Programs and Proofs

Anthony Wang (xy)

January 16, 2026

Cool Lean projects

- Raytracer
- Webring generator
- HouLean
- Video player
- SciLean
- Functorio
- Rupert
- Equational theories
- Mathlib
- Analysis textbook
- Erdős 707
- LeanTeX

History of formalized math

- 1910: Principia Mathematica

***54·43.** $\vdash :. \alpha, \beta \in 1 . \supset : \alpha \cap \beta = \Lambda . \equiv . \alpha \cup \beta \in 2$

Dem.

$\vdash . *54{\cdot}26 . \supset \vdash :. \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . x \neq y .$

$[*51{\cdot}231] \qquad \equiv . \iota`x \cap \iota`y = \Lambda .$

$[*13{\cdot}12] \qquad \equiv . \alpha \cap \beta = \Lambda \qquad (1)$

$\vdash . (1) . *11{\cdot}11{\cdot}35 . \supset$

$\vdash :. (\exists x, y) . \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . \alpha \cap \beta = \Lambda \qquad (2)$

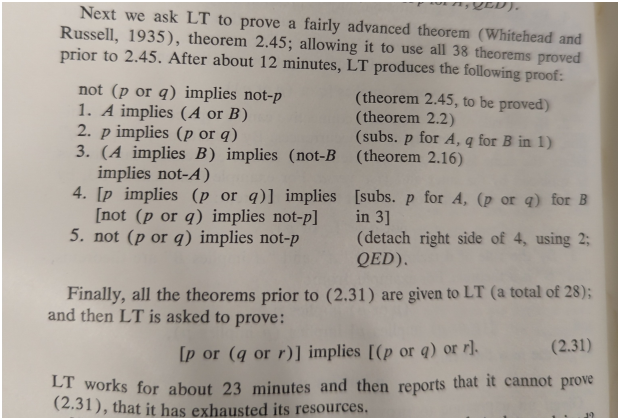
$\vdash . (2) . *11{\cdot}54 . *52{\cdot}1 . \supset \vdash .$ Prop

From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

- 1931: Gödel's incompleteness theorems

History (cont.)

- 1936: Entscheidungsproblem proven undecidable
- 1956: Logic Theorist ("first AI program")



Next we ask LT to prove a fairly advanced theorem (Whitehead and Russell, 1935), theorem 2.45; allowing it to use all 38 theorems proved prior to 2.45. After about 12 minutes, LT produces the following proof:

not (p or q) implies not-p	(theorem 2.45, to be proved)
1. A implies (A or B)	(theorem 2.2)
2. p implies (p or q)	(subs. p for A, q for B in 1)
3. (A implies B) implies (not-B implies not-A)	(theorem 2.16)
4. [p implies (p or q)] implies [not (p or q) implies not-p]	[subs. p for A, (p or q) for B in 3]
5. not (p or q) implies not-p	(detach right side of 4, using 2; *QED*).

Finally, all the theorems prior to (2.31) are given to LT (a total of 28); and then LT is asked to prove:

$$[p \text{ or } (q \text{ or } r)] \text{ implies } [(p \text{ or } q) \text{ or } r]. \qquad (2.31)$$

LT works for about 23 minutes and then reports that it cannot prove (2.31), that it has exhausted its resources.

History (cont.)

- 1976: Four color theorem proved using brute force (verified in Coq in 2005)
- 1989: Coq (Rocq) released

ITPs vs ATPs

- Two main paradigms
- ITP = Interactive theorem prover, uses tactics, ex: Rocq, Lean
- ATP = Automated ..., uses SMT, ex: Dafny

ITP foundations

- Set theory (Mizar, Metamath)
- Simple type theory (Isabelle/HOL)
- Dependent type theory (Lean, Rocq, Agda, Idris)

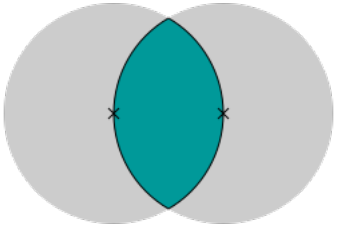
Lean bio

- 2013: Created by Leo de Moura at Microsoft, previously created Z3
- 2023: Lean 4 released, rewritten in Lean (except type checker)
- Not named after the drug

Why Lean?

- Most popular proof assistant
- Mathlib
- Automation (grind, etc)
- AI: Harmonic's Aristotle, AlphaProof
- Fun!

Challenges



- "Invisible math"
- Terry Tao: Writing Lean is 10x more time than conventional proofs
- Not many programming libraries
- Hard to learn