The Science and Engineering of Autoformalizing Mathematics

A Case Study in Euclidean Geometry

Xujie Si

Assistant Professor, University of Toronto
Vector Institute
Mila – Quebec AI Institute

IJCAI workshop on
User-Aligned Assessment of Adaptive AI Systems
Aug 18, 2025 Montreal

AI Revolution for X = {Mathematics, ... }

AI|MO

ARTIFICIAL INTELLIGENCE MATHEMATICAL OLYMPIAD

$10mn AI Mathematical Olympiad Prize Launches

Advanced version of Gemini with Deep Think officially achieves gold-medal standard at the International Mathematical Olympiad

21 JULY 2025

Thang Luong and Edward Lockhart

DeepMind

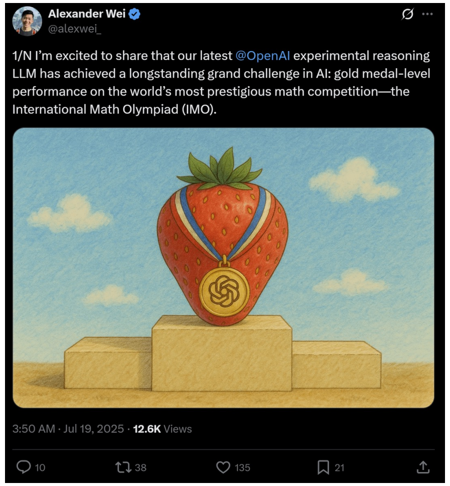
Mathematical Superintelligence

Harmonic

FrontierMath

A math benchmark testing the limits of AI

EPOCH AI

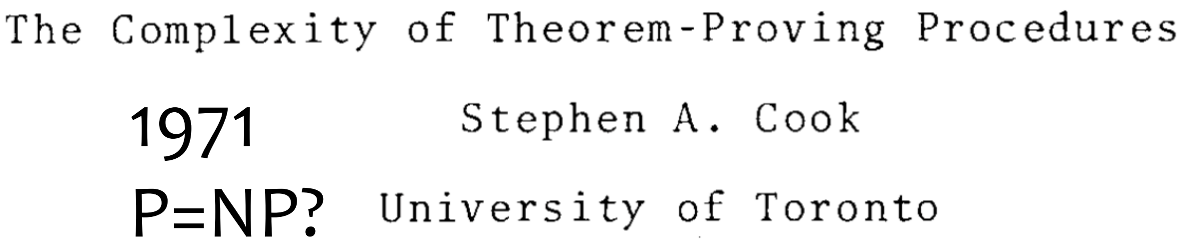


Alexander Wei
@alexwei_

1/N I'm excited to share that our latest @OpenAI experimental reasoning LLM has achieved a longstanding grand challenge in AI: gold medal-level performance on the world's most prestigious math competition—the International Math Olympiad (IMO).

3:50 AM · Jul 19, 2025 · **12.6K** Views

AI for Math isn’t that new …

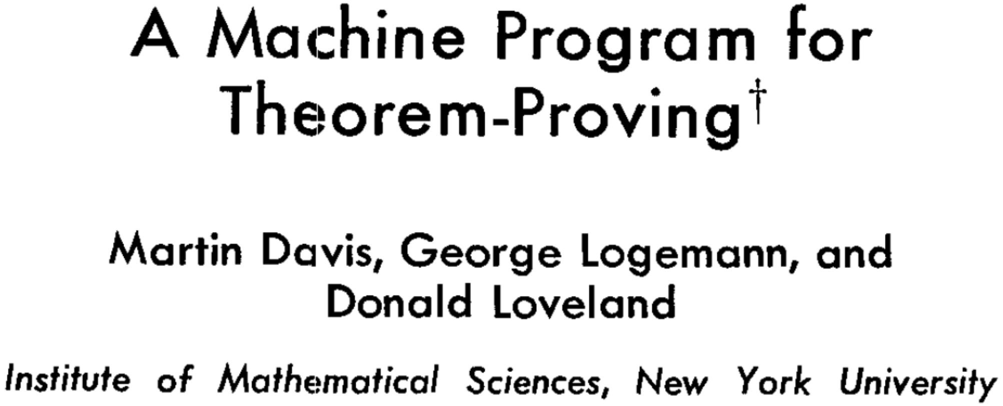

The Complexity of Theorem-Proving Procedures

Stephen A. Cook

University of Toronto

1971
P=NP?

1960s, DPLL


A Machine Program for Theorem-Proving†

Martin Davis, George Logemann, and Donald Loveland

Institute of Mathematical Sciences, New York University

Logic for Computable Functions (LCF)

1972
Robin Milner

AUTOMATH
1972 - 1977

N. G. de Bruijn

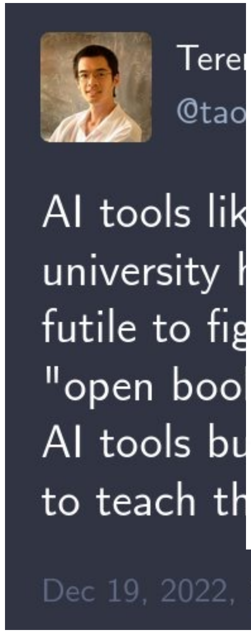
Formalizing Context: Expanded notes

McCarthy et al, 1997

Why is it so different this time?

Ch

Released on
Nov 30, 2022



Tere
@tao

AI tools lik
university l
futile to fig
"open boo
AI tools bu
to teach th

Dec 19, 2022, 19:21 · Web · 143 · 214

Microsoft
Research

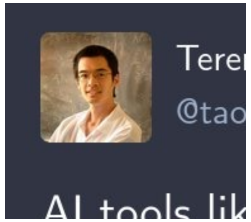
Sparks of Artificial General Intelligence:
Early experiments with GPT-4

Sébastien Bubeck Varun Chandrasekaran Ronen Eldan Johannes Gehrke
Eric Horvitz Ece Kamar Peter Lee Yin Tat Lee Yuanzhi Li Scott Lundberg
Harsha Nori Hamid Palangi Marco Tulio Ribeiro Yi Zhang

