CS 957, System-2 AI
Neuro-Symbolic AI

Mohammad Hossein Rohban | Feb 2025

Sharif University of Technology

Article

Solving olympiad geometry without human demonstrations

https://doi.org/10.1038/s41586-023-06747-5

Received: 30 April 2023

Accepted: 13 October 2023

Published online: 17 January 2024

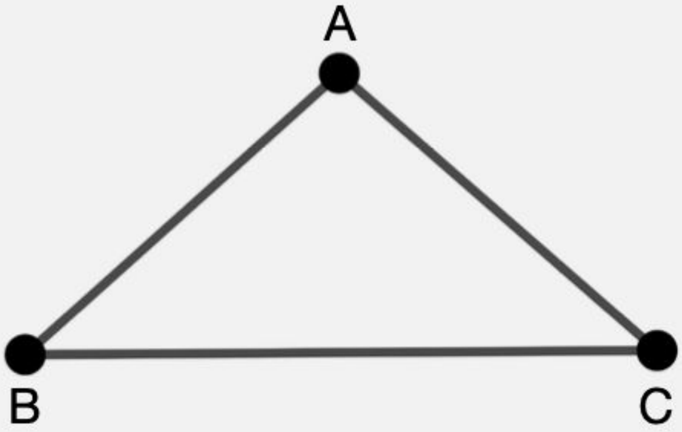
Open access

Check for updates

Trieu H. Trinh[1,2]✉, Yuhuai Wu[1], Quoc V. Le[1], He He[2] & Thang Luong[1]✉

Proving mathematical theorems at the olympiad level represents a notable milestone in human-level automated reasoning[1–4], owing to their reputed difficulty among the world's best talents in pre-university mathematics. Current machine-learning approaches, however, are not applicable to most mathematical domains owing to the high cost of translating human proofs into machine-verifiable format. The problem is

Simple Problem Formulation



"Let ABC be any triangle with AB = AC.
Prove that $\angle ABC = \angle BCA$."

Symbolic Solver

What we have:

- Symbolic solvers whose knowledge base are of the Horn clause form: $Q \leftarrow P_1(x) \wedge \ldots \wedge P_k(x)$
- Contains a whole lot of deduction rules (deductive database or DD) in 2D euclidean geometry.
  - If two AB and BC are collinear, then ∡ABC is 180 degrees.
  - AB = DE, AC = DF, and ∡CAB = ∡EDF, then ABC = DEF.
  - ...
- How does the solver work?
  - Picks one of the rules at a time whose premises ($P_1(x)$, ..., $P_k(x)$) are known to be true.
  - Add the conclusion Q to the knowledge base and repeat.

Algebraic Rules (AR)

AR is necessary to perform angle, ratio and distance chasing.

First, convert the input linear equations to a matrix of their coefficients.

Applies Gaussian Elimination

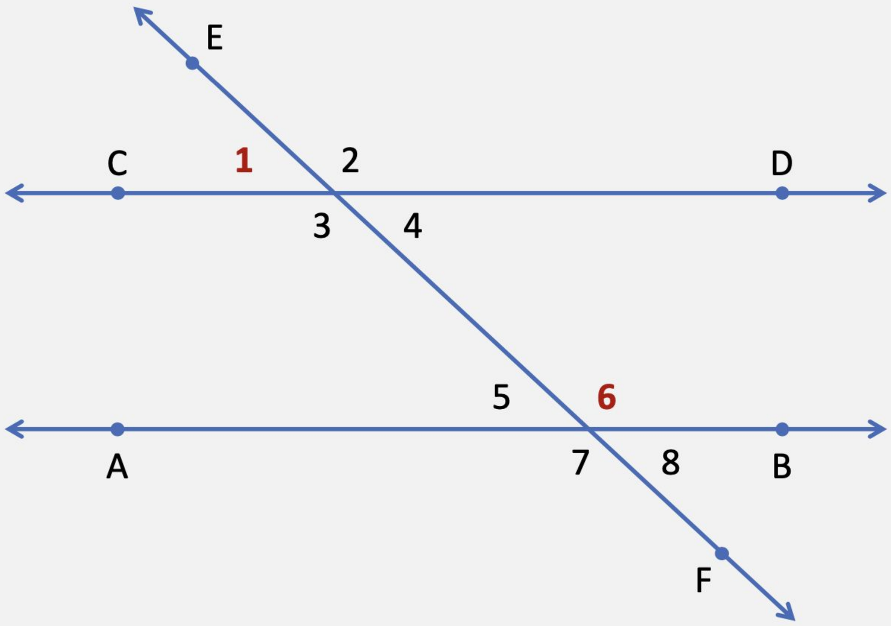
∠ABC = ∠XYZ is represented as s(AB) – s(BC) = s(XY) – s(YZ), in which s(AB) is the angle between AB and the x-direction, modulo pi.

Similarly, ratios AB:CD = EF:GH are represented as log(AB) – log(CD) = log(EF) – log(GH), in which log(AB) is the log of the length of segment AB.

$$\begin{pmatrix} a & b & c & d & e \\ 1 & -2 & 1 & 0 & 0 \\ -1 & 0 & -1 & 2 & 0 \\ 0 & 1 & -2 & 0 & 1 \end{pmatrix} \xrightarrow{GE} \begin{pmatrix} a & b & c & d & e \\ 1 & 0 & 0 & -1.5 & 0.5 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -0.5 & -0.5 \end{pmatrix} \Rightarrow \begin{cases} a = 1.5d - 0.5e \\ b = d \\ c = 0.5d + 0.5e \end{cases}$$

