OK Geometry Plus

Reference for OK Geometry Plus (v.22)

Zlatan Magajna

June 2026

Contents

1 Introduction

OK Geometry Plus is an extension of OK Geometry Basic. It consists of several independent modules that are functionally integrated into OK Geometry Basic. This document should therefore be thus considered a continuation of the OK Geometry Basic references.

The first module is dedicated to **triangle geometry**. It deals with many objects and operations commonly used in triangle geometry. It contains also elements of geometry of quadrilaterals. Not only can you construct such objects, but you can also observe how a geometric object relates to an extensive set of triangle objects.

The second module provides a more **in-depth query** of objects in a dynamic construction. It relates a given object not only to other objects in the configuration, but also to many other objects derived from the given object. Thus, it can be of great use in hypothesising solutions to construction problems.

The third module **observes algebraic relationships** between geometric quantities in dynamic constructions. It can help you hypothesise how a geometric quantity (distance, radius of a circle, area of a triangle, etc.) is related to other geometric quantities in the configuration.

The fourth module **deductively proves properties** of geometric constructions. The proving technique is based on the Geometric Deductive Database (GDD) method by Chou, Gao and Zhang. The proving mechanism, when successful, leads to a readable deductive proof.

The fifth module deals with **generic constructions**. You can think of generic constructions as families of constructions obtained with different operations at certain construction steps. This module allows you to create families of dozens or even hundreds of different constructions at the same time. The entire family of constructions can be observed simultaneously, you can perform triangle analysis or algebraic observation on the family, and you can identify hard-to-prove properties within all the family.

Finally, the Plus version introduces **archives of dynamic constructions**, a useful and manageable way to store a large number of dynamic constructions.

The Plus mode of OK Geometry can be accessed via the command *Configure|Working mode|Plus* in the main menu (Figure 1).

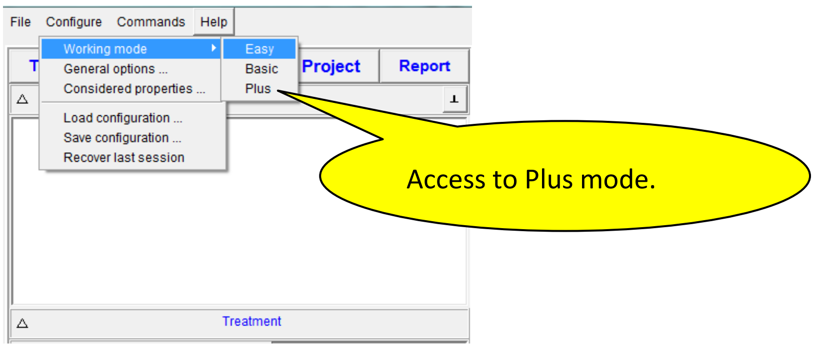


Figure 1

2 Observing and creating triangle objects

The core of OK Geometry Plus is a large database of

- triangle centres (e.g. symmedian point, currently about 60 000 of them),
- triangle related triangles (e.g. orthic triangle, currently more than 230 of them),
- triangle related lines (e.g. Euler line, currently 25 lines, more that 2500 including those of related triangles),
- triangle related circles (e.g. 9-point circle, currently 92 circles, more than 7500 including the characteristic circles of related triangles),
- triangle related conics (e.g. Feuerbach hyperbola of a triangle, currently 39 conics),
- triangle cubics (e.g., Neuberg cubic of a triangle, currently more than 1000 of them),
- triangle operations that associate a point, line, circle, triangle, or conic to a given point (e.g. isogonal conjugation of a point, currently 49 operations),
- quadrilateral objects (e.g., Euler-Poncelet point, Newton line, currently about 100 objects).

The database elements can be used directly in constructions as well as in observation of objects. The Plus mode comprises three main procedures:

- Simple observation of triangle objects
- Using triangle objects in constructions, cyclic constructions
- Advanced observation of triangle objects

In the final section we present some examples.

