

# GeoGebra Automatisierte Begründungswerkzeuge

## Ein Tutorial

Zoltán Kovács, Tomás Recio und M. Pilar Vélez  
Übersetzt von Katharina Schiffler  
**work in progress**

24. März 2017

## 1 Einführung

Das Softwaretool GeoGebra (<http://www.geogebra.org>) ist geeignet zur Unterstützung des Unterrichts der zweidimensionalen euklidischen Geometrie Theorien mit symbolischen Berechnungen. Zu diesem Zweck wurden einige Werkzeuge für den automatischen Beweis und zur Entdeckung von geometrischen Lehrsätzen entwickelt.

Die neuen Technologien im Klassenzimmer befinden sich noch in einer experimentellen Phase. Dieses Dokument fasst die technischen Möglichkeiten zusammen und zeigt auch einige Beispiele.

## 2 Starting GeoGebra

GeoGebra ist auf vielen Plattformen verfügbar, wie zum Beispiel

- Computer mit verschiedenen Betriebssystemen
- Tablets, und
- Smartphones.

Darüber hinaus sind eingebettete GeoGebra Applets auf Websites verfügbar, speziell auf GeoGebra Materials (<https://www.geogebra.org/materials/>) mit Millionen kostenlos verfügbarer Unterrichtsmaterialien.

Die verfügbaren Werkzeuge können unter Umständen variieren. Auch die Benutzererfahrung kann auf den verschiedenen Plattformen unterschiedlich sein: Die symbolischen Berechnungen können ein hohes Ausmaß an Kalkulationen benötigen und die unterliegenden Hardware- und Softwarekomponenten unterstützen möglicherweise nicht alle Schritte vollständig.

Die vorgeschlagene Plattform für Klassen kann variieren. Die schnellsten Resultate können mit schnellen Computern erzielt werden, aber in diesem Fall muss die Software vom Benutzer heruntergeladen und installiert werden. Manche Beispiele in diesem Tutorial funktionieren unter Umständen nur in der „Desktop“ Version (welche für Microsoft Windows, Apple Macintosh und Linux Computer verfügbar ist). Auf der anderen Seite benötigt die „Web“ Version keine Installation durch den Benutzer: Diese funktioniert mit einem aktuellen Web Browser und die Lehrerin/der Lehrer kann zum Beispiel eine Liste von Beispielen als GeoGebra Applets für die Klasse unter Verwendung von GeoGebra Materials vorbereiten. Die „Web“ Version ist allerdings langsamer: Die symbolische Berechnung kann etwas langsamer sein.

GeoGebra funktioniert mittlerweile auch auf Tablets und Smartphones. In manchen Fällen können diese Plattformen schnellere Ergebnisse liefern als die Web Version, aber der kleinere Bildschirm könnte die Benutzer davon abhalten, geometrische Lehrsätze im Detail zu untersuchen. Lehrerinnen und Lehrer sollen bestärkt werden, mit dem Umgang dieser modernen Hilfsmittel zu experimentieren, aber der Einsatz für automatische Beweisführung ist noch nicht erforscht.

An den Werkzeugen zur automatischen Beweisführung wird kontinuierlich gearbeitet. Es ist zu bevorzugen, immer die aktuellste Version zu verwenden. Ein wöchentliches Update kann generell für alle Versionen empfohlen werden, abgesehen vom Mac App Store, welcher ungefähr monatlich aktualisiert wird. Die Liste der neuesten Änderungen kann auf <http://dev.geogebra.org/trac/timeline>—eingesehen werden -das ist nur für fortgeschrittene Benutzer und Entwickler.

### 3 Automatische Beweisführung Werkzeuge

Automatische Beweisführung Werkzeuge sind eine Sammlung von GeoGebra Werkzeugen und Befehlen zur Mutmaßung, Entdeckung, Abstimmung und Beweis geometrischer Axiome in einer dynamischen geometrischen Konstruktion.

Am Anfang muss der Benutzer eine geometrische Figur mit bestimmten Werkzeugen auf dem Hauptfenster in GeoGebra zeichnen. Nach der Konstruktion der Figur kann GeoGebra auf verschiedene Arten die geometrischen Eigenschaften mit verschiedenen Werkzeugen und Einstellungen prüfen:

1. Mit *freie Objekte ziehen* ihre abhängigen Objekte können sichtbar erforscht werden.
2. Das *Beziehung* Werkzeug hilft beim Vergleichen von Objekten und erkennen von Beziehungen.
3. Mit dem Setzen der *Spur* eines konstruierten Objektes mit/ohne Bewegung eines Objektes wird sichtbar wenn seine vorhergehenden Objekte geändert werden.
4. Das *Ortslinie* Werkzeug zeigt die Spur eines Objektes für alle möglichen Positionen eines vorhergehenden Objektes (wenn es sich auf einem Pfad bewegt).
5. Mit dem Eintippen von *Beziehung* oder *Ortslinie* Befehl in der GeoGebra Befehlszeile können genauere Informationen ausgewählt werden.

Diese Methoden sind normalerweise bekannt in der GeoGebra community, deshalb sind sie gut dokumentiert und es können einige Beispiele auf GeoGebra Materials (<https://www.geogebra.org/materials/>) gefunden werden. Auf der anderen Seite bietet GeoGebra zurzeit auch symbolische automatische Beweisführung Werkzeuge für vermutete geometrische Eigenschaften:

6. *Beziehung* Werkzeug und Befehl können zum symbolischen Nachrechnen verwendet werden,
7. Der *Ortsliniengleichung* Befehl berechnet das Ergebnis der Ortsliniengleichung mit dem Zeigen der algebraischen Gleichung der graphischen Darstellung genauer.
8. Der *Ortsliniengleichung* Befehl kann *implicit loci* prüfen.
9. Der *Einhüllende* Befehl berechnet die Gleichung der Tangente mehrerer Objekte während ein bestimmter Vorgänger sich auf einer Bahn bewegt.

#### 3.1 Einfache und schwierige Werkzeuge

GeoGebra liefert die obenstehenden schwierigen Methoden um geometrische Theorien zu unterstützen. Diese Icons sind als schwierige Werkzeuge gedacht, weil sie aufgrund ihrer einfachen Benutzung direkt in der Klasse gezeigt werden können. Es gibt auch andere Wege, um mehr über den mathematischen Hintergrund zu lernen oder einfach zur Fehleranalyse. Diese einfachen Methoden sind im Zusatz aufgelistet und daher nicht empfohlen für den direkten Einsatz mit Schülerinnen und Schülern.

Offensichtlich sind einige der aufgelisteten Methoden einfach und andere schwierige. Bei der Befehlszeile in GeoGebra kann überlegt werden als ein schwierigerer Weg für die meisten Benutzer. Daher kann eine Lehrerin/ein Lehrer zuerst die einfacheren Methoden zeigen und nach einiger Erfahrung der Schülerinnen und Schüler die schwierigeren.

#### 3.2 Werkzeuge mit symbolischer Unterstützung

Wie oben erwähnt liefern einige automatische Beweisführung Werkzeuge symbolische Unterstützung. Dieses Feature erlaubt die Verifizierung einem mathematisch rigorosem Weg allgemeine Aussagen der Elementargeometrie, die vom Benutzer vermutet werden.

