Implementing theorem proving in GeoGebra by using a Singular webservice, or by exact check of a statement in a bounded number of test cases

F. Botana[1] Z. Kovács[2] T. Recio[3] S. Weitzhofer[4]

[1]University of Vigo

[2,4]Department of Mathematics Education
Johannes Kepler University of Linz

[3]University of Cantabria

XIII. Encuentro de Álgebra Computacional y Aplicaciones.
June 13th–15th, 2012. Alcalá de Henares (Madrid), Spain.

Acknowledgment

This work was partially supported by
the Spanish Ministerio de Economia y Competitividad,
grants number MTM2008-04699-C03-03
and MTM2011-25816-C02-02, and by
the European Regional Development Fund (ERDF).

Outline

1. Motivation
   - Dynamic Geometry
   - Numerical Checking
   - DG Software
2. Decision Mechanism in GeoGebra
   - Automatic Mode and Configuration
   - Proving via Singular WebService
   - Proving via exact checks
3. Summary
   - Benchmarking
   - Conclusions

Outline

1 Motivation
- Dynamic Geometry
- Numerical Checking
- DG Software

2 Decision Mechanism in GeoGebra
- Automatic Mode and Configuration
- Proving via Singular WebService
- Proving via exact checks

3 Summary
- Benchmarking
- Conclusions

What is Dynamic Geometry? (DG)

Models built by computer software that can be changed dynamically

- Dynamic transformation
- Dynamic measurement
- Free dragging
- Animation
- Locus generation

What is Dynamic Geometry? (DG)

Models built by computer software that can be changed dynamically

- Dynamic transformation
- Dynamic measurement
- Free dragging
- Animation
- Locus generation

What is Dynamic Geometry? (DG)

Models built by computer software that can be changed dynamically

- Dynamic transformation
- Dynamic measurement
- Free dragging
- Animation
- Locus generation

What is Dynamic Geometry? (DG)

Models built by computer software that can be changed dynamically

- Dynamic transformation
- Dynamic measurement
- Free dragging
- Animation
- Locus generation

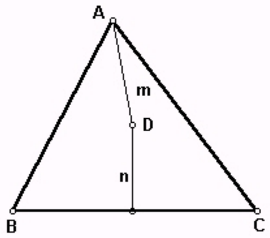
What is Dynamic Geometry? (DG)

Models built by computer software that can be changed dynamically

- Dynamic transformation
- Dynamic measurement
- Free dragging
- Animation
- Locus generation

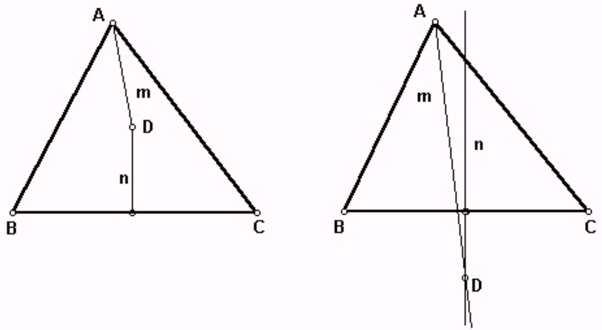
Visual Proofs

All triangles are isosceles I



Visual Proofs

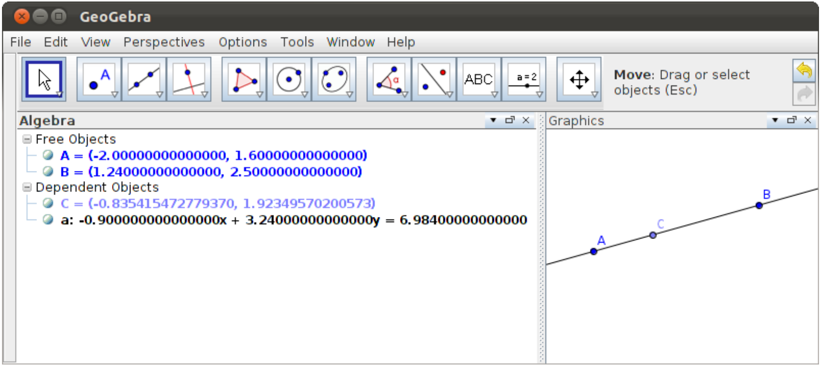
All triangles are isosceles II



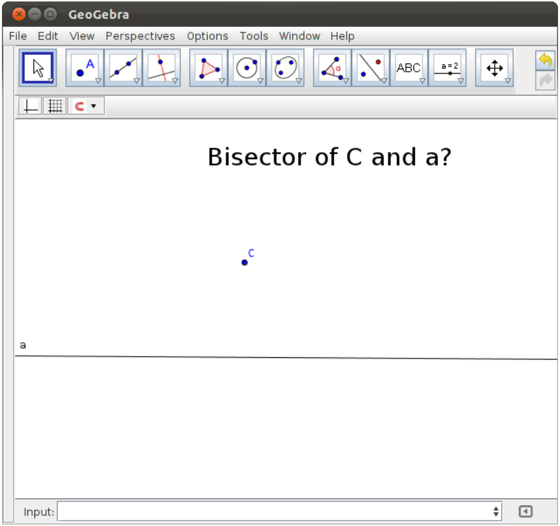
Outline

1. Motivation
   - Dynamic Geometry
   - Numerical Checking
   - DG Software
2. Decision Mechanism in GeoGebra
   - Automatic Mode and Configuration
   - Proving via Singular WebService
   - Proving via exact checks
3. Summary
   - Benchmarking
   - Conclusions