Triangle geometry involves a multitude of concepts (various objects and operations). In order to facilitate the work, a help system was developed. It is explained in the Section 2.1.

2.1 The help system

The help system of OK Geometry Plus consists of two tools: *Triangle commands* and (triangle) *Glossary*. Both tools are accessible in all triangle related texts in OK Geometry, including in texts inside the tools themselves.

2.1.1 The Glossary

The (triangle) Glossary can be accessed with the command *Help|Glossary* or with the **F1** key. For a given term the glossary provides a list of concepts related to that term. We explain how to use of glossary with an example.

Suppose we come across the term ‘Gergonne point’. We may not know what it is and where to look in OK Geometry for the related commands. We call up the Glossary (by pressing **F1**) and enter the (approximate) term - for example ‘gergone’.

We now explain the form that appears (see Figure 2).

All entries related to the term under consideration are grouped into three categories (see the red arrow in Figure 2):

Commands. Here are grouped the standard commands of dynamic geometry that relate to the term under consideration (e.g. Perpendicular connector, Arc 3pts). Clicking on an entry displays the help for the command. With a right-click on the description you can also access the command. In our case, the group of commands referring to 'gergone' (or something similar) is empty.

Special. Here are grouped the entries from **triangle geometry** that refer to the term under consideration. In most cases the entries correspond to a command from the Sketch Editor menu Special, but some entries are only explanation of a term. The group includes also some basic **quadrilateral objects** (see section 2.8) from the *Encyclopedia of quadric-figures*[1], their names begin with QA-, QL or QG-. In our case, 12 entries were found. In the first 4 of them, 'gergone' (or something similar) is included in the name of the entry/command. The remaining entries have a '-' prefix since 'gergone' (or something similar) is included only in the explanatory text of the entry. With a right-click on the description you can access each command.