Ein allgemeiner Hinweis, dass der Benutzer sollte GeoGebra zur Inbetriebnahme mit "Grafikrechner" Modus starten. Diese Einstellung zeigt die Bezeichnung auf jedem neu hinzugefügten Objekt. Das kann für Beziehungswerkzeug und -befehl entscheidend sein, wenn die verschiedenen Beziehungen ausgewertet werden.

In den meisten Fällen sind die Achsen nicht wirklich notwendig: Diese können ausgeschaltet werden, wenn das *Bewegen* Werkzeug aktiv ist. (das ist das linke Icon, das einen Cursor zeigt) mit rechtsklick in der *Grafik Ansicht* und Auswahl der *Achsen* Einstellung.

In einigen Fällen ist es nicht notwendig, die *Algebra Ansicht* anzuzeigen. Wenn die Gleichungen der impliziten Kurven nicht im Detail betrachtet werden, kann dies auch durch die Änderung der Bezeichnung erreicht werden,

sodass auch der Wert angezeigt wird. (Um das durchzuführen, muss ein Rechtsklick und die Auswahl *Objekt Eigenschaften* durchgeführt werden, der Benutzer muss *Beschriftung anzeigen* zu *Wert* an dem *Basis* auswählen.)

GeoGebra Applets können komfortabel genutzt werden, wenn sie auf GeoGebra Materials hochgeladen sind. Wenn die Algebra Ansicht verwendet wird, kann es hilfreich sein, die Weite zu vergrößern, bevor das Applet auf GeoGebra Materials hochgeladen wird. Sonst könnte es schwierig für den Benutzer sein, die Befehle einzugeben. Nach dem hochladen in GeoGebra Materials ist es wichtig, die Funktionen des Applets *Erweiterte Einstellungen*, *Zeige Werkzeugleiste* und *Zeige Befehlszeile* zu überprüfen. Danach sollte eine adäquate Größe für das Applet gewählt werden.

### 3.2.1 Das Beziehungswerkzeug und Befehle

GeoGebra's Beziehungswerkzeug und Befehle zeigt ein Fenster, das dem Benutzer Informationen über die Beziehung zwischen zwei oder mehr Objekten gibt. Dieser Befehl erlaubt dem Benutzer, numerisch herauszufinden, ob (für die gezeichnete Konstruktion mit zugehörigen Koordinaten)

- zwei Geraden normal aufeinander stehen,
- zwei Geraden parallel sind,
- zwei (oder mehr) Objekte (Punkte, Streckenlängen, Polygonflächen) gleich sind,
- ein Punkt auf einer Linie oder Kurve liegt,
- eine Gerade eine Tangent oder eine Passante einer Kurve ist,
- drei Punkte kollinear sind,
- drei Geraden übereinstimmend sind (oder parallel),
- vier Punkte auf einer Kreisbahn liegen (oder kollinear sind).

Manche dieser Überprüfungen können auch symbolisch ausgeführt werden, also der Satz kann generell verifiziert werden (mit beliebigen Koordinaten) und nicht nur für die gezeichnete konkrete Konstruktion.

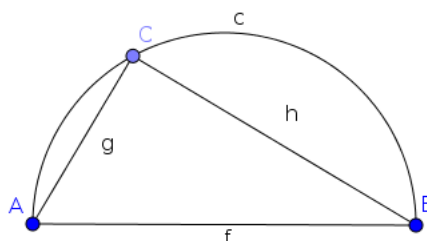
Mit der Verwendung des Beziehungs *werkzeug* muss der Benutzer auf zwei Objekte klicken um die Mitteilungsbbox zu sehen. Alternativ können zwei, drei oder vier Objekte mit dem Auswahlrechteck ausgewählt werden, um die Mitteilungsbbox zu erhalten. Um nicht ungewollte Objekte im Applet auszuwählen ist es möglich, diese von der Auswahl auszunehmen, indem man mit einem Rechtsklick und der Auswahl *Objekt Eigenschaften Erweitert* auswählt und das Häkchen bei *Auswahl erlaubt* entfernt.

Bei der Verwendung des Beziehungs *befehls* gibt der Benutzer eine der folgenden Formlen in die Befehlszeile ein:

- `Beziehung[ <Objekt>, <Objekt> ]`
- `Beziehung[ { <Objekt>, <Objekt> } ]`
- `Beziehung[ { <Objekt>, <Objekt>, <Objekt> } ]`
- `Beziehung[ { <Objekt>, <Objekt>, <Objekt>, <Objekt> } ]`

Wenn die Mitteilungsbbox eines oder mehrere zutreffende numerische Aussagen zu dem Objekt zeigt, gibt es unter Umständen eine Schaltfläche "Mehr...", welche die symbolische Unterstützung für die gegebene Aussage zeigt. Bei Klick auf "Mehr...", wird in Kürze die numerische Aussage zu einer allgemeinen Aussage aktualisiert.

#### Beispiel (Thaleskreis)



1. Mit dem *Strecke* Werkzeug wird  $AB$  konstruiert.
2. Mit der Auswahl *Halbkreis durch zwei Punkte* Werkzeug wird  $c$  erstellt.
3. Der Punkt  $C$  wird auf  $c$  erstellt.
4. Konstruieren Sie die Strecken  $AC$  und  $BC$  und benennen Sie diese  $g$  und  $h$ .
5. Vergleichen Sie  $g$  und  $h$  unter Verwendung des Beziehungswerkzeugs durch Anklicken von  $g$  und  $h$ , oder durch Eintippen von `Beziehung[g,h]` in der Befehlszeile. Die folgende Mitteilung wird angezeigt:
6. Klicken Sie auf “Mehr...” — die Mitteilung wird sich wie folgt ändern:

Beachten Sie, dass *Beziehung* (Schritt 5) sucht nach Beziehungen zwischen  $g$  und  $h$  von den Koordinaten und Gleichungen abhängig von der gezeichneten Konstruktion. Durch Klicken auf “Mehr...” (Schritt 6) kann verifiziert werden, dass  $g$  und  $h$  normal aufeinander stehen und für alle Punkte kann  $A$  und  $B$  kann man Schritt 1 auswählen.

*Beziehung* zwischen bestimmten Objekten kann unter bestimmten Bedingungen wahr sein, aber nicht zwingend “immer”. In diesen Fällen werden genug Voraussetzungen angezeigt. Sonst zeigt GeoGebra nur die Aussage “unter bestimmten Voraussetzungen”. Das muss als “allgemein wahr” verstanden werden, aber in manchen Einzelfällen (welche eine ‘minimale Anzahl von Fällen’ ist, verglichen mit den allgemeinen Fällen) kann die Aussage falsch sein.

Das symbolische Ergebnis der *Beziehung* kann auch negativ sein, auch wenn die numerische Überprüfung positiv ist. Zum Beispiel, wenn man zwei Punkte  $P = (0,0)$  und  $Q = (0,0)$  definiert, kann die *Beziehung* diese numerisch vergleichen, aber das symbolische Ergebnis kann ergeben, dass “ $P$  und  $Q$  gleich sind (aber nicht allgemein wahr)”.