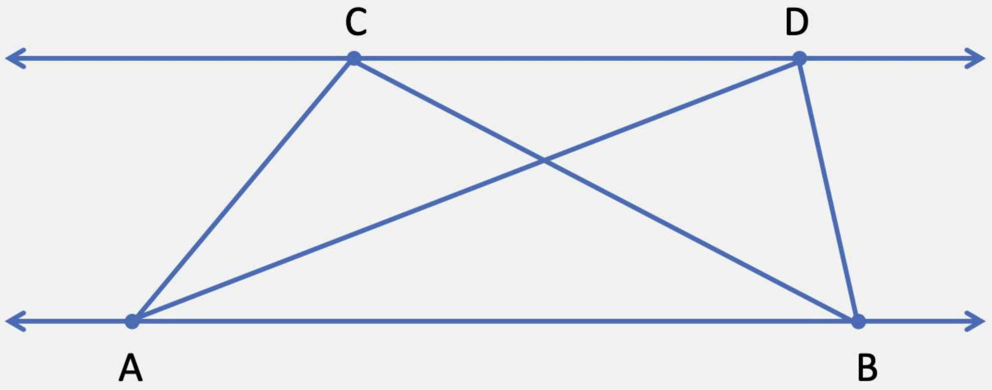
Example

- We want to prove that angle 1 and angle 6 are supplementary, given parallel lines AB and CD.
- First, symbolic engine might prove that angle 1 equals angles 4, 5, and 8
- Then, it might prove that angle 6 and angle 8 are supplementary.
- Combining the above two, we have shown that angle 1 and angle 6 are supplementary.



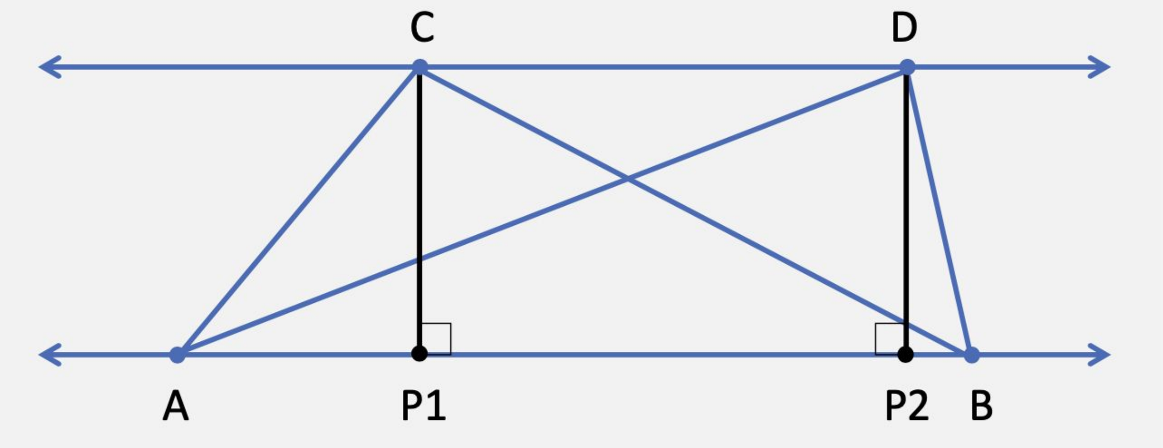
Another Example

We want to prove that triangle ABC and triangle ABD have the same area, given parallel lines AB and CD

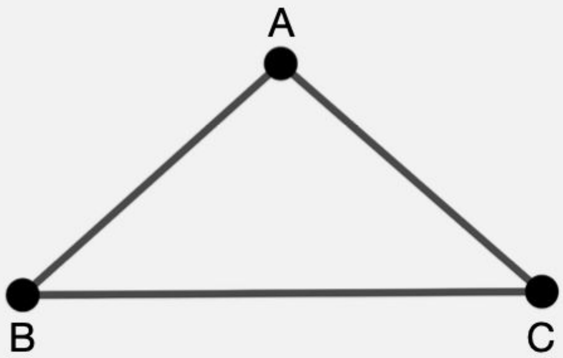


Adding Auxiliary Points

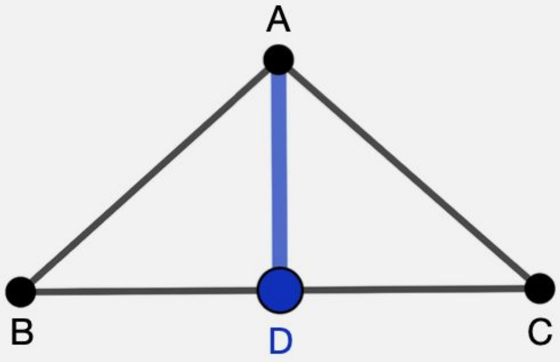
- Adding two points P1 and P2 allows us to show the heights of both triangles are the same
- Since both share the same base, the areas are the same



Adding Constructs (point D) Helps a lot!



"Let ABC be any triangle with AB = AC.
Prove that ∠ABC = ∠BCA."



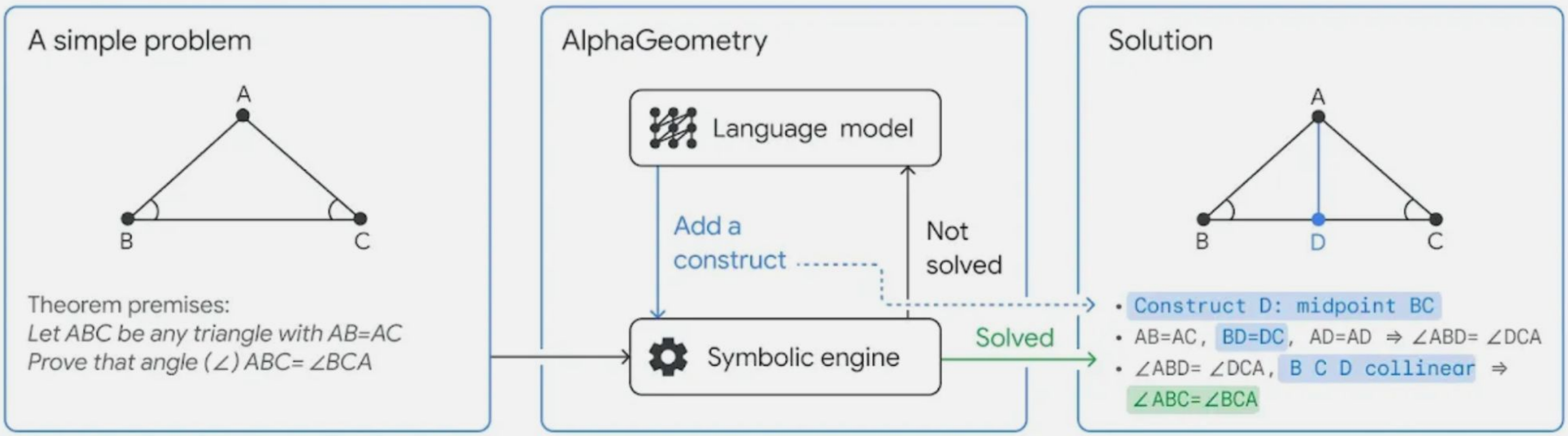
- Construct D: midpoint BC,
  AB=AC, BD = DC, AD=AD ⇒ ∠ABD=∠DCA [1]
  [1], B C D collinear ⇒ ∠ABC=∠BCA