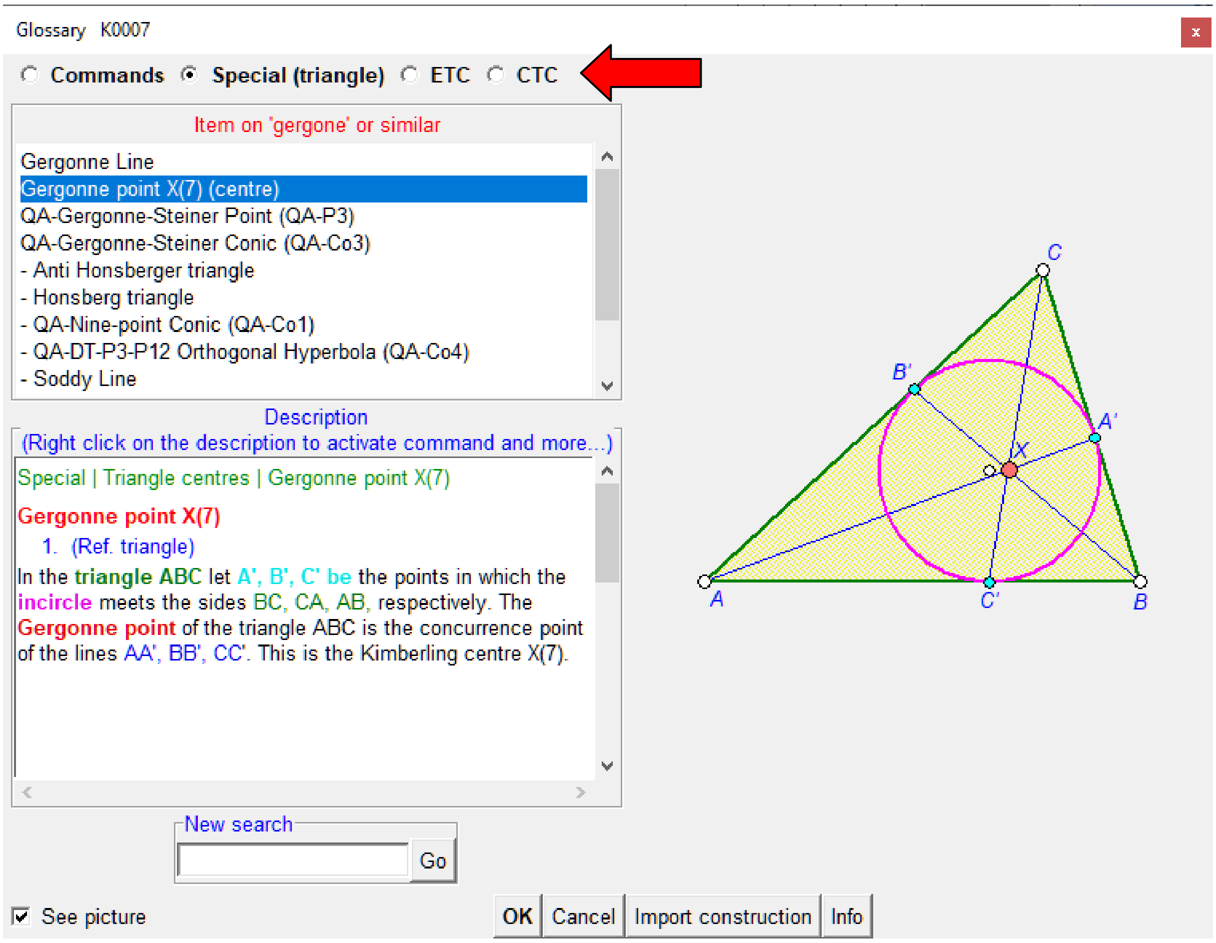


Figure 2

ETC. Here are grouped the centres from the *Encyclopedia of triangle centers*[2] that contain the string 'gergone' (or something similar) in their name. There is no explanatory text for these entries, but you can access the commands for constructions these centres via the glossary. In our case this group consists of about 35 entries.

[1] https://chrisvantienhoven.nl/mathematics/encyclopedia

[2] https://faculty.evansville.edu/ck6/encyclopedia/etc.html

CTC. Here are grouped those triangle cubics from the *Catalogue of triangle cubics*[3] that contain the string 'gergone' (or something similar) in their name. In most cases there is no explanatory text for these entries, but you can access the commands for constructions these cubics via the glossary. In our case this group consists of 3 entries.

So, entries of the glossary for triangle geometry can be found under the category **Special**. Clicking any article (entry) will take you to the information in that article. In our case, we click on the 2nd entry ('Gergonne point'), and the form is filled in, see Figure 2.

The form contains a **description** and an **illustration** of the selected term. Since there is a command related to the Gergonne point, the description also contains the location of the command in the menu (first line) and the arguments of the command.

A right click on the description leads to a menu with further help.

Explain terms turns the cursor into **a question mark with an arrow**. A click of the arrow on a term in the description (e.g. 'incircle') shows an explanation of the meaning of the term (possibly including the adjacent words).

Glossary help turns the cursor into **a spider**. A click of the spider on a term in the description activates the glossary for that term.

Execute command activates the considered command (some commands can be executed only in the Sketch Editor). In our case the list of commands for triangle centres appears with the command *Gergonne point* selected.

2.1.2 Triangle commands

The *Help|Triangle commands* displays the list of all commands for the creation of triangle related objects and operations. The list can be restricted to commands that create a specific type of objects. In the Filter entry write the exact word to look for (e.g., 'gergonne') or an approximate one with a question mark at the end (e.g., 'gergone?'). For each command there is a description, which has the same functionality as the description in the glossary. The command can be therefore activated by a right click on the description.