Eine komplette Übersicht der verschiedenen Ergebnisse der *Beziehung* könnene im Anhang gefunden werden. 6.1.3 auf der Seite 15.

### 3.2.2 Die Ortsliniengleichung Ortsliniengleichung Befehl

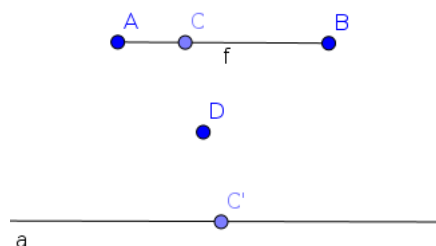
Dieser Befehl berechnet die Gleichung einer Ortslinie und zeichnet diese als implizite Kurve. Es gibt zwei Möglichkeiten der Nutzung:

- **Explizite Ortslinie.** Gegeben ist ein Anfangspunkt  $\mathcal{I}$  auf einer Strecke  $\mathcal{P}$ , einige Konstruktionsschritte und ein Endpunkt  $\mathcal{O}$ . Die Aufgabe ist, die Gleichung von  $\mathcal{E}$  of  $\mathcal{O}$  zu bestimmen, wenn sich  $\mathcal{I}$  auf  $\mathcal{P}$  bewegt, und dann  $\mathcal{E}$  zu bestimmen.  $\mathcal{I}$  ist für gewöhnlich *beweger* genannt,  $\mathcal{O}$  ist die *Spur*.  $\mathcal{E}$  heißt *Ortsliniengleichung* und ihre grafische Visualisierung ist die *Ortslinie*.

Die Schreibweise des Befehls ist:

`Ortsliniengleichung[ <Punkt Spur>, <Punkt Beweger> ].`

#### Beispiel



1. Unter Verwendung des *Strecke* Werkzeug wird  $AB$  konstruiert. Es wird eine automatische Strecke  $f$  konstruiert.

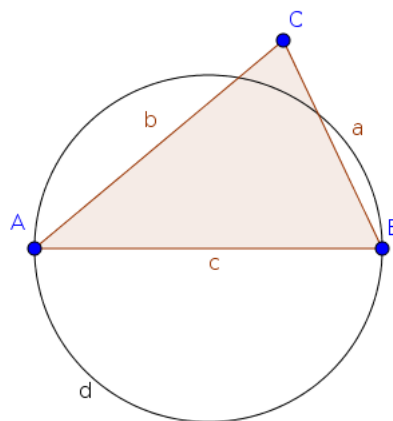
2. Setzen Sie Punkt  $C$  auf  $f$ .
3. Erstellen Sie Punkt  $D$  mit dem *Punkt* Werkzeug.
4. Mit dem *über Punkt spiegeln* Werkzeug spiegeln Sie  $C$  über  $D$ . Damit erstellen Sie Punkt  $C'$ .
5. Geben Sie Ortsliniengleichung  $[C', C]$  in die Befehlszeile ein. Nun wird die implizite Kurve  $a$  berechnet und gezeichnet. Das sollte eine Strecke (das Spiegelbild von  $f$  über  $D$  sein), aber GeoGebra zeichnete  $f$  als Gerade statt einer Strecke (aus algebraischen geometrischen Gründen), da das Spiegelbild ebenfalls eine Gerade ist.
6. Versuchen Sie, die verschiebbaren Objekte zu ziehen. Es kann gezeigt werden, dass das Spiegelbild einer Strecke über einen Punkt immer parallel zum Urbild ist.

- **Implizite Ortslinie.** Gegeben ist ein Anfangspunkt  $\mathcal{J}$ , entweder als beliebiger Punkt oder auf einer Strecke  $\mathcal{P}$ . Es sind auch einige Konstruktionsschritte gegeben. Der Benutzer behauptet eine Boolesche Bedingung  $\mathcal{C}$  auf einige Objekte der Konstruktion. Die Aufgabe ist, eine Gleichung  $\mathcal{E}$  für alle Punkte  $\mathcal{J}'$  aufzustellen, sodass  $\mathcal{J} = \mathcal{J}'$ ,  $\mathcal{C}$  erfüllt. Wieder heißt  $\mathcal{E}$  Ortsliniengleichung und ihre grafische Interpretation ist die Ortslinie.

Die Schreibweise des Befehls ist:

Ortsliniengleichung[ <Boolescher Ausdruck>, <Punkt> ].

### Beispiel



1. Mit dem *Polygon* Werkzeug wird ein Dreieck  $ABC$  konstruiert. Die Strecken  $a$ ,  $b$  und  $c$  werden automatisch von GeoGebra erstellt.
2. Es wird Ortsliniengleichung  $a^2 + b^2 == c^2, C]$  in der Werkzeugleiste eingegeben. Jetzt wird die implizite Kurve  $d$  berechnet und gezeichnet, welche wie ein Kreis aussieht. Bedenken Sie, dass *zwei* gleiche Zeichen eingegeben werden müssen; eine andere Möglichkeit ist,  $\stackrel{?}{=}$  zu verwenden. (durch anklicken von  $\alpha$  Icon, oder durch Einfügen des Symbols von einer externen Applikation mit Copy and Paste).
3. Versuchen Sie nun, die beweglichen Objekte zu ziehen. Es kann gezeigt werden, dass wenn  $C$  auf einem Kreis liegt, dessen Durchmesser auf  $AB$  liegt, wegen der rechtwinkligen Eigenschaft des Dreiecks— $a^2 + b^2 = c^2$  gilt.

Eine Boolesche Bedingung kann sein:

- Eine Gleichung mit der Bezeichnung der Strecken, e.g.  $a^2 + b^2 == c^2$ .
- Eine Gleichheit von zwei Geometrischen Objekten, e.g.  $A == B$ . Beachten Sie, dass *zwei* gleiche Zeichen eingegeben werden müssen; es gibt auch andere Möglichkeiten
  - $\stackrel{?}{=}$  (durch Anklicken des  $\alpha$  Icon, oder durch Einfügen des Symbols von einer externen Applikation mit Copy and Paste).

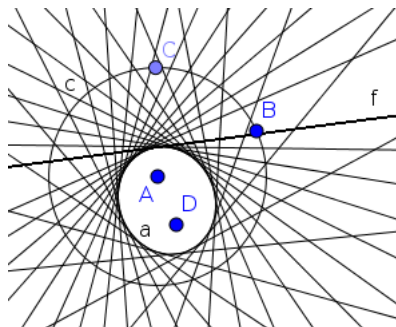
- Alternativ, `SindGleich1 [A,B]` für die gesamte Boolesche Bedingung.
- Eine Überprüfung, ob zwei geometrische Objekte kongruent sind, e.g. `SindKongruent [c,d]`.
- Eine Überprüfung, ob ein Punkt auf einer Strecke, einer Gerade oder einem Kreis liegt, e.g.  $A \in c$ .
- Eine Überprüfung, ob zwei Geraden oder Strecken parallel sind, e.g.  $p \parallel q$ . Hier kann auch `SindParallel [p,q]` verwendet werden.
- Eine Überprüfung, ob zwei Geraden oder Strecken normal aufeinander stehen, e.g.  $p \perp q$ . Hier kann auch `ArePerpendicular [p,q]` verwendet werden.
- `SindKollinear [A,B,C]` prüft, ob die Punkte  $A, B$  und  $C$  kollinear sind.
- `SindKongruent [d,e,f]` prüft, ob die Geraden  $d, e$  und  $f$  kongruent sind.
- `SindKonzentrisch [A,B,C,D]` prüft, ob die Punkte  $A, B, C$  und  $D$  konzentrisch sind.