Let's brainstorm how to go about solving this problem

How to incorporate auxiliary points?

General Strategy of AlphaGeometry

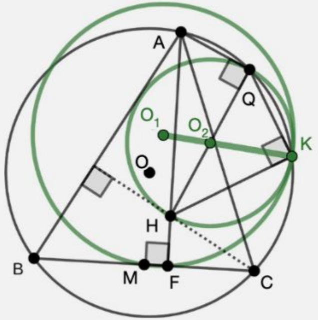
The space of auxiliary construction is **huge**



A closer look into the solution

e IMO 2015 P3

"Let ABC be an acute triangle. Let (O) be its circumcircle, H its orthocenter, and F the foot of the altitude from A. Let M be the midpoint of BC. Let Q be the point on (O) such that QH ⊥ QA and let K be the point on (O) such that KH ⊥ KQ. Prove that the circumcircles (O_1) and (O_2) of triangles FKM and KQH are tangent to each other."



→ Alpha-Geometry →

f Solution

...

Construct D: midpoint BH [a]

[a], O_2 midpoint HQ ⇒ BQ // O_2D [20]

...

Construct G: midpoint HC [b] ...

$\angle$GMD = $\angle$GO_2D ⇒ M O_2 G D cyclic [26]

...

[a],[b] ⇒ BC // DG [30]

...

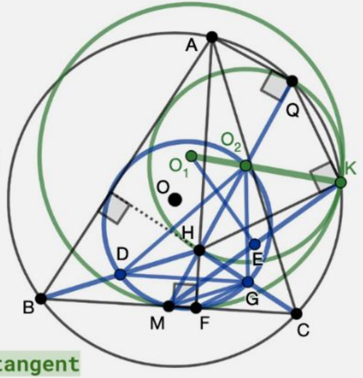
Construct E: midpoint MK [c]

..., [c] ⇒ $\angle$KFC = $\angle$KO_1E [104]

...

$\angle$FKO_1=$\angle$FKO_2 ⇒ KO_1 // KO_2 [109]

[109] **⇒ O_1O_2K collinear ⇒ (O_1)(O_2) tangent**



IMO 2004 P1:
"Let ABC be an acute-angled triangle with AB ≠ AC. The circle with diameter BC intersects the sides AB and AC at M and N respectively. Denote by O the midpoint of the side BC. The bisectors of the angles ∠BAC and ∠MON intersect at R. Prove that the circumcircles of the triangles BMR and CNR have a common point lying on the side BC."

Translate

Premise

```
A B C O M N R P : Points
mid_point(O,B,C) [--]
same_line(B,M,A) [00] OM=OB [01]
same_line(N,C,A) [02] ON=OB [03]
∠BAR=∠RAC [04] ∠MOR=∠RON [05]
circle(B,M,R,P) [06] circle(C,N,R,P) [07]
```

Goal

```
same_line(P, B, C)
```

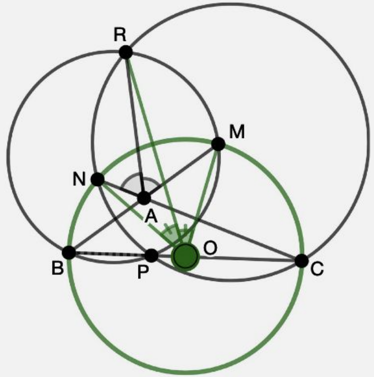
Solve

Proof

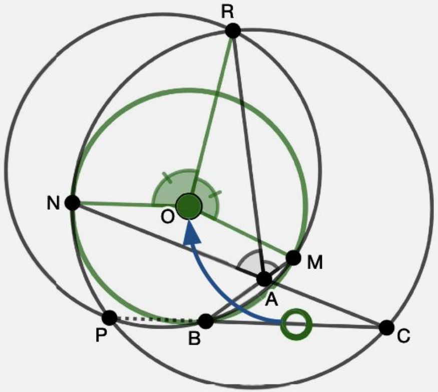
```
[01][03] ⇒ ∠ONM=∠NMO [08]
[01][03][05] ⇒ RN=RM [09]
[01][03][09] ⇒ NM ⊥ OR [10]
AUXILIARY POINT K : KM = KN
[01][03] KM = KN ⇒ MN ⊥ KO [12]
AUXILIARY POINT L : KL = KA, OL = OA
KL = KA,OL = OA ⇒ KO⊥AL [15] ∠AKO=∠OKL [16]
[15][12][10][16][13] ⇒ RA=RL [17]
OL = OA ⇒ ∠OAL=∠ALO [18]
angle-chase:[12][15][08][18]⇒∠NOA=∠LOM [19]
[01][03]OL = OA[19] ⇒ AN=LM [21]
[17][21][09] ⇒ ∠NAR=∠RLM [22]
[02][04][00][22] ⇒ circle(L,M,A,R) [23]
similar ⇒ circle(R,L,N,A) [24]
[23][24] ⇒ ∠RMA=∠RNA [25]
[06] ⇒ ∠BPR=∠BMR [26]
[07] ⇒ ∠NCP=∠NRP [27]
[00][02][25][26][27] ⇒ PC // BP
⇒ same_line(B,P,C)
```

