Formalización Matemática en Lean y su relación con asistentes inteligentes.

Seminario del curso de teoría de Tipos

Docente: Juan Carlos Agudelo
Autor: Sebastián Zapata Álzate

Universidad de Antioquia

2025

Tabla de contenidos

Introducción

La formalización y verificación de teorías matemáticas por medio de asistentes computacionales ha experimentado un crecimiento significativo debido a varios aspectos. Entre ellos, el avance que han tenido ciertas teorías como la de tipos para construir diferentes formalizadores (Lean, Rocq, Agda) además de algunas situaciones que han acontecido en los últimos años dentro de la comunidad matemática que le han dado protagonismo a los formalizadores y por ende la conexión hacia otras herramientas y tecnologías que potencien su uso.

Conjetura abc

Sean $a, b, c \in \mathbb{Z}^+$ coprimos tales que $a + b = c$ y fijémonos en los factores primos que dividen cualquiera de los tres números. Por ejemplo, para la ecuación $5 + 16 = 21$ los factores primos son 5, 2, 3 y 7 cuyo producto es 210, que es grande comparado con el mayor de los números $c = 21$.
Por otro lado, para la tripleta $5 + 27 = 32$ cuyos factores son 5, 3 y 2 y el producto 30 en este caso es menor que $c = 32$. La razón por la que el producto es tan pequeño es que los factores primos se repiten varias veces.
Si empezamos a jugar con tripletas (a, b, c) que cumplan las condiciones que acabamos de describir vamos a encontrar que el segundo escenario es extremadamente raro, pues si consideramos las 3044 de estas tripletas que podemos formar para a,b y c entre 1 y 100 existen solo 7 de ellas en las que el producto de los primos es menor que c.
La conjetura abc da una formulación precisa del por qué este tipo de tripletas son tan escasas.

Enunciado de la conjetura

Conjetura (abc)

Sean $a, b, c \in \mathbb{Z}^+$ coprimos tales que $a + b = c$ entonces para cada $\epsilon > 0$ existe una cantidad finita de tripletas (a, b, c) tales que:

$$c > rad(abc)^{1+\epsilon}$$

donde $rad(n)$ denota el producto de los primos diferentes que dividen a n

Volviendo a nuestro ejemplo de $5 + 27 = 32$, note que 32 es mayor que 30 pero solo por una cantidad pequeña, más aún, 32 es menor que $30^{1.02} = 32.11$ lo cuál está codificado de manera general en la conjetura con la posibilidad de escoger el ϵ arbitrariamente pequeño.

Importancia de la conjetura

- La conjetura está íntimamente relacionada con el comportamiento de las operaciones de suma y producto, a su vez que establece relaciones profundas de los factores primos de un numero entero positivo.
- Se ha demostrado que una respuesta afirmativa a esta conjetura permite brindar una nueva demostración a teoremas como el ultimo teorema de Fermat (del cual volveremos más adelante)
- Al igual que con Fermat, la resolución de esta conjetura ha dado y sigue dando lugar a nuevas herramientas y resultados en otras áreas de las matemáticas como la teoría analítica de números o la geometría aritmética.

Algo no cuadra...

La búsqueda por esta conjetura presentó en el año 2012 un hito sorprendente cuando el matemático japonés Shinichi Mochizuki publicó en línea una serie de artículos en los que afirmaba haber probado la conjetura.

