Alphageometry in Practice

Anna Petiurenko

University of the National Education Commission, Krakow

Article | Open access | Published: 17 January 2024

Solving olympiad geometry without human demonstrations

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He & Thang Luong

Nature **625**, 476–482 (2024) | Cite this article

An Author Correction to this article was published on 23 February 2024

This article has been updated

Associated content

This AI just figured out geometry — is this a step towards artificial reasoning?

Nature | **Nature Podcast** | 17 Jan 2024

Language

List of actions to construct the random premises in Alphageometry

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
X = incenter(A, B, C)	Construct X as the incenter of ABC
X,Y,Z,I = incenter2(A, B, C)	Construct I as the incenter of ABC with touchpoints X, Y, Z
X = excenter(A, B, C)	Construct X as the excenter of ABC
X,Y,Z,I = excenter2(A,B,C)	Construct X as the excenter of ABC with touchpoints X,Y,Z
X = centroid(A,B,C)	Construct X as the centroid of ABC
X,Y,Z,I = midpointcircle(A,B,C)	Construct X, Y, Z as the midpoints of triangle ABC, and I as the circumcenter of XYZ
A,B,C = isos()	Construct A, B, C such that AB = AC
X = tangent(O, A)	Construct X such that OA is perpendicular to AX
X = midpoint(A, B)	Construct X as the midpoint of AB

List of actions to construct the random premises in Alphageometry

Construction	Description
X = mirror(A, B)	Construct X such that B is the midpoint of AX
X = rotate90(A, B)	Construct X such that AXB is a right isosceles triangle
X = on_aline(A, B, C, D, E)	Construct X such that ∠XAB = ∠CDE
X = on_bline(X, A, B)	Construct X on the perpendicular bisector of AB
X = on_circle(O, A)	Construct X such that OA = OX
X = on_line(A, B)	Construct X on line AB
X = on_pline(A, B, C)	Construct X such that XA is parallel to BC
X = on_tline(A, B, C)	Construct X such that XA is perpendicular to BC
X = orthocenter(A, B, C)	Construct X as the orthocenter of ABC
X = parallelogram(A, B, C)	Construct X such that ABCX is a parallelogram
A, B, C, D, E = pentagon()	Construct pentagon ABCDE
A, B, C, D = quadrilateral()	Construct quadrilateral ABCD
A, B, C, D = trapezoid()	Construct right trapezoid ABCD
A, B, C = r_triangle()	Construct right triangle ABC
A, B, C, D = rectangle()	Construct rectangle ABCD
X = reflect(A, B, C)	Construct X as the reflection of A about BC
A, B, C = risos()	Construct right isosceles triangle ABC
X = angle(A, B, α)	Construct X such that ∠ABX = α
A, B = segment()	Construct two distinct points A, B
X = shift(B, C, D)	Construct point X such that XB=CD and XC=BD
X Y = square(A, B)	Construct X, Y such that XYAB is a square

List of actions to construct the random premises in Alphageometry

Construction	Description
A, B, C, D = init_square()	Construct square ABCD
A, B, C, D = trapezoid()	Construct trapezoid ABCD
A, B, C = triangle()	Construct triangle ABC
A, B, C = triangle12()	Construct trianglel ABC with AB:AC = 1:2
X,Y,Z,I = 2L1C(A, B, C, O)	Construct circle center I that touches line AC and line BC and circle (O, A) at X, Y, Z
X, Y, Z = 3PEQ(A, B, C)	Construct X, Y, Z on three sides of triangle ABC such that Y is the midpoint of XZ
X, Y = trisect(A, B, C)	Construct X, Y on AC such that BX and BY trisect ∠ABC
X, Y = trisegment(A, B)	Construct X, Y on segment AB such that AX=XY=YB
X = on_dia(A, B)	Construct point X such that AX is perpendicular to BX
A, B, C = ieqtriangle()	Construct equilateral triangle ABC
X, Y, Z, T = cc_tangent(O, A, W, B)	Construct common tangents of circles (O, A) and (W, B) with touchpoints X, Y for one tangent and Z, T for the other.
X = eqangle3(A, B, D, E, F)	Construct point X such that ∠AXB = ∠EDF
X, Y = tangent(A, O, B)	Construct points X, Y as the tangent touch points from A to circle (O, B)
X = intersect(f, g)	Construct point X as the intersection of two functions f() and g(), where f() and g() is any of the above functions that returns more than one possible construction.

„After running this process on 100,000 CPU workers for 72 h, we obtained roughly 500 million synthetic proof examples. We reformat the proof statements to their canonical form (for example, sorting arguments of individual terms and sorting terms within the same proof step, etc.) to avoid shallow deduplication against itself and against the test set. At the end, we obtain 100 million unique theorem–proof examples. A total of 9 million examples involves at least one auxiliary construction. We find no IMO-AG-30 problems in the synthetic data. On the set of geometry problems collected in JGEX17, which consists mainly of problems with moderate difficulty and well-known theorems, we find nearly 20 problems in the synthetic data. This suggests that the training data covered a fair amount of common knowledge in geometry, but the space of more sophisticated theorems is still much larger."

What is the starting point?

https://github.com/google-deepmind/alphageometry/blob/main/rules.txt

Rules

1. perp A B C D, perp C D E F, ncoll A B E => para A B E F
2. cong O A O B, cong O B O C, cong O C O D => cyclic A B C D
3. eqangle A B P Q C D P Q => para A B C D
4. cyclic A B P Q => eqangle P A P B Q A Q B
5. eqangle6 P A P B Q A Q B, ncoll P Q A B => cyclic A B P Q
6. cyclic A B C P Q R, eqangle C A C B R P R Q => cong A B P Q
7. midp E A B, midp F A C => para E F B C
8. para A B C D, coll O A C, coll O B D => eqratio3 A B C D O O
9. perp A B C D, perp E F G H, npara A B E F => eqangle A B E F C D G H
10. eqangle a b c d m n p q, eqangle c d e f p q r u => eqangle a b e f m n r u
11. eqratio a b c d m n p q, eqratio c d e f p q r u => eqratio a b e f m n r u
12. eqratio6 d b d c a b a c, coll d b c, ncoll a b c => eqangle6 a b a d a d a
13. eqangle6 a b a d a d a c, coll d b c, ncoll a b c => eqratio6 d b d c a b a c c
14. cong O A O B, ncoll O A B => eqangle O A A B A B O B
15. eqangle6 A O A B B A B O, ncoll O A B => cong O A O B

Rules

16.cong O A O B, ncoll O A B => eqangle O A A B A B O B
17.eqangle6 A O A B B A B O, ncoll O A B => cong O A O B
18.circle O A B C, perp O A A X => eqangle A X A B C A C B
19.circle O A B C, eqangle A X A B C A C B => perp O A A X
20.circle O A B C, midp M B C => eqangle A B A C O B O M
21.circle O A B C, coll M B C, eqangle A B A C O B O M => midp M B C
22.perp A B B C, midp M A C => cong A M B M
23.circle O A B C, coll O A C => perp A B B C
24.cyclic A B C D, para A B C D => eqangle A D C D C D C B
25.cyclic A B C D, para A B C D => eqangle A D C D C D C B
26.midp M A B, perp O M A B => cong O A O B
27.cong A P B P, cong A Q B Q => perp A B P Q
28.cong A P B P, cong A Q B Q, cyclic A B P Q => perp P A A Q
29.midp M A B, midp M C D => para A C B D
30.midp M A B, para A C B D, para A D B C => midp M C D
31.eqratio O A A C O B B D, coll O A C, coll O B D, ncoll A B C, sameside A O C B O D => para A B C D
32.para A B A C => coll A B C