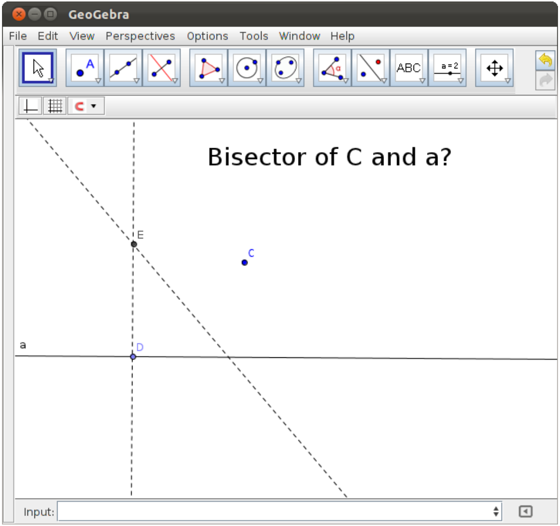
Numerical Checking: Collinear points



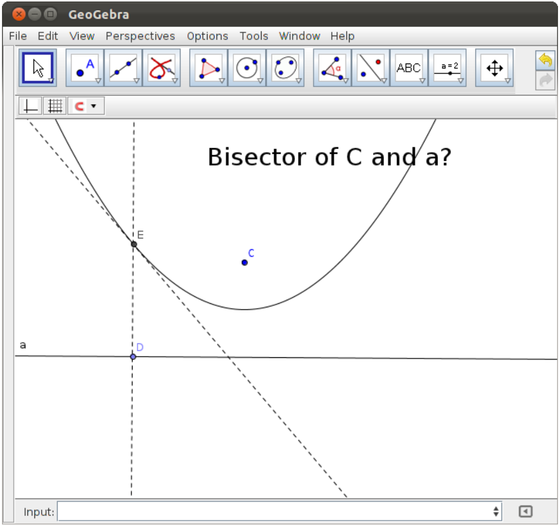
Numerical Checking: A parabola I



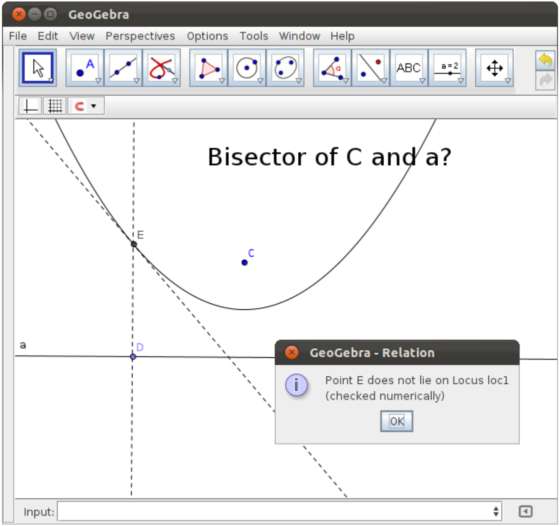
Numerical Checking: A parabola II



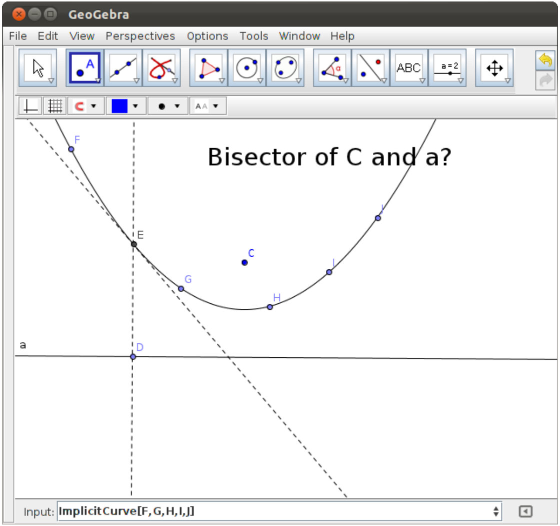
Numerical Checking: A parabola III



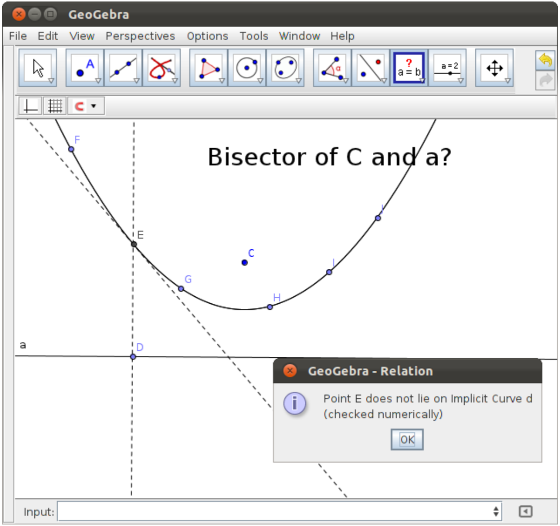
Numerical Checking: A parabola IV



Numerical Checking: A parabola V



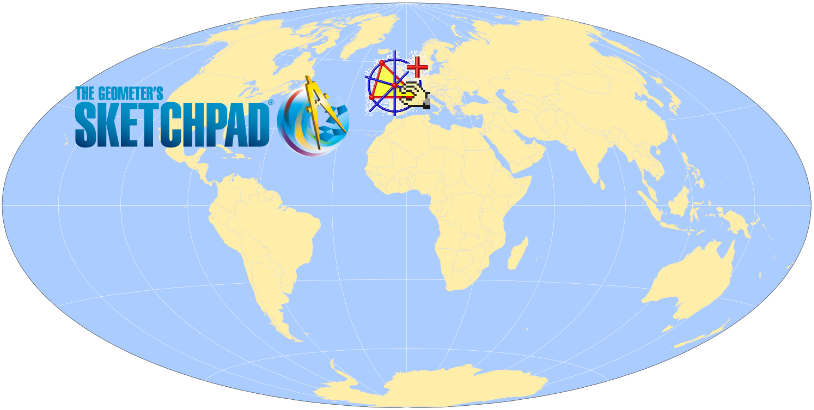
Numerical Checking: A parabola VI



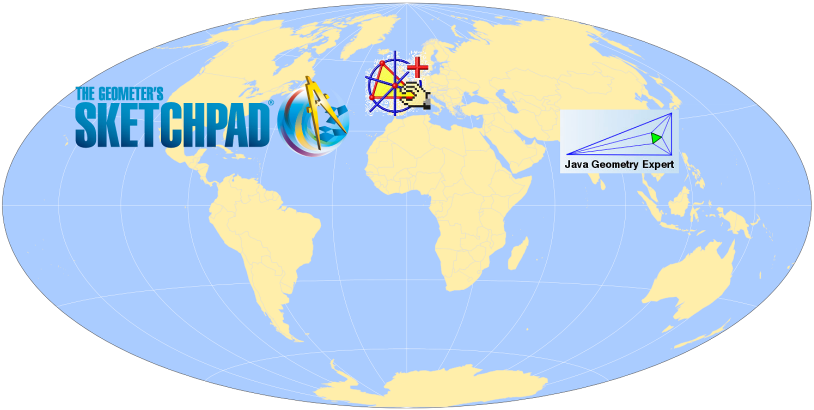
Outline

1. Motivation
   - Dynamic Geometry
   - Numerical Checking
   - DG Software
2. Decision Mechanism in GeoGebra
   - Automatic Mode and Configuration
   - Proving via Singular WebService
   - Proving via exact checks
3. Summary
   - Benchmarking
   - Conclusions

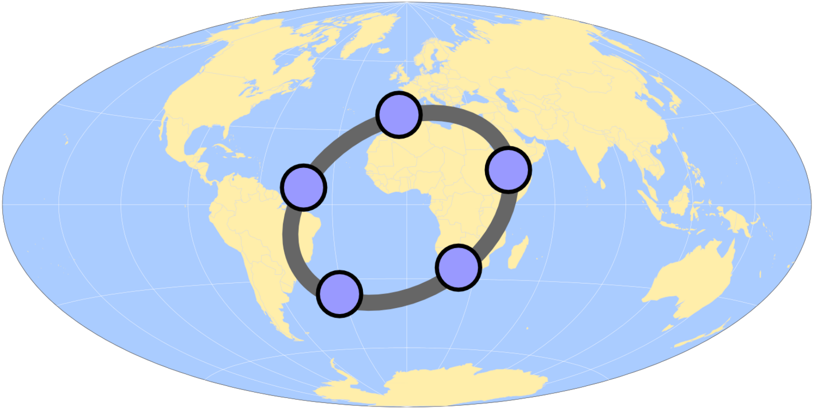
DG Software I



DG Software II



DG Software III



Turning a Dream into Reality

[...] the interaction of such a tool with our method could provide an intelligent, interactive environment for learning Euclidean geometry.
[...] Maybe this is too much for a future dream [...]. But it is, anyhow, a promising research, in our opinion.

Turning a Dream into Reality

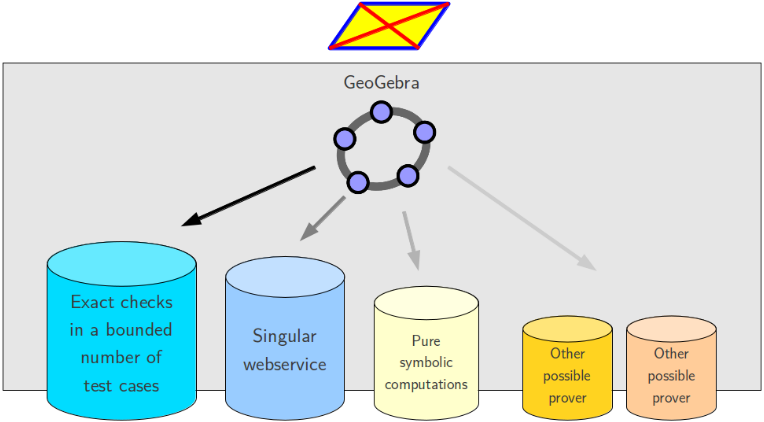
[...] the interaction of such a tool with our method could provide an intelligent, interactive environment for learning Euclidean geometry.
[...] Maybe this is too much for a future dream [...]. But it is, anyhow, a promising research, in our opinion.

T. Recio and M.P. Vélez (1996)

Outline

1. Motivation
   - Dynamic Geometry
   - Numerical Checking
   - DG Software
2. Decision Mechanism in GeoGebra