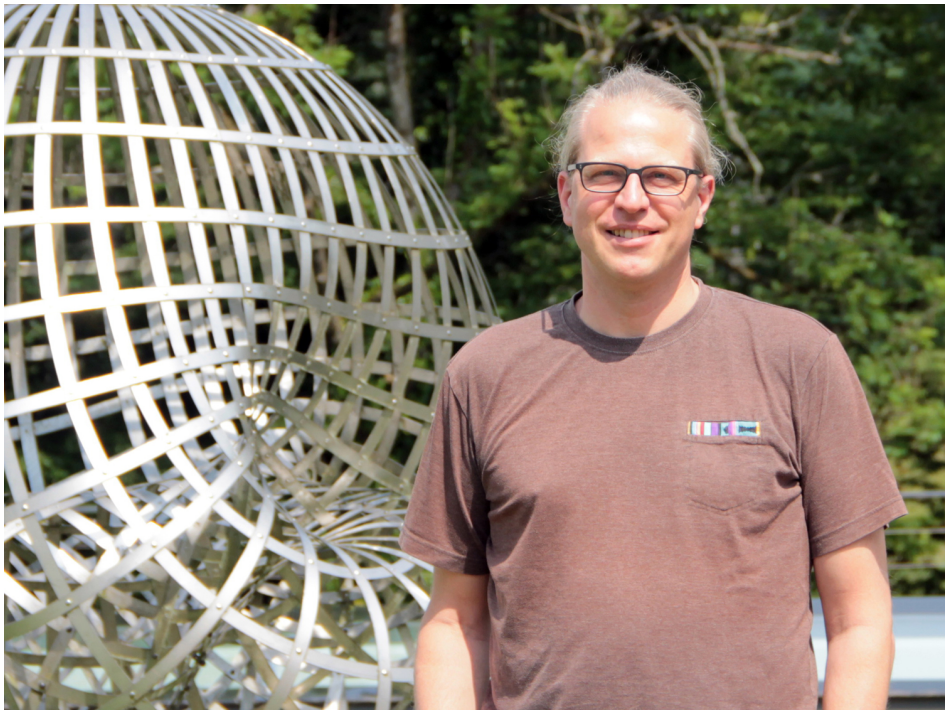
INTER-UNIVERSAL TEICHMÜLLER THEORY I: CONSTRUCTION OF HODGE THEATERS

SHINICHI MOCHIZUKI

May 2020

ABSTRACT. The present paper is the first in a series of four papers, the goal of which is to establish an *arithmetic* version of *Teichmüller theory* for **number fields** equipped with an **elliptic curve** — which we refer to as **"inter-universal Teichmüller theory"** — by applying the theory of *semi-graphs of anabelioids*, *Frobenioids*, the *étale theta function*, and *log-shells* developed in earlier papers by the author. We begin by fixing what we call *"initial Θ-data"*, which consists of an *elliptic curve* E_F over a *number field* F, and a *prime number* $l \geq 5$, as well as some other technical data satisfying certain technical properties. This data determines various *hyperbolic orbicurves* that are related via finite étale coverings to the

Conjetura abc



Conjetura abc

Según varios matemáticos (Además de Scholze y Stix) que han hecho un estudio sistemático de la prueba afirman que las problemáticas que la prueba presenta son las siguientes:

- La notación que Mochizuki adopta en dichos artículos hace que la lectura sea realmente difícil de seguir.
- En los artículos, Mochizuki hace amplia referencia a una gran cantidad de artículos suyos anteriores (también difíciles y largos) para justificar construcciones en la serie de abc
- Cadenas largas de definiciones que abarcan paginas enteras seguidas por teoremas iguales de largos cuyas demostraciones unicamente dicen "*Se sigue directo por la definición*" Sin que para dichos estudiosos lo sea.

Una falla fundamental

El "golpe final" de Scholze llega cuando se encuentra con el **corolario 3.12** del tercer articulo de Mochizuki el cual tiene una demostracion de 9 paginas. Cuando Scholze y Stix estudiaron a fondo la prueba captaron que había una parte del razonamiento que no estaba correcta, consultaron con otros matemáticos expertos del área y se dieron cuenta que tenían el mismo presentimiento. El problema de esto es que la prueba de abc se sigue directamente de este corolario 3.12

Según Mochizuki el problema de Scholze y Stix está en "la falta de tiempo suficiente para reflexionar en profundidad sobre las matemáticas objeto de debate" y además "una profunda sensación de incomodidad, o desconocimiento, de nuevas formas de pensar sobre objetos matemáticos familiares".

A día de hoy 2025 la prueba **no** ha sido aceptada por la comunidad matemática y la conjetura sigue abierta...