Traceback

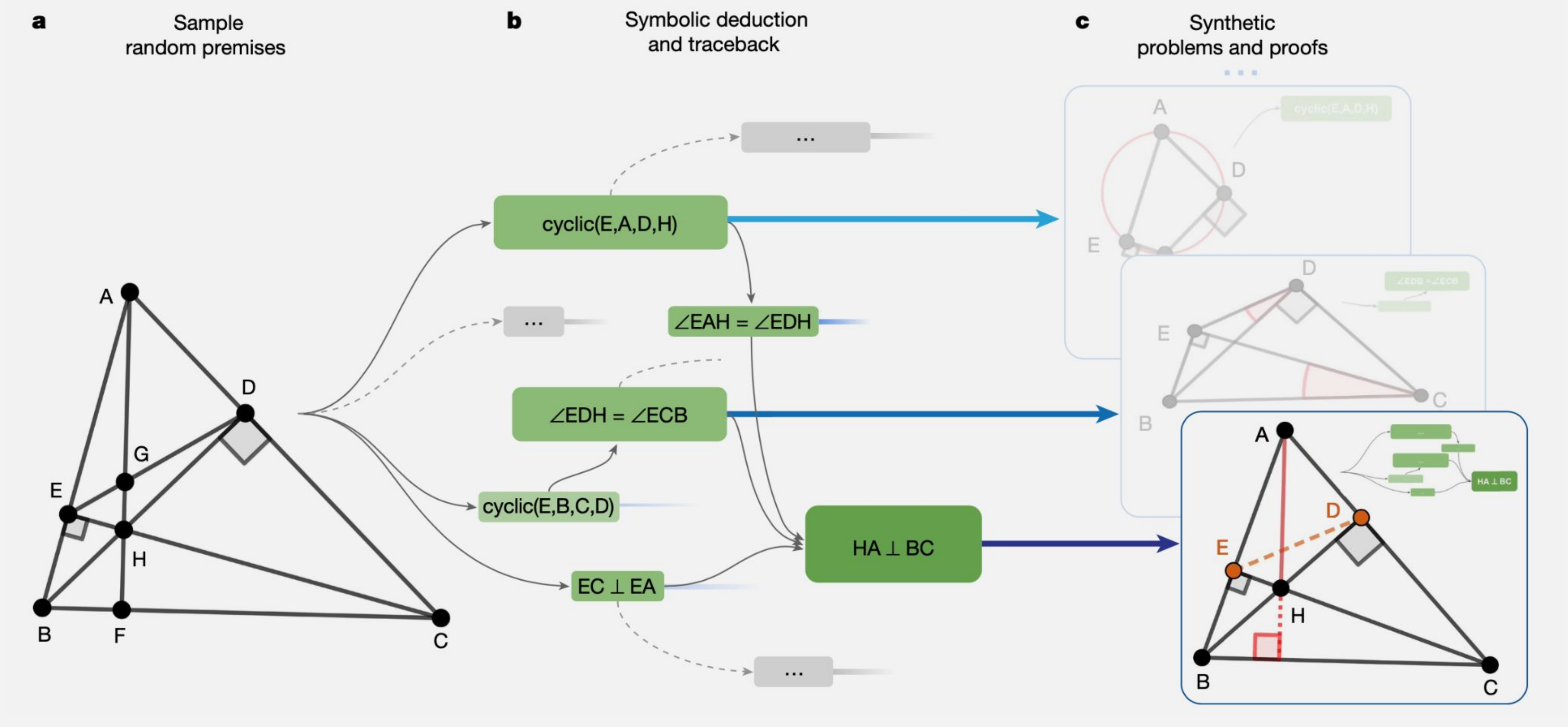
ABC Unused premise
ABC Used premises
ABC Neural net output
ABC Symbolic solver output



Generalize



How to make synthetic data?

