Theorem proving components in GeoGebra

Francisco Botana
University of Vigo, Spain
http://fbotana.webs.uvigo.es

MTM2014-54141-P

ThEdu'17 – Theorem Proving components for Educational Software

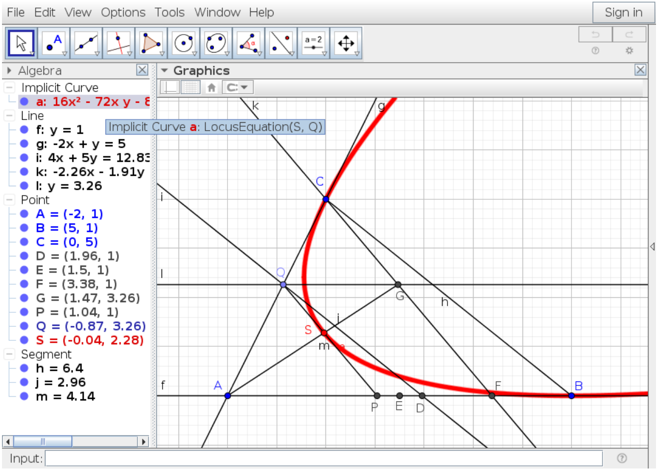
Summary

- GeoGebra
- Automated Reasoning Tools (ART) in GeoGebra
    - Deriving relations
    - Proving
    - Discovering

What is GeoGebra?

- Hohenwarter, M. (2002). "Ein Softwaresystem für dynamische Geometrie und Algebra der Ebene". Master's thesis. Salzburg University.
- GeoGebra is dynamic mathematics software for all levels of education that brings together geometry, algebra, spreadsheets, graphing, statistics and calculus in one easy-to-use package.
- In 2013, Bernard Parisse's Giac was integrated into GeoGebra's CAS view.
- GeoGebra is a rapidly expanding community of about forty millions users, located in just about every country. Available in many languages. GeoGebra Materials: 1 million resources (April 2016)
- Desktop, web, tablet, smartphone versions.
- Open source software freely available for non-commercial users
- `https://en.wikipedia.org/wiki/GeoGebra`

GeoGebra


File Edit View Options Tools Window Help
Sign in
Algebra
Graphics
Implicit Curve
a: 16x² - 72x y - 8
Line
f: y = 1
g: -2x + y = 5
i: 4x + 5y = 12.83
k: -2.26x - 1.91y
l: y = 3.26
Point
A = (-2, 1)
B = (5, 1)
C = (0, 5)
D = (1.96, 1)
E = (1.5, 1)
F = (3.38, 1)
G = (1.47, 3.26)
P = (1.04, 1)
Q = (-0.87, 3.26)
S = (-0.04, 2.28)
Segment
h = 6.4
j = 2.96
m = 4.14
Implicit Curve a: LocusEquation(S, Q)
Input:

GeoGebra ART

GeoGebra **ART**: Automated Reasoning Tools

- Automated derivation
- Automated proving
- Automated discovery
- Locus: mover-tracer, boolean, envelopes, etc.

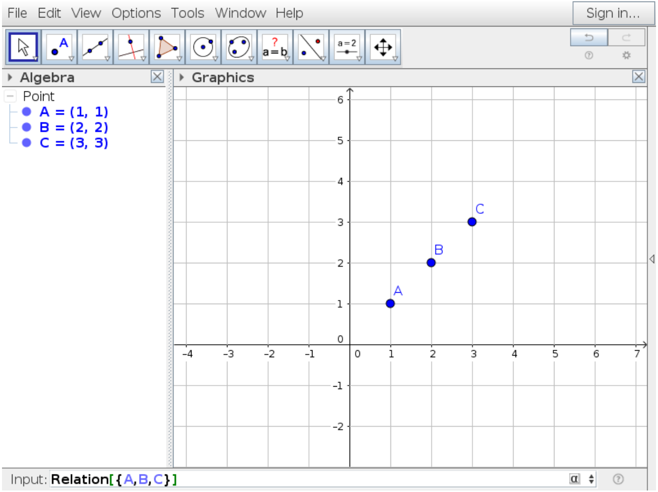
T. Recio, University of Cantabria; Z. Kovács, The Private University College of Education of the Diocese of Linz; F.B.

Some references

- https://www.researchgate.net/project/Theorem-proving-tools-in-GeoGebra
- Abánades, M.; Botana, F.; Montes, A.; Recio, T.: "An algebraic taxonomy for locus computation in dynamic geometry". CAD 56(2014) 22-33.
- Botana, F.; Hohenwarter, M.; Janicic, J.; Kovács, Z.; Recio, T.; Petrovic, I.; Weitzhofer, S.: "Automated Theorem Proving in GeoGebra: Current Achievements". JAR 55(2015) 39-59.
- Kovács Z.: Computer Based Conjectures and Proofs in Teaching Euclidean Geometry, Ph. Dissertation. JKU, 2015.
- Abánades, M.; Botana, F.; Kovács, Z.; Recio, T.; Solyom-Gecse, C.: "Development of automatic reasoning tools in GeoGebra". ACM Comm Comput Alg 50(2016) 85-88.
- Botana, F.; Recio, T.; Vélez, M. P.: "The role of automated reasoning of geometry statements in mathematics instruction". Poster at CERME 10, Dublin, February, 2017.