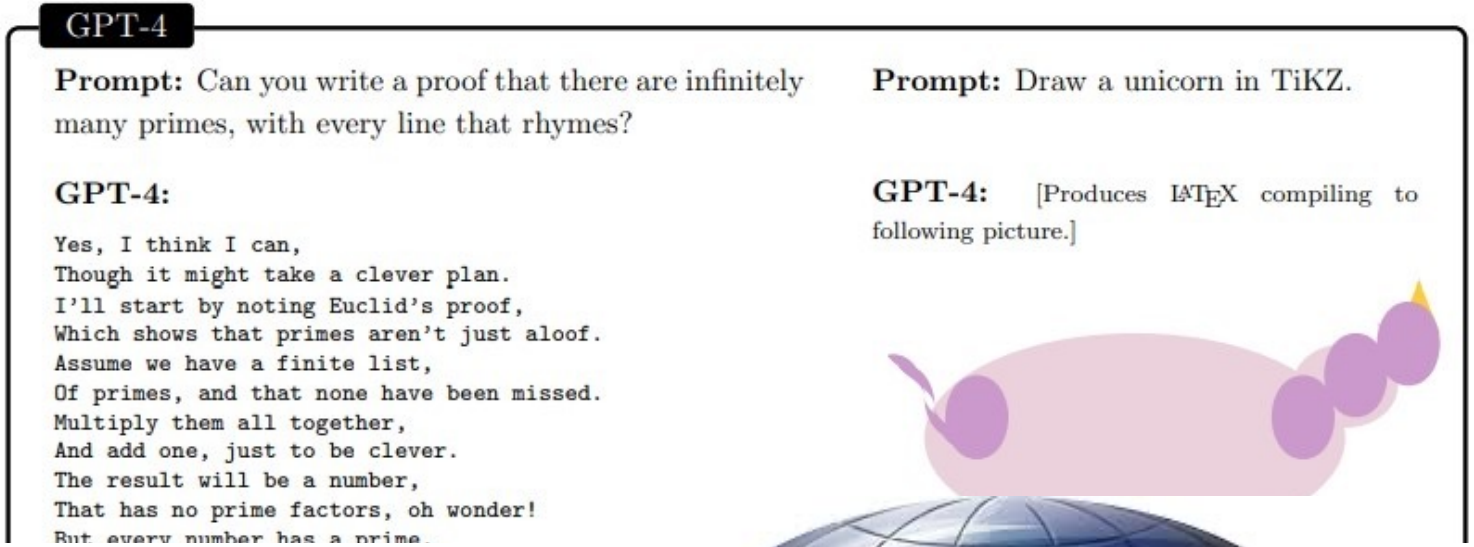
Microsoft Research

Why is it so different this time?

Ch



Released on
Nov 30, 2022


GPT-4

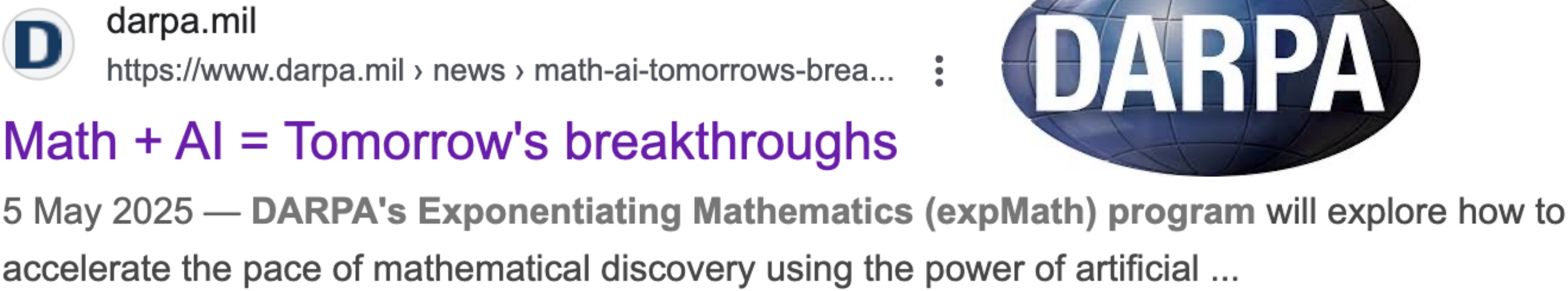
Prompt: Can you write a proof that there are infinitely many primes, with every line that rhymes?

GPT-4:

```
Yes, I think I can,
Though it might take a clever plan.
I'll start by noting Euclid's proof,
Which shows that primes aren't just aloof.
Assume we have a finite list,
Of primes, and that none have been missed.
Multiply them all together,
And add one, just to be clever.
The result will be a number,
That has no prime factors, oh wonder!
But every number has a prime.
```

Prompt: Draw a unicorn in TiKZ.

GPT-4: [Produces LaTeX compiling to following picture.]


darpa.mil
https://www.darpa.mil › news › math-ai-tomorrows-brea...

Math + AI = Tomorrow's breakthroughs

5 May 2025 — **DARPA's Exponentiating Mathematics (expMath) program** will explore how to accelerate the pace of mathematical discovery using the power of artificial ...

DARPA

Autoformalization: Challenges & Promises

①

Natural Language (NL)
Descriptions
(Human Comprehensible)

grounding

③

②

Formal Language (FL)
Representations
(Machine Checkable)

Informal math theorems & proofs

Formalized theorems & proofs

Fermat's
Last Theorem

$$\forall n \in \mathbb{Z},\, n > 2$$
$$\neg\exists a, b, c \in \mathbb{Z}^{+} \text{ such that } a^n + b^n = c^n$$

Law in natural languages

Law in theorem prover??