Objetividad vs Comunidad

"*¿Qué es una demostración matemática? Tendemos a pensar que es la revelación de una verdad eterna, pero quizá sea mejor entenderla como una construcción social.*" Jordana Cepelewicz
Matemáticos como Kevin Buzzard del Imperial College de Londres, respecto a esta polémica han destacado los formalizadores computacionales no solo como una solución, sino la necesidad misma de trasladar el conocimiento matemático a estos ecosistemas.
Liquid project:
`https://github.com/leanprover-community/lean-liquid`
Blog de Buzzard :
`https://xenaproject.wordpress.com/`

Lean

- **¿Qué es?**
  Lean es un asistente de demostraciones matemáticas y un lenguaje de programación funcional creado por Leonardo de Moura en 2013 auspiciado en ese entonces por Microsoft Research.
- **Caracteristicas principales**
  Lean está basado en el calculo de construcciones con tipos inductivos, lo que permita representar una amplia variedad de conocimiento matemático. En particular, Lean posee una libreria llamada **mathlib** donde se hospeda la gran mayoria de matemáticas que han sido formalizadas en este lenguaje. Además de esto Lean ha presentado un crecimiento significativo en los ultimos años, con matemáticos de alto renombre como Terry Tao aportando a su desarrollo.

El reto de los 100 Teoremas

Isabelle	92
HOL Light	89
Coq	79
Lean	79
Metamath	74
Mizar	69
nqthm/ACL2	47
ProofPower	43
PVS	26
Megalodon	12
Naproche	10
NuPRL/MetaPRL	8

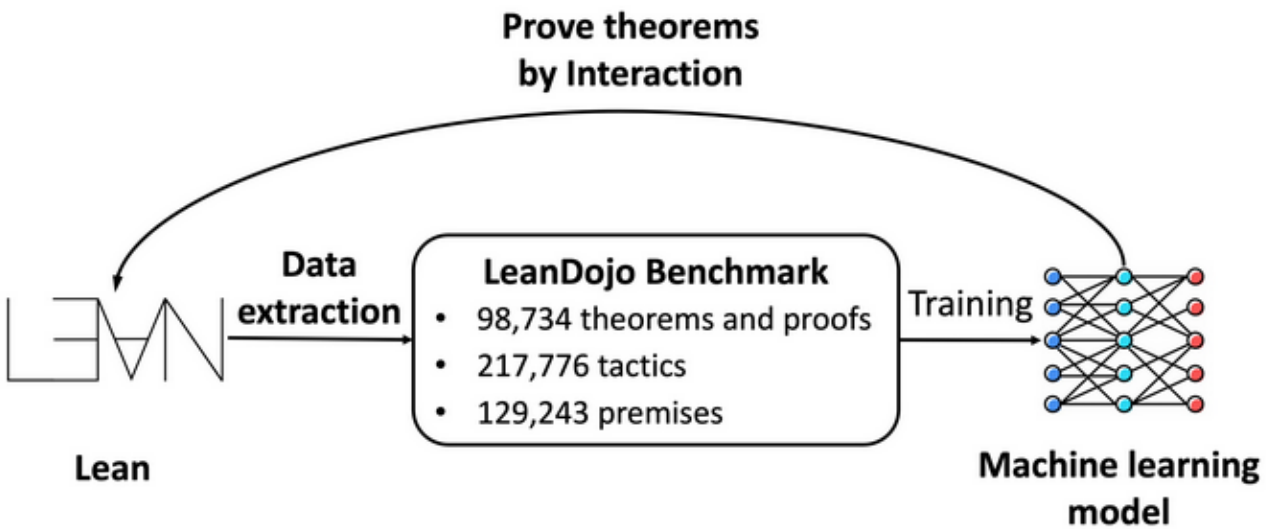
Recuperado de: `https://www.cs.ru.nl/~freek/100/`

Ejemplo de implementación

A continuación vamos a mostrar un ejemplo de implementación de código en Lean. Vamos a definir el objeto de polinomios con coeficientes reales aprovechándonos de los tipos inductivos en Lean.