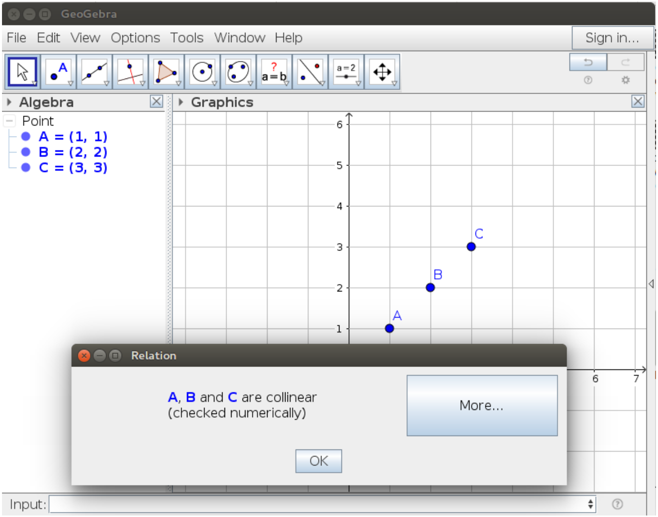
Automated derivation

aka *property checker*



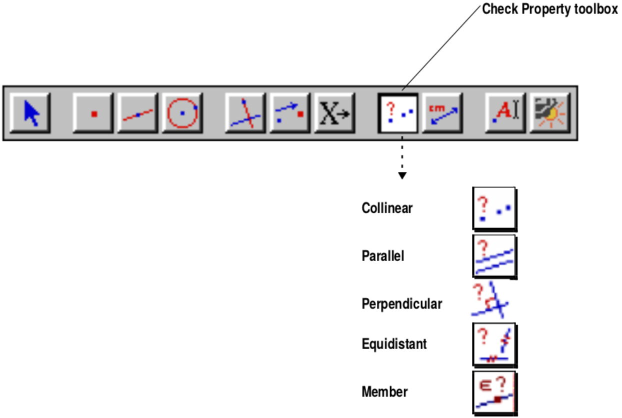
Automated derivation

aka *property checker*



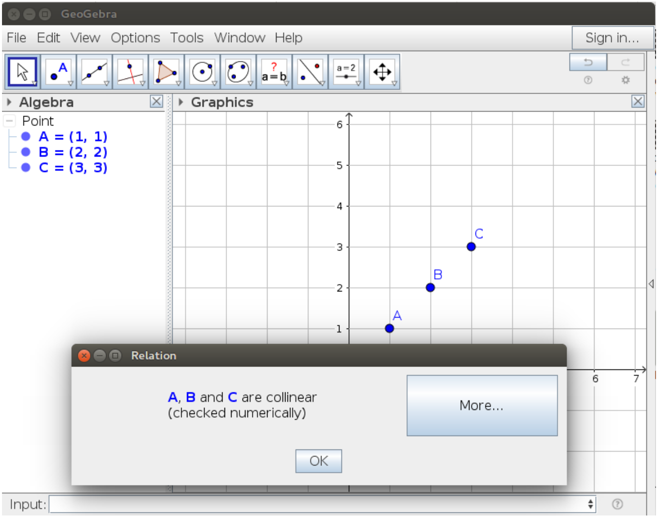
Automated derivation

aka *property checker*



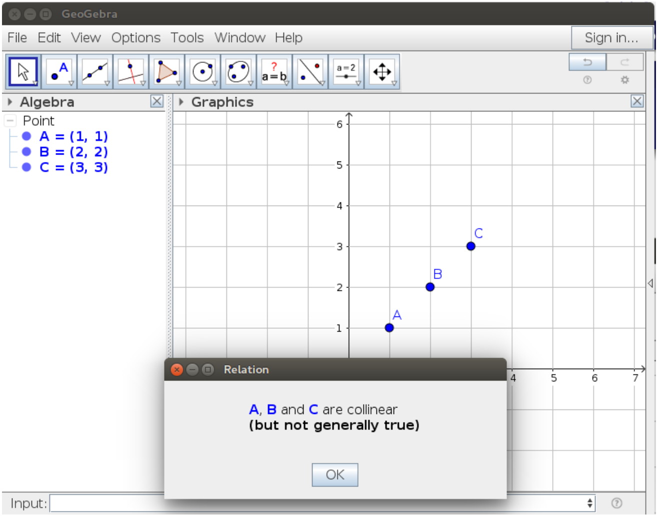
Automated derivation

aka *property checker*



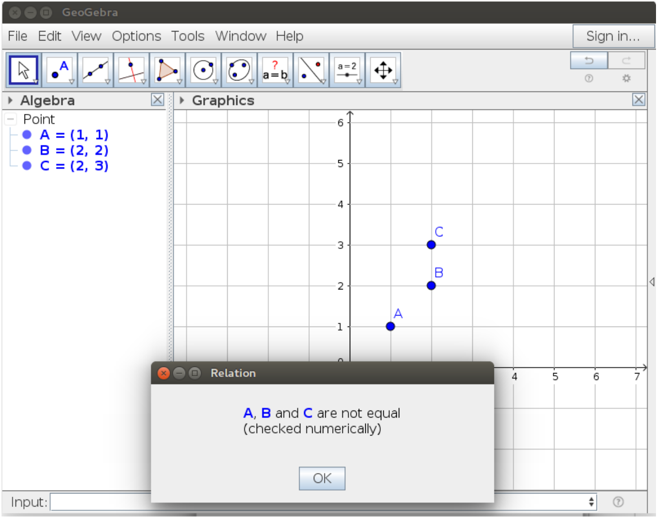
Automated derivation

aka *property checker*



Automated derivation

aka *property checker*



The limited expression power of Relation

https://wiki.geogebra.org/en/Relation_Command

GeoGebra + Materials Down

Relation Command

Relation[<List>]

