library: a formalization of homotopy type theory in Coq. In *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs*, pages 164–172, 2017.
- [3] Julio Ernesto Solís Daun and Yolanda Torres Falcón. *Lógica matemática*. UAM, Unidad Iztapalapa, 1995.
- [4] Alan G Hamilton. *Logic for mathematicians*. Cambridge University Press, 1988.