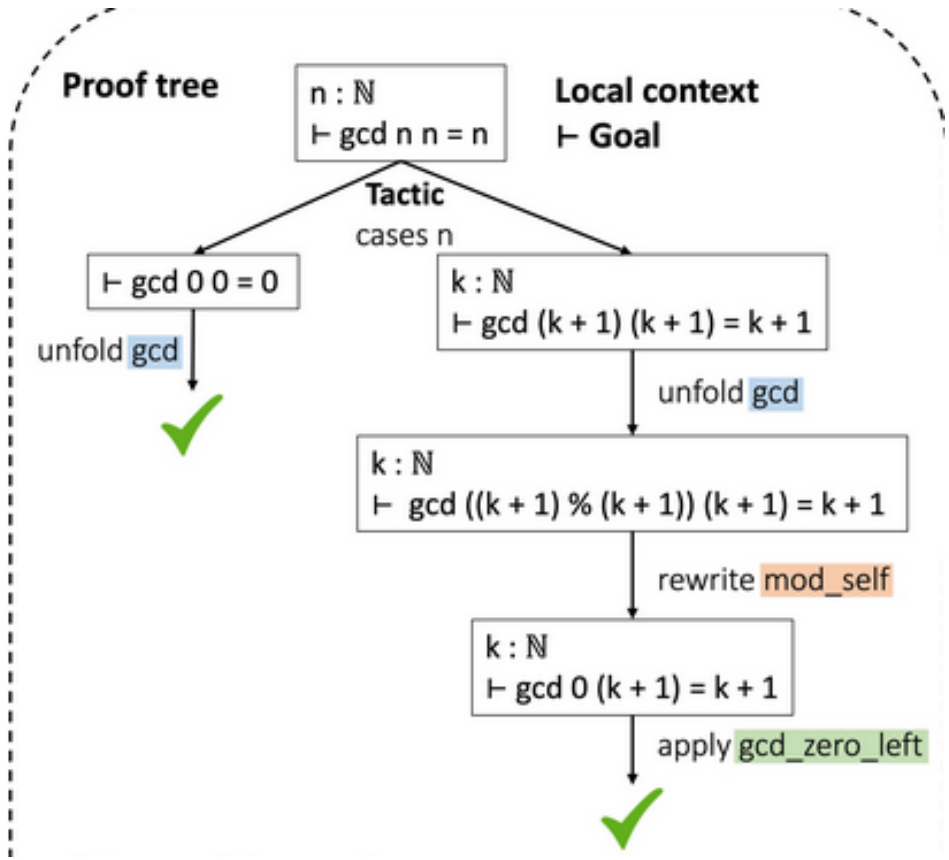
LeanDojo

LeanDojo es un entorno computacional en el que algunos modelos de inteligencia artifical de Deep Learning y modelos extensos del lenguaje han resultado muy prometedores para automatizar el proceso de formalización en Lean.