Autoformalization: Challenges & Promises

Natural Langu
Descriptions
(Human Com

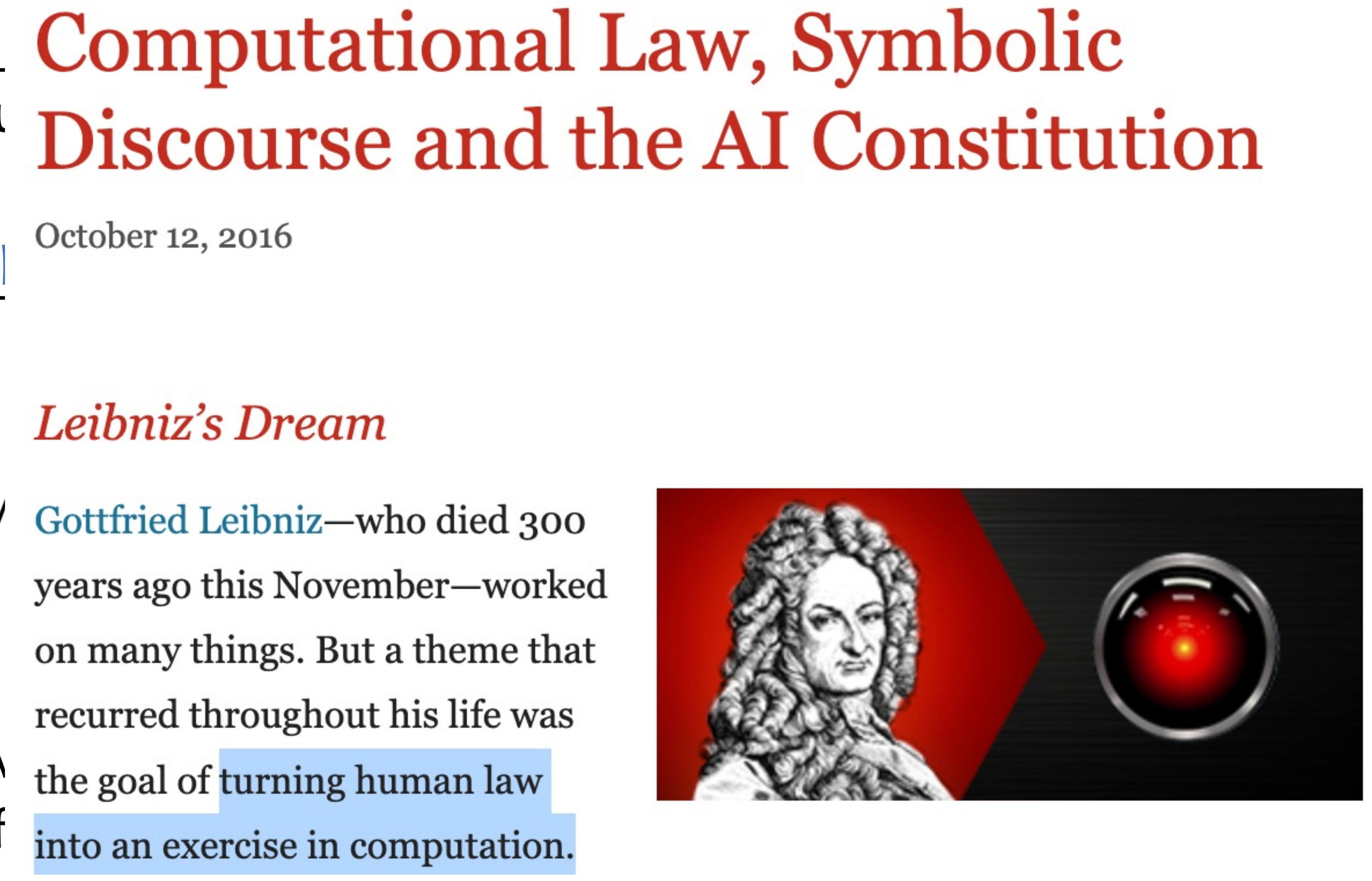
Informal math

Fern
Last

In M

Law in na

\
f



Computational Law, Symbolic Discourse and the AI Constitution

October 12, 2016

Leibniz's Dream

Gottfried Leibniz—who died 300 years ago this November—worked on many things. But a theme that recurred throughout his life was the goal of turning human law into an exercise in computation.

Source: https://writings.stephenwolfram.com/2016/10/computational-law-symbolic-discourse-and-the-ai-constitution/

②

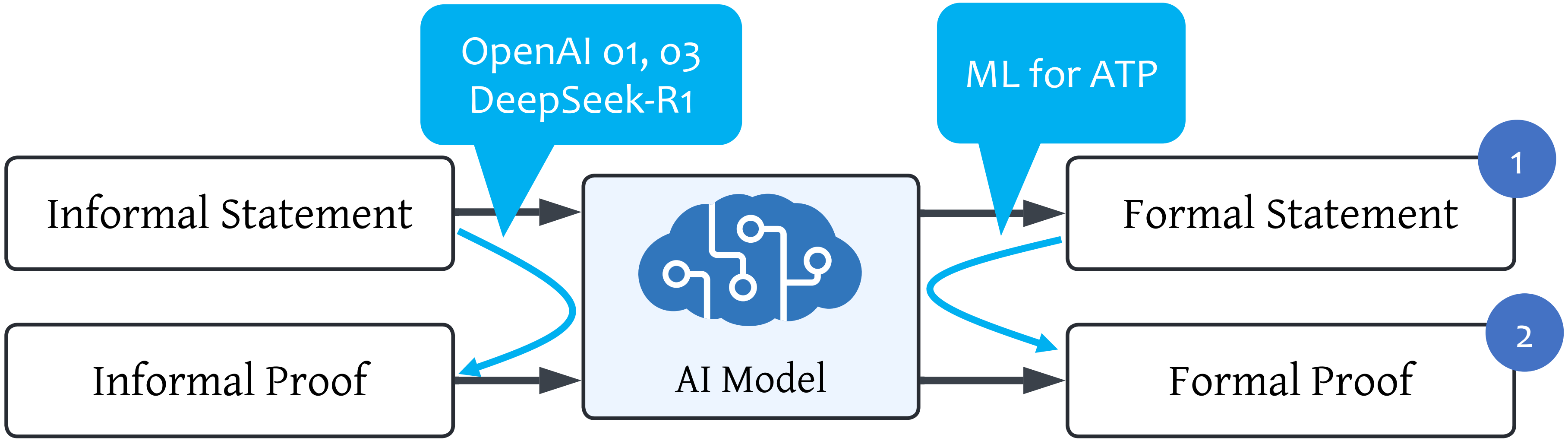
ge (FL)
ns
kable)

ems & proofs

c^n

prover??