2.2 Reference triangle

Triangle objects refer to a specific triangle under consideration. Operations involving triangle objects therefore require information about the reference triangle. In some cases, the reference triangle is specified as a parameter of the operation, but, in general, the operations use (assumed) reference triangle, which is set in advance using the *Commands|Set reference triangle command* or in the Sketch Editor using the *Set reference triangle* command (Figure 3). The reference triangle is declared by specifying its (three) vertices, which must be labelled. An empty entry means there will be no (assumed) reference triangle. The default reference triangle can be changed at any time. On the display, the reference triangle appears pale yellow.

[3] http://bernard-gibert.fr

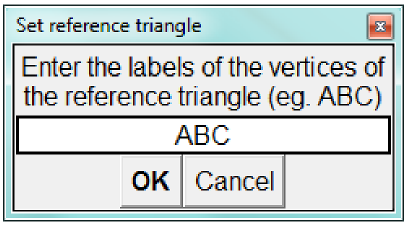


Figure 3

If a triangle is the main object of investigation, it is good practice to <u>set the reference triangle at the very beginning of a construction</u>. This is done with the command *A triangle*, which creates a triangle ABC (and also sets its cyclic structure, see Section 2.5). You can choose between two common ways of labelling the vertices of the reference triangle: A left or A top (see Figure 4). At the same time you can also choose the way of labelling triangle centres, e.g., X(5), X5, or simply 5. Both options can be set with the *Configure|General options|Sketch|Triangle centre notation* command.

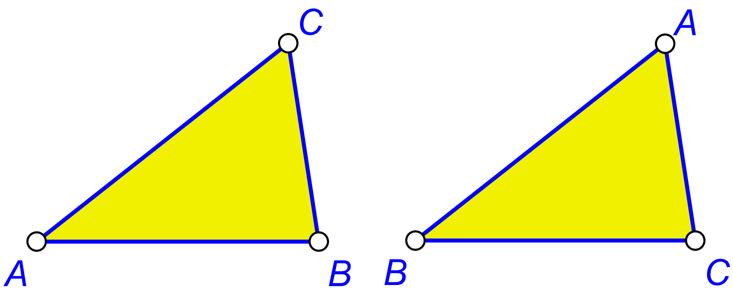


Figure 4

2.3 Simple observation of triangle objects

We describe here how to relate constructed objects to a reference triangle. We explain this with a simple example[4].

Example 1

Let A', B', C' be the base points of the altitudes from the vertices A, B, C of the triangle ΔABC. Furthermore, let A'', B'', C'' be the midpoints of B'C', C'A', and A'B'. By observing the configuration or otherwise, we note that the lines AA'', BB'', CC'' (apparently) meet at a common point P. We wonder how is this point related to ΔABC (Figure 5).

[4] OKExamples\OKG_Plus\Triangle_01.p

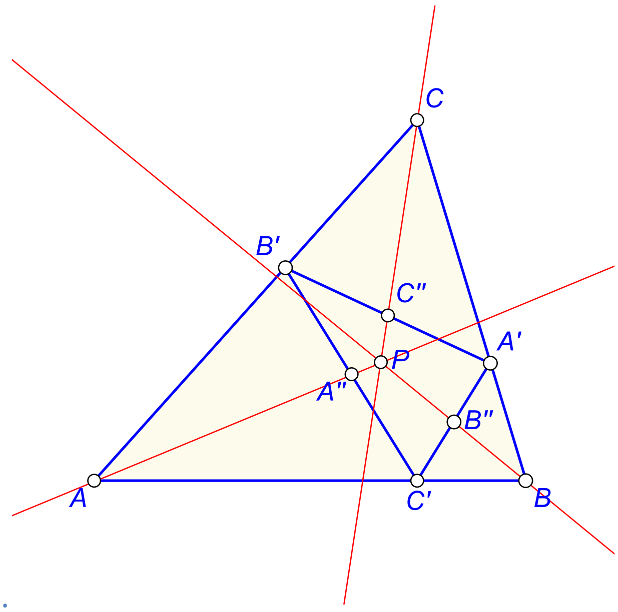


Figure 5

We can easily create the described configuration with standard commands of Sketch Editor.
If we are not in Plus mode, we need to switch to it – from the menu at the top select the command *Configure|Working mode|Plus* .