Rules

33. midp M A B, midp N C D => eqratio M A A B N C C D
34. eqangle A B P Q C D U V, perp P Q U V => perp A B C D
35. eqratio A B P Q C D U V, cong P Q U V => cong A B C D
36. cong A B P Q, cong B C Q R, cong C A R P, ncoll A B C => contri* A B C P Q R
37. cong A B P Q, cong B C Q R, eqangle6 B A B C Q P Q R, ncoll A B C => contri* A B C P Q R
38. eqangle6 B A B C Q P Q R, eqangle6 C A C B R P R Q, ncoll A B C => simtri A B C P Q R
39. eqangle6 B A B C Q R Q P, eqangle6 C A C B R Q R P, ncoll A B C => simtri2 A B C P Q R
40. eqangle6 B A B C Q P Q R, eqangle6 C A C B R P R Q, ncoll A B C, cong A B P Q => contri A B C P Q R
41. eqangle6 B A B C Q R Q P, eqangle6 C A C B R Q R P, ncoll A B C, cong A B P Q => contri2 A B C P Q R
42. eqratio6 B A B C Q P Q R, eqratio6 C A C B R P R Q, ncoll A B C => simtri* A B C P Q R
43. eqratio6 B A B C Q P Q R, eqangle6 B A B C Q P Q R, ncoll A B C => simtri* A B C P Q R
44. eqratio6 B A B C Q P Q R, eqratio6 C A C B R P R Q, ncoll A B C, cong A B P Q => contri* A B C P Q R
45. para a b c d, coll m a d, coll n b c, eqratio6 m a m d n b n c, sameside m a d n b c => para m n a b
46. para a b c d, coll m a d, coll n b c, para m n a b => eqratio6 m a m d n b n c

How Alphageometry Solves Problems

•**Deductive Reasoning (DD)**: The system starts by applying geometric rules to draw new conclusions. For example, it might deduce that two lines, AB and CD, are parallel.

•**Algebraic Reasoning (AR)**: The information deduced by DD, such as "AB is parallel to CD," is then translated into algebraic form. In this case, it means that the slopes of lines AB and CD will be set as equal in a matrix of equations managed by AR.

•**Algebraic Processing**: The system uses methods like Gaussian elimination to manipulate these equations and find new relationships between variables. These new relationships are then passed back to the deductive part (DD).

•**Repetitive Process**: This process of passing conclusions back and forth between DD and AR continues until the system can't make any more new deductions (when the "deduction closure" stops expanding).

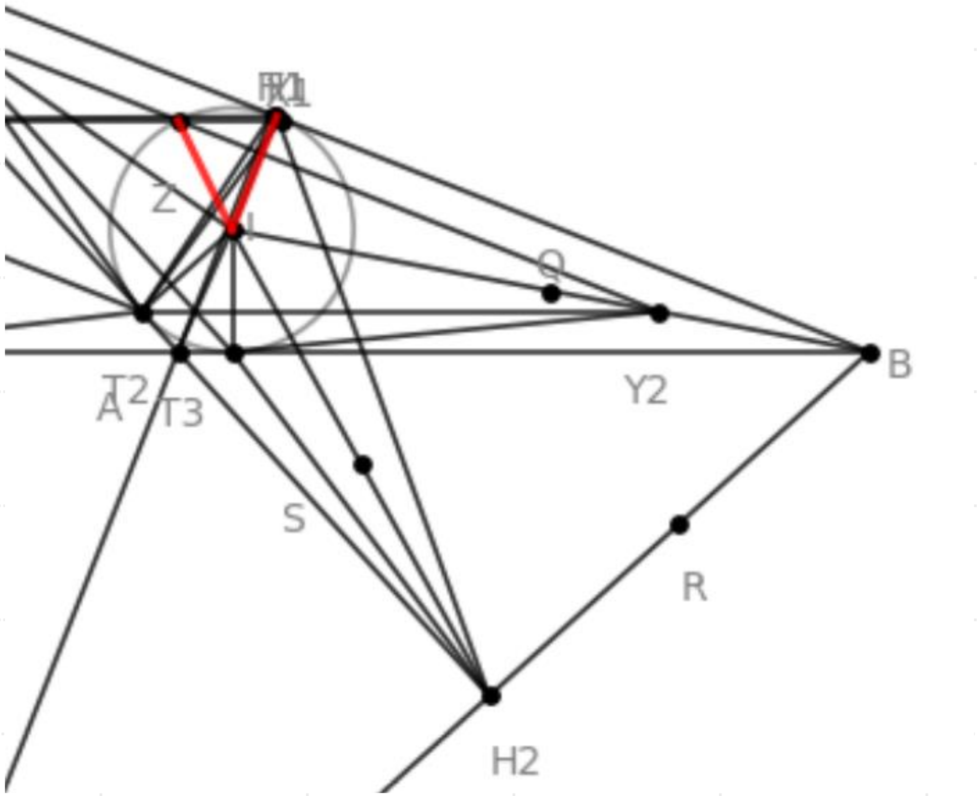
Our paper shows that language models can learn to come up with auxiliary constructions from synthetic data, in which problem statements and auxiliary constructions are randomly generated together and then separated using the traceback algorithm to identify the dependency difference.

Additional constructs used in solving IMO tasks

IMO 2006 P6

Construct point Q as the midpoint of BI.

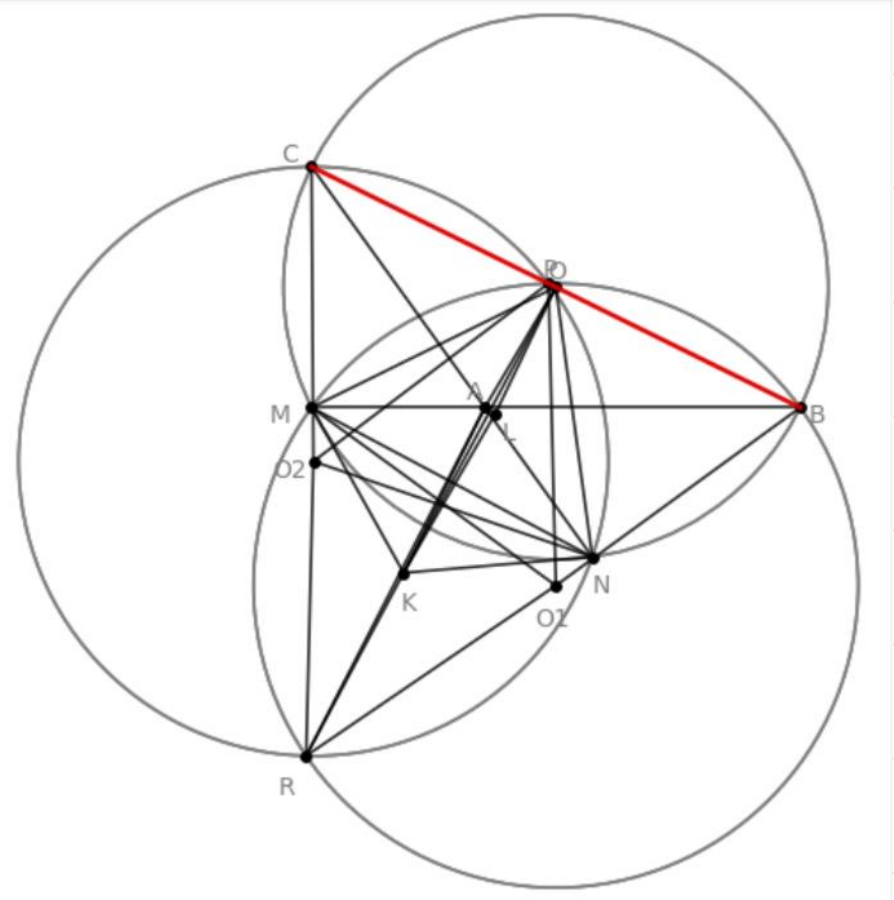
Construct point S as the midpoint of IH2.