### 3.2.3 Der Einhüllende Befehl

Dieser Befehl berechnet die Gleichung einer Tangente zu einer Gruppe von Objekten mit einem bestimmten Vorgänger auf einer Strecke.

Präziser gesagt, ist ein Anfangspunkt  $\mathcal{I}$  auf einer Strecke  $\mathcal{P}$ , einige Konstruktionsschritte und ein Endpunkt  $\mathcal{O}$  gegeben, entweder eine Gerade oder ein Kreis. Die Aufgabe besteht darin, die Gleichung  $\mathcal{E}$  einer Kurve  $\mathcal{C}$ , welche die Tangente zu  $\mathcal{O}$  ist, aufzustellen, wenn sich  $\mathcal{I}$  auf  $\mathcal{P}$ . Am Ende ergibt sich  $\mathcal{E}$ .  $\mathcal{I}$  wird der Beweger genannt.  $\mathcal{E}$  wird die *Einhüllende Gleichung* genannt, und ihre graphische Darstellung ist die *Einhüllende*.

#### Beispiel



1. Mit dem *Kreis mit Mittelpunkt durch Punkt Werkzeug*, wird ein Kreis  $c$  mit dem Mittelpunkt  $A$  und dem Umkreispunkt  $B$  konstruiert.
2. Setzen Sie Punkt  $C$  auf  $c$ .
3. Erstellen Sie einen beliebigen Punkt  $D$  in  $c$ .
4. Konstruieren Sie die *Mittelsenkrechte*  $f$  der Strecke  $CD$  unter Verwendung der Endpunkte.
5. Geben Sie `Einhüllende[f,C]` in der Befehlszeile ein. Die implizite Kurve  $a$  wird berechnet und gezeichnet. Diese sieht aus wie eine Ellipse.

### 3.3 Technische Notizen

Die folgenden Notizen sind wichtige Einschränkungen für jedes Automatische Begründungswerkzeug in GeoGebra, symbolische Berechnungen nutzt:

- Es werden nicht alle GeoGebra Werkzeuge und Konstruktionsschritte unterstützt.
- Die unterstützten Werkzeuge funktionieren nur für eine beschränkte Menge an geometrischen Objekten, zum Beispiel die Verwendung von Punkten, Geraden, Kreisen, Kegel.
- Abschnitte von Strahlen und Geraden werden als unendliche Geraden behandelt. Kreisbögen werden wie Kreise behandelt.

- Zu komplizierte Ortslinien- oder Einhüllende-Berechnungen können in der Algebra Ansicht als undefiniert dargestellt werden.
- Beziehungs-Beweise, deren Inhalt zu kompliziert ist, werden die Mitteilung möglicherweise allgemein wahr anzeigen. Die Interpretation dafür ist, dass es GeoGebra nicht möglich ist, zu entscheiden, ob die Beziehung im allgemeinen wahr ist, aber die numerischen Ergebnisse lassen darauf schließen. Das heißt, die Beziehung kann auch im allgemeinen falsch sein.
- Wenn es keine Ortslinie oder Einhüllende gibt, ist die implizite Kurve die leere Menge  $0 = 1$ . Zum Beispiel: Für einen beliebigen Punkt  $P$

`Ortsliniengleichung[false,P]`

zeigt die leere Menge.

- Wenn die Ortslinie oder die Einhüllende die ganze Ebene ist, hat die implizite Kurve die Gleichung  $0 = 0$ . Zum Beispiel: Für einen beliebigen Punkt  $P$

`Ortsliniengleichung[true,P]`

zeigt die ganze Ebene.

- Manchmal werden zusätzliche Verzweigungen erscheinen, die nicht in der ursprünglichen Ortslinie oder Einhüllenden waren.
- Der Graph der impliziten Kurve kann in manchen Fällen ungenau sein.

## 4 Verwendung in der Klasse: Vermutung, Beweis und Verallgemeinerung

Technisch gesehen ist das Beziehungswerkzeug das einfachsten symbolische Werkzeug aus der obigen Liste. Auf der anderen Seite können manche Unterrichtssituationen den Einsatz andere Werkzeuge, oder den Einsatz mehr als eines Werkzeuges erfordern, möglicherweise auch in einer anderen Reihung als in der obigen Liste.

### 4.1 Satz des Thales

In vielen traditionellen Mathematikstunden wird der Satz des Thales in einer expliziten Form unterrichtet: Wenn  $C$  auf einem Halbkreis liegt, sind die Strecken  $g$  und  $h$  normal aufeinander. Tatsächlich kann dieser Lehrsatz mit einer offenen Frage formuliert werden: *Sei  $ABC$  ein beliebiges Dreieck. Was ist die geometrische Ortslinie von  $C$ , wenn der Winkel bei  $C$  ein rechter Winkel sein soll?*

Bei diesem Zugang kann es sinnvoller sein, den technisch anspruchsvolleren `Ortsliniengleichung[g⊥h,C]` Befehl zuerst zu verwenden und dann die Konstruktion mit dem Beziehungswerkzeug oder -befehl zu vollenden. Der Output des `Ortsliniengleichung` Befehl kann den Schülerinnen und Schülern zeigen, dass die Kurve ein Kreis ist. Die Algebra Ansicht zeigt die Gleichung der Kurve, das könnte unter Umständen für jüngere Schülerinnen und Schüler verwirrend sein.

Schließlich kann der Satz des Thales durch das eingeschriebene Dreieck in einen Halbkreis verallgemeinert werden. In diesem Fall ist die Bedingung nicht mehr  $g \perp h$ , sondern der Winkel der beiden veränderlichen Strecken zu einer fixen. GeoGebra unterstützt diese Art der Erhebung mit der Schreibweise.

`Ortsliniengleichung[SindKongruent[α,β],C]`

wenn  $\alpha$  fix ist und  $\beta = \angle ACB$ .

Zusammenfassend, bei diesem Zugang

1. wird eine implizite Ortslinie mit GeoGebra berechnet,
2. wird von den Schülerinnen und Schülern eine Vermutung aufgestellt, wie die Kurve aussieht,
3. wird die Vermutung mit dem Beziehungswerkzeug oder -befehl in GeoGebra überprüft,
4. kann der Beweis von den Schülerinnen und Schülern optional handschriftlich durchgeführt werden,
5. wird der Satz des Thales mit den impliziten Ortslinien in GeoGebra verallgemeinert - als zukünftiges Experiment für die Schülerinnen und Schüler.

## 4.2 weitere Beispiele

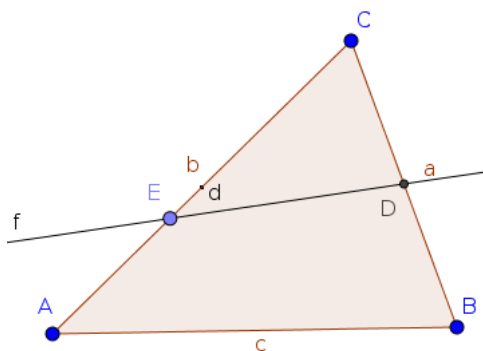
Die Dreiecksungleichung kann in eine Gleichung übersetzt werden, die in eine Erforschung der degenerierten Dreiecke geändert werden kann. Als Verallgemeinerung kann die künstliche Definition von konischen Segmenten erwähnt werden.