Autoformalizing Mathematics



OpenAI o1, o3
DeepSeek-R1

ML for ATP

Informal Statement

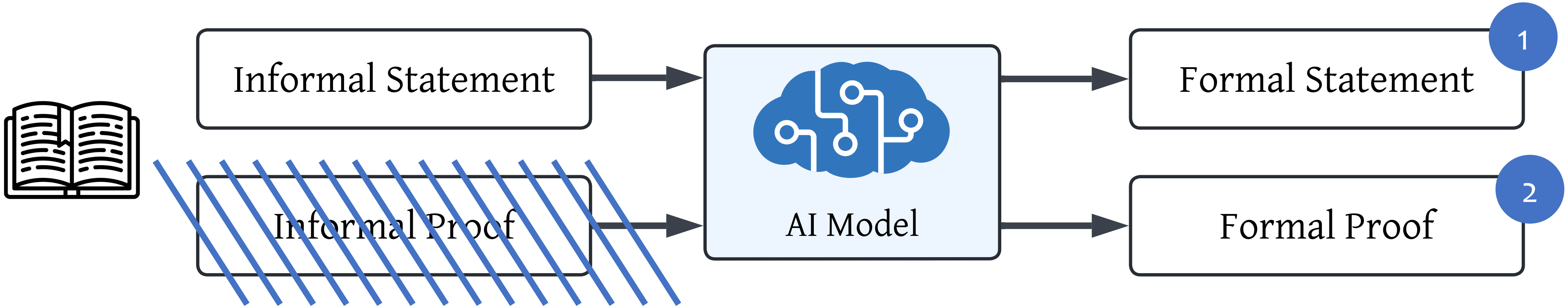
Informal Proof

AI Model

Formal Statement

Formal Proof

Autoformalizing Mathematics



Informal Statement

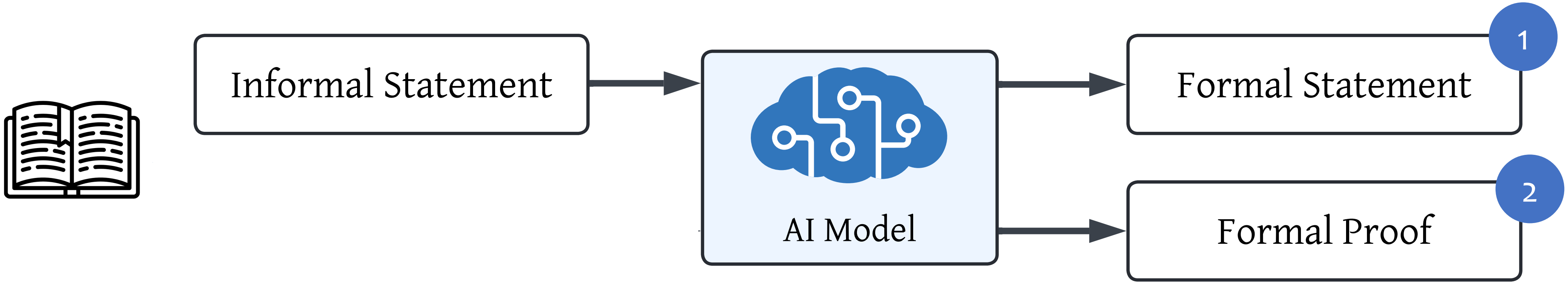
~~Informal Proof~~

AI Model

Formal Statement

Formal Proof

Autoformalizing Mathematics



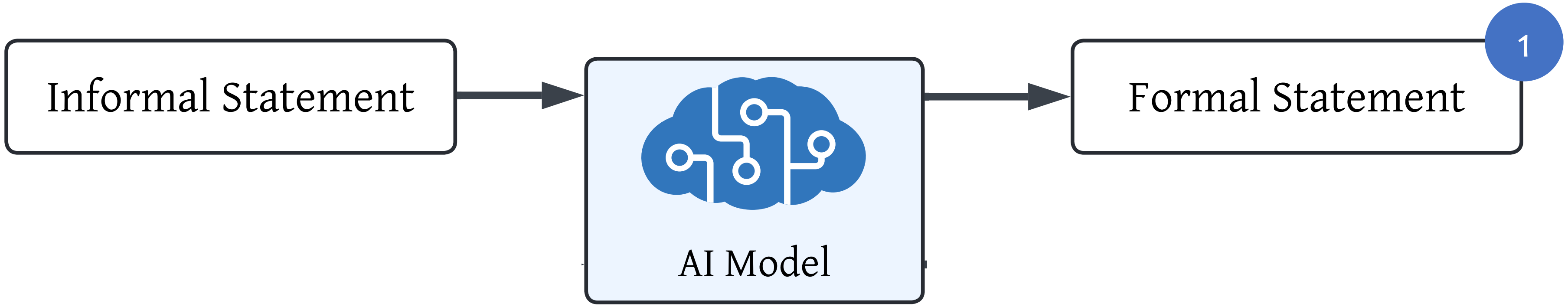
Informal Statement

AI Model

Formal Statement

Formal Proof

Autoformalizing Mathematics



Informal Statement

AI Model

Formal Statement

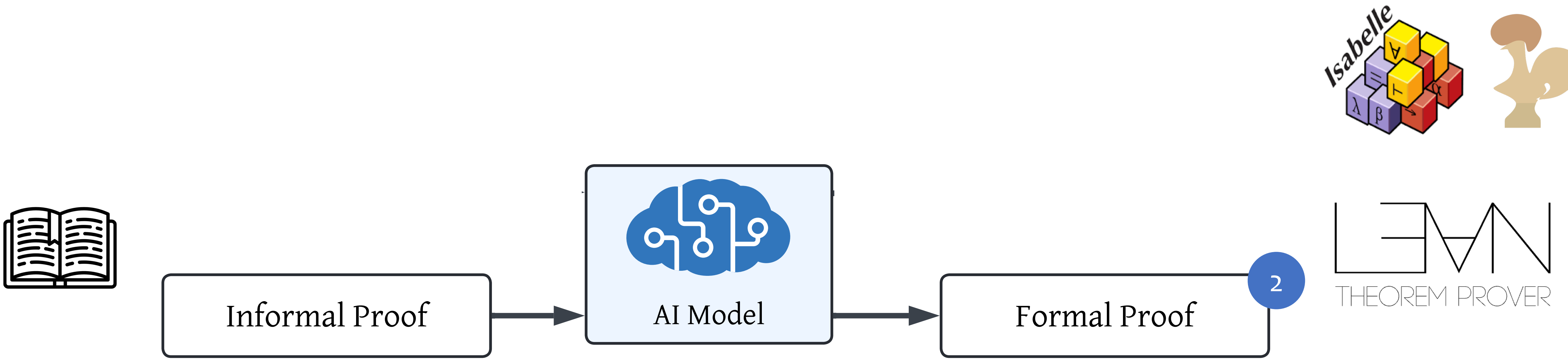
1

Isabelle

LEAN

THEOREM PROVER

Autoformalizing Mathematics



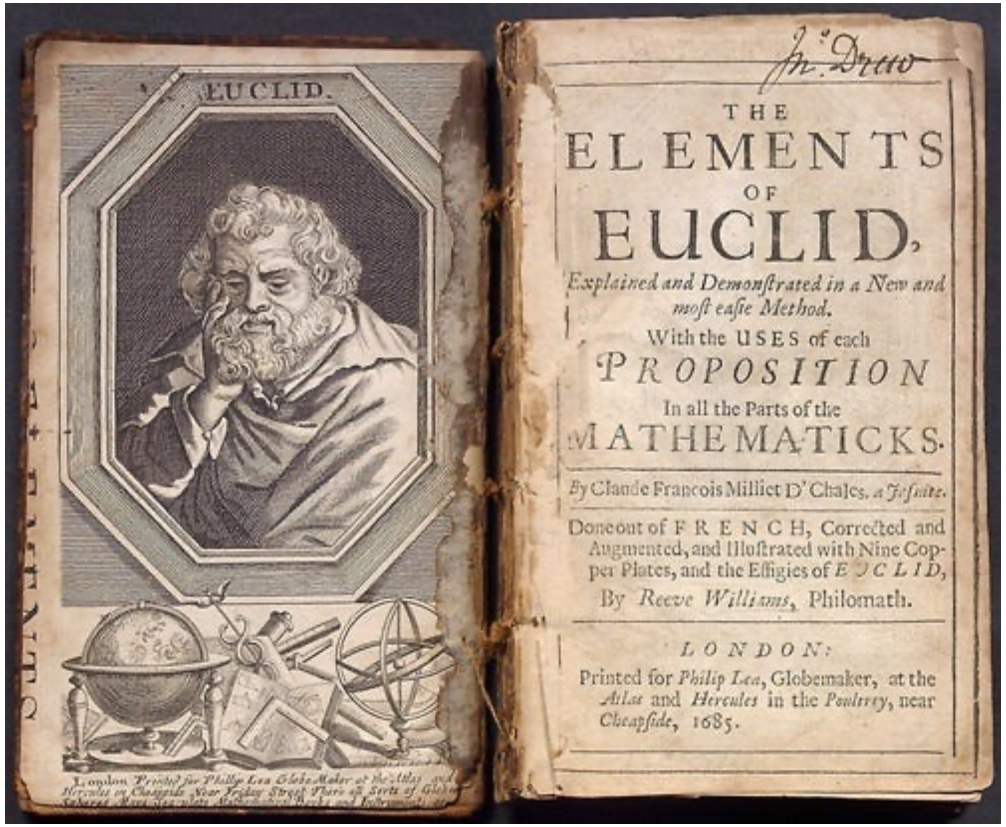
Informal Proof

AI Model

Formal Proof

2

Autoformalizing Euclidean Geometry

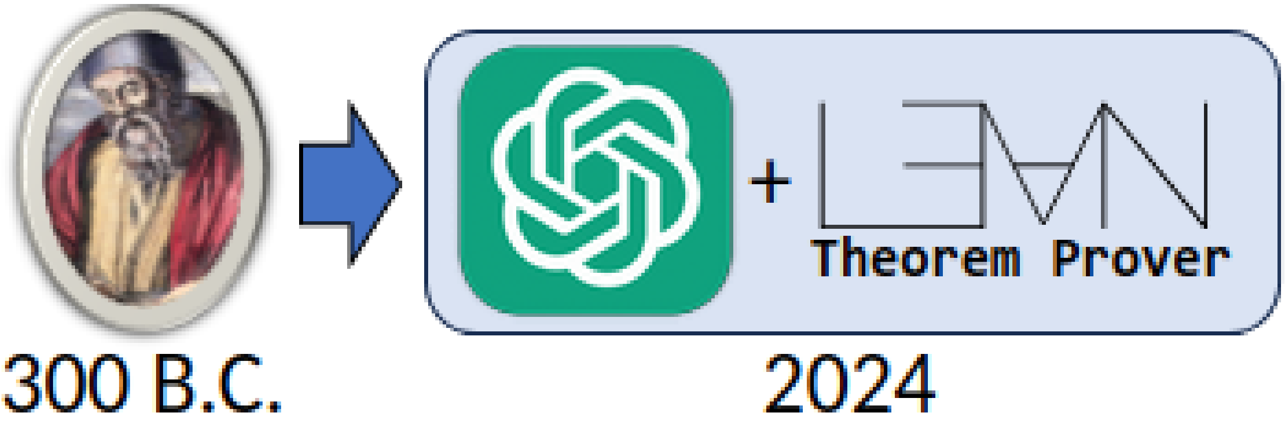

EUCLID.
ELEMENTS

Jn: Drew

THE
ELEMENTS
OF
EUCLID,
Explained and Demonſtrated in a New and moſt eaſie Method.
With the USES of each
PROPOSITION
In all the Parts of the
MATHEMATICKS.
By Claude Francois Milliet D'Chales, *a Jeſuite.*
Done out of FRENCH, Corrected and Augmented, and Illuſtrated with Nine Copper Plates, and the Effigies of *EUCLID*,
By *Reeve Williams*, Philomath.

