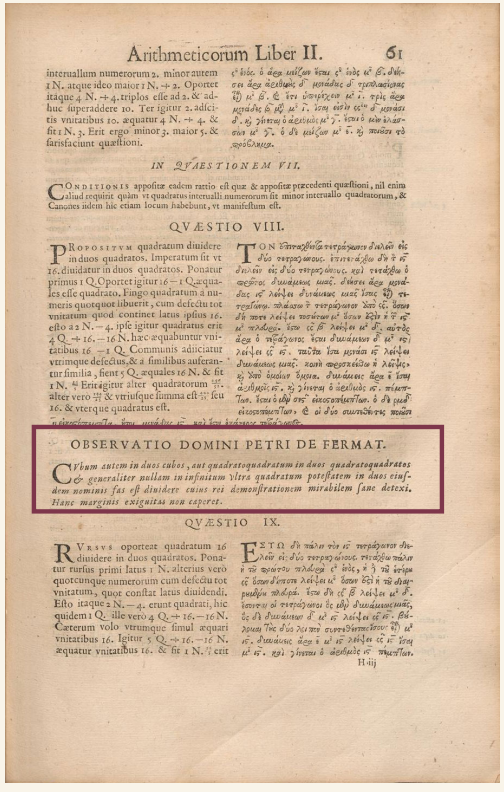
What is...deautoformalization?

Or...translating AI math back to human math

I first heard this idea at Kevin Buzzard's talk at SMRI on Thursday, 12 November 2026.

What can AI formalize in 2026?



Arithmeticorum Liber II. 61

interuallum numerorum 2. minor autem 1 N. atque ideo maior 1 N. + 2. Oportet itaque 4 N. + 4. triplos esse ad 2. & adhuc superaddere 10. Ter igitur 2. adsciti vnitatibus 10. æquatur 4 N. + 4. & fit 1 N. 3. Erit ergo minor 3. maior 5. & satisfaciunt quæstioni.

IN QVÆSTIONEM VII.

CONDITIONIS appositæ eadem ratio est quæ & appositæ præcedenti quæstioni, nil enim aliud requirit quàm vt quadratus interualli numerorum sit minor interuallo quadratorum, & Canones idem hic etiam locum habebunt, vt manifestum est.

QVÆSTIO VIII.

PROPOSITVM quadratum dividere in duos quadratos. Imperatum sit vt 16. dividatur in duos quadratos. Ponatur primus 1 Q. Oportet igitur 16 − 1 Q. æquales esse quadrato. Fingo quadratum à numeris quotquot libuerit, cum defectu tot vnitatum quod continet latus ipsius 16. esto 2 N. − 4. ipse igitur quadratus erit 4 Q. + 16. − 16 N. hæc æquabuntur vnitatibus 16 − 1 Q. Communis adiiciatur vtrimque defectus, & a similibus auferantur similia, fient 5 Q. æquales 16 N. & fit 1 N. $\frac{4}{5}$. Erit igitur alter quadratorum $\frac{256}{25}$. alter vero $\frac{144}{25}$ & vtriusque summa est $\frac{400}{25}$ seu 16. & vterque quadratus est.

OBSERVATIO DOMINI PETRI DE FERMAT.

CVbum autem in duos cubos, aut quadratoquadratum in duos quadratoquadratos & generaliter nullam in infinitum vltrà quadratum potestatem in duos eiusdem nominis fas est dividere cuius rei demonstrationem mirabilem sane detexi. Hanc marginis exiguitas non caperet.

QVÆSTIO IX.

RVRSVS oportet quadratum 16 dividere in duos quadratos. Ponatur rursus primi latus 1 N. alterius verò quotcunque numerorum cum defectu tot vnitatum, quot constat latus dividendi. Esto itaque 2 N. − 4. erunt quadrati, hic quidem 1 Q. ille verò 4 Q. + 16. − 16 N. Cæterum volo vtrumque simul æquari vnitatibus 16. Igitur 5 Q. + 16. − 16 N. æquatur vnitatibus 16. & fit 1 N. $\frac{4}{5}$ erit