Shows a message box that gives you information about the relation between two or more (up to 4) objects.

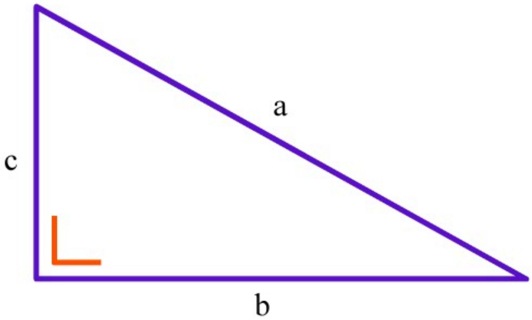
Relation[<Object>, <Object>]

Shows a message box that gives you information about the relation between two objects.

This command allows you to find out whether

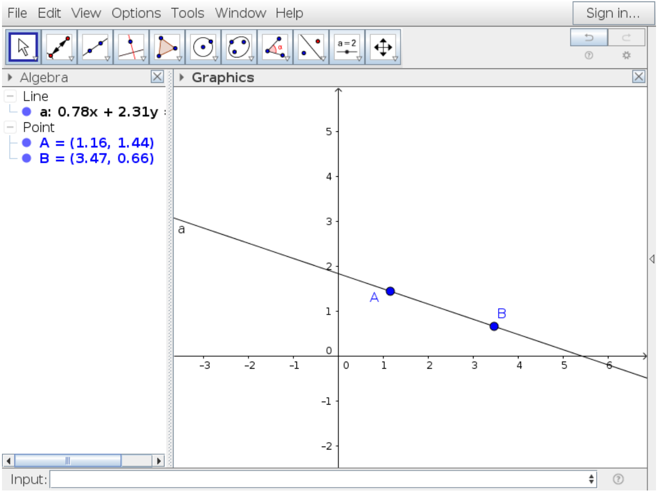
- two lines are perpendicular
- two lines are parallel
- two (or more) objects are equal
- a point lies on a line or conic
- a line is tangent or a passing line to a conic
- three points are collinear
- three lines are concurrent (or parallel)
- four points are concyclic (or collinear).

Automated proving:
Pythagorean theorem through Relation?