IMO 2004 P1

Construct point K such that KM = KN.

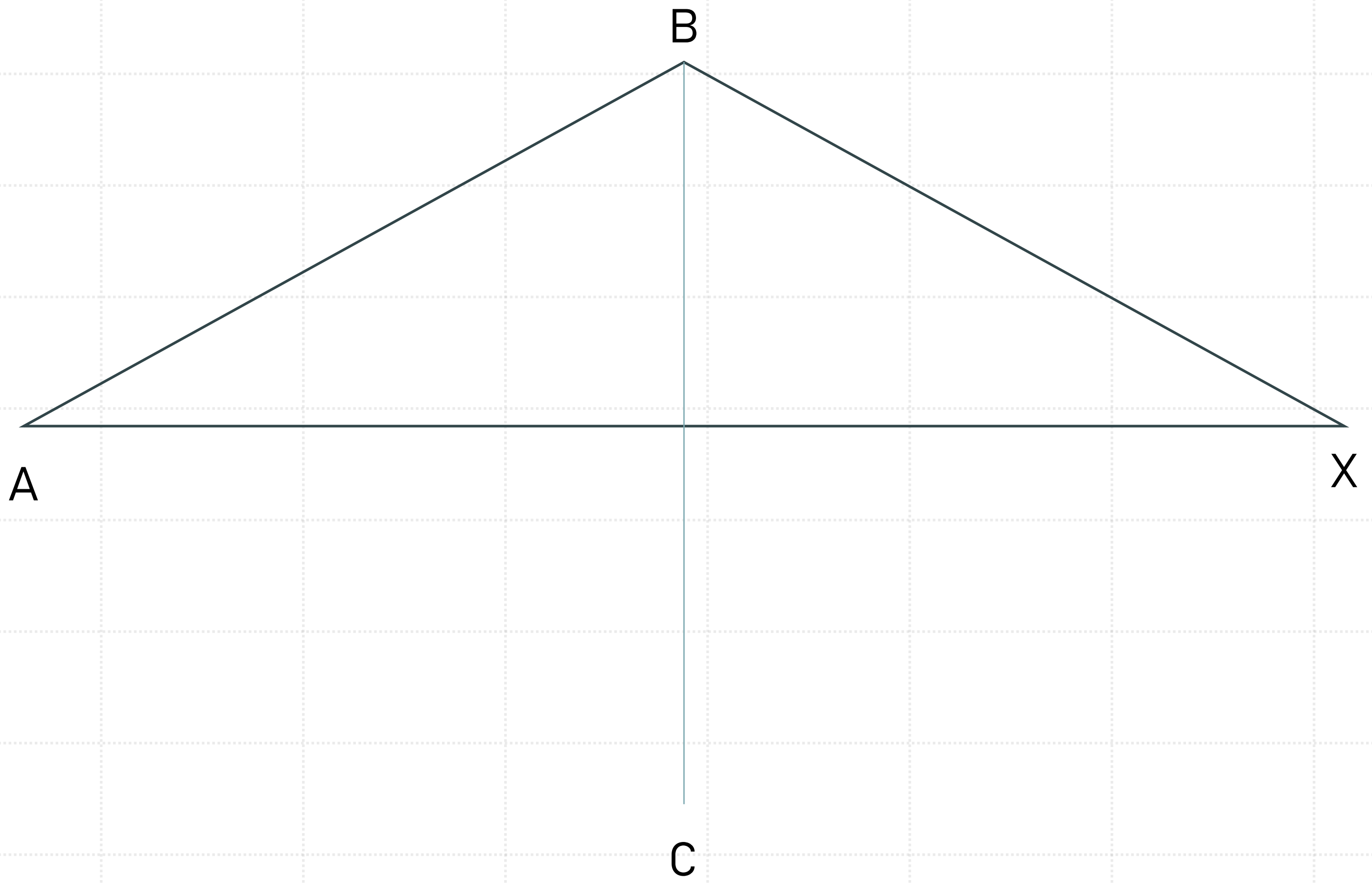
Construct point L as the intersection of circles (K, A) and (O, A).



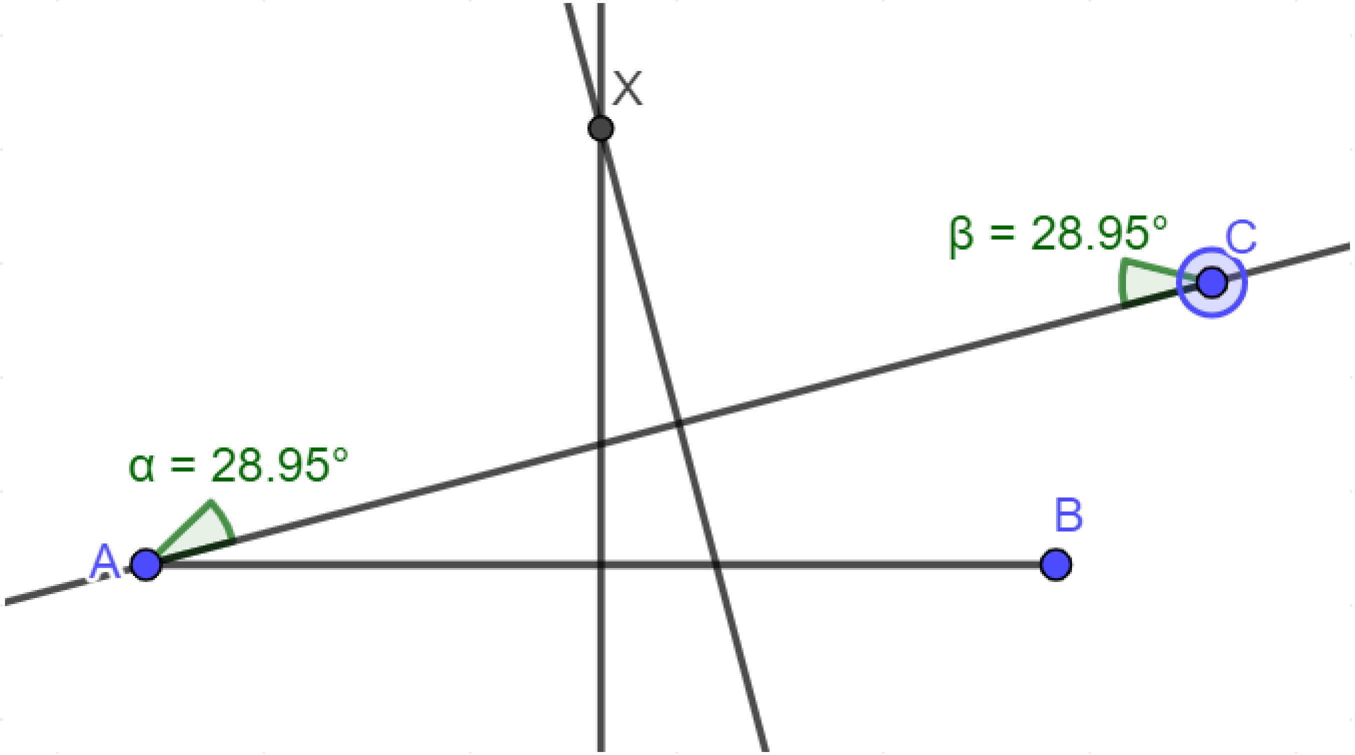
Additional constructs

- Construct point X as the midpoint of segment AB
- Construct point X as the mirror of A through B.
- Construct point X as the intersection of circles (O, A) and (O1, A).
- Construct point X as the circumcenter of triangle ABC.
- Construct point X as the orthocenter of triangle ABC.
- Construct point X such that XA = XB.
- Construct point X such that AB = BX and AX is perpendicular to BC.
- Construct point X such that AX = BX and <CAX = <XCA.
- Construct point X such that AB = CX and AX = CB.
- Construct point X such that AB is parallel to CX and AX is parallel to CD.
- Construct point X as the foot of A on line BC.