H iij

1670 **A note printed in the Arithmetica**

$$a^n + b^n \neq c^n$$

Positive integers a, b, c and integers $n > 2$

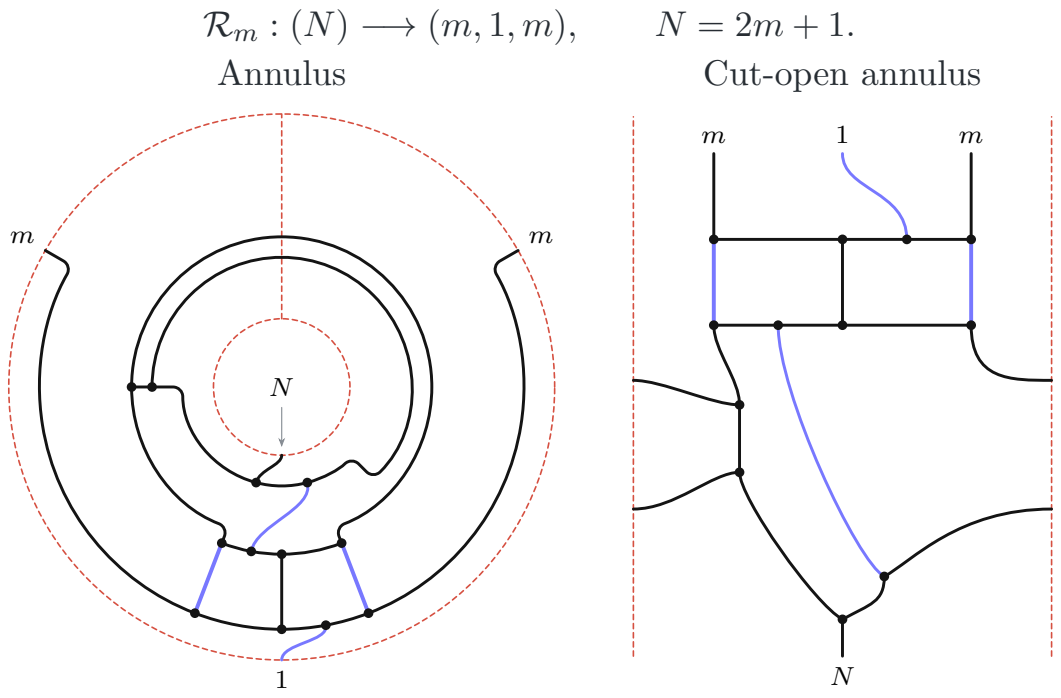
2026 **The conclusion in Lean**

```
a ^ n + b ^ n ≠ c ^ n
```

Arithmetica, p. 61 · Wikimedia Commons, public domain

- Fermat No two positive nth powers add up to a third when $n > 2$
- September 2026 Anthropic released an AI-generated Lean proof
- How? With human guidance, the agents followed a known proof, filling in lemmas and checking them in Lean

You can formalize your own papers in Lean in 2026



Goertzen, Lacabanne, Tubbenhauer and Vaz, *Annular webs and Guralnick's conjecture*, Fig. 4

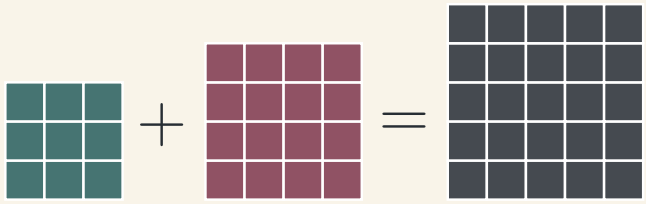
- Your paper Work theorem by theorem. Check that the Lean definitions match yours
- Imported results List the cited theorems the proof is allowed to assume
- Reward hacking A file can compile because the AI assumed the very lemma you asked it to prove

How to make Fermat's Last Theorem suspiciously easy

The requested claim

$$a^n + b^n \neq c^n$$

All positive a, b, c, $\quad n > 2$



For squares, $3^2 + 4^2 = 5^2$

An easier translation

$$a^n + b^n \neq a^n$$

The extra hypothesis is $c = a$



$3^2 + 4^2$ is plainly bigger than 3^2

- A translation slip Suppose the AI quietly adds the hypothesis $c = a$
- A correct proof Now $b^n > 0$ settles the question, even for $n = 2$
- Another pitfall The file can also compile with a missing proof marked `sorry`

How do you double-check a Lean proof?