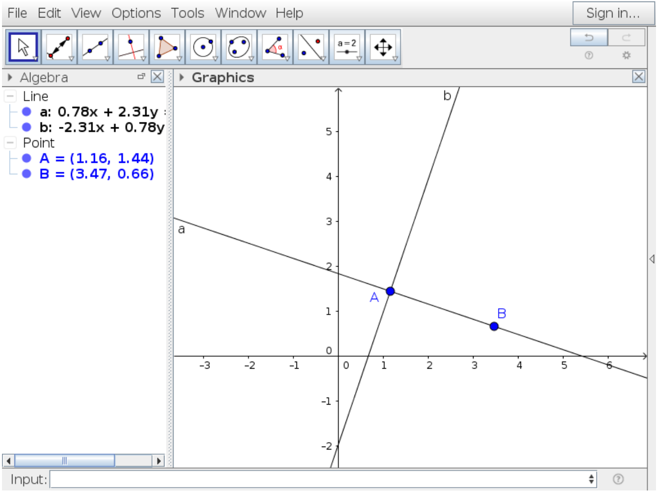


$$a^2 = b^2 + c^2$$

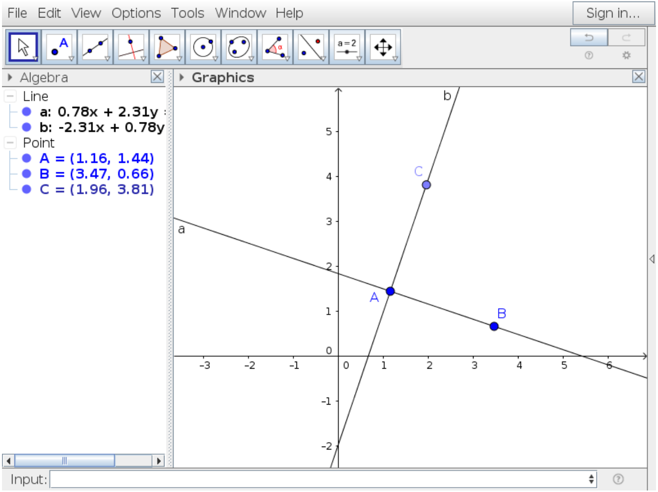
Automated proving: Pythagorean theorem



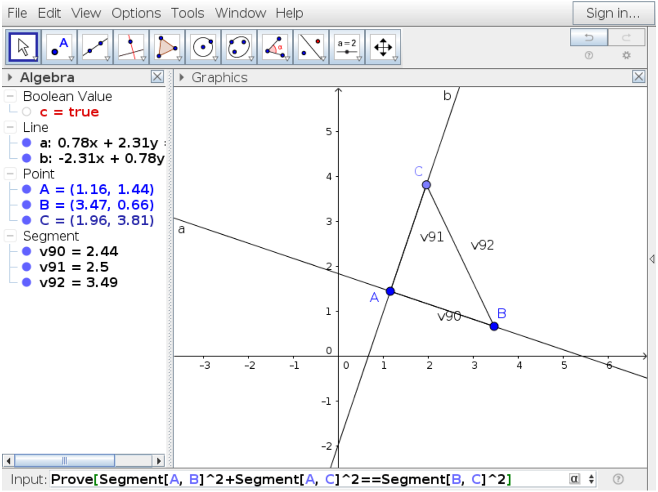
Automated proving: Pythagorean theorem



Automated proving: Pythagorean theorem



Automated proving: Pythagorean theorem



Automated proving: Behind the curtain

Construction Protocol - Pythagoras.ggb

No.	Name	Definition
1	Point A	
2	Point B	
3	Line a	Line through A, B
4	Line b	Line through A perpendicular to a
5	Point C	Point on b
6	Boolean Value c	Prove[Segment[A, B]² + Segment[A, C]² ≟ Segment[B, C]²]

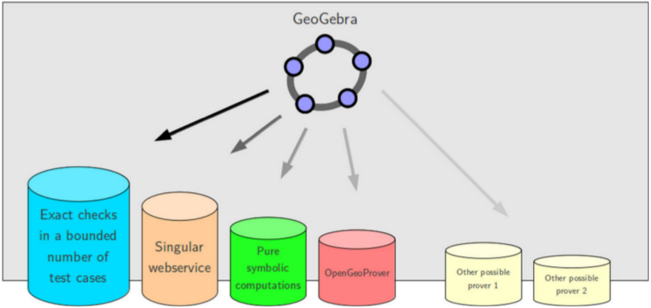
Automated proving: Behind the curtain

```
l_10 [style=dashed]; v92_4 -> null_10 [style=dashed]; a_1 -> b_2; v90_1 -> null_
10 [style=dashed]; B_0 -> v90_1; A_0 -> a_1; A_0 -> b_2; b_2 -> C_3; B_0 -> a_1;
 A_0 -> v90_1; C_3 -> v91_4; A_0 -> v91_4; C_3 -> v92_4; }
DEBUG: Using AUTO
DEBUG: Using RECIOS_PROVER
DEBUG: Using BOTANAS_PROVER
DEBUG: A = (1.16, 1.44) /* free point */
DEBUG: // Free point A(v55,v56)
DEBUG: B = (3.47, 0.66) /* free point */
DEBUG: // Free point B(v57,v58)
DEBUG: a = Line[A, B] /* Line through A, B */
DEBUG: b = OrthogonalLine[A, a] /* Line through A perpendicular to a */
DEBUG: Hypotheses:
DEBUG: 1. -1*v60+v57+v56+-1*v55
DEBUG: 2. -1*v59+-1*v58+v56+v55
DEBUG: C = Point[b] /* Point on b */
DEBUG: // Constrained point C(v61,v62)
DEBUG: Hypotheses:
DEBUG: 3. 1*v61*v60+-1*v62*v59+-1*v61*v56+1*v59*v56+1*v62*v55+-1*v60*v55
DEBUG: v90 = Segment[A, B] /* Segment [A, B] */
DEBUG: v91 = Segment[A, C] /* Segment [A, C] */
DEBUG: v92 = Segment[B, C] /* Segment [B, C] */
DEBUG: Processing numerical object
DEBUG: Hypotheses have been processed.
DEBUG: giac evalRaw input: evalfa(expand(((ggbtmpvarv90)^(2))+((ggbtmpvarv91)^(2
))))
DEBUG: giac evalRaw output: ggbtmpvarv90^2+ggbtmpvarv91^2
DEBUG: input = expand(((ggbtmpvarv90)^(2))+((ggbtmpvarv91)^(2)))
DEBUG: result = ggbtmpvarv90^2+ggbtmpvarv91^2
DEBUG: eliminate([((ggbtmpvarv90)^(2))+((ggbtmpvarv91)^(2))-((ggbtmpvarv92)^(2))
=0,ggbtmpvarv91^2=v93^2,ggbtmpvarv90^2=v94^2,ggbtmpvarv92^2=v95^2],[ggbtmpvarv91
,ggbtmpvarv90,ggbtmpvarv92])
DEBUG: giac evalRaw input: evalfa(eliminate([((ggbtmpvarv90)^(2))+((ggbtmpvarv91
)^(2))-((ggbtmpvarv92)^(2))=0,ggbtmpvarv91^2=v93^2,ggbtmpvarv90^2=v94^2,ggbtmpva
rv92^2=v95^2],[ggbtmpvarv91,ggbtmpvarv90,ggbtmpvarv92]))
Running a probabilistic check for the reconstructed Groebner basis. If successfu
ll, error probability is less than 1e-07 and is estimated to be less than 10^-18
. Use proba_epsilon:=0 to certify (this takes more time).
// Groebner basis computation time 0.000311 Memory -1e-06M
```