   - Automatic Mode and Configuration
   - Proving via Singular WebService
   - Proving via exact checks
3. Summary
   - Benchmarking
   - Conclusions

Automatic Mode



Command Line Arguments

```
$ java -jar geogebra.jar --proverhelp

  --prover=OPTIONS              set options for the prover subsystem

  where OPTIONS is a comma separated list, formed with the following available settings
  (defaults in brackets):

  engine:ENGINE             set engine (Auto|Recio|Botana|PureSymbolic) [Auto]
  timeout:SECS              set the maximum time attributed to the prover (in seconds) [5]
  fpnevercoll:BOOLEAN       assume three free points are never collinear [true] (Botana only)
  usefixcoords:BOOLEAN      use fix coordinates for the first points [true] (Botana only)
  singularWS:BOOLEAN        use Singular WebService when possible [true]
  singularWSremoteURL:URL   set the remote server URL for Singular WebService
                                                             [http://ggb1.idm.jku.at:8085/]
  singularWStimeout:SECS    set the timeout for SingularWebService [5]

  Example: --prover=engine:Botana,timeout:10,fpnevercoll:false
```

Outline

1 Motivation
- Dynamic Geometry
- Numerical Checking
- DG Software

2 Decision Mechanism in GeoGebra
- Automatic Mode and Configuration
- Proving via Singular WebService
- Proving via exact checks

3 Summary
- Benchmarking
- Conclusions

Theoretical Background

- Hilbert's Strong Nullstellensatz (1893)
- Rabinowitsch's Trick (1929)
- Gröbner bases, Buchberger's Algorithm (1965)
- Geometry theorem proving: Kapur, Kutzler, Stifter, Chou (1986-1988)
- Faugère's F5 Algorithm (2002)