Eine andere Anwendung ist eine Verzweigung der Ortsliniengleichung im Dreieck  $ABC$  mit der Bedingung  $a \stackrel{?}{=} b$ , dazu muss  $C$  gefunden werden (step 1). Natürlich muss  $C$  auf der Streckensymmetrale  $AB$  liegen (step 2). Dass  $C$  explizit auf der Streckensymmetrale liegt, bestätigt GeoGebra, als  $AC = BC$ , wenn man die Berechnung mit dem Beziehungswerkzeug startet (step 3). Nach dem Beweis des Satzes mit traditionellen Methoden (step 4) kann eine Verallgemeinerung durch das Eintippen des Befehls `Ortsliniengleichung[a==2b,C]` erhalten werden. Dies kann auch ein interessantes Experiment für fortgeschrittene Schülerinnen und Schüler sein. (step 5).

## 4.3 Ein ausgearbeitetes Beispiel: Das A worked out example: The midline theorem

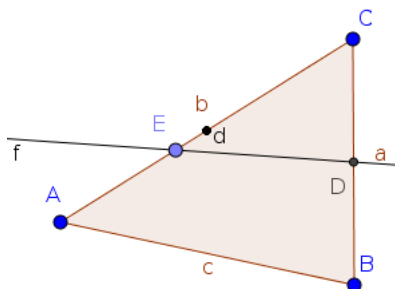
Here step-by-step instructions are provided on a possible way on investigating the midline theorem by using GeoGebra's automated reasoning tools.

### Step 1



1. By using the *Polygon* tool, construct triangle  $ABC$ . This will automatically create segments  $a$ ,  $b$  and  $c$ .
2. By using the *Midpoint or Center* tool, create the midpoint  $D$  of  $a$ .
3. Put point  $E$  on  $b$ .
4. Create line  $f$  which joins  $D$  and  $E$ .
5. Ask GeoGebra on the requirement for  $E$  in order to have  $f$  parallel to  $c$ : type `LocusEquation[c||f,E]` in the Input Bar. Now the implicit curve  $d$  will be computed and plotted, and it seems to be a single point. Note: it may be useful to change the line thickness of the implicit curve  $d$ , and also to increase its layer number to ensure that other objects do not hide it. Both settings can be changed in the Object properties window.

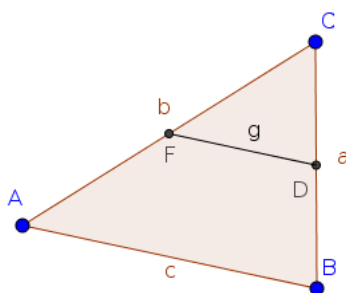
### Step 2



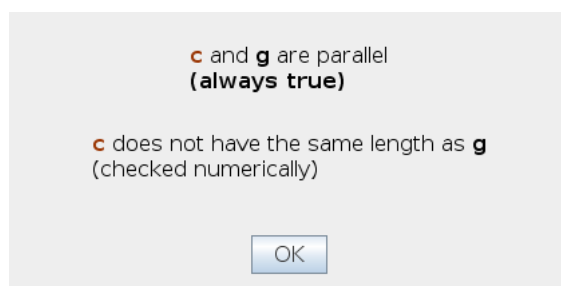
6. Drag the free objects and conjecture that  $E$  must be the midpoint of  $b$ .
7. To confirm this conjecture create midpoint  $F$  of segment  $b$  (and align labels of  $d$  and  $F$  to avoid overlapping). Drag the free objects again.



### Step 3



8. Make the objects  $E$ ,  $f$  and  $d$  invisible by hiding them.
9. Join  $D$  and  $F$  by segment  $g$ .
10. Use the *Relation* tool to compare  $c$  and  $g$ . They seem to be parallel.
11. Click “More...” in the popup window and check symbolically that they are indeed parallel.



The pupils may continue with step 4 if they need an elegant way to prove this statement, or just stop here if there is no time for further work in the classroom.

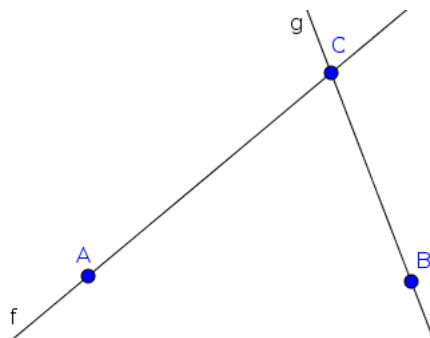
Also, in step 5 further questions can be raised.  $c$  and  $g$  do not have the same length—but can  $g$  be computed by using the length of  $c$ ? Maybe  $c = 1.5 \cdot g$ , or maybe more? The GeoGebra command `Relation[c, 1.5g]` will result in the answer that  $c$  and  $1.5g$  are not equal, but maybe there is another constant than 1.5 which results in a positive answer... Even if there is no time for further work in the classroom, some pupils may find these questions interesting and they can continue thinking on them alone or in groups—but in some sense *independently*, using the computer as an expert system.

## 5 Limitations: a case study of Thales' circle theorem

Intuitive use of GeoGebra Automated Reasoning Tools may result in unexpected outputs in some cases. This subsection explains some common mistakes during their use.

Different approaches will be discussed when investigating Thales' circle theorem.

### Approach 1

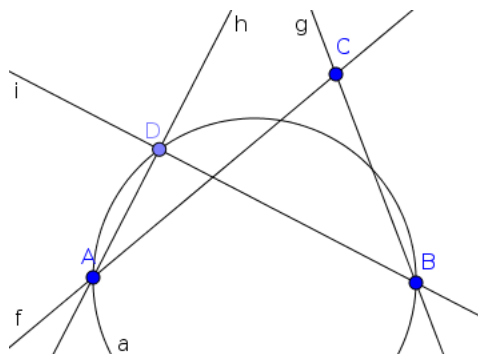


1. Create points  $A$ ,  $B$  and  $C$ .
2. Create lines  $f = AC$ ,  $g = BC$ .
3. Check the result of the command `Relation[f,g]`: “ $f$  intersects with  $g$ ”.
4. Ask GeoGebra about geometric prerequisites of  $f \perp g$ :

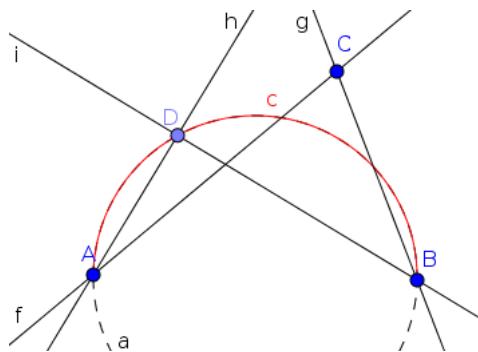
`LocusEquation[f⊥g,C]`.