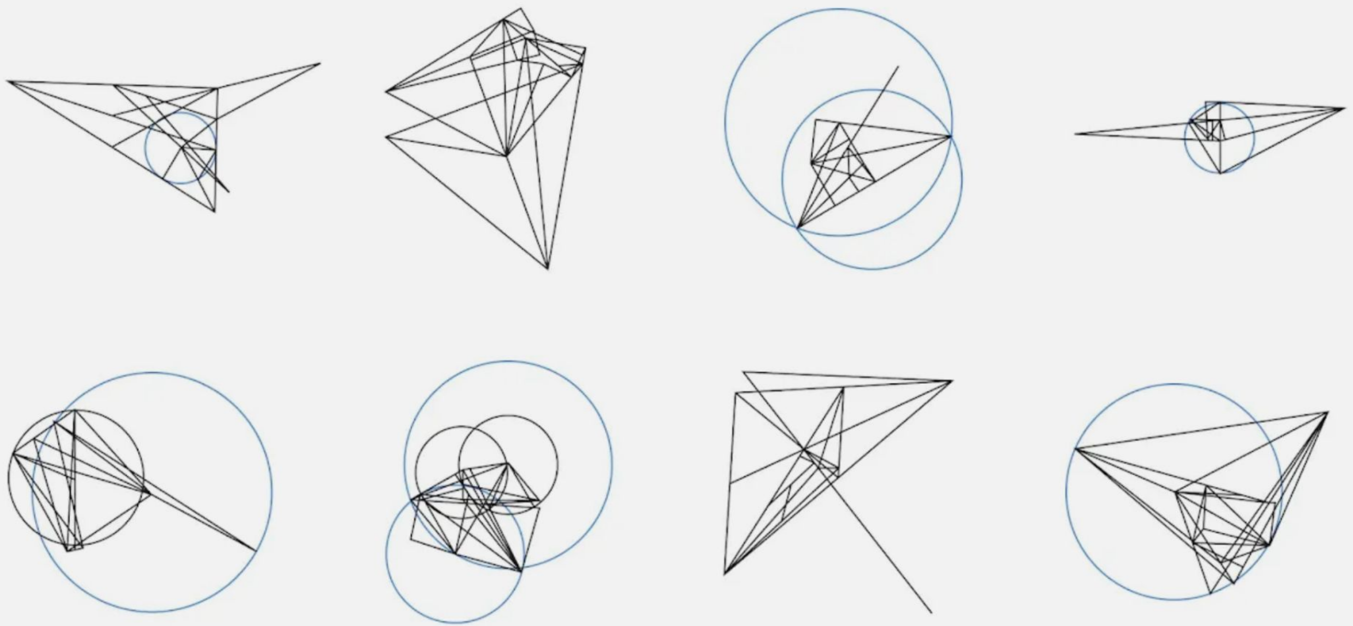


How to generate premises?

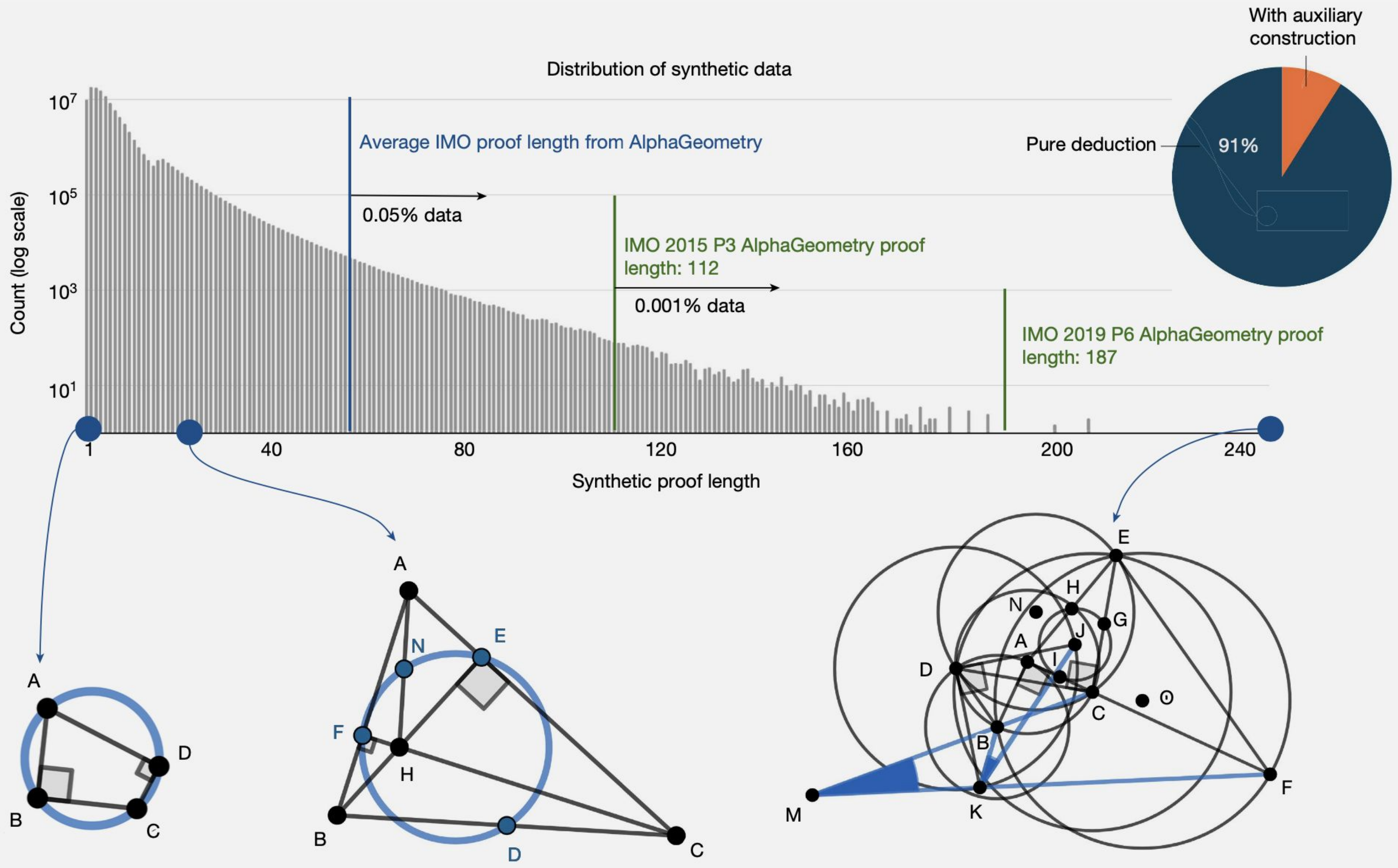
- A small set of rules to make random premises once applying them consecutively.

Construction	Description
X = angle bisector(A, B, C)	Construct a point X on the angle bisector of ∠ABC
X = angle mirror(A, B, C)	Construct a point X such that BC is the bisector of ∠ABX
X = circle(A, B, C)	Construct point X as the circumcenter of A, B, C
A, B, C, D = eq_quadrilateral()	Construct quadrilateral ABCD with AD = BC
A, B, C, D = eq_trapezoid()	Construct trapezoid ABCD with AD = BC
X = eqtriangle(B, C)	Construct X such that XBC is an equilateral triangle
X = eqangle2(A, B, C)	Construct X such that ∠BAX = ∠XCB
A,B,C,D = eqdia_equadrilateral()	Construct quadrilateral ABCD with AC = BD
X = eqdistance(A, B, C)	Construct X such that XA = BC
X = foot(A, B, C)	Construct X as the foot of A on BC
X = free	Construct a free point X

Random Premises



Visual representations of the synthetic data generated by AlphaGeometry



Trivial theorem
Proof length = 1

Well-known theorem
Proof length = 20

Most complex synthetic theorem
Proof length = 247

How to train the LLM?