Theoretical Background

- Hilbert's Strong Nullstellensatz (1893)
- Rabinowitsch's Trick (1929)
- Gröbner bases, Buchberger's Algorithm (1965)
- Geometry theorem proving: Kapur, Kutzler, Stifter, Chou (1986-1988)
- Faugère's F5 Algorithm (2002)

Theoretical Background

- Hilbert's Strong Nullstellensatz (1893)
- Rabinowitsch's Trick (1929)
- Gröbner bases, Buchberger's Algorithm (1965)
- Geometry theorem proving: Kapur, Kutzler, Stifter, Chou (1986-1988)
- Faugère's F5 Algorithm (2002)

Theoretical Background

- Hilbert's Strong Nullstellensatz (1893)
- Rabinowitsch's Trick (1929)
- Gröbner bases, Buchberger's Algorithm (1965)
- Geometry theorem proving: Kapur, Kutzler, Stifter, Chou (1986-1988)
- Faugère's F5 Algorithm (2002)

Theoretical Background

- Hilbert's Strong Nullstellensatz (1893)
- Rabinowitsch's Trick (1929)
- Gröbner bases, Buchberger's Algorithm (1965)
- Geometry theorem proving: Kapur, Kutzler, Stifter, Chou (1986-1988)
- Faugère's F5 Algorithm (2002)

Technical Background I

- GeoGebra platform, Java (2002-)
- Apache HTTP server (1995-)
- Varnish HTTP cache (2006-)
- PHP (1995-)
- Singular, Univ. of Kaiserslautern (1984-)

Technical Background I

- GeoGebra platform, Java (2002-)
- Apache HTTP server (1995-)
- Varnish HTTP cache (2006-)
- PHP (1995-)
- Singular, Univ. of Kaiserslautern (1984-)

Technical Background I

- GeoGebra platform, Java (2002-)
- Apache HTTP server (1995-)
- Varnish HTTP cache (2006-)
- PHP (1995-)
- Singular, Univ. of Kaiserslautern (1984-)

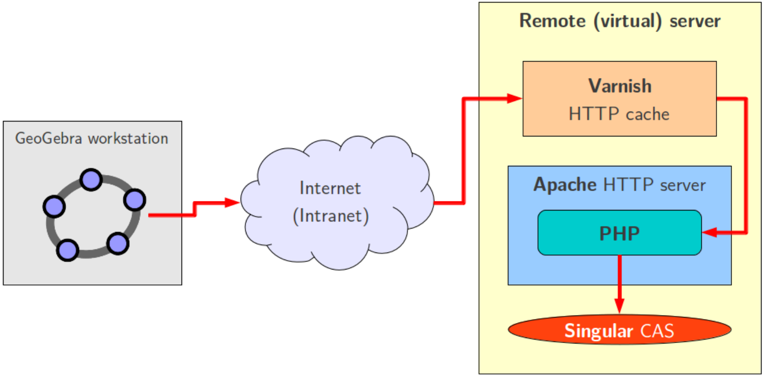
Technical Background I

- GeoGebra platform, Java (2002-)
- Apache HTTP server (1995-)
- Varnish HTTP cache (2006-)
- PHP (1995-)
- Singular, Univ. of Kaiserslautern (1984-)

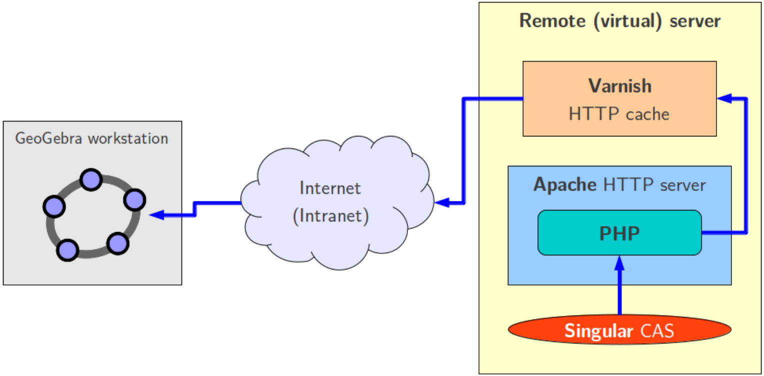
Technical Background I

- GeoGebra platform, Java (2002-)
- Apache HTTP server (1995-)
- Varnish HTTP cache (2006-)
- PHP (1995-)
- Singular, Univ. of Kaiserslautern (1984-)

Technical Background II



Technical Background II



Set up the Equation System

Hypotheses

- Collect parent points in-depth
  - Set up variable pairs (x_i, y_i)
- Collect parent objects
  - Set up polynomials

NDG conditions

- Assume no sets of three free points are collinear

Fixed points

- $(x_1, y_1) \mapsto (0, 0)$
- $(x_2, y_2) \mapsto (0, 1)$

Negated thesis

- Set up polynomials
  - Rabinowitsch's trick for one of them

Set up the Equation System

Hypotheses

- Collect parent points in-depth
  - Set up variable pairs (x_i, y_i)
- Collect parent objects
  - Set up polynomials

NDG conditions

- Assume no sets of three free points are collinear

Fixed points

- $(x_1, y_1) \mapsto (0, 0)$
- $(x_2, y_2) \mapsto (0, 1)$

Negated thesis

- Set up polynomials
  - Rabinowitsch's trick for one of them

Set up the Equation System

Hypotheses

- Collect parent points in-depth
  - Set up variable pairs (x_i, y_i)
- Collect parent objects
  - Set up polynomials