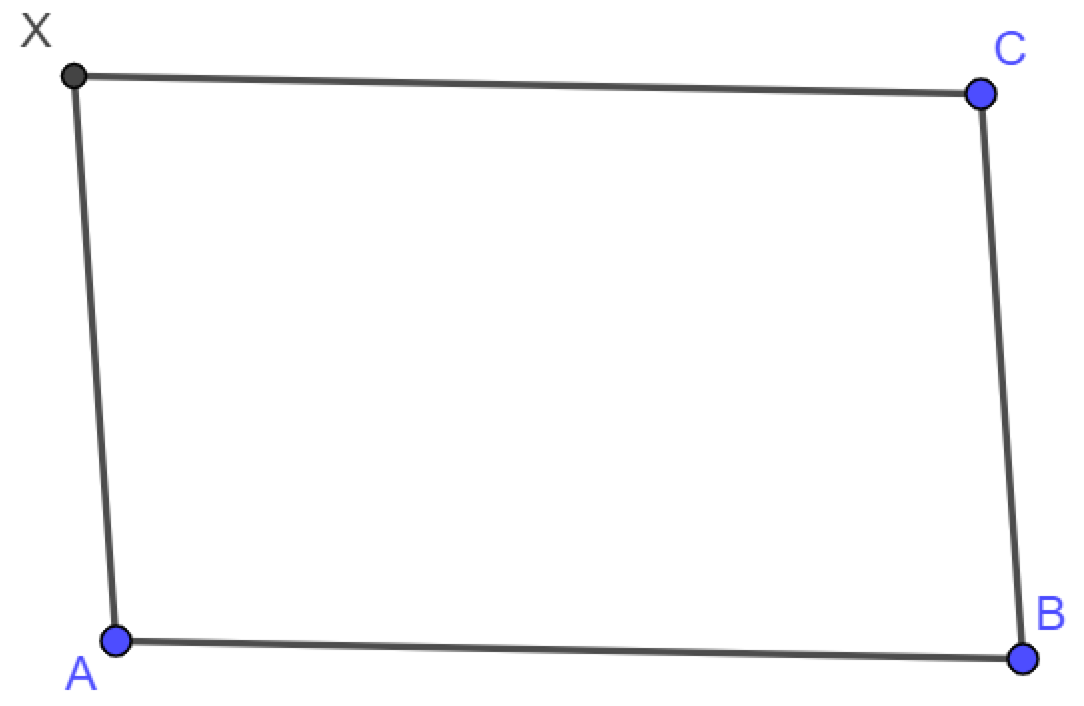
- Construct point X such that AB = BX and AX is perpendicular to BC.



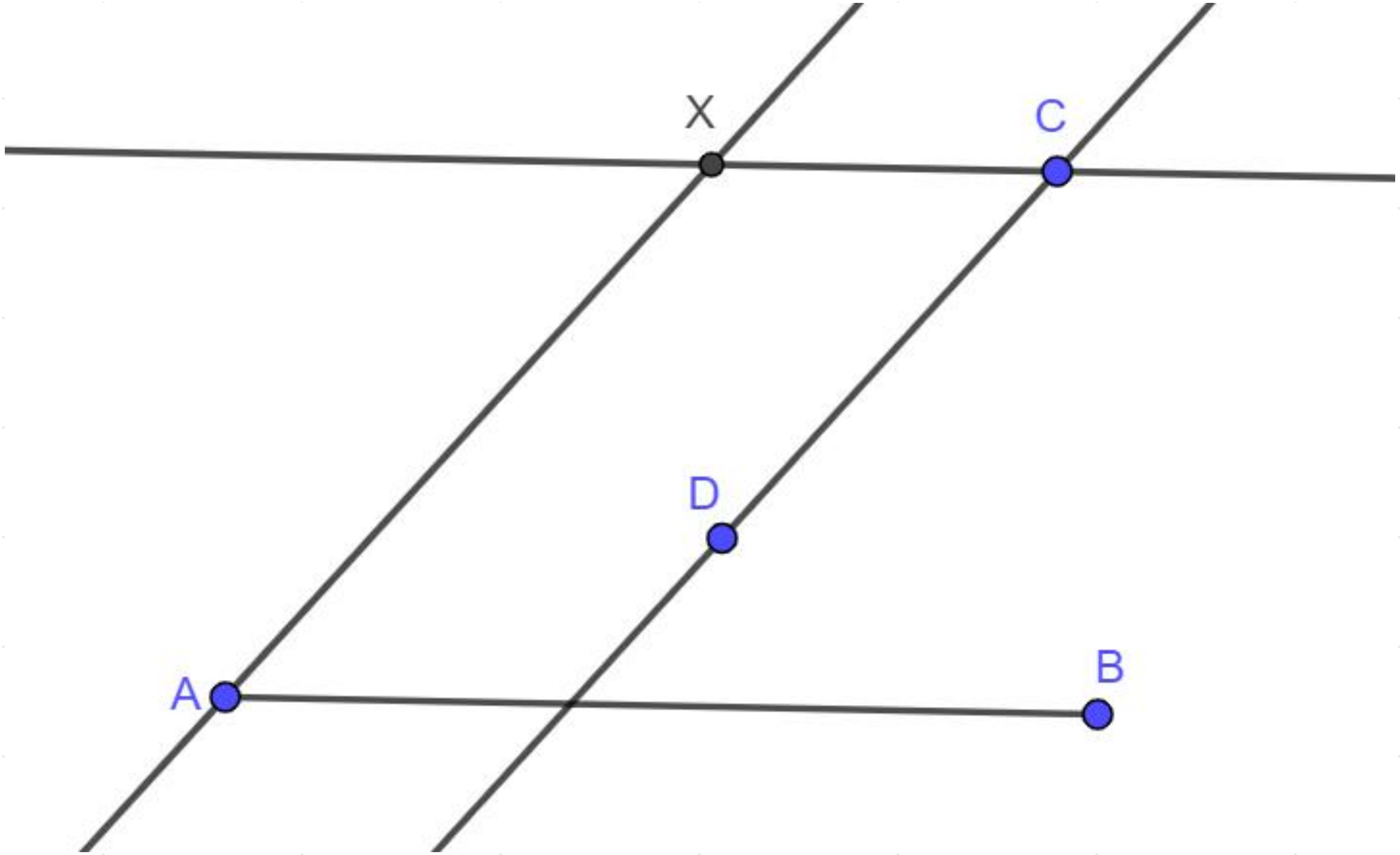
Construct point X such that AX = BX and <CAX = <XCA