To observe, click on the command 

on the taskbar on the right or *Analyse object wrt. triangle*. Then click on the object(s) you want to know something about, in our case the point P.

If the reference triangle has not been defined yet, you need first set the reference triangle (see Section 2.2).

After a short calculation a form appears with the observational results (Figure 6).

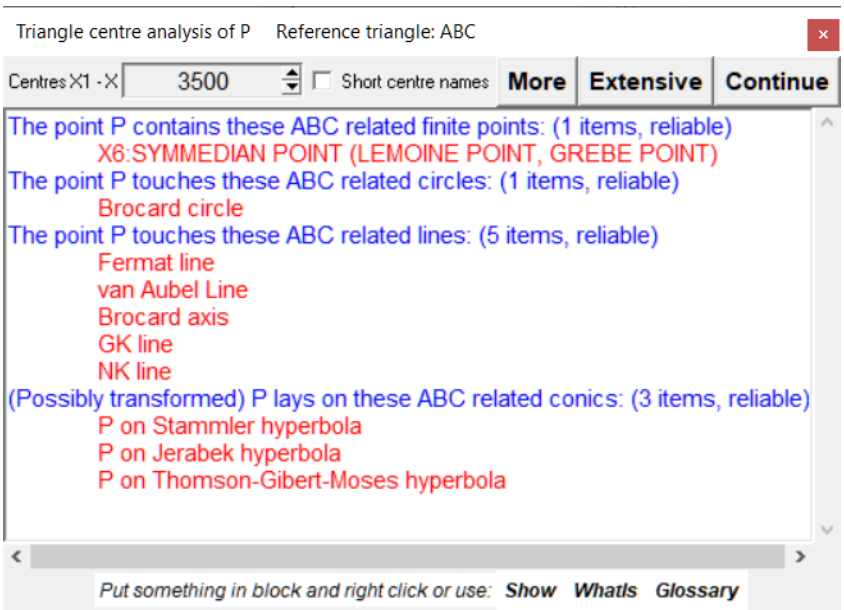


Figure 6

OK Geometry observes that the point P appears to be the symmedian point (commonly denoted as X6) of the triangle ∆ABC and presents some additional observed facts related to the point P. To view more information about the object P (more properties and more considered objects), click on the ***More*** button. The ***Extensive*** button will display an extensive list of properties involving a large set of triangle objects.

In the form (Figure 6) we can specify the range of centres (from the ETC list of triangle centres[5]) to be used in the analysis. For the most exhaustive analysis choose the option 16342+bic, which takes into account all the first 16341 centres, about 200 bicentres and many centres (up to the index 60000). The option *59992 takes into account the first 59992 centres, but performs only a limited and less reliable analysis.

The objects that are mentioned in the list of properties can be readily displayed. For example, to display the Fermat line, put the text Fermat line in a block (or simply position the cursor on the text) and click the ***Show*** command – the Fermat line will appear on the construction (Figure 7). To add at once several objects mentioned the list to the construction, simply put the objects in a block and click the ***Show*** command.