An implicit curve  $a$  which seems like a circle will be shown.

5. Move  $C$  in the neighborhood of the implicit curve as close as possible. Now `Relation[f,g]` may still report that “ $f$  intersects with  $g$ ”. **Why? Because the point  $C$  may be not accurately on the circle. We need to exactly state that it is on the circle.**
  - (a) Try attaching point  $C$  on implicit curve  $a$  by using the *Attach / Detach Point* tool. **This is not allowed in GeoGebra, because by definition  $a$  depends on  $C$ , and circular dependency would make no sense.**
  - (b) Instead, create a new point  $D$  by putting it on the implicit curve. This is allowed in GeoGebra. Create also lines  $h = AD$ ,  $i = BD$ .

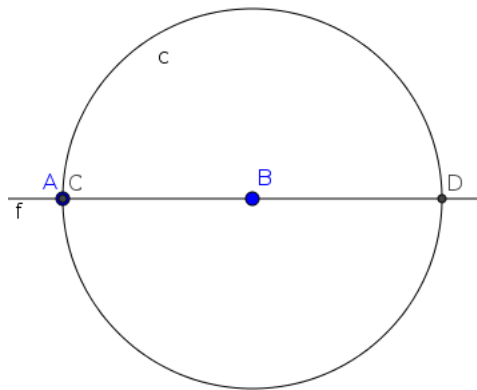


- (c) Check the result of the command `Relation[h,i]`: “ $h$  and  $i$  are perpendicular” when checked numerically. By clicking “More...” the result is however “possibly generally true”. **Why? Because GeoGebra interprets the underlying implicit curve as the result of a particular setup of the construction. In other words: an implicit curve is a numerical object, it does not have a symbolic representation. That is, symbolic checks based on an implicit curve are not possible. Here GeoGebra was just optimistic about the truth of the conjecture, but the software was actually unable to prove it.**
- (d) The proper way to finalize the steps in this approach is to create the circle with diameter  $AB$  with a Circle tool, for example by using the *Semicircle through 2 Points* tool, after detaching  $D$  from  $a$  and making  $a$  invisible. Now  $D$  can be attached to the semicircle.

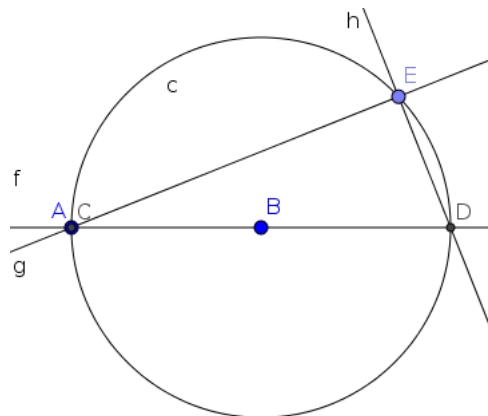


(Optionally the implicit curve can be set to visible by displaying it with a different style. In this example another style was used for the semicircle as well.) Finally `Relation[h,i]` will now result in positive outputs both numerically and symbolically.

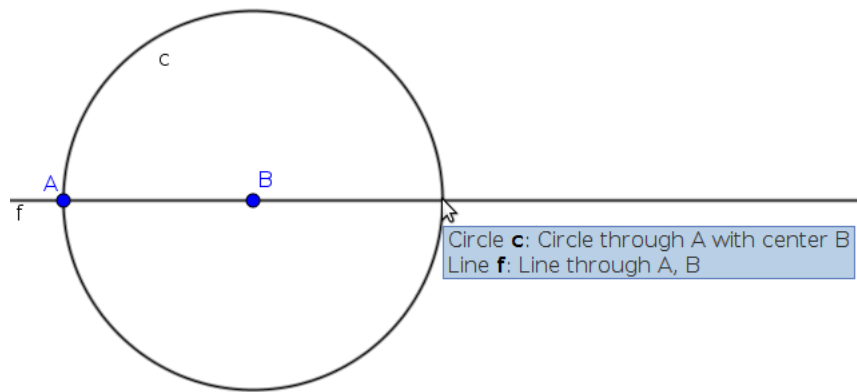
## Approach 2



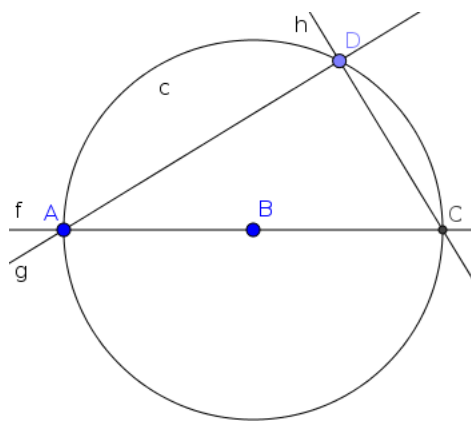
1. Create points  $A$  and  $B$ .
2. Create circle  $c$  with center  $B$  through  $A$ .
3. Draw line  $f$  by joining points  $A$  and  $B$ .
4. Create the intersection points  $C$  and  $D$  of  $c$  and  $f$ . (You may want to move the label of  $A$  away to avoid overlapping with the label of  $C$ .)
5. Put point  $E$  on  $c$ .
6. Draw lines  $g = AE$ ,  $h = DE$ .



7. Compare  $g$  and  $h$  by using  $\text{Relation}[g, h]$ . The numerical result is correct, but GeoGebra reports that the perpendicularity of  $f$  and  $g$  is not generally true. **Why? When  $C$  and  $D$  were created, it was not defined which is which. That is, the visual precondition that  $A = C$  is actually false—the other case  $A = D$  must be also allowed as a possible precondition. In that other case, of course,  $f$  and  $g$  are not perpendicular, but identical.**
  - (a) Instead, change the definition of  $g$  to  $g = CE$ . Now the relationship between  $g$  and  $h$  will be correctly reported in the symbolic case as well.
  - (b) Another option is to not create both intersection points  $C$  and  $D$ , but just one of them (which differs from  $A$ ). This is possible by clicking on the neighborhood of the sought intersection point in the diagram.



Now continuing with the creation of  $D$  (by putting it on  $c$ ),  $g = AD$  and  $h = CD$ ,  $\text{Relation}[f, g]$  will report the correct results both numerically and symbolically.



**Why? In this diagram GeoGebra assumes that  $C$  differs from  $A$  in general (the special case  $A = C$  will be excluded from the symbolic proof of the perpendicularity—but even in this side case the statement on perpendicularity will be true, that is the symbolic result will be “always true”).**

## 6 Appendix

### 6.1 Low level GeoGebra tools

Automated reasoning tools in GeoGebra are completed by some low level tools prepared for learning more and in a more accurate way about geometric properties.