Construct point X such that AB = CX and AX = CB



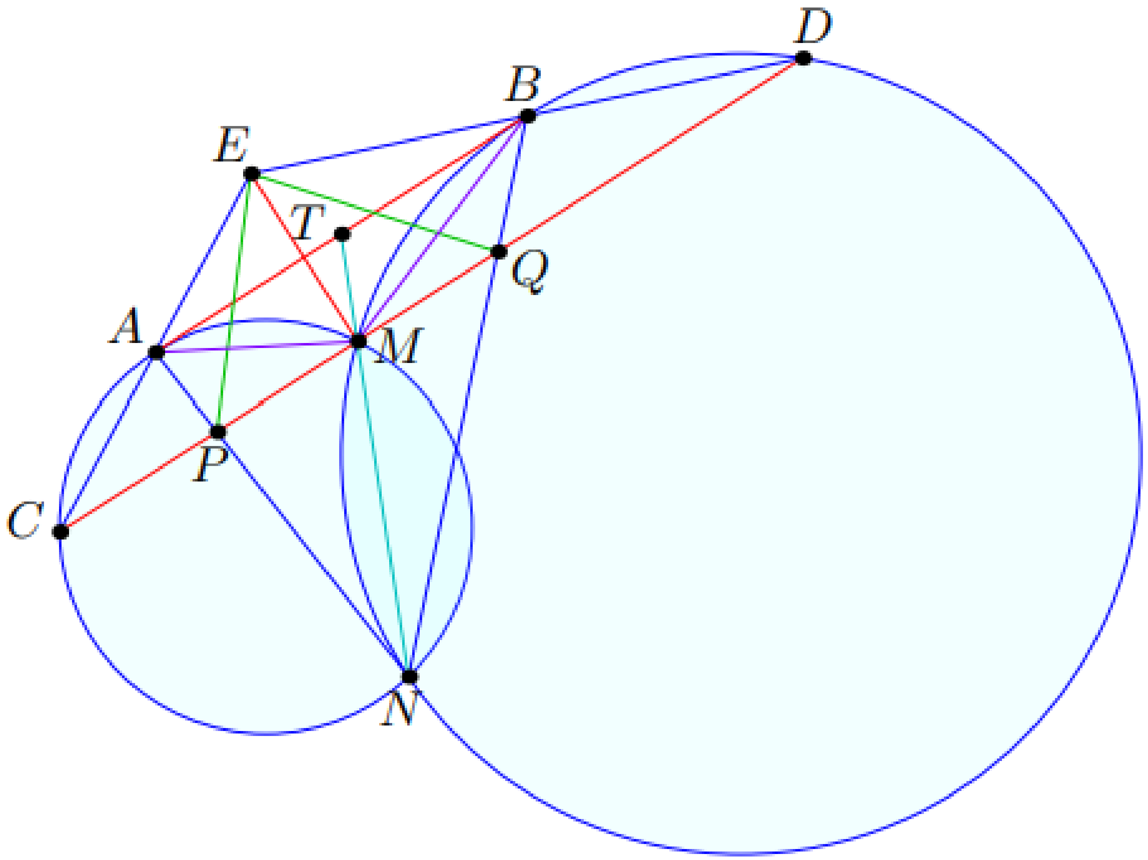
Construct point X such that AB is parallel to CX and AX is parallel to CD



Sample tasks: IMO 2000 P1

Original:

Two circles G1 and G2 intersect at two points M and N. Let AB be the line tangent to these circles at A and B, respectively, so that M lies closer to AB than N. Let CD be the line parallel to AB and passing through the point M, with C on G1 and D on G2. Lines AC and BD meet at E; lines AN and CD meet at P; lines BN and CD meet at Q. Show that EP = EQ.



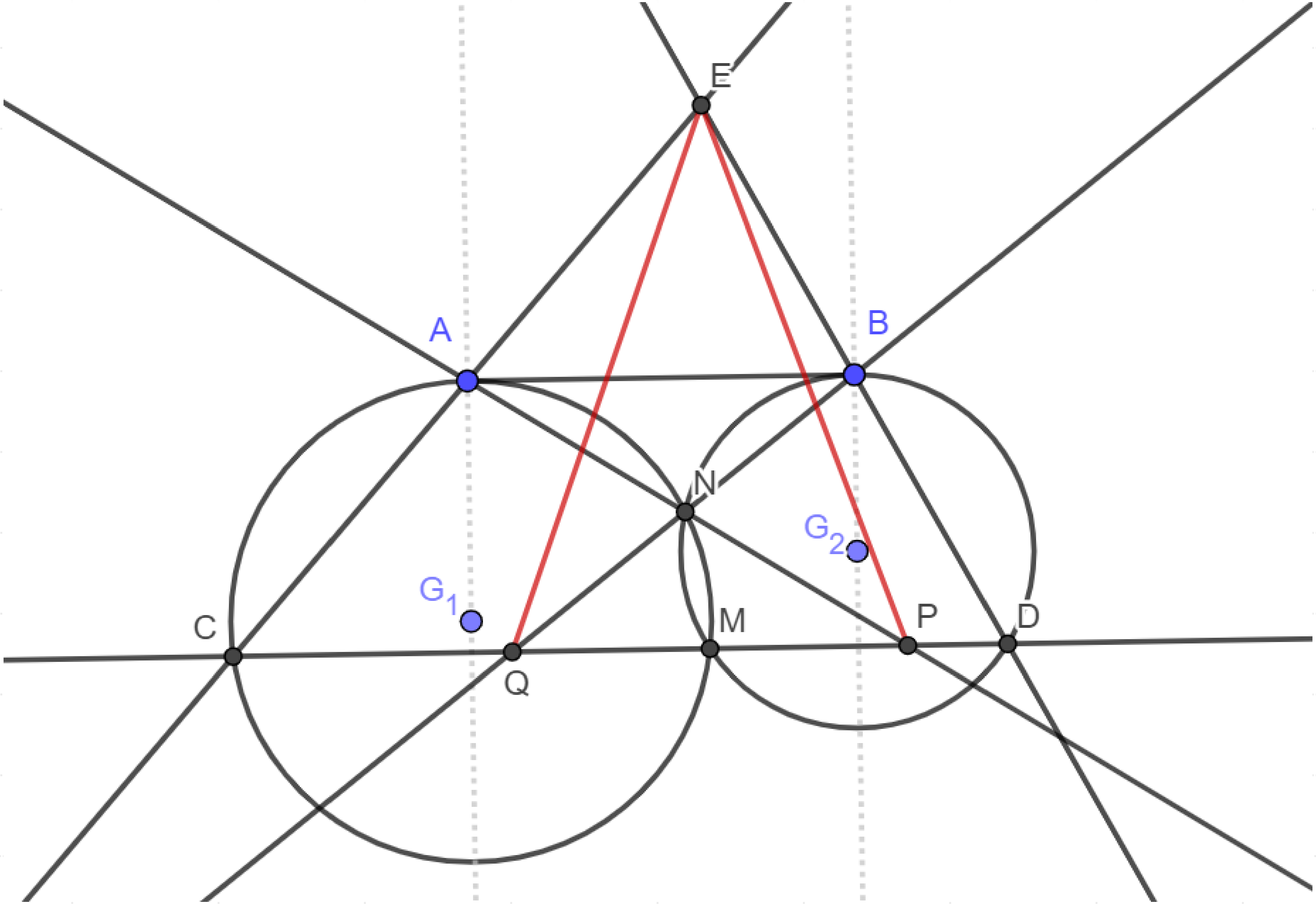
Sample tasks: IMO 2000 P1

Original:

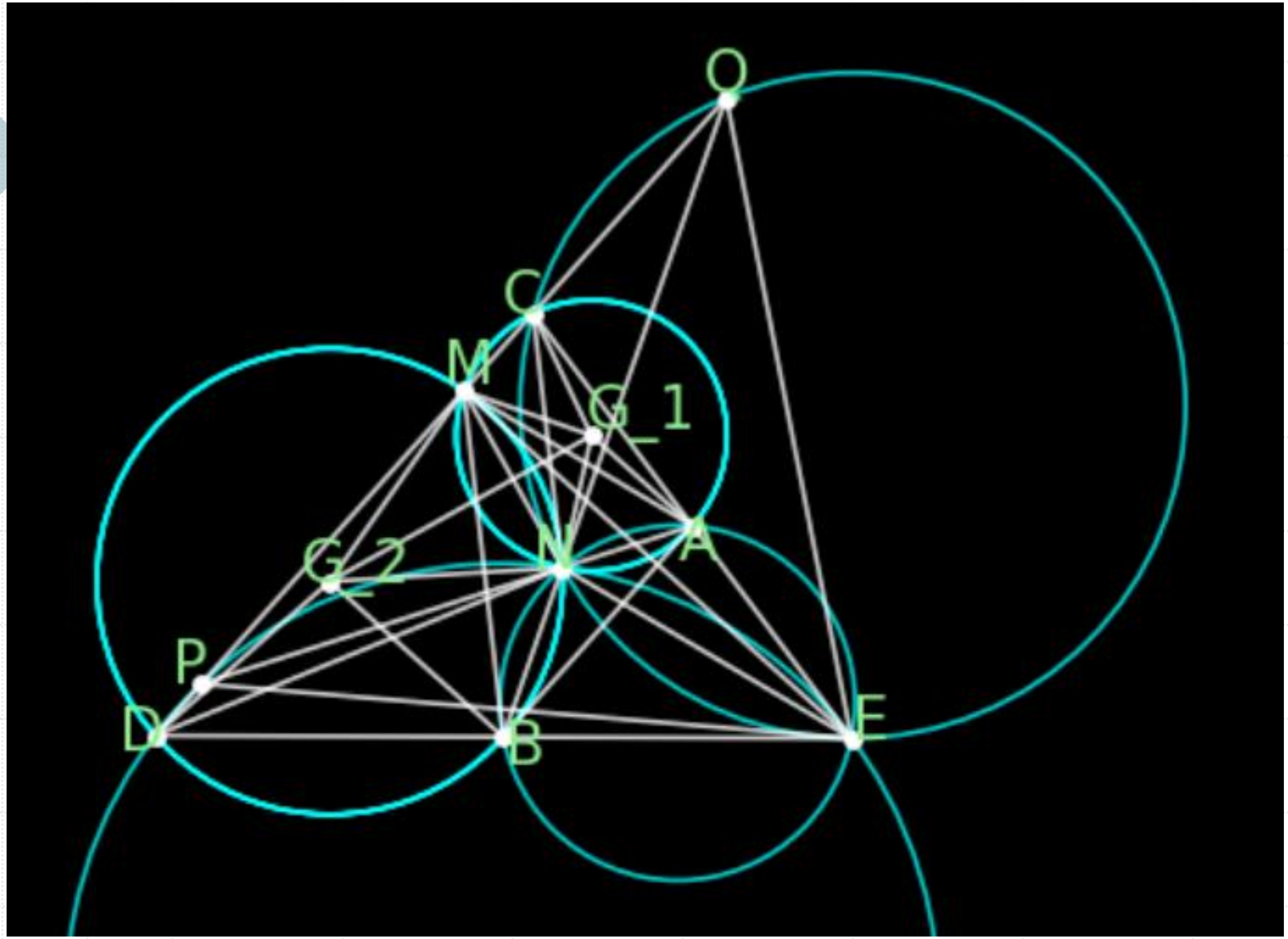
Two circles G1 and G2 intersect at two points M and N. Let AB be the line tangent to these circles at A and B, respectively, so that M lies closer to AB than N. Let CD be the line parallel to AB and passing through the point M, with C on G1 and D on G2. Lines AC and BD meet at E; lines AN and CD meet at P; lines BN and CD meet at Q. Show that EP = EQ.

Translated (in paper):

Let A and B be any two distinct points. Define point G1 such that AB is perpendicular to AG1. Define point G2 such that AB is perpendicular to BG2. Define point M as the intersection of circles (G1, A) and (G2, B). Define point N as the intersection of circles (G1, A) and (G2, B). Define point C on circle (G1, A) such that AB is parallel to CM. Define point D on circle (G2, B) such that AB is parallel to DM. Define point E as the intersection of lines AC and BD. Define point P as the intersection of lines AN and CD. Define point Q as the intersection of lines BN and CD. Prove that EP = EQ


E
A
B
N
G2
G1
C
Q
M
P
D

Sample tasks: IMO 2000 P1



Alphageometry:

a b = segment a b;
g1 = on_tline g1 a a b;
g2 = on_tline g2 b b a;
m = on_circle m g1 a, on_circle m g2 b;
n = on_circle n g1 a, on_circle n g2 b;
c = on_pline c m a b, on_circle c g1 a;
d = on_pline d m a b, on_circle d g2 b;
e = on_line e a c, on_line e b d;
p = on_line p a n, on_line p c d;
q = on_line q b n, on_line q c d
? cong e p e q

Sample tasks: IMO 2004 P5A

Original (in paper):

In a convex quadrilateral ABCD, the diagonal BD bisects neither the angle ABC nor the angle CDA. The point P lies inside ABCD and satisfies <PBC = <DBA and <PDC = <BDA. Prove that AP=CP given ABCD is a cyclic quadrilateral.

Original: Prove that ABCD is a cyclic quadrilateral if and only if AP = CP.

Translated:

Let ABC be a triangle. Define point O as the circumcenter of triangle CBA. Let D be any point on circle (O, A). Define point P such that <ABD = <PBC and <ADB = <PDC. Prove that AP = CP

Alphageometry:
a b c = triangle a b c;
o = circle o a b c;
d = on_circle d o a;
p = on_aline p b c a b d, on_aline p d c a d b
? cong a p c p

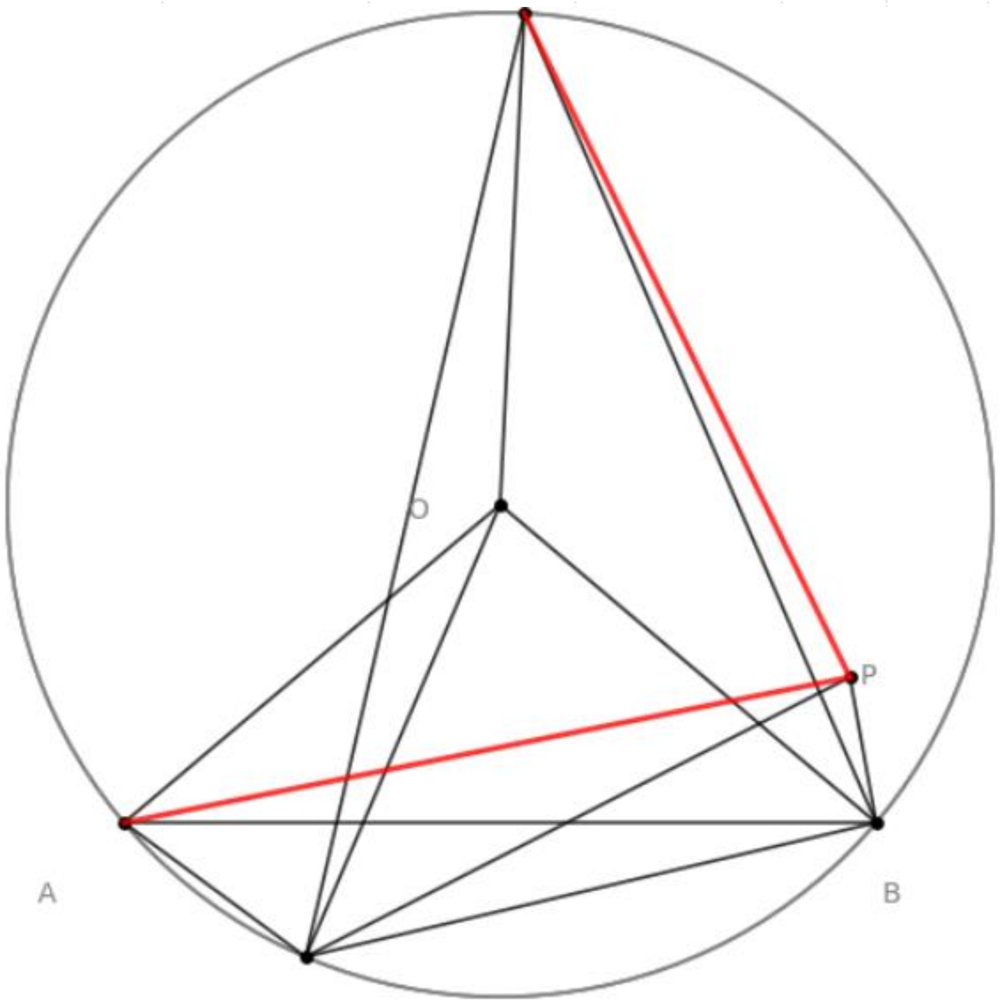


Figure by Alphageometry

Sample tasks: IMO 2004 P5A

Step 1. AO = BO, AO = DO and BO = CO ⇒ A, B, C, D are cyclic.