Automated proving: Behind the curtain

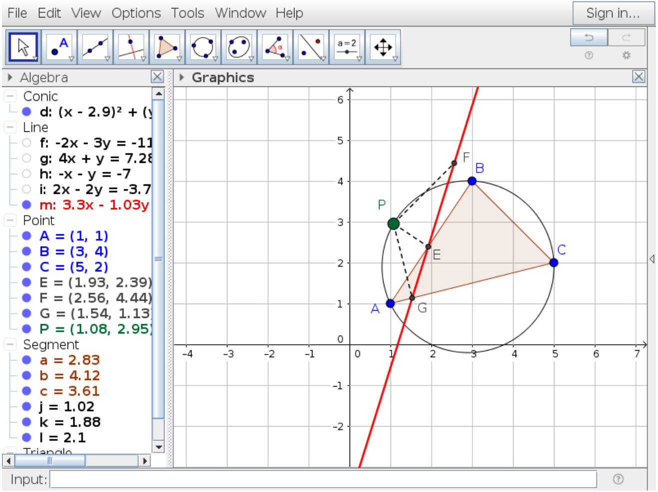
```
ll, error probability is less than 1e-07 and is estimated to be less than 10^-18
. Use proba_epsilon:=0 to certify (this takes more time).
// Groebner basis computation time 0.00039 Memory -1e-06M
DEBUG: giac evalRaw output: {v94^2+v93^2-v95^2}
DEBUG: input = eliminate([((ggbtmpvarv90)^(2))+((ggbtmpvarv91)^(2))-((ggbtmpvarv
92)^(2))=0,ggbtmpvarv91=v93,ggbtmpvarv90=v94,ggbtmpvarv92=v95],[ggbtmpvarv91,ggb
tmpvarv90,ggbtmpvarv92])
DEBUG: result = {v94^2+v93^2-v95^2}
DEBUG: giac evalRaw input: evalfa(simplify({v93^2+v94^2-v95^2}/{v94^2+v93^2-v95^
2}))
DEBUG: giac evalRaw output: {1}
DEBUG: input = simplify({v93^2+v94^2-v95^2}/{v94^2+v93^2-v95^2})
DEBUG: result = {1}
DEBUG: giac evalRaw input: evalfa(factor(1))
DEBUG: giac evalRaw output: 1
DEBUG: input = factor(1)
DEBUG: result = 1
DEBUG: Thesis equations (non-denied ones):
DEBUG: 4. 1*v93^2+-1*v62^2+-1*v61^2+2*v62*v56+-1*v56^2+2*v61*v55+-1*v55^2
DEBUG: 5. 1*v94^2+-1*v58^2+-1*v57^2+2*v58*v56+-1*v56^2+2*v57*v55+-1*v55^2
DEBUG: 6. 1*v95^2+-1*v62^2+-1*v61^2+2*v62*v58+-1*v58^2+2*v61*v57+-1*v57^2
DEBUG: Thesis reductio ad absurdum (denied statement), product of factors:
DEBUG: (-1*v95^2+1*v94^2+1*v93^2)*v97-1
DEBUG: that is,
DEBUG: 7. -1+-1*v97*v95^2+1*v97*v94^2+1*v97*v93^2
DEBUG: substitutions: {v55=0, v56=0}
TRACE: ring r=(0,v55,v58,v57,v56,v61),(v97,v59,v93,v95,v94,v62,v60),dp;ideal i=-
1+-1*v97*v95^2+1*v97*v94^2+1*v97*v93^2,1*v61*v60+-1*v62*v59+-1*v61*v56+1*v59*v56
+1*v62*v55+-1*v60*v55,-1*v59+-1*v58+v56+v55,-1*v60+v57+v56+-1*v55,1*v95^2+-1*v62
^2+-1*v61^2+2*v62*v58+-1*v58^2+2*v61*v57+-1*v57^2,1*v94^2+-1*v58^2+-1*v57^2+2*v5
8*v56+-1*v56^2+2*v57*v55+-1*v55^2,1*v93^2+-1*v62^2+-1*v61^2+2*v62*v56+-1*v56^2+2
*v61*v55+-1*v55^2;i=subst(i,v55,0,v56,0);groebner(i)!=1; -> singular
DEBUG: Waiting for the prover: 1
DEBUG: Waiting for the prover: 2
TRACE: singular -> 0
DEBUG: Statement is GENERALLY TRUE
DEBUG: Benchmarking: 102 ms
DEBUG: STATEMENT IS TRUE
DEBUG: OUTPUT for Prove: null = true
```