NDG conditions

- Assume no sets of three free points are collinear

Fixed points

- $(x_1, y_1) \mapsto (0, 0)$
- $(x_2, y_2) \mapsto (0, 1)$

Negated thesis

- Set up polynomials
  - Rabinowitsch's trick for one of them

Set up the Equation System

Hypotheses

- Collect parent points in-depth
  - Set up variable pairs (x_i, y_i)
- Collect parent objects
  - Set up polynomials

NDG conditions

- Assume no sets of three free points are collinear

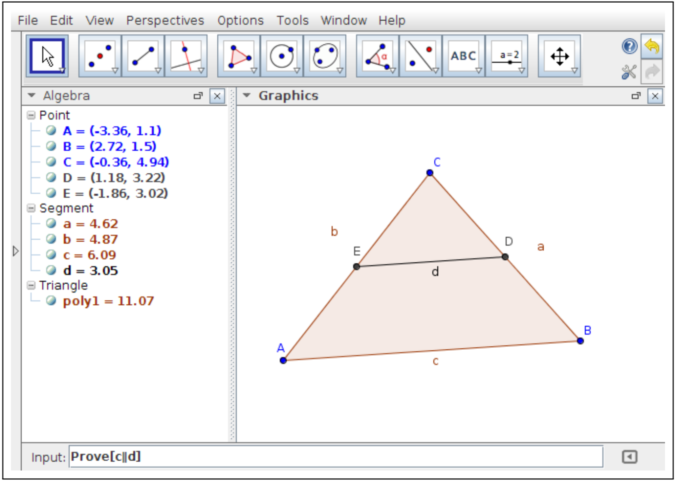
Fixed points

- $(x_1, y_1) \mapsto (0, 0)$
- $(x_2, y_2) \mapsto (0, 1)$

Negated thesis

- Set up polynomials
  - Rabinowitsch's trick for one of them

Example: The Midline of a Triangle is Parallel to a Side I



Example: The Midline of a Triangle is Parallel to a Side II

Hypotheses

- Collect parent points in-depth
  - $D(v_1, v_2)$, $B(v_3, v_4)$, $C(v_5, v_6)$, $E(v_7, v_8)$, $A(v_9, v_{10})$
- Collect parent objects
  - $2v_1 - v_3 - v_5, 2v_2 - v_4 - v_6$
  - $2v_7 - v_5 - v_9, 2v_8 - v_6 - v_{10}$

NDG conditions

- $v_{11}(v_9 v_6 - v_9 v_4 - v_{10} v_5 + v_4 v_5 + v_3 v_{10} - v_6 v_3) - 1$

Fixed points

- $(v_5, v_6) \mapsto (0, 0)$
- $(v_3, v_4) \mapsto (0, 1)$

Negated thesis

- $v_{12}((v_9 - v_3) \cdot (v_8 - v_2) - (v_{10} - v_4) \cdot (v_7 - v_1)) - 1$

Example: The Midline of a Triangle is Parallel to a Side II

Hypotheses

- Collect parent points in-depth
  - $D(v_1, v_2)$, $B(v_3, v_4)$, $C(v_5, v_6)$, $E(v_7, v_8)$, $A(v_9, v_{10})$
- Collect parent objects
  - $2v_1 - v_3 - v_5, 2v_2 - v_4 - v_6$
  - $2v_7 - v_5 - v_9, 2v_8 - v_6 - v_{10}$

NDG conditions

- $v_{11}(v_9 v_6 - v_9 v_4 - v_{10} v_5 + v_4 v_5 + v_3 v_{10} - v_6 v_3) - 1$

Fixed points

- $(v_5, v_6) \mapsto (0, 0)$
- $(v_3, v_4) \mapsto (0, 1)$

Negated thesis

- $v_{12}((v_9 - v_3) \cdot (v_8 - v_2) - (v_{10} - v_4) \cdot (v_7 - v_1)) - 1$

Example: The Midline of a Triangle is Parallel to a Side II

Hypotheses

- Collect parent points in-depth
  - $D(v_1, v_2)$, $B(v_3, v_4)$, $C(v_5, v_6)$, $E(v_7, v_8)$, $A(v_9, v_{10})$
- Collect parent objects
  - $2v_1 - v_3 - v_5, 2v_2 - v_4 - v_6$
  - $2v_7 - v_5 - v_9, 2v_8 - v_6 - v_{10}$

NDG conditions

- $v_{11}(v_9 v_6 - v_9 v_4 - v_{10} v_5 + v_4 v_5 + v_3 v_{10} - v_6 v_3) - 1$

Fixed points

- $(v_5, v_6) \mapsto (0, 0)$
- $(v_3, v_4) \mapsto (0, 1)$

Negated thesis

- $v_{12}((v_9 - v_3) \cdot (v_8 - v_2) - (v_{10} - v_4) \cdot (v_7 - v_1)) - 1$

Example: The Midline of a Triangle is Parallel to a Side II

Hypotheses

- Collect parent points in-depth
  - $D(v_1, v_2)$, $B(v_3, v_4)$, $C(v_5, v_6)$, $E(v_7, v_8)$, $A(v_9, v_{10})$
- Collect parent objects
  - $2v_1 - v_3 - v_5, 2v_2 - v_4 - v_6$
  - $2v_7 - v_5 - v_9, 2v_8 - v_6 - v_{10}$

NDG conditions

- $v_{11}(v_9 v_6 - v_9 v_4 - v_{10} v_5 + v_4 v_5 + v_3 v_{10} - v_6 v_3) - 1$

Fixed points

- $(v_5, v_6) \mapsto (0, 0)$
- $(v_3, v_4) \mapsto (0, 1)$

Negated thesis

- $v_{12}((v_9 - v_3) \cdot (v_8 - v_2) - (v_{10} - v_4) \cdot (v_7 - v_1)) - 1$

Example: The Midline of a Triangle is Parallel to a Side III

```
ring r=0,(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12),dp;
ideal i=
  -1*v5+-1*v3+2*v1,
  -1*v6+-1*v4+2*v2,
  -1*v9+2*v7+-1*v5,
  -1*v10+2*v8+-1*v6,
  -1+v11*v9*v6+-1*v11*v10*v5+-1*v11*v9*v4
    +v11*v5*v4+v11*v10*v3+-1*v11*v6*v3,
  -1+v12*v9*v8+-1*v12*v10*v7+v12*v7*v4+-1*v12*v8*v3
    +-1*v12*v9*v2+v12*v3*v2+v12*v10*v1+-1*v12*v4*v1;
i=subst(i,v3,0,v4,1,v5,0,v6,0);
groebner(i)!=1;
```