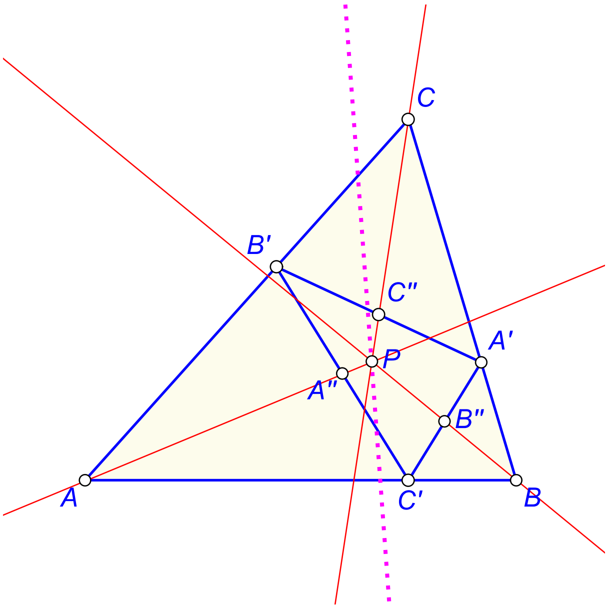


Figure 7

You can also obtain an explanation of the terms in the list of properties. For example, to obtain some basic information of the Fermat line (or some other term), click on the ***WhatIs*** command (a question mark with arrow will appear) and then click on the term with the (question mark) pointer. An explanation of the Fermat line will appear (Figure 8).

Finally, there is a glossary that relates the terms in the list of properties to other similar terms in triangle analysis. For example, to find articles related to symmedian point, click on the ***Glossary*** command (a spider pointer will appear) and then click on the term Symmedian point with the (spider) pointer. The list of articles related to symmedian will appear (Figure 9).

[5] ETC is acronym for Encyclopedia of Triangle Centers (https://faculty.evansville.edu/ck6/encyclopedia/ETC.html).

Some additional commands related to the form (Figure 6) are available with a right-click on the Description field.

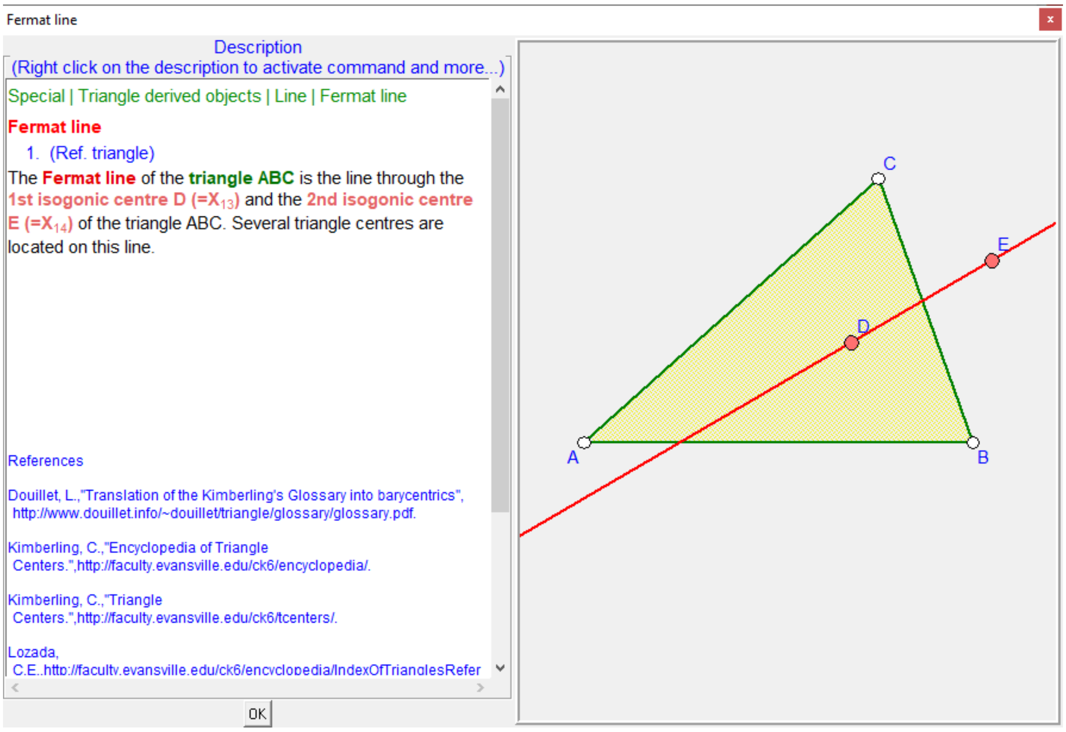


Figure 8