Automated proving: Provers and tests



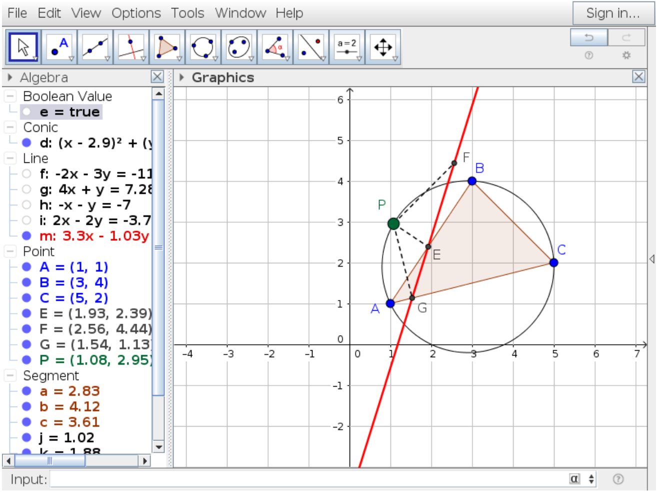
http://dev.geogebra.org/trac/browser/trunk/geogebra/test/scripts/benchmark/prover/tests

Automated proving: Wallace-Simson theorem



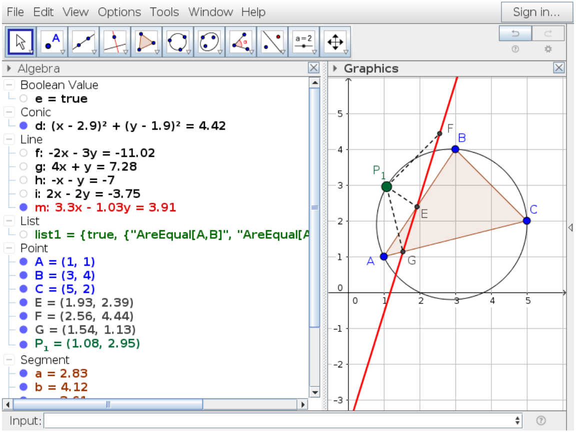
Automated proving: Wallace-Simson theorem

Prove[AreCollinear[E, F, G]]



Automated proving: Wallace-Simson details

ProveDetails[AreCollinear[E, F, G]]

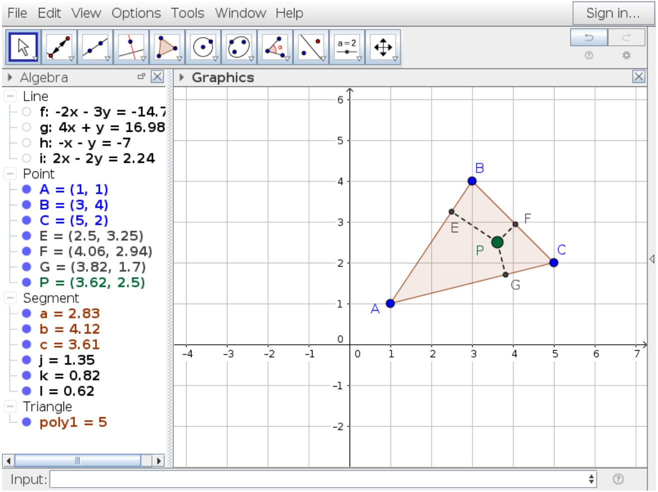


list1 = { true,

{“AreEqual[A,B]”,“AreEqual[A,C]”,“AreEqual[B,C]” }}

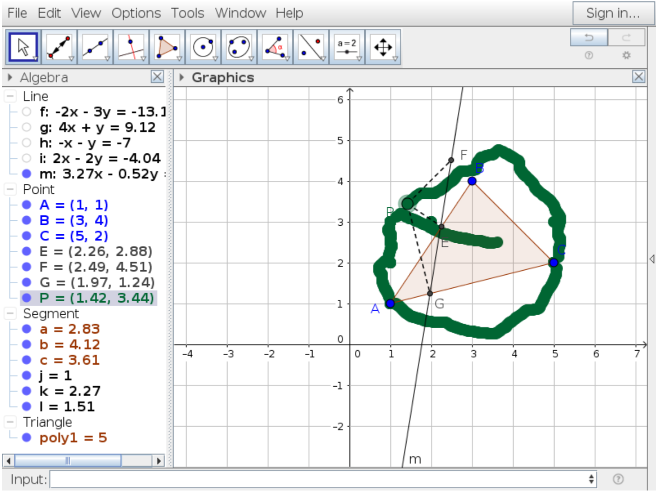
Automated discovering: Wallace-Simson (again)