Challenge.lean

Fermat's Last Theorem

All positive a, b, c, every $n > 2$. *Question fixed in advance*

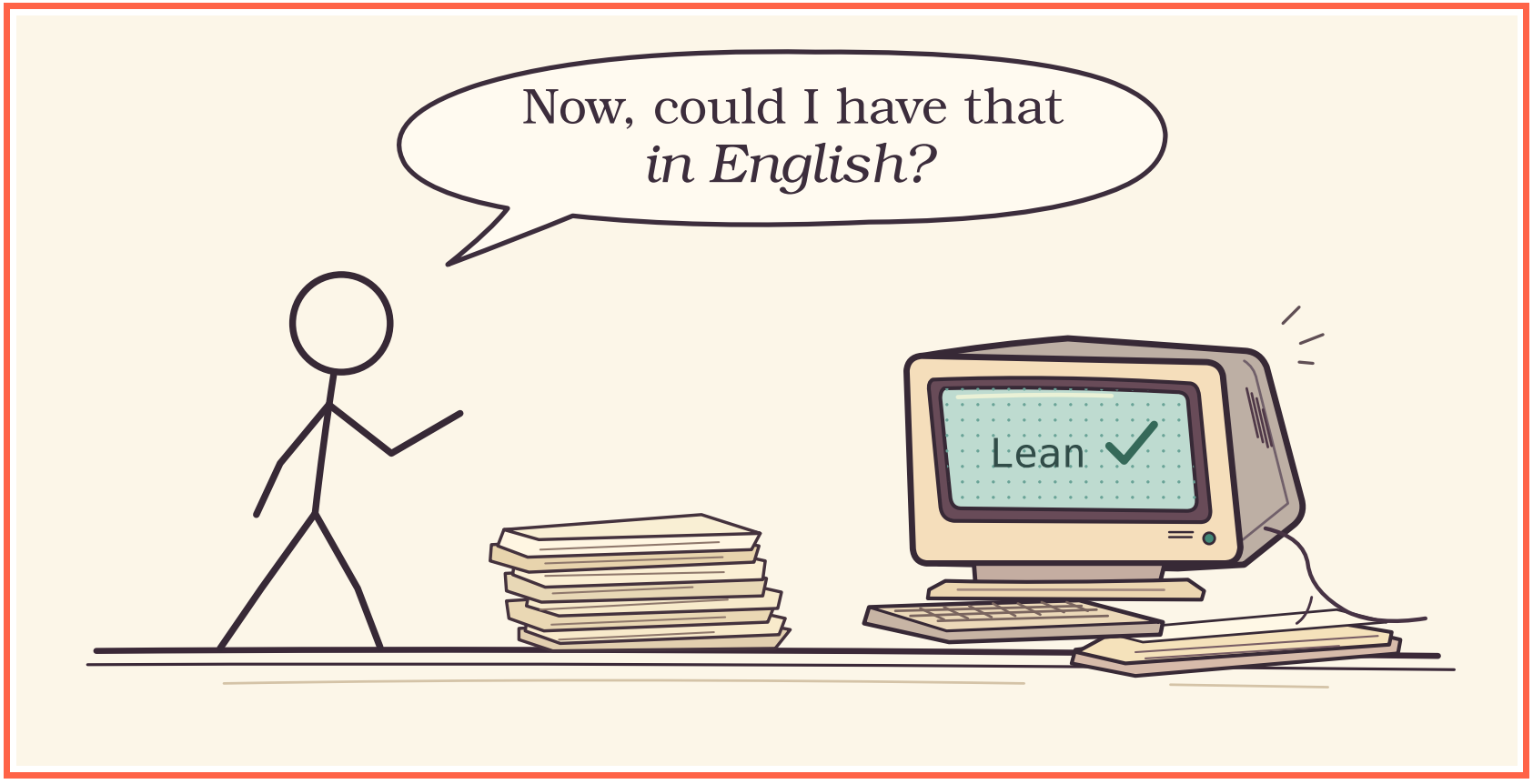
```
Comparator      Your solution is okay!

Nanoda          Checked 1052234 declarations with no errors
```

Published verification logs · Anthropic FLT release, 2026

- Fix the question Read the Lean statement and its definitions against the original
- Comparator Checks that the theorem matches and uses only the allowed axioms
- Replay the proof The Fermat team ran the proof through Nanoda, an independent checker written in Rust

A half-serious case for deautoformalization



- The AI-written PDF Hard to read, and small mistakes are easy to miss
- The AI-written Lean That can go wrong too, but now we can check it
- Deautoformalize! Give the AI the checked Lean proof. Ask it to write the mathematics you would actually like to read

Thank you for your attention!

Next time: Back to the regular programme!