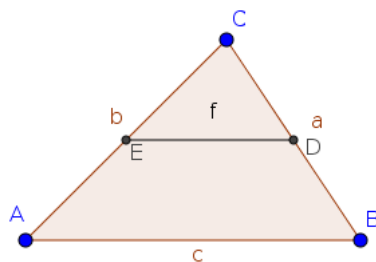
#### 6.1.1 The Prove command

The Prove command decides if a geometric statement is true in general. It has three possible outputs:

- *true* means that the statement is always true, or true under some non-degeneracy conditions.
- *false* means that the statement is false in general. GeoGebra uses algebraic geometry in many cases to decide such questions. In algebraic geometry “generally true” (true in “most” cases) and “generally false” (false in “most” cases) are not opposite properties, that is, a statement can be not “generally true” and not “generally false” at the same time. GeoGebra interprets this special case as *false* (since it is not generally true).
- *undefined* means that GeoGebra cannot decide because of some reason:
  - The statement cannot be translated into a model which can be further investigated. This usually means that algebraization of the statement failed because of
    - \* theoretical impossibility (e.g. using a transcendent function as a construction step, for example, sine of  $x$ ),
    - \* missing implementation in GeoGebra.

- The translated statement in algebraic geometry is too difficult to solve. This means that either there are too many variables, or the equations are hard to handle by the solver algorithm. This results in either a timeout or an out of memory error.
- The solver algorithm was able to investigate the situation, but the result is ambiguous: either the statement is false, or it is true under certain conditions—but the algorithm was not able to decide which case is present.
- There was an internal error in GeoGebra during the computations.

## Example



1. Construct the triangle  $ABC$  by using the *Polygon* tool.
2. Construct the midpoints  $D$  and  $E$  of sides  $a$  and  $b$ , respectively, by using the *Midpoint or Center* tool.
3. By using the *Segment* tool, create  $f$  by joining  $D$  and  $E$ .
4. Type `Prove[f||c]` to obtain *true* in the Algebra View as Boolean Value  $d$ . Note that the parallel sign must be entered by using either
  - the list of the mathematical symbols by clicking the  $\alpha$  icon in the Input Bar, or
  - inserting this symbol externally by using Copy and Paste.
  - Alternatively,  $f||c$  can be substituted by `AreParallel[f,c]` also.
5. Type `Prove[c==3f]`. Now the answer is *undefined*, because GeoGebra cannot decide if the statement is false or it is true under certain conditions. In such cases the `ProveDetails` command can help (see below). Note that *two* equal signs must be entered; another possibilities are to use
  - `Prove[c?=3f]`, or
  - `Prove[AreEqual[c,3f]]`.

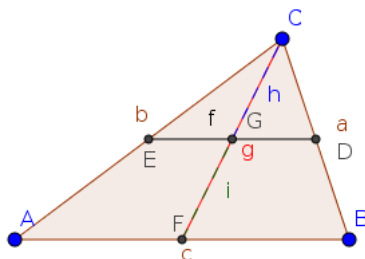
### 6.1.2 The ProveDetails command

The `ProveDetails` command has similar behavior like the `Prove` command has, but it may use different algorithms in the decision process, and may provide more information on the results. It has three possible outputs:

- $\{true\}$  means that the statement is always true.
- $\{true, \{\dots\}\}$  if the statement is true under some non-degeneracy or essential conditions: these conditions are listed in the internal braces. (If the list remains “...”, it means that no synthetic translation could be found.) If the conjunction of the negated conditions is true, then the statement is true.
- $\{false\}$  means that the statement is false in general. See the comments at the `Prove` command for more details on this.

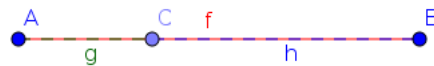
## Example (continued)

6. Type `ProveDetails[c==3f]`. Now the answer is  $\{false\}$ .
7. Type `ProveDetails[c==2f]`. Now the answer is  $\{true\}$ .



8. Now let  $F$  be the midpoint of  $c$ , and let us denote segment  $CF$  by  $g$ . Let  $G$  be the intersection point of  $f$  and  $g$ . Finally, let us denote segments  $CG$  and  $FG$  by  $h$  and  $i$ , respectively. In this case `ProveDetails[h==i]` returns  $\{true, \{“AreCollinear[A,B,C]”\}\}$  which means that if  $A$ ,  $B$  and  $C$  are not collinear, then  $h = i$ .

### Another example



1. Let  $AB$  a segment, denoted by  $f$ .
2. Let  $C$  be a point on  $f$ .
3. Let us denote segments  $AC$  and  $BC$  by  $g$  and  $h$ , respectively.
4. Type `ProveDetails[f==g+h]`. Now the answer is

$$\{true, \{ "g+f=h", "h+f=g" \} \}$$

which means that if  $g + f \neq h$  and  $h + f \neq g$ , then  $f = g + h$ .

### 6.1.3 A comparison of Prove, ProveDetails and Relation

The following table explains the meanings of the outputs of the commands `Prove`, `ProveDetails` and `Relation`.

GeoGebra outputs			Conclusion
Prove	ProveDetails	Relation's symbolic window	
true	{true}	always true	The statement is true.
	{true,{conditions}}	true under <i>conjunction of specified conditions</i>	The statement is true if the specified <i>conditions</i> hold. These conditions are sufficient, but maybe not necessary. There may be other sufficient conditions to make the statement true.
	{true,{...}}	true under certain conditions	The statement is true if certain equations hold. These equations have no visually clear geometric meanings for GeoGebra.
	{}	generally true	The statement is true if certain conditions hold. GeoGebra was unable to find these conditions due to computational difficulties.
false	{false}	not generally true	The statement is false.
	{}	possibly generally true	GeoGebra was unable to decide if the statement is true or false. The numerical check confirms the truth, but the symbolic check was unsuccessful due to computational difficulties.
undefined	{true}	always true	The statement is true.
	{true,{conditions}}	true under <i>conjunction of specified conditions</i>	The statement is true if the specified <i>conditions</i> hold. These conditions are sufficient, but maybe not necessary. There may be other sufficient conditions to make the statement true.
	{true,{...}}	true under certain conditions	The statement is true if certain conditions hold. These equations have no visually clear geometric meanings for GeoGebra.
	{false}	generally false	The statement is false.
	{}	possibly generally true	GeoGebra was unable to decide if the statement is true or false. The numerical check confirms the truth, but the symbolic check was unsuccessful due to computational difficulties, or the symbolic check for the given statement is not yet implemented in GeoGebra.

## 6.2 Debugging

Starting GeoGebra via command line there are more possibilities to investigate the results. Here the method on a typical Linux installation is demonstrated.