Step 2. A, B, C, D are cyclic ⇒ <BAD = <BCD and <BAC = <BDC.

Step 3. AO = BO, AO = DO and BO = CO ⇒ CO = DO.

Step 4. CO = DO ⇒ <CDO = <OCD.

Step 5. BO = CO ⇒ <BCO = <OBC.

Step 6. <BAD = <BCD, <ABD = <PBC, <BCO = <OBC, <ADB = <PDC and <CDO = <OCD ⇒ by angle chasing: <BOD = <BPD.

Step 7. <BOD = <BP D ⇒ B, D, O, P are cyclic.

Step 8. B, D, O, P are cyclic ⇒ <BDP = <BOP.

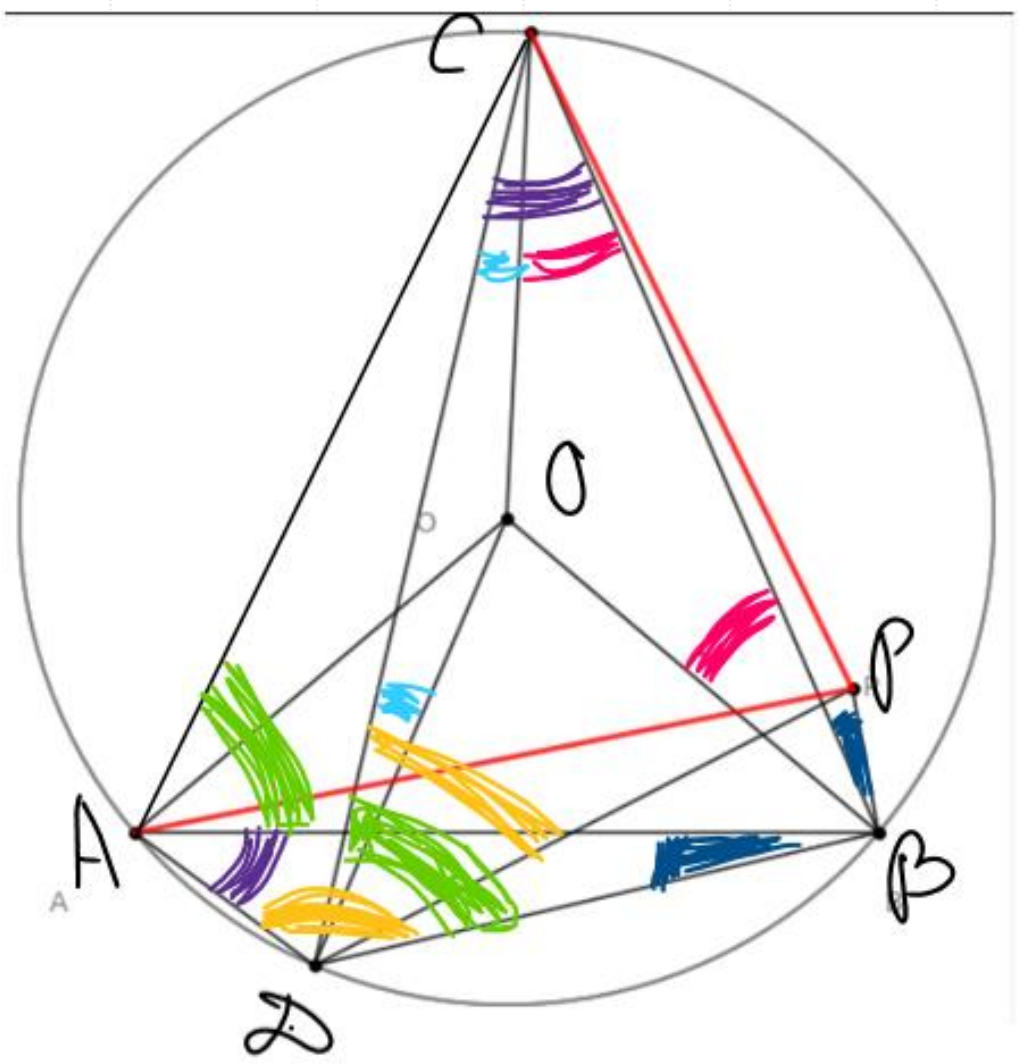
Step 9. AO = BO and BO = CO ⇒ AO = CO.

Step 10. AO = CO ⇒ <ACO = <OAC.

Step 11. AO = BO ⇒ <ABO = <OAB.

Step 12. <BAC = <BDC, <BAD = <BCD, <ABO = <OAB, <ACO = <OAC, <BCO = <OBC, <ADB = <PDC and <BDP = <BOP ⇒ by angle chasing: OP is the bisector of <AOC.

Step 13. AO = CO and OP is the bisector of <AOC ⇒ AP = CP



https://www.cut-the-knot.org/pythagoras/IMO2004-5.shtml

My tasks

1. a b c = triangle a b c; d = on_pline d c b a, on_pline d b c a ? equangle a b c c d a