LONDON:
Printed for *Philip Lea*, Globemaker, at the *Atlas* and *Hercules* in the *Poultrey*, near *Cheapſide*, 1685.

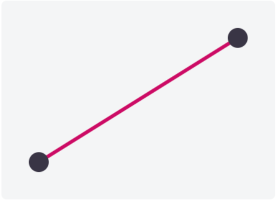
The Elements (Book I)



300 B.C.

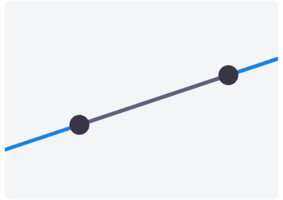
2024

Autoformalizing Euclidean Geometry



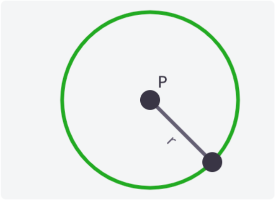
First Axiom

You can join any two points using exactly one straight line segment.



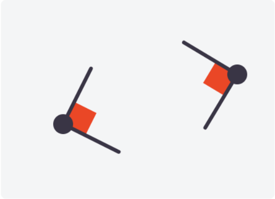
Second Axiom

You can extend any line segment to an infinitely long line.



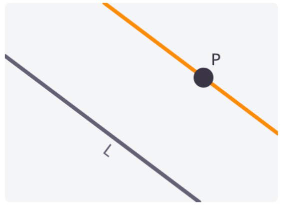
Third Axiom

Given a point *P* and a distance *r*, you can draw a circle with centre *P* and radius *r*.



Fourth Axiom

Any two right angles are congruent.



Fifth Axiom

Given a line *L* and a point *P* not on *L*, there is exactly one line through *P* that is parallel to *L*.

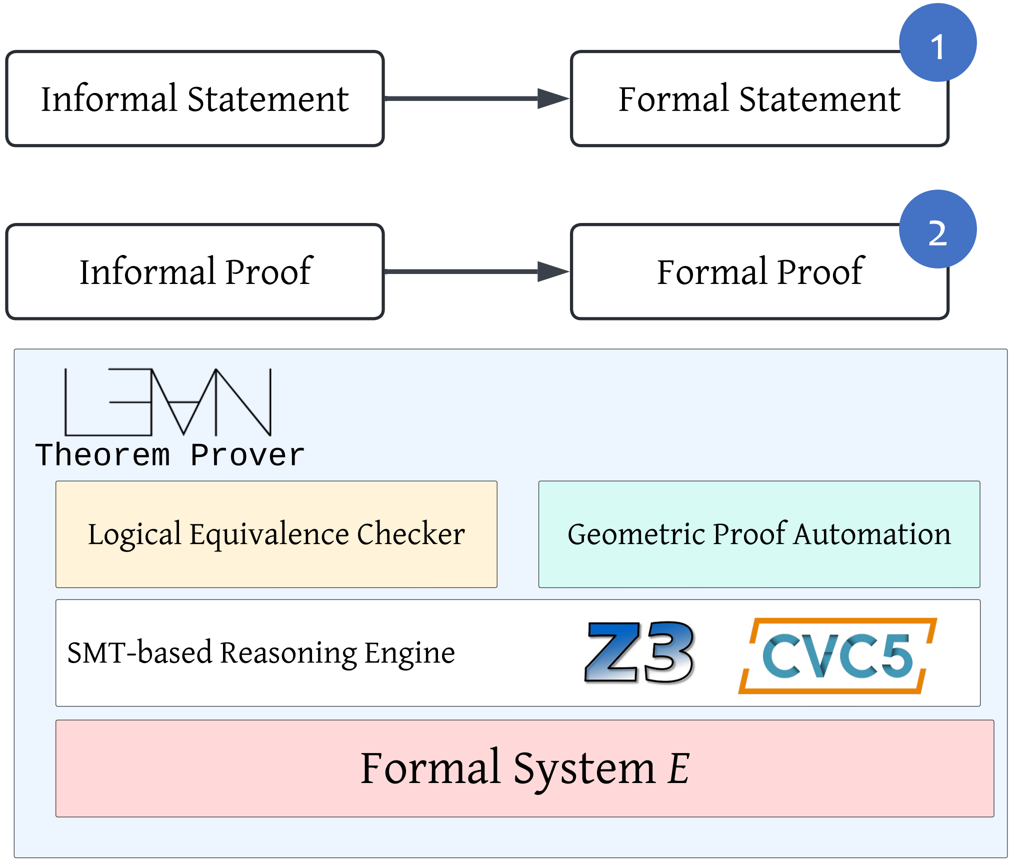
The Elements (Book I)

+ LEAN
Theorem Prover

300 B.C.

2024

The Science & Engineering behind



Informal Statement → Formal Statement ①

Informal Proof → Formal Proof ②

LEAN Theorem Prover

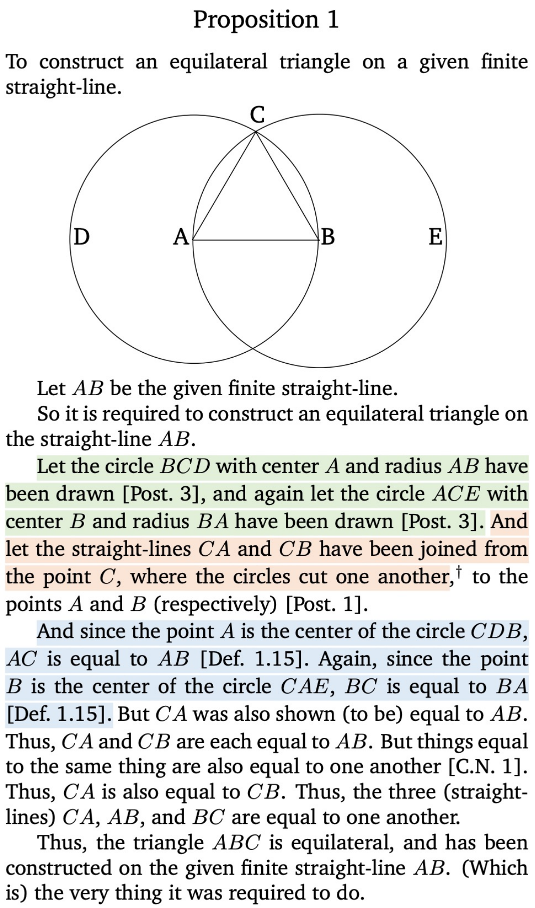
Logical Equivalence Checker

Geometric Proof Automation

SMT-based Reasoning Engine Z3 CVC5

Formal System *E*

- Axiomatic Systems
  - Pasch (1882), Peano (1889), Hilbert (1899)
  - Tarski (1959)
  - Avigad et al., System E (2009)
- Formal Theorem Language
  - Embed System E in Lean 4
  - Design equivalence checkers
- Formal Proof Language
  - Design domain-specific tactics for geometry proofs
  - Design wrappers of SMT solvers to achieve better automation



Proposition 1

To construct an equilateral triangle on a given finite straight-line.