The user needs to start GeoGebra by the following command:

```
geogebra --logfile=/dev/stdout --logshowcaller=false \
--logshowtime=false --logshowlevel=false
```

A typical output looks like as follows:

```
Using AUTO
Using BOTANAS_PROVER
A = (3.42, 1.86) /* free point */
// Free point A(v1,v2)
B = (10.48, 3.1) /* free point */
// Free point B(v3,v4)
f = Segment[A, B] /* Segment [A, B] */
C = Point[f] /* Point on f */
// Constrained point C(v5,v6)
Hypotheses:
1. -v5*v4+v6*v3+v5*v2-v3*v2-v6*v1+v4*v1
g = Segment[A, C] /* Segment [A, C] */
h = Segment[C, B] /* Segment [C, B] */
Processing numerical object
Hypotheses have been processed.
giac evalRaw input: evalfa(expand(ggbtmpvarf))
giac evalRaw output: ggbtmpvarf
input = expand(ggbtmpvarf)
result = ggbtmpvarf
eliminate([ggbtmpvarf-((ggbtmpvarg)+(ggbtmpvarh))=0,ggbtmpvarh^2=v11^2,ggbtmpvarg^2=v12^2,ggbtmpvarf^2=v13^2],[ggbtmpvarh,
ggbtmpvarg,ggbtmpvarf])
giac evalRaw input: evalfa(eliminate([ggbtmpvarf-((ggbtmpvarg)+(ggbtmpvarh))=0,ggbtmpvarh^2=v11^2,ggbtmpvarg^2=v12^2,
ggbtmpvarf^2=v13^2],[ggbtmpvarh,ggbtmpvarg,ggbtmpvarf]))
Running a probabilistic check for the reconstructed Groebner basis. If successfull, 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 0.000448 Memory -1e-06M
giac evalRaw output: {v11^4-2*v11^2*v12^2+v12^4-2*v11^2*v13^2-2*v12^2*v13^2+v13^4}
input = eliminate([ggbtmpvarf-((ggbtmpvarg)+(ggbtmpvarh))=0,ggbtmpvarh^2=v11^2,ggbtmpvarg^2=v12^2,ggbtmpvarf^2=v13^2],[
ggbtmpvarh,ggbtmpvarg,ggbtmpvarf])
result = {v11^4-2*v11^2*v12^2+v12^4-2*v11^2*v13^2-2*v12^2*v13^2+v13^4}
giac evalRaw input: evalfa(eliminate([ggbtmpvarf-((ggbtmpvarg)+(ggbtmpvarh))=0,ggbtmpvarh=v11,ggbtmpvarg=v12,ggbtmpvarf=
v13],[ggbtmpvarh,ggbtmpvarg,ggbtmpvarf]))
Running a probabilistic check for the reconstructed Groebner basis. If successfull, 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 0.000592 Memory -1e-06M
giac evalRaw output: {v11+v12-v13}
input = eliminate([ggbtmpvarf-((ggbtmpvarg)+(ggbtmpvarh))=0,ggbtmpvarh=v11,ggbtmpvarg=v12,ggbtmpvarf=v13],[ggbtmpvarh,
ggbtmpvarg,ggbtmpvarf])
result = {v11+v12-v13}
giac evalRaw input: evalfa(simplify({v11^4-2*v11^2*v12^2+v12^4-2*v11^2*v13^2-2*v12^2*v13^2+v13^4}/{v11+v12-v13}))
giac evalRaw output: {v11^3-v11^2*v12+v11^2*v13-v11*v12^2-2*v11*v12*v13-v11*v13^2+v12^3+v12^2*v13-v12*v13^2-v13^3}
input = simplify({v11^4-2*v11^2*v12^2+v12^4-2*v11^2*v13^2-2*v12^2*v13^2+v13^4}/{v11+v12-v13})
result = {v11^3-v11^2*v12+v11^2*v13-v11*v12^2-2*v11*v12*v13-v11*v13^2+v12^3+v12^2*v13-v12*v13^2-v13^3}
giac evalRaw input: evalfa(factor(v11^3-v11^2*v12+v11^2*v13-v11*v12^2-2*v11*v12*v13-v11*v13^2+v12^3+v12^2*v13-v12*v13^2-
v13^3))
giac evalRaw output: (v11-v12-v13)*(v11-v12+v13)*(v11+v12+v13)
input = factor(v11^3-v11^2*v12+v11^2*v13-v11*v12^2-2*v11*v12*v13-v11*v13^2+v12^3+v12^2*v13-v12*v13^2-v13^3)
result = (v11-v12-v13)*(v11-v12+v13)*(v11+v12+v13)
Trying to detect polynomial -v13-v12+v11
-v13-v12+v11 means h = f + g
Trying to detect polynomial v13-v12+v11
v13-v12+v11 means f + h = g
Trying to detect polynomial v13+v12+v11
v13+v12+v11 means f + g + h = 0, uninteresting
Thesis equations (non-denied ones):
2. v11^2-v6^2-v5^2+2*v6*v4-v4^2+2*v5*v3-v3^2
3. v12^2-v6^2-v5^2+2*v6*v2-v2^2+2*v5*v1-v1^2
4. v13^2-v4^2-v3^2+2*v4*v2-v2^2+2*v3*v1-v1^2
Thesis reductio ad absurdum (denied statement), product of factors:
(v13^4-2*v13^2*v12^2+v12^4-2*v13^2*v11^2-2*v12^2*v11^2+v11^4)*v14-1
that is,
5. -1+v14*v13^4-2*v14*v13^2*v12^2+v14*v12^4-2*v14*v13^2*v11^2-2*v14*v12^2*v11^2+v14*v11^4
substitutions: {v1=0, v2=0}
Eliminating system in 8 variables (5 dependent)
giac evalRaw input: evalfa([ff:=\"\",[aa:=eliminate2([v12^2-v6^2-v5^2,v11^2-v6^2-v5^2+2*v6*v4-v4^2+2*v5*v3-v3^2,-1+v14*
v13^4-2*v14*v13^2*v12^2+v14*v12^4-2*v14*v13^2*v11^2-2*v14*v12^2*v11^2+v14*v11^4,v13^2-v4^2-v3^2,-v5*v4+v6*v3],
revlist([v6,v11,v12,v13,v14]))],[bb:=size(aa)],[for ii from 0 to bb-1 do ff+=\"[\"+(ii+1)+\"]\": [i]:
unicode95uunicode91u1]=1\");cc:=factors(aa[ii]);dd:=size(cc);for jj from 0 to dd-1 by 2 do ff+=\"[\"
unicode95uunicode91u1]+(jj/2+2)+\"]=\"+cc[jj]); od; ff+=\"[2]: \" +cc[1]);for kk from 1 to dd-1 by 2 do ff+=\"[,\"+
cc[kk]);od;od],[if(ff=\\\"\\\") begin ff:=[0] end],[ff][5])
Running a probabilistic check for the reconstructed Groebner basis. If successfull, error probability is less than 1e-07
and is estimated to be less than 10^-7. Use proba_epsilon:=0 to certify (this takes more time).
// Groebner basis computation time 0.000249 Memory -1e-06M
giac evalRaw output: "[1]: [1]: unicode95uunicode91u1]=1 unicode95uunicode91u2]=1 [2]: 1,1"
```



```
Considering NDG 1...
Found a better NDG score (0.0) than Infinity
Statement is GENERALLY TRUE
Benchmarking: 38 ms
STATEMENT IS TRUE (yes/no: TRUE)
OUTPUT for ProveDetails: null = {true, {"f + h = g", "h = f + g"}}
```

There is intentionally no easier way to show the users this type of output. However, the last few lines of the debug information is available in GeoGebra in the *Help* menu, by choosing *About/License*, and clicking *System Information*—this copies the latest debug messages into the clipboard.

### 6.3 Translation of GeoGebra commands

The names of GeoGebra automated reasoning tools may need to be translated to other languages. For example, the German translation of Prove can be Prüfe. To learn the translated command names the following steps are recommended:

1. Create a GeoGebra file which contains the required commands in the Algebra View.
2. Change the language in GeoGebra in the *Options* menu by choosing *Language*.
3. The command names will be automatically changed in the Algebra View.
4. Move the mouse over a command in the Algebra View and read its translated name off.