Outline

1. Motivation
   - Dynamic Geometry
   - Numerical Checking
   - DG Software

2. Decision Mechanism in GeoGebra
   - Automatic Mode and Configuration
   - Proving via Singular WebService
   - Proving via exact checks

3. Summary
   - Benchmarking
   - Conclusions

Proving via Exact Checks I

Situation

- Many statements can be expressed by a polynomial.

$$p \equiv 0 \Leftrightarrow \text{statement is true}$$

- Calculating the polynomial - expensive.
- Statement holds on fixed coordinates - cheap.
- Upper bound for degree of p - cheap.

Proving via Exact Checks I

Situation

- Many statements can be expressed by a polynomial.

$$p \equiv 0 \Leftrightarrow \text{statement is true}$$

- Calculating the polynomial - expensive.
- Statement holds on fixed coordinates - cheap.
- Upper bound for degree of p - cheap.

Proving via Exact Checks I

Situation

- Many statements can be expressed by a polynomial.

$$p \equiv 0 \Leftrightarrow \text{statement is true}$$

- Calculating the polynomial - expensive.
- Statement holds on fixed coordinates - cheap.
- Upper bound for degree of p - cheap.

Proving via Exact Checks I

Situation

- Many statements can be expressed by a polynomial.

$$p \equiv 0 \Leftrightarrow \text{statement is true}$$

- Calculating the polynomial - expensive.
- Statement holds on fixed coordinates - cheap.
- Upper bound for degree of p - cheap.

Proving via Exact Checks II

Basic idea

$p \in \mathbb{R}[x_1, \ldots, x_n]$:

$$\begin{aligned} p &\equiv 0 \\ &\Updownarrow \\ \mathbb{V}(p) &\equiv \mathbb{R}^n \end{aligned}$$

Proving via Exact Checks (2d Case) I

Bézout's theorem

Let X and Y be plane projective algebraic curves which do not have a component in common.

$$\#\{\text{intersection points}(X, Y)\} \leq \deg X \cdot \deg Y$$

Proving via Exact Checks (2d Case) II

Using Bézout's theorem

For $p \in \mathbb{R}[x_1, x_2]$, $\mathbb{V}(p)$ is a plane projective algebraic curve, or if $p \equiv 0$, the whole plane.
The number of intersection points of a line with the curve cannot be greater than the degree d of p, or the line and the curve have a component in common.

- A line consists only of one component.
- If the curve contains more than d points in a line, it contains the whole line.

Proving via Exact Checks (2d Case) II

Using Bézout's theorem

For $p \in \mathbb{R}[x_1, x_2]$, $\mathbb{V}(p)$ is a plane projective algebraic curve, or if $p \equiv 0$, the whole plane.
The number of intersection points of a line with the curve cannot be greater than the degree d of p, or the line and the curve have a component in common.

- A line consists only of one component.
- If the curve contains more than d points in a line, it contains the whole line.

Proving via Exact Checks (2d Case) II

Using Bézout's theorem

For $p \in \mathbb{R}[x_1, x_2]$, $\mathbb{V}(p)$ is a plane projective algebraic curve, or if $p \equiv 0$, the whole plane.
The number of intersection points of a line with the curve cannot be greater than the degree d of p, or the line and the curve have a component in common.

- A line consists only of one component.
- If the curve contains more than d points in a line, it contains the whole line.

Proving via Exact Checks (2d Case) III

Consequences

- The union of $d+1$ distinct lines is a variety of degree $d+1$.
- If the variety $\mathbb{V}(p)$ of degree less than d contains $d+1$ distinct lines, it cannot be an algebraic curve. It has to be the whole plane $\Rightarrow p \equiv 0$.

Proving via Exact Checks (2d Case) III

Consequences

- The union of $d+1$ distinct lines is a variety of degree $d+1$.
- If the variety $\mathbb{V}(p)$ of degree less than d contains $d+1$ distinct lines, it cannot be an algebraic curve. It has to be the whole plane $\Rightarrow p \equiv 0$.

Proving via Exact Checks (2d Case) IV

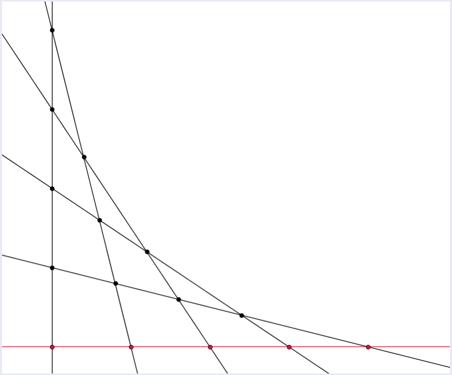
Summary

For $p \in \mathbb{R}[x_1, x_2]$, the variety $\mathbb{V}(p)$ is a plane projective algebraic curve or the whole plane.

- If the variety contains $d+1$ collinear points, it contains the whole line.
- If the variety contains $d+1$ different lines, it cannot be a curve, but has to be the whole plane.

Proving via Exact Checks (2d Case) V

A possible arrangement



$d = 4$

One possibility of $d+1$ points on each of $d+1$ lines:
Take $d+2$ lines in general position and use the

$$\binom{d+2}{2}$$

intersection points.

Proving via Exact Checks (2d Case) VI

Algorithm 1

Input:

- An upper limit d for the degree of an unknown polynomial p.
- A method to test whether $p[x,y]$ evaluates to zero or not.