Find points P s.t. E, F, G are collinear

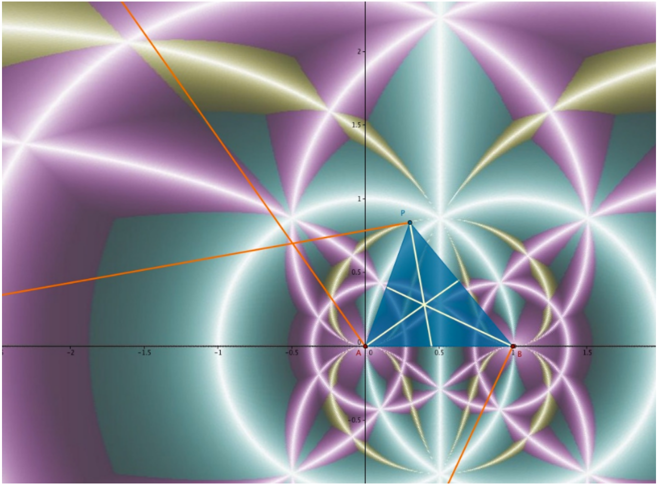


Automated discovering: Wallace-Simson by hand

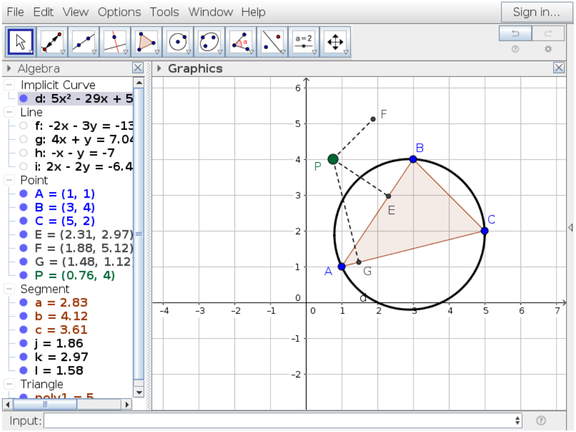
Find points P s.t. E, F, G are collinear



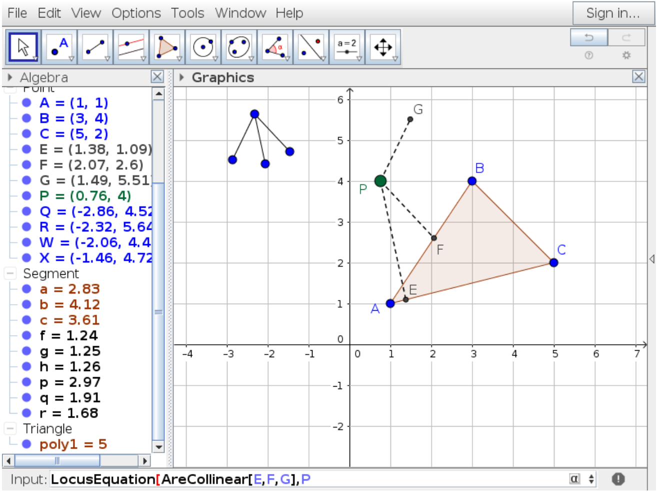
Automated discovering: Brute force



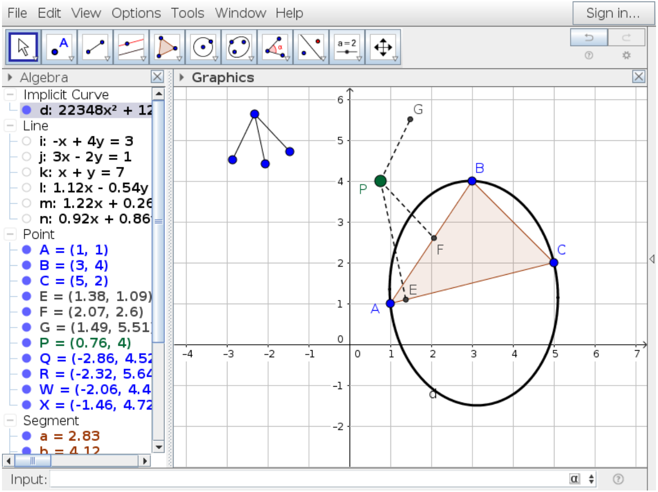
Since points *P* are *a posteriori* constrained, specify the condition LocusEquation[AreCollinear[E,F,G],P]