LeanDojo

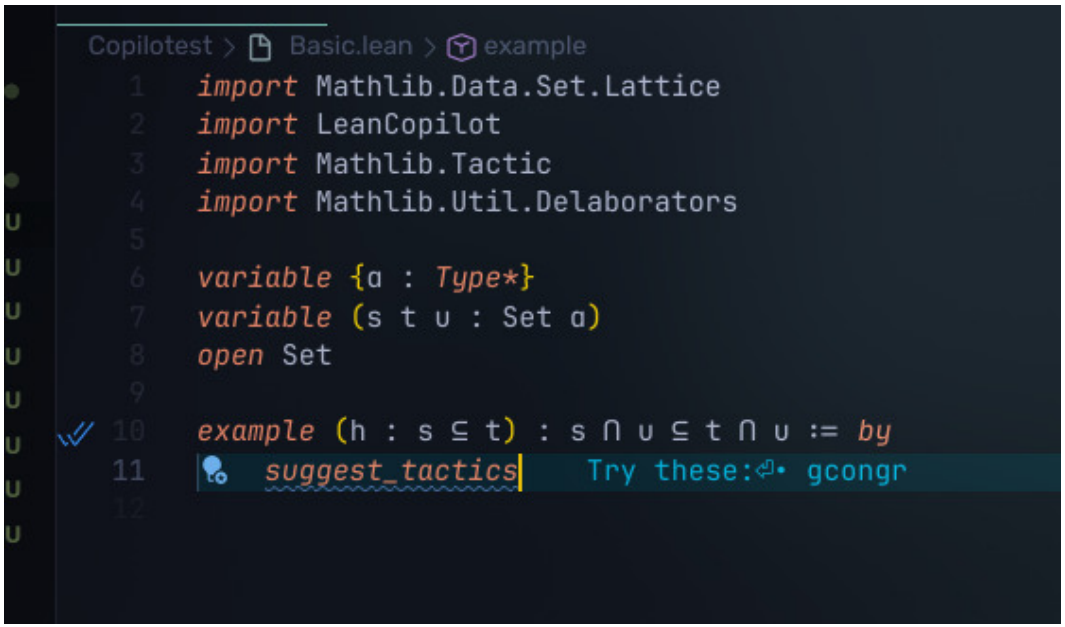
Por ejemplo, una de las caracteristicas de los modelos extensos del lenguaje (LLms) es que utilizan el concepto de **encoders** los cuales son una forma de codificar informacion a traves de vectores numéricos para entrenar modelos.



LeanCopilot

LeanCopilot es un framework de codigo abierto en el cual se utilizan diferentes modelos de IA para automatizar y optimizar el proceso de formalización en Lean

1. **SuggestTactics**
   Esta herramienta que implementa LeanCopilot sugiere tacticas que ayudan o incluso demuestran teoremas en Lean, su enlace con la libreria principal Mathlib la vuelve una herramienta muy robusta



LeanCopilot

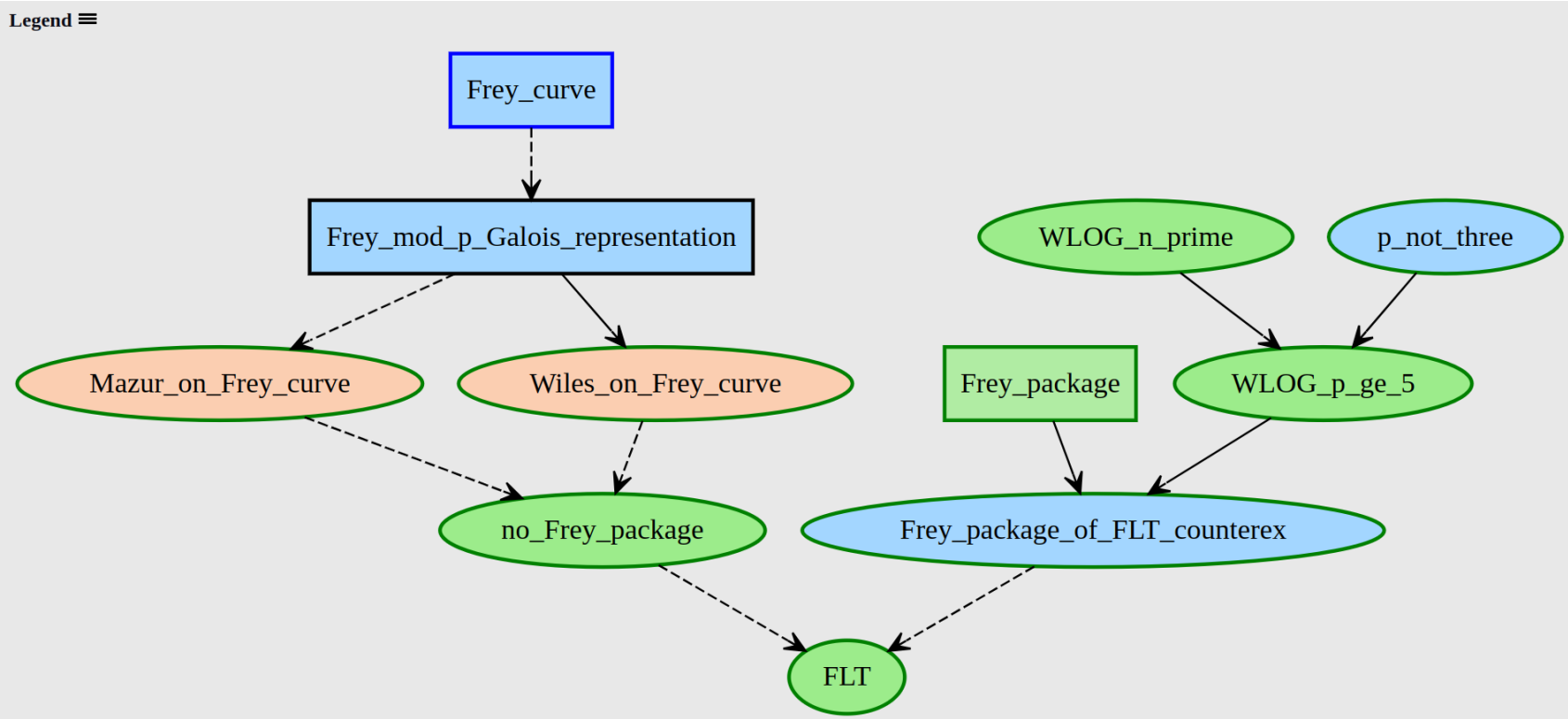
1. **SearchProof**
   Esta funcionalidad de Copilot permite identificar si al momento de construir la prueba de un teorema en Lean, dicho teorema ya existe dentro de su biblioteca estándar o de Mathlib, además también puede contribuir a buscar nuevas demostraciones.