Output:

- True if $p \equiv 0$, false otherwise.

Proving via Exact Checks (2d Case) VII

Algorithm 1

1. Choose $d+2$ lines in general position.
2. Get all the intersection points of the lines.
3. If for each point p evaluates to zero return true, else return false.

Proving via Exact Checks III

Algorithm 2

Input:

- A construction and a statement which can be expressed with a polynomial equation.
- A method to test if the statement holds for some fixed points.

Output:

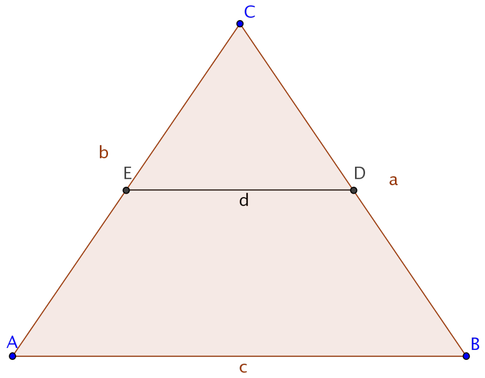
- True if the statement is generally true, false otherwise.

Proving via Exact Checks IV

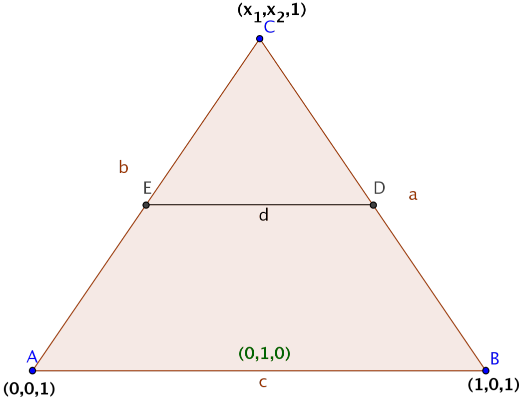
Algorithm 2

1. Fix two points (optional).
2. Find the number of free variables of the statement.
3. Find an upper bound for the degree of the statement.
4. Use algorithm 1 to test if the describing polynomial is constantly zero.
   Return true if it is constantly zero, false otherwise.

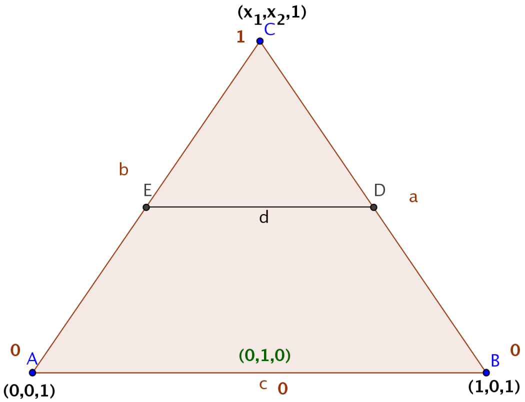
Example



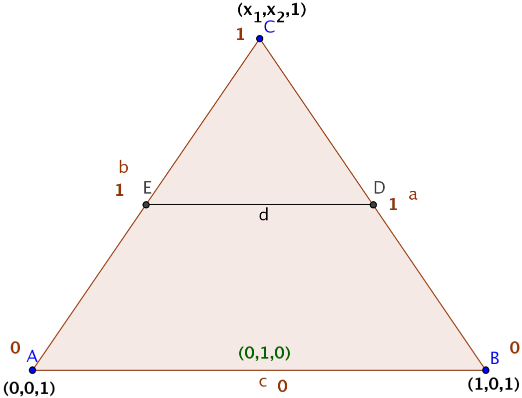
Example



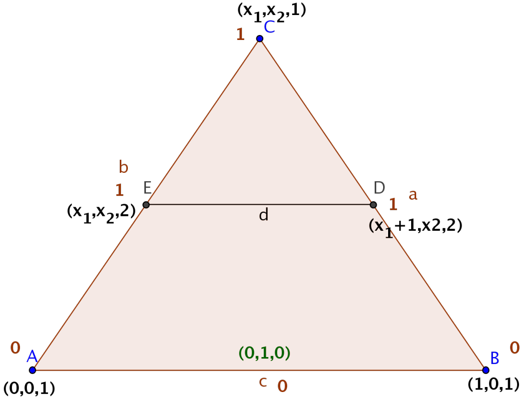
Example



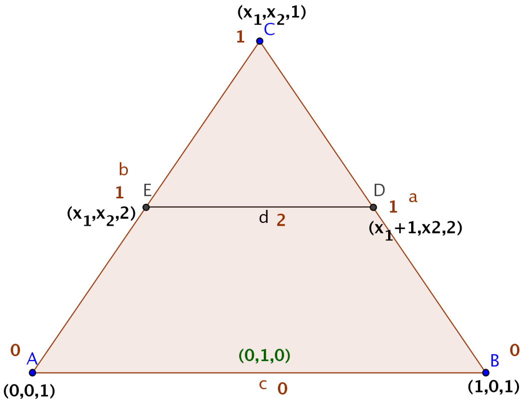
Example



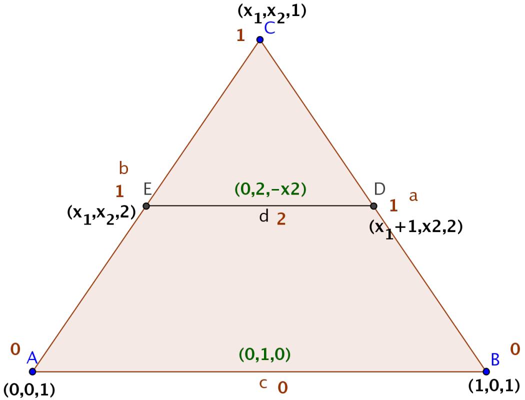
Example



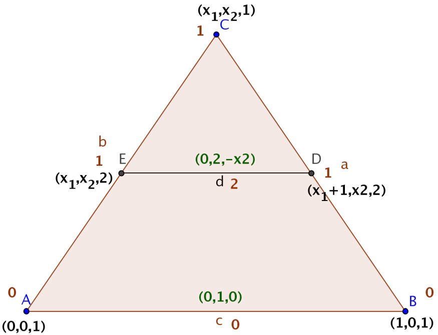
Example



Example



Example



Evaluations

Deg. statement:
$\deg(d) + \deg(c) = 2$

Nr. of evaluations:
$\binom{2+2}{2} = 6$

Outline

1 Motivation
- Dynamic Geometry
- Numerical Checking
- DG Software

2 Decision Mechanism in GeoGebra
- Automatic Mode and Configuration
- Proving via Singular WebService
- Proving via exact checks