Randomly make a premise.

Make a deduction closure.

Pick one of the conclusions as a target.

Trace back to the premise so that necessary premises to deduce the selected target.

Remove some constructs $X_1, \ldots, X_k$ that do not appear in the target.

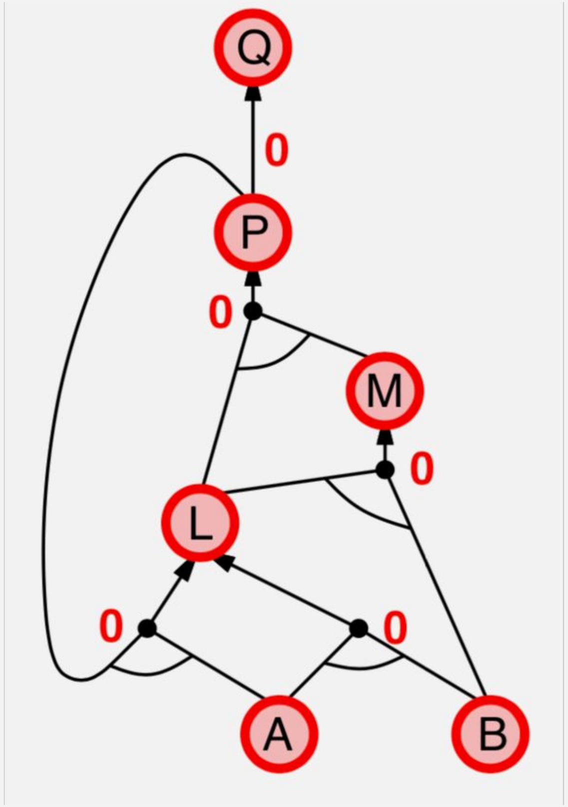
Add few sentences to the proof "Add construct X_i" for all i.

Train an LLM with all proofs

Fine tune the LLM with modified proofs containing added constructs.

Forward Chaining to get the closure

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Language Model Architecture

12 layers Transformer with ~151 M weights

Embedding dim = 1,024

8 attention heads

ReLU activation

Context length = 1,024

Vocab. Size = 757

LLM Inference

Takes as input the <premise> <conclusion> <proof> format.

Adding constructs

Performs beam search to produce k=512 top (highest probable) next tokens

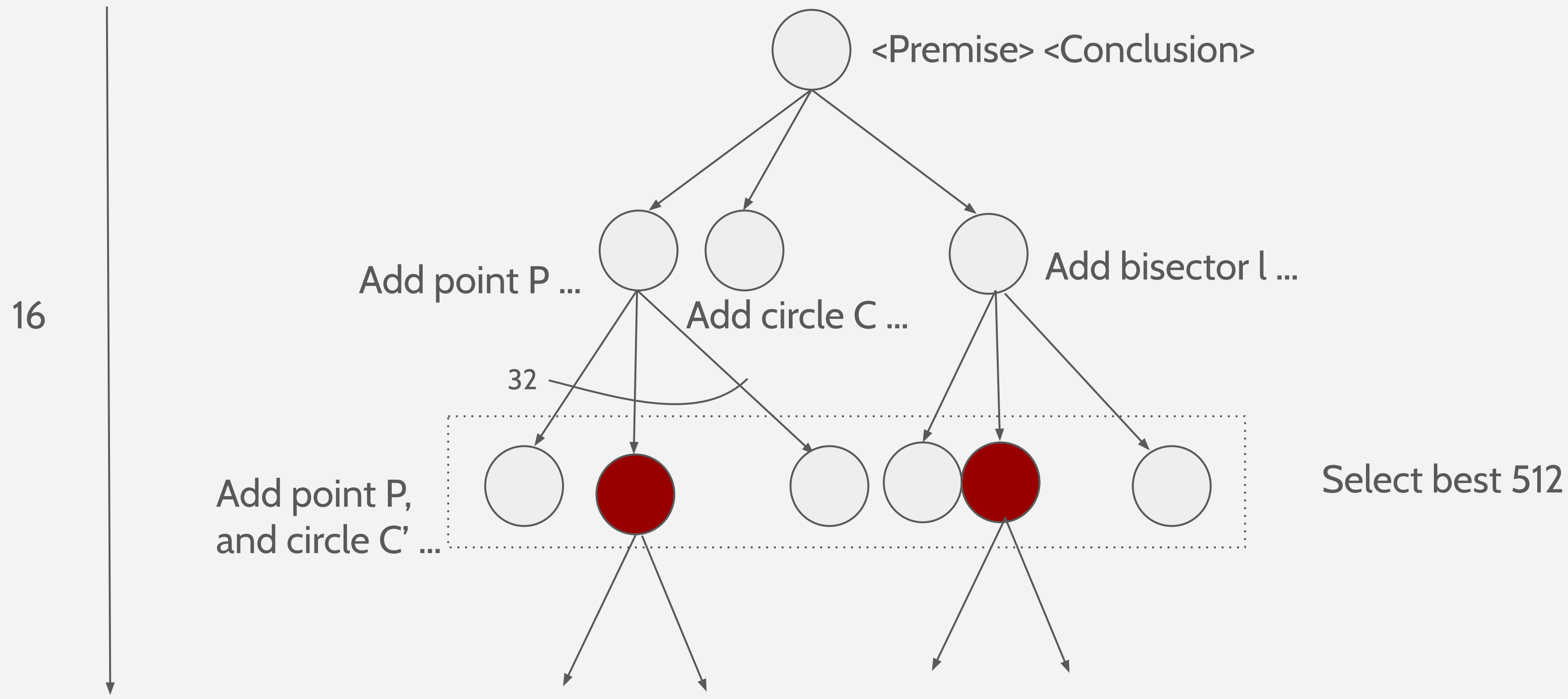
Branching width = 32

Each node containing a set of constructs is fed into the symbolic solver.