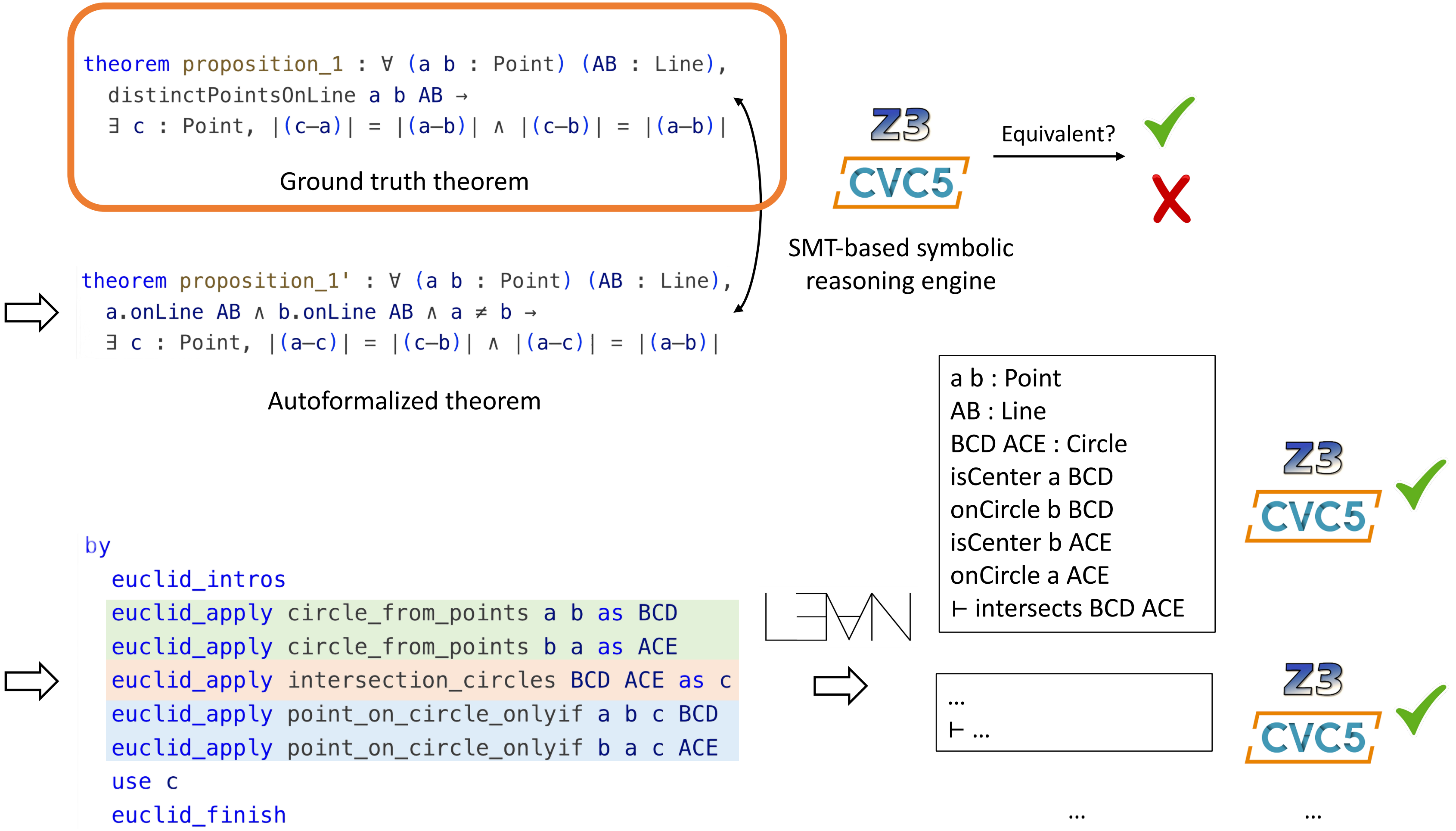
Let AB be the given finite straight-line.

So it is required to construct an equilateral triangle on the straight-line AB.

Let the circle BCD with center A and radius AB have been drawn [Post. 3], and again let the circle ACE with center B and radius BA have been drawn [Post. 3]. And let the straight-lines CA and CB have been joined from the point C, where the circles cut one another,† to the points A and B (respectively) [Post. 1].

And since the point A is the center of the circle CDB, AC is equal to AB [Def. 1.15]. Again, since the point B is the center of the circle CAE, BC is equal to BA [Def. 1.15]. But CA was also shown (to be) equal to AB. Thus, CA and CB are each equal to AB. But things equal to the same thing are also equal to one another [C.N. 1]. Thus, CA is also equal to CB. Thus, the three (straight-lines) CA, AB, and BC are equal to one another.

Thus, the triangle ABC is equilateral, and has been constructed on the given finite straight-line AB. (Which is) the very thing it was required to do.



```
theorem proposition_1 : ∀ (a b : Point) (AB : Line),
  distinctPointsOnLine a b AB →
  ∃ c : Point, |(c─a)| = |(a─b)| ∧ |(c─b)| = |(a─b)|
```

Ground truth theorem

```
theorem proposition_1' : ∀ (a b : Point) (AB : Line),
  a.onLine AB ∧ b.onLine AB ∧ a ≠ b →
  ∃ c : Point, |(a─c)| = |(c─b)| ∧ |(a─c)| = |(a─b)|
```

Autoformalized theorem

SMT-based symbolic reasoning engine

```
by
  euclid_intros
  euclid_apply circle_from_points a b as BCD
  euclid_apply circle_from_points b a as ACE
  euclid_apply intersection_circles BCD ACE as c
  euclid_apply point_on_circle_onlyif a b c BCD
  euclid_apply point_on_circle_onlyif b a c ACE
  use c
  euclid_finish
```