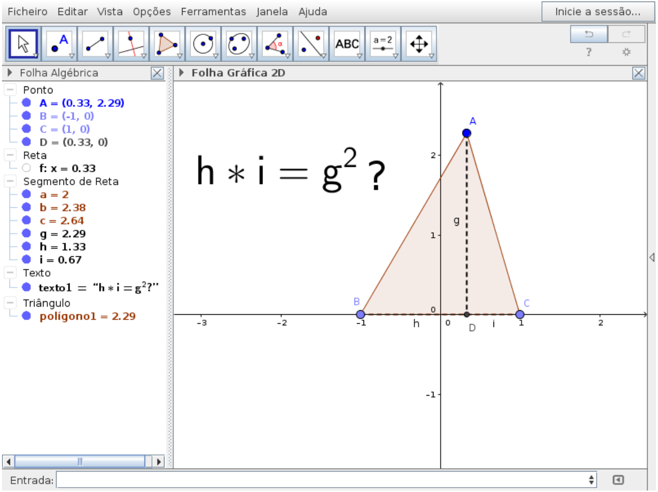
Automated discovering: Guess...



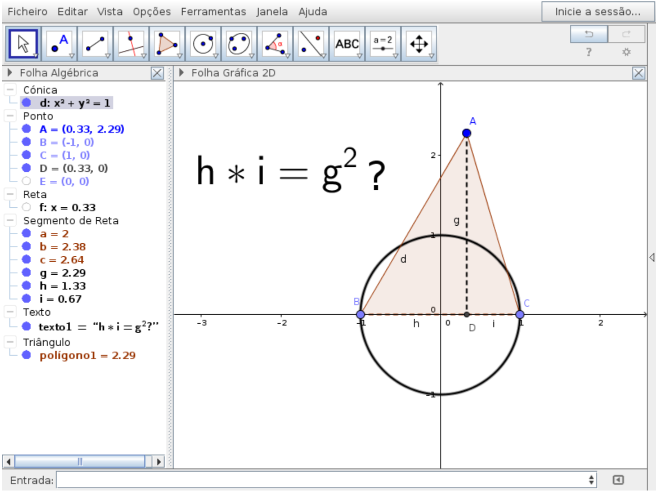
Automated discovering: A generalization of W-S



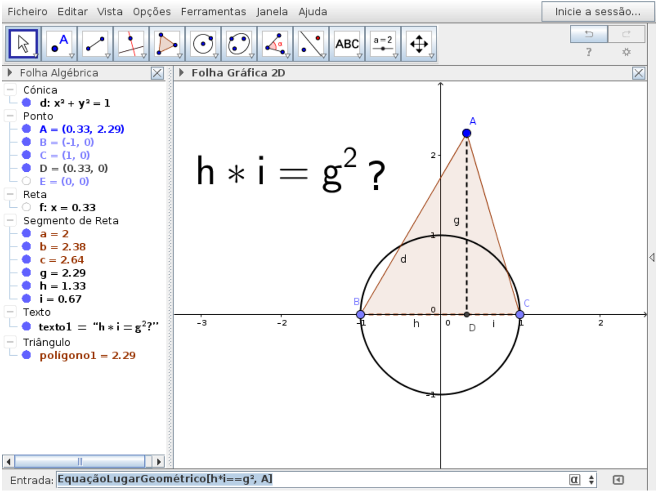
Automated discovering



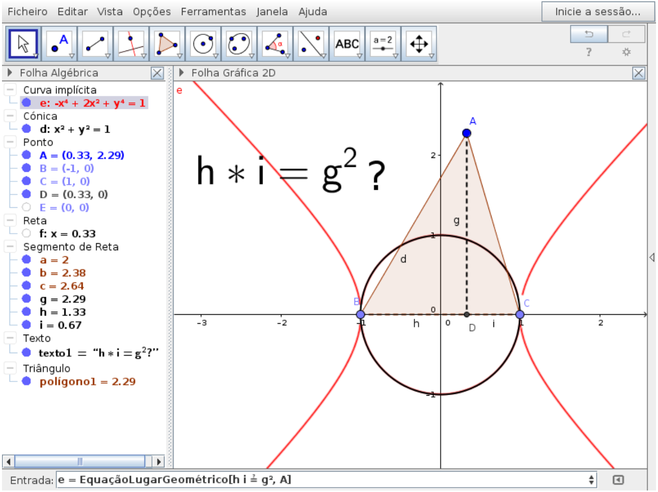
Automated discovering



Automated discovering



Automated discovering



Conclusion

- What could be the role of knowing geometric properties if a simple, widely accessible, free tool can automatically
  - FIND
  - VERIFY
  - DISCOVER

  properties well beyond our personal ability?
- What is the pedagogical role of knowing facts?

Conclusion

- As an auxiliary tool, what opportunities, what differences involve using GeoGebra ART?
- Can guide student exploration, provide hints, answer (partially) questions...
- Can help building up diagrams, locus...
- Helping teachers!

Thank you!