3 Summary
- Benchmarking
- Conclusions

Benchmarking Results I

Prover benchmark for GeoGebra 4.1.90.0 (r17932)

on 2012-06-08 22:57 at kovzol-dell

```
Intel(R) Core(TM) i5 CPU       M 480  @ 2.67GHz, 1197 MHz
Intel(R) Core(TM) i5 CPU       M 480  @ 2.67GHz, 1197 MHz
Intel(R) Core(TM) i5 CPU       M 480  @ 2.67GHz, 1197 MHz
Intel(R) Core(TM) i5 CPU       M 480  @ 2.67GHz, 2661 MHz
```

Test file	Auto		Recio		SingularWS		PureSymbolic	
	Result	Speed	Result	Speed	Result	Speed	Result	Speed
foot-exists.ggb	undefined	32	undefined	5	undefined	16	undefined	20
lines-parallel.ggb	false	12	false	11	false	168	false	8
points-collinear.ggb	false	8	false	10	false	98	false	7
points-equal.ggb	false	17	false	23	false	207	false	21
bisector-midpoint.ggb	true	12	true	11	true	160	true	18
centroid-median-ratio.ggb	true	55	undefined	8	true	61	undefined	3
circumcenter1.ggb	true	10	true	15	true	49	true	40
circumcenter2.ggb	true	15	true	16	true	44	true	38

Benchmarking Results II

circumcenter5.ggb	true	58	undefined	8	true	71	undefined	4
def-line-perpline-perpline.ggb	true	10	true	11	true	93	true	7
def-points-on-a-circle1.ggb	true	81	undefined	7	true	77	undefined	5
def-points-on-a-circle2.ggb	true	101	undefined	4	true	103	undefined	8
def-points-on-a-line.ggb	true	11	true	10	true	113	true	18
Desargues.ggb	true	2730	true	2799	false	156	true	59372
EulerLine.ggb	true	20	true	25	true	68	true	920
line-circle-intersection.ggb	true	45	undefined	6	true	221	undefined	9
nine-points-circle.ggb	true	50	true	43	true	83	true	16347
orthocenter1.ggb	true	15	true	17	true	51	true	147
orthocenter2.ggb	true	13	true	12	true	283	true	13
orthocenter3.ggb	true	15	true	11	true	62	true	36
orthocenter4.ggb	true	13	true	14	true	69	true	38
orthocenter5.ggb	true	20	true	20	true	397	true	80
orthocenter6.ggb	true	18	true	18	true	47	true	42

Benchmarking Results III

point-equal.ggb	true	23	true	22	true	161	true	23
regular-triangle.ggb	true	229	undefined	6	true	52	undefined	5
Simson1.ggb	false	23711	undefined	6	false	22725	undefined	6
Simson2.ggb	false	20870	undefined	6	false	12805	undefined	5
Thales1.ggb	true	66	undefined	7	true	55	undefined	3
Thales2.ggb	false	75	undefined	3	false	99	undefined	3
Thales3.ggb	true	60	undefined	7	true	47	undefined	4
triangle-medians.ggb	true	11	true	16	true	76	true	47
triangle-midsegment1.ggb	true	7	true	10	true	45	true	15
triangle-midsegment2.ggb	true	13	true	13	true	99	true	13
triangle-midsegment3.ggb	true	12	true	12	true	84	true	15
triangle-midsegment4.ggb	true	12	true	12	true	71	true	10
triangle-midsegment5.ggb	true	17	true	14	true	50	true	14
true.ggb	true	1	true	1	true	2	true	1
Varignon.ggb	true	17	true	17	true	45	true	15
Summary (of 43)	**39**		**31**		**38**		**31**	

Outline

1. Motivation
   - Dynamic Geometry
   - Numerical Checking
   - DG Software
2. Decision Mechanism in GeoGebra
   - Automatic Mode and Configuration
   - Proving via Singular WebService
   - Proving via exact checks
3. Summary
   - Benchmarking
   - Conclusions

Conclusions

- **Prove[<Boolean Expression>]** and **ProveDetails[<Boolean Expression>]** commands in GeoGebra 5 (2013)
  - live demo available at `http://www.geogebra.org/web/web_gui`
- Relation Tool: *numerical* → *symbolic*
- Computer Aided Assessment
  - automatic check of open ended exercises in Euclidean geometry

References I

S.-C. Chou
Mechanical geometry theorem proving.
D. Reidel Publishing Co., 1988

D. Kapur.
Using Gröbner bases to reason about geometry problems.
J. Symb. Comput. 2:399–408, 1986.

References II

Wikipedia Contributors
Bézout's theorem.
In Wikipedia, The Free Encyclopedia. Retrieved 21:02, May 29, 2012, from
http://en.wikipedia.org/wiki/Bézout's_theorem

T. Recio
La mecánica de la demostración y la demostración mecánica.
Actas X Jornadas para el Aprendizaje y la Enseñanza de las Matemáticas, Vol. I, Sociedad Aragonesa de Profesores de Matemáticas. ICE Universidad de Zaragoza, Zaragoza, pp. 189-212. 2002.