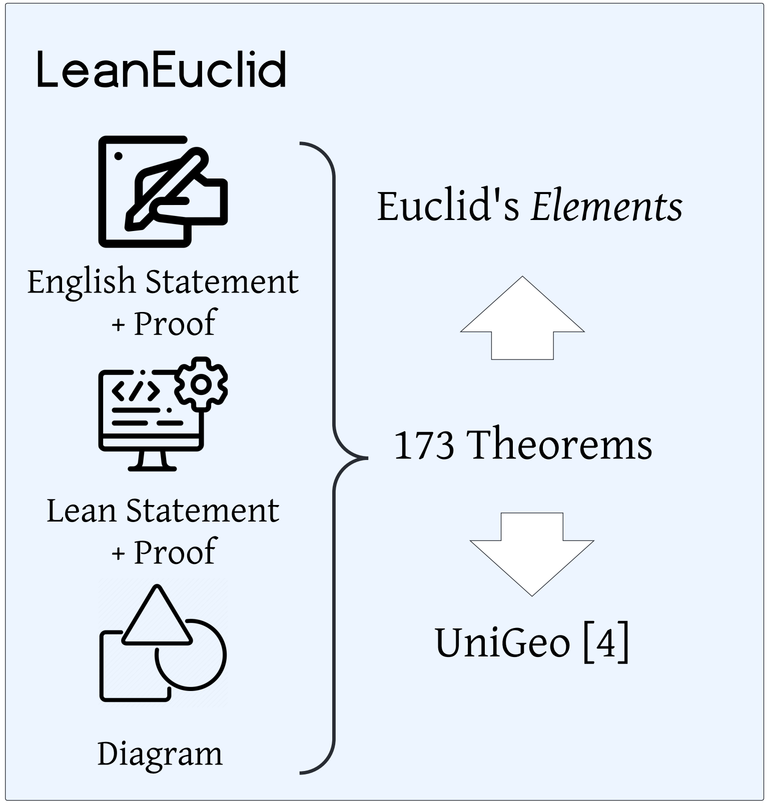
a b : Point
AB : Line
BCD ACE : Circle
isCenter a BCD
onCircle b BCD
isCenter b ACE
onCircle a ACE
⊢ intersects BCD ACE

…
⊢ …

… …

Logan Murphy*, Kaiyu Yang*, et al., *Autoformalizing Euclidean Geometry*, ICML 2024

Evaluation of State-of-the-art LLMs



LeanEuclid

English Statement + Proof

Lean Statement + Proof

Diagram

Euclid's *Elements*

173 Theorems

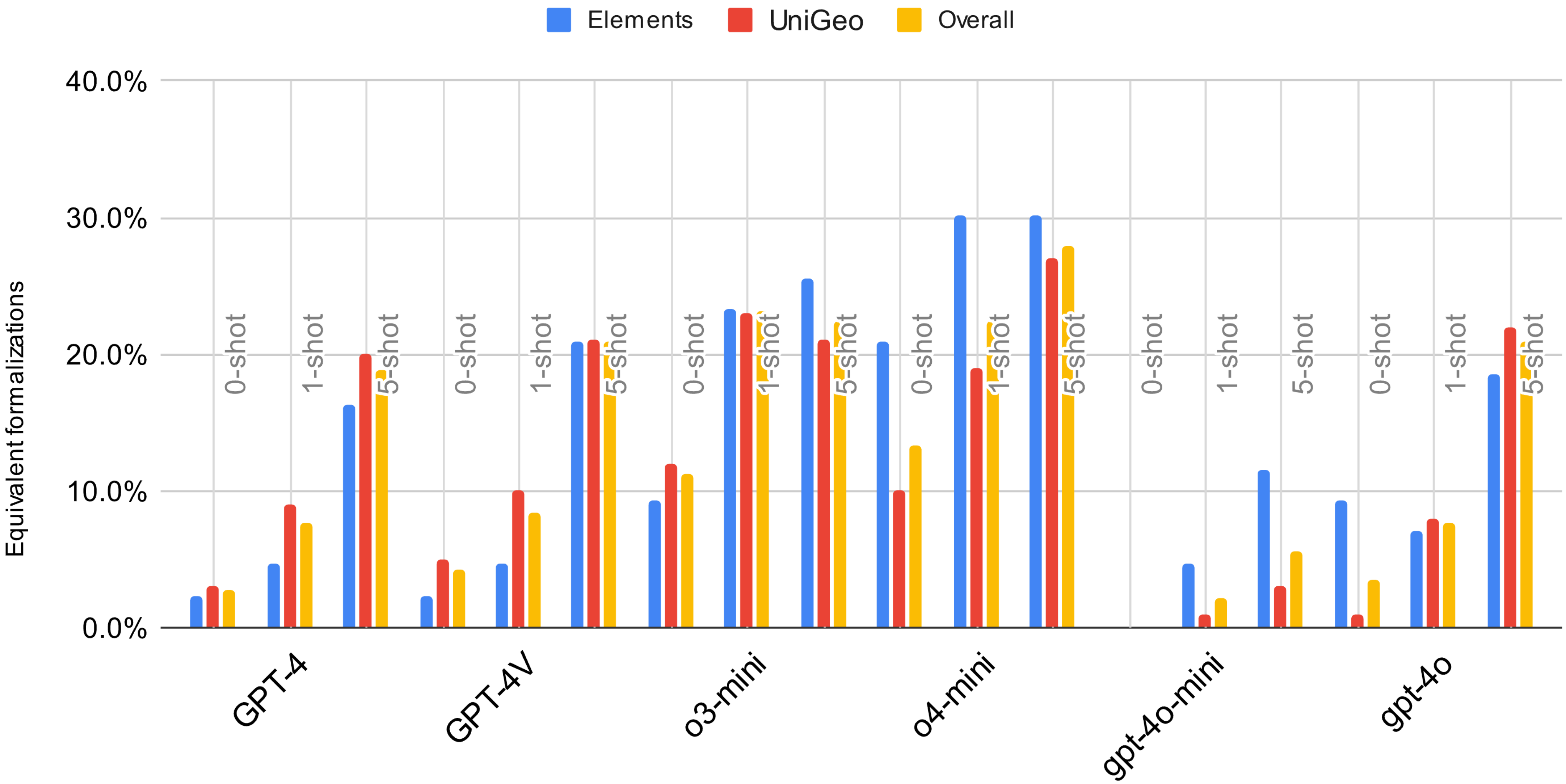
UniGeo [4]

	GPT-4			GPT-4V		
Dataset	0-shot	1-shot	5-shot	0-shot	1-shot	5-shot
Elements	2.3%	4.7%	16.3%	2.3%	4.7%	20.9%
UniGeo	3.0%	9.0%	20.0%	5.0%	10.0%	21.0%
Overall	2.8%	7.7%	18.9%	4.2%	8.4%	21.0%

Experiments conducted in ***Jan, 2024***
gpt-4-1106-preview
gpt-4-1106-vision-preview

Chen et al., *UniGeo: Unifying Geometry Logical Reasoning via Reformulating Mathematical Expression,* EMNLP 2022