Direcciones a futuro

Para darle continuación al proyecto nos proponemos los siguientes objetivos:

- Ahondar en los fundamentos teóricos de estos modelos de inteligencia artificial con la finalidad de poder expandir y potenciar su uso en Lean.
- Profundizar en el tema de optimización de estos modelos ya que el tiempo de ejecución es muy alto, los entornos para correr son difíciles de configurar etc.
- Indagar como la teoría de tipos puede servir para el aprovechamiento de estos modelos i.e ¿Tiene alguna ventaja los tipos dependientes o los tipos inductivos para trabajar con Llms?

Ultimo Teorema de Fermat



Recuperado de:
`https://xenaproject.wordpress.com/2024/01/20/lean-in-2024/`

Conclusiones

- Situaciones como las que acabamos de ver en la conjetura abc, son un fuerte indicio de que la formalización se va a convertir en un indispensable de todo matemático.
- La intersección entre los fundamentos teóricos de los modelos de inteligencia artificial(Arquitecturas, matemáticas base, etc.) y su aplicación a Lean abre puertas a la colaboración entre personas del sector computacional y del área de matemáticas, con posibles beneficios en ambas direcciones.
- Resulta de gran interes la manera en como se despliega el proceso de poder demostrar teoremas muy sofisticados en Lean como UTF, además, como esto resulta en una motivación para formalizar otros aspectos teóricos relevantes en la prueba de estos teoremas.

Bibliografía

E. Klarreich. "Titans of Mathematics Clash Over Epic Proof of ABC Conjecture." *Quanta Magazine*, 20 septiembre 2018. https://www.quantamagazine.org/titans-of-mathematics-clash-over-epic-proof-of-abc-conjec

P. Scholze , J. Stix. *Why abc is still a conjecture*, 2018. https://ncatlab.org/nlab/files/why_abc_is_still_a_conjecture.pdf.

Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar, "LeanDojo: Theorem Proving with Retrieval-Augmented Language Models," in *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2023.

Bibliografía

- Jeremy Avigad, Leonardo de Moura, y Floris van Doorn. *Mathematics in Lean*. Lean Prover Community, 2024.
- The Lean Community. *Theorem Proving in Lean 4*. Lean Prover Community, 2023.
- Peiyang Song, Kaiyu Yang, y Anima Anandkumar. *Lean copilot: Large language models as copilots for theorem proving in Lean*. arXiv preprint arXiv:2404.12534, 2024.