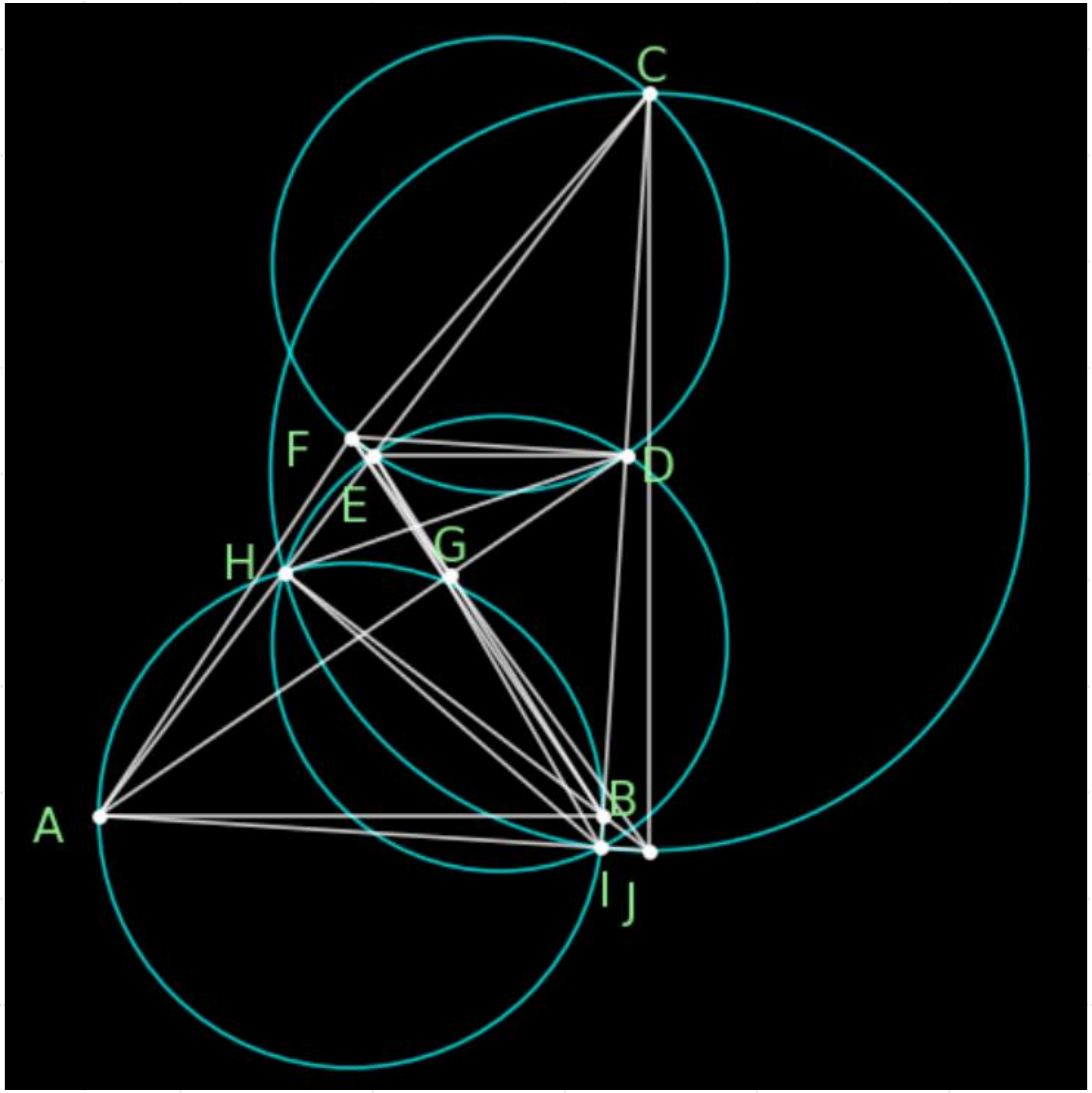
2. a b = segment; c = on_tline c a b a; d = on_tline d b b a, on_tline d c b a ? eqdistance d a c b

3. a b c = triangle a b c; d = midpoint d a b; e = foot e b c d; f = foot f a c d ? eqdistance a f e b

4. c b a = triangle c b a; d = parallelogram c b a d; o = circumcenter o a d b; l = intersection_lc l c o d; k = intersection_lc k b o c; n = mirror n a o ? eqdistance n k c n

https://www.tandfonline.com/eprint/MKJWRQFEGGEA6HN3RRWB/full?target=10.1080/0020739X.2024.2377724#d1e288

My task



a b c = triangle a b c;
d = midpoint d c b;
e = midpoint e c a;
f = on_bline f a c, on_bline f c b;
g = intersection_ll g a d b e;
h = foot h b a c; i = foot i a b c;
j = intersection_ll j a i b h
? coll j g f

Citations from the article

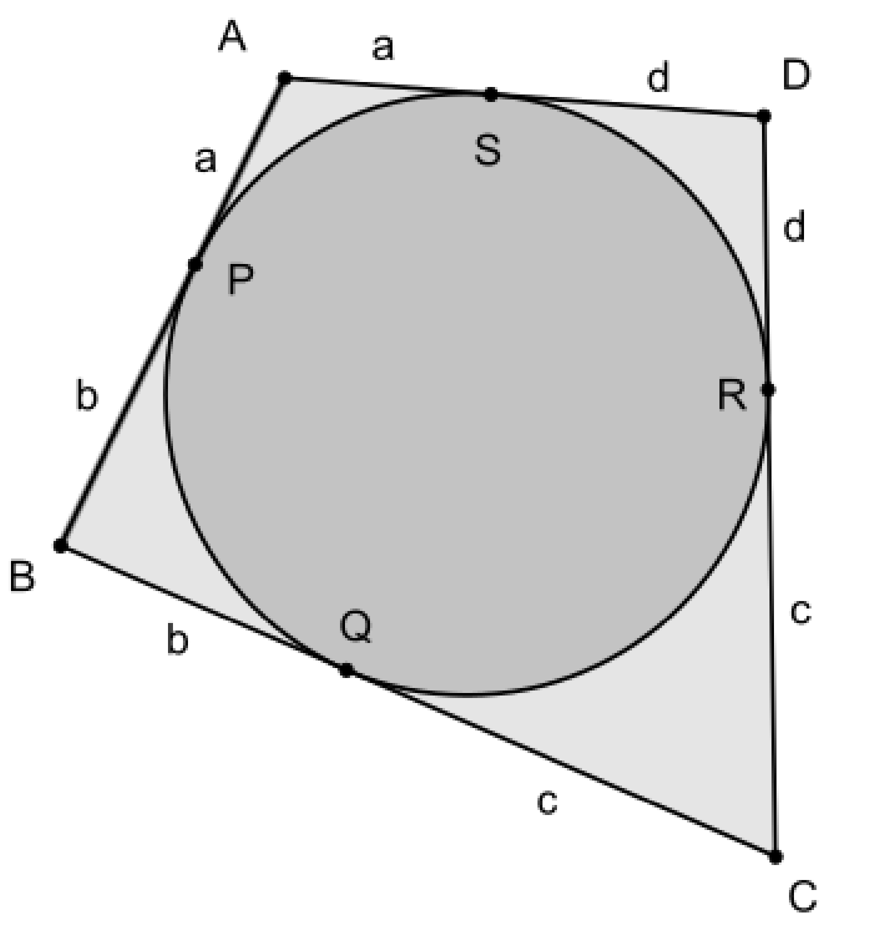
This auxiliary construction can be found quickly with the knowledge of Reim's theorem, which is not included in the deduction rule list used by the symbolic engine during synthetic data generation. Including such high-level theorems into the synthetic data generation can greatly improve the coverage of synthetic data and thus improve auxiliary construction capability. Further, higher-level steps using Reim's theorem also cut down the current proof length by a factor of 3.

AlphaGeometry constructs point K to materialize this axis, whereas humans simply use the existing point R for the same purpose. This is a case in which proof pruning itself cannot remove K and a sign of similar redundancy in our synthetic data.

This human proof uses four auxiliary constructions (diameters of circles W1 and W2) and high-level theorems such as the Pitot theorem and the notion of homothety. These high-level concepts are not available to our current version of the symbolic deduction engine both during synthetic data generation and proof search. Again, this suggests that enhancing the symbolic engine with more powerful tools that IMO contestants are trained to use can improve both the synthetic data and the test-time performance of AlphaGeometry.

Pitot theorem

The Pitot theorem in geometry states that in a tangential quadrilateral the two pairs of opposite sides have the same total length.



$$
\begin{aligned}
&|AB| + |CD| \\
=&(a+b)+(c+d) \\
=&(b+c)+(a+d) \\
=&|BC|+|DA|
\end{aligned}
$$

https://en.wikipedia.org/wiki/Pitot_theorem

(Reim's Theorem). Choose points A, B, X, Y on circle $\omega 1$ and let C and D be points on AX and BY . Then AB || CD if X, Y, C, D are concyclic.

AlphaGeometry excels at solving problems involving cyclic quadrilaterals, with cyclic points appearing in 24 out of 25 solved tasks.

This suggests that the system is particularly strong when working with such geometric structures. However, this raises the possibility that the tasks were selected to play to the system's strengths. It's conceivable that a different set of problems, without reliance on cyclic figures, could be more challenging for AlphaGeometry but potentially solvable by other provers.