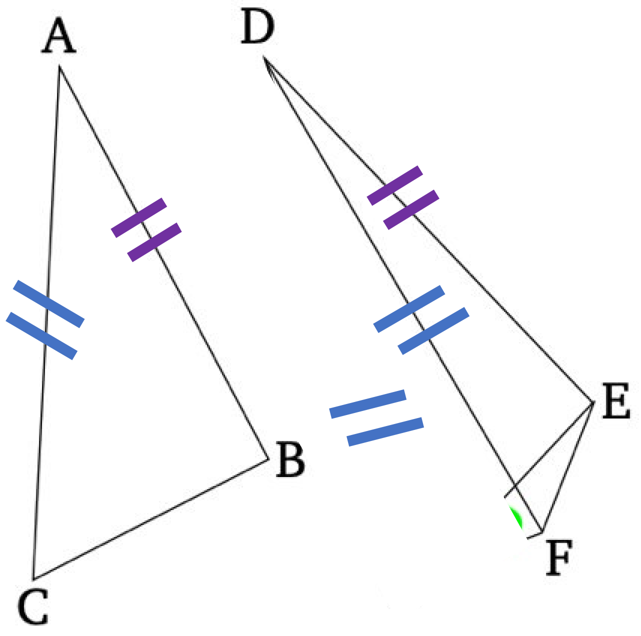
Re-evaluation in ***June, 2025***

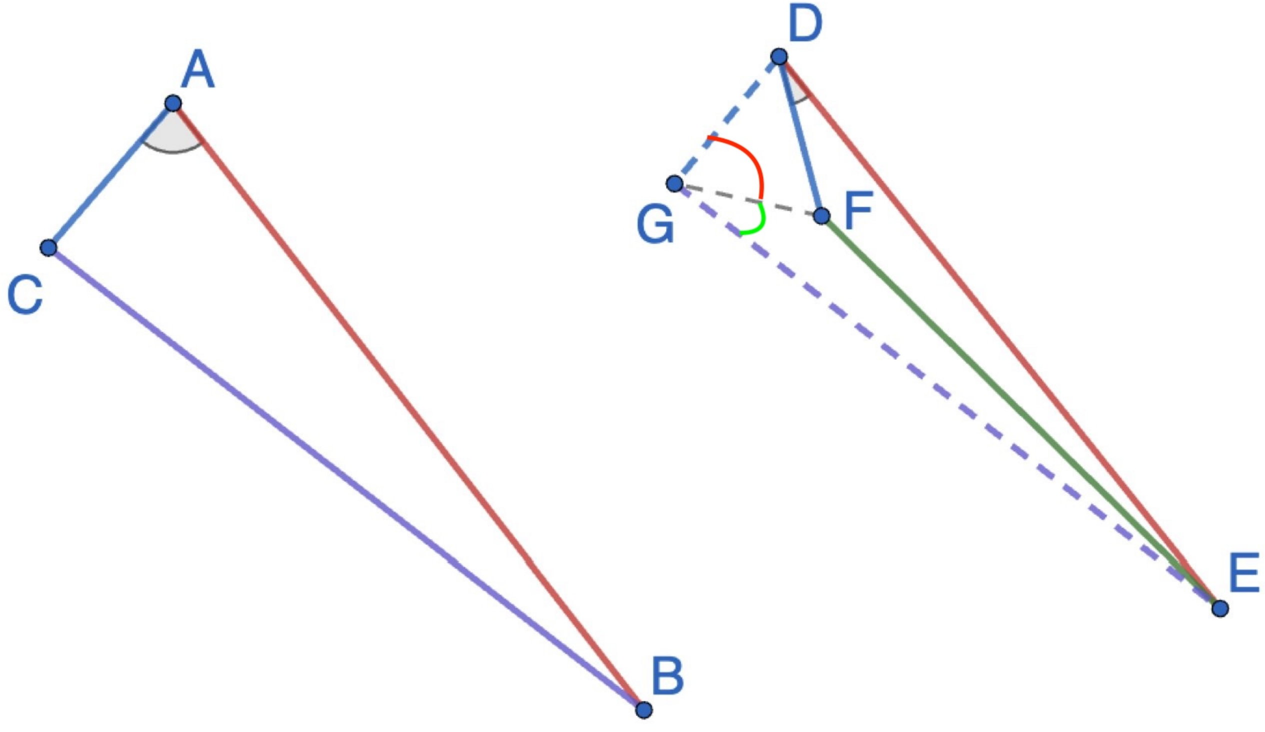


the non-trivial case missed by Euclid

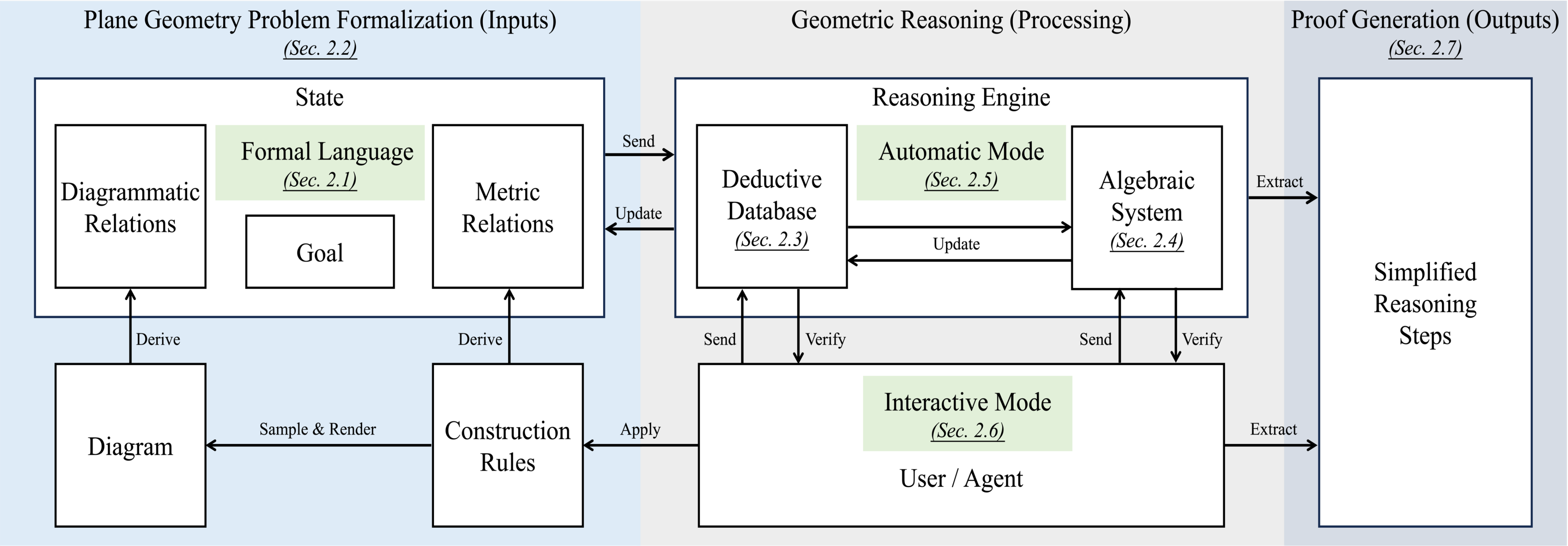
Proposition 24

If two triangles have two sides equal to two sides, respectively, but (one) has the angle encompassed by the equal straight-lines greater than the (corresponding) angle (in the other), then (the former triangle) will also have a base greater than the base (of the latter).





Better Proof automation with **PyEuclid**



Li et al., *PyEuclid: A Versatile Formal Plane Geometry System in Python*, CAV 2025

PyEuclid Performance

- **JGEX-AG-231 (AlphaGeometry benchmark)**
  - DD (152), Wu's method (173), DDAR (198), **PyEuclid (203)**
  - AlphaGeometry (228)

- **Geometry3K (more general benchmark)**
  - PyEuclid solves 529 problems first (out of 599 theorem proving problems)
  - Identifies 38 buggy problems
    - 21 with incorrect or contradictory values
    - 17 with missing or incorrect relations
  - Solved 567 (529 + 38) in total

Open Challenges

- How to assure the correctness/alignment of auto-formalization?
- How to automatically design the formal languages?
- How to automatically design/learn high-level proof tactics?
- How to *auto-**in**formalize* a giant proof?
- How to conjecture new interesting theorems?
- How to scale to textbook-level auto-formalization?
- How to scale to research article level auto-formalization?
- From peer review to machine review?

Takeaways

- Auto-formalization is full of interesting challenges

- Innovations & impacts can be made in many areas of research
  - Math foundations, Formal Methods, Machine Learning, NLP
  - Programming languages, Software Engineering, HCI
  - AI safety, Interpretability, Alignment