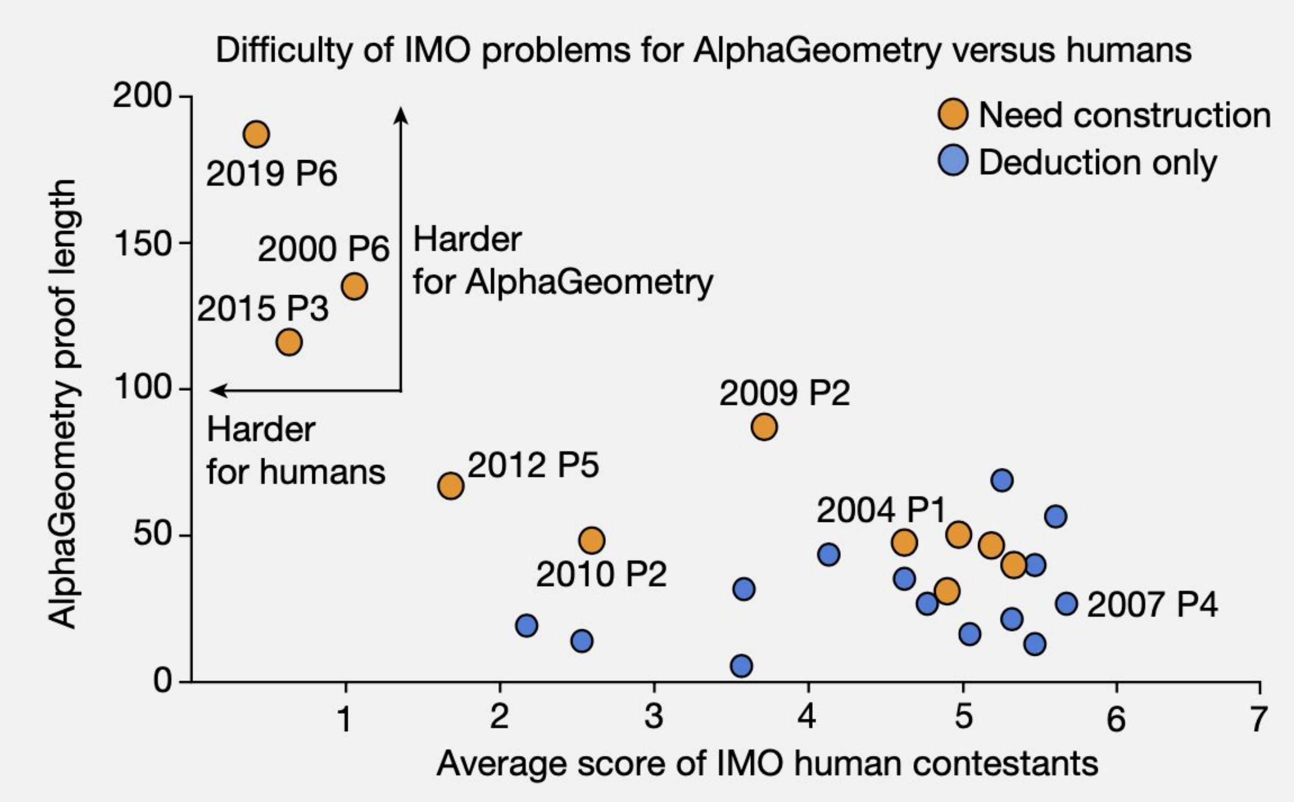
If solved then stop.

Search to depth = 16

LLM Inference with beam search



Longer proof problems are harder

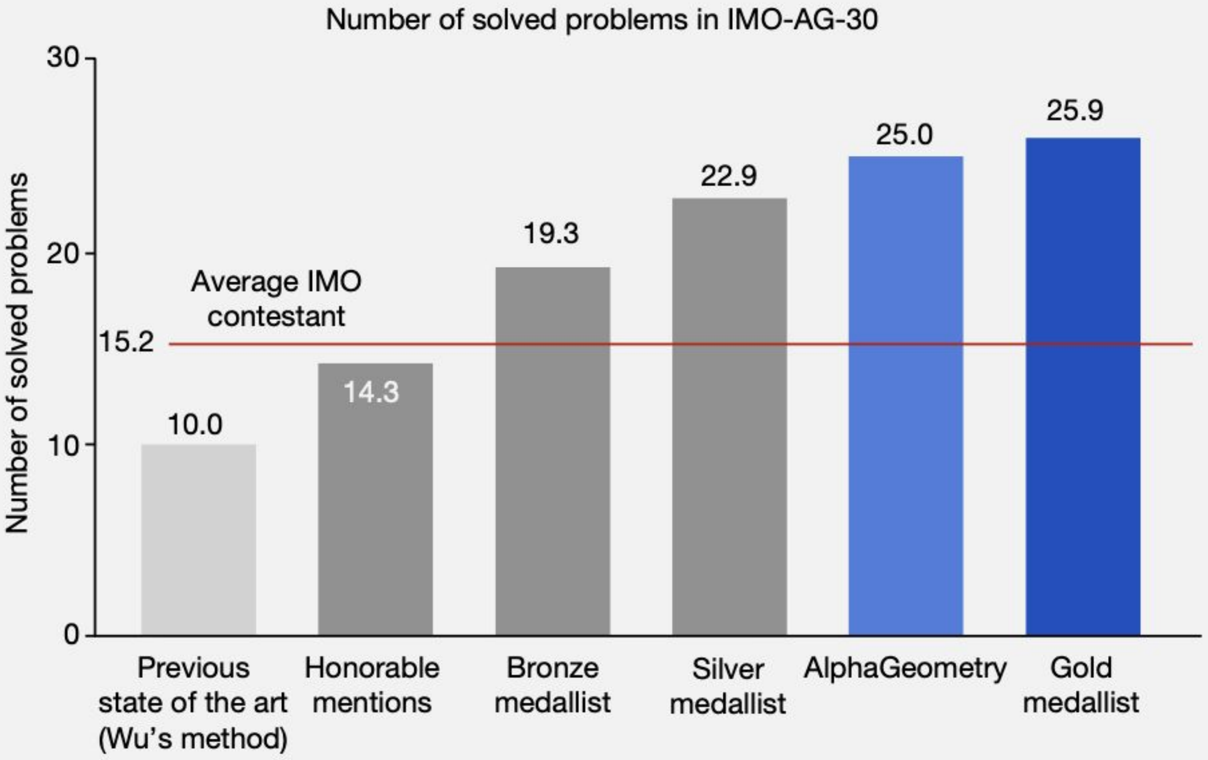


Results

Method		Problems solved (out of 30)
Computer algebra	Wu's method[21] (previous state of the art)	10
	Gröbner basis[20]	4
Search (human-like)	GPT-4 (ref. 25)	0
	Full-angle method[30]	2
	Deductive database (DD)[10]	7
	DD + human-designed heuristics[17]	9
	DD + AR (ours)	14
	DD + AR + GPT-4 auxiliary constructions	15
	DD + AR + human-designed heuristics	18
	AlphaGeometry	25
	• Without pretraining	21
	• Without fine-tuning	23

We compare AlphaGeometry to other state-of-the-art methods (computer algebra and search approaches), most notably Wu's method. We also show the results of DD + AR (our contribution) and its variants, resulting in the strongest baseline DD + AR + human-designed heuristics. Finally, we include ablation settings for AlphaGeometry without pretraining and fine-tuning.

On par with gold medalist



Ablation

a.

Training data size	Solved / 30
100M	25
80M	24
60M	23
40M	23
20M	21

b.

Method	Solved / 231
Wu	173
DD	152
DD+human heuristics	160
DD+AR	198
DD+AR+human heuristics	213
AlphaGeometry	228

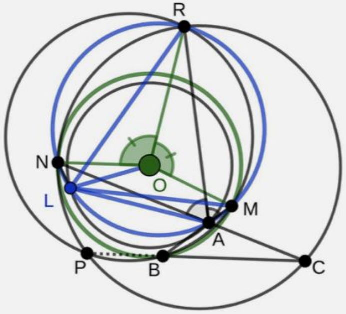
c.

Beam size	Solved / 30
512	25
128	25
32	24
8	21
2	16

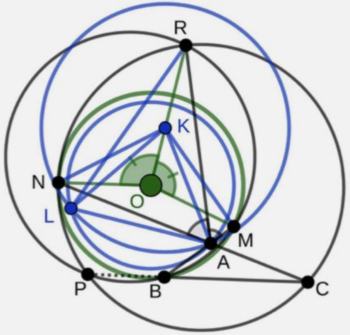
d.

Search depth	Solved / 30
16	25
8	25
4	25
2	21
1	16

Original problem statement:
Let ABC be an acute-angled triangle with AB ≠ AC. Let O be any point. The circle with diameter BC intersects the sides AB and AC at M and N respectively. Denote by O the midpoint of the side BC. The bisectors of the angles ∠BAC and ∠MON intersect at R. Prove that the circumcircles of the triangles BMR and CNR have a common point lying on the side BC.



Adapted problem statement:
Let ABC be a triangle. Let O be any point. Define point M as the intersection of circle (O, B) and line AB. Define point N as the intersection of circle (O,B) and line AC. Define point R such that AR is the bisector of ∠BAC and OR is the bisector of ∠MON. Define point O1 as the circumcenter of triangle BRM. Define point O2 as the circumcenter of triangle NRC. Define point P as the intersection of circles (O1, R) and (O2, R). Prove that B, C, P are collinear.



Human proof:

Let L be the reflection of A about OR — **Minimal construction**

∠RLM = ∠NAR (LN is the reflection of AM about OR)
= ∠RAM (AR is bisector of ∠NAM)
⇒ L,M,A,R is cyclic

Short, high-level deductions

Similarly, ANLR is cyclic

⇒ RMNA is cyclic
So BPR = BMR (BMRP is cyclic)
= AMR
= ANR (RMNA is cyclic)
= CNR
= CPR (CNRP is cyclic)
⇒ BP // CP

Readable algebraic steps

⇒ B, P, and C is collinear.

AlphaGeometry proof:

```
AUXILIARY POINT K : KM=KN
AUXILIARY POINT L : KL=KA, OL=OA
```

Redundant

```
[01][03] ⇒ ∠ONM=∠NMO [08]
[01][03][05] ⇒ RN=RM [09]
[09][01][03] ⇒ NM ⊥ OR [10]
[01][03][KM=KN] ⇒ MN ⊥ KO [12]
[KL=KA][OL=OA] ⇒ KO ⊥ AL [15] ∠AKO=∠OKL [16]
[15][12][10][16][KL=KA] ⇒ RA=RL [17]
[OL=OA] ⇒ ∠OAL=∠ALO [18]
angle-chase:[12][15][08][18]⇒ ∠NOA=∠LOM [19]
[19][01][03][OL=OA] ⇒ AN=LM [21]
[17][21][09] ⇒ ∠NAR=∠RLM [22]
[22][02][04][00] ⇒ circle(L,M,A,R) [23]
```

Verbose, low-level steps

```
similar ⇒ circle(R,L,N,A) [24]
```

```
[23][24] ⇒ ∠RMA=∠RNA [25]
[06] ⇒ ∠BPR=∠BMR [26]
[07] ⇒ ∠NCP=∠NRP [27]
angle-chase:[25][00][02][26][27] ⇒ PC // BP
```

Low readability

```
⇒ same_line(B,P,C)
```

Neuro-Symbolic Architectures so far ...

Neural | Symbolic uses a neural architecture to interpret perceptual data as symbols and relationships that are reasoned about symbolically. NS-VQA, and Neural-Concept Learner are examples.

Symbolic[Neural] is exemplified by AlphaGo, and AlphaGeometry, where symbolic techniques are used to invoke neural techniques. In this case, the symbolic approach is beam search together with symbolic proof engine, and the neural techniques is the LLM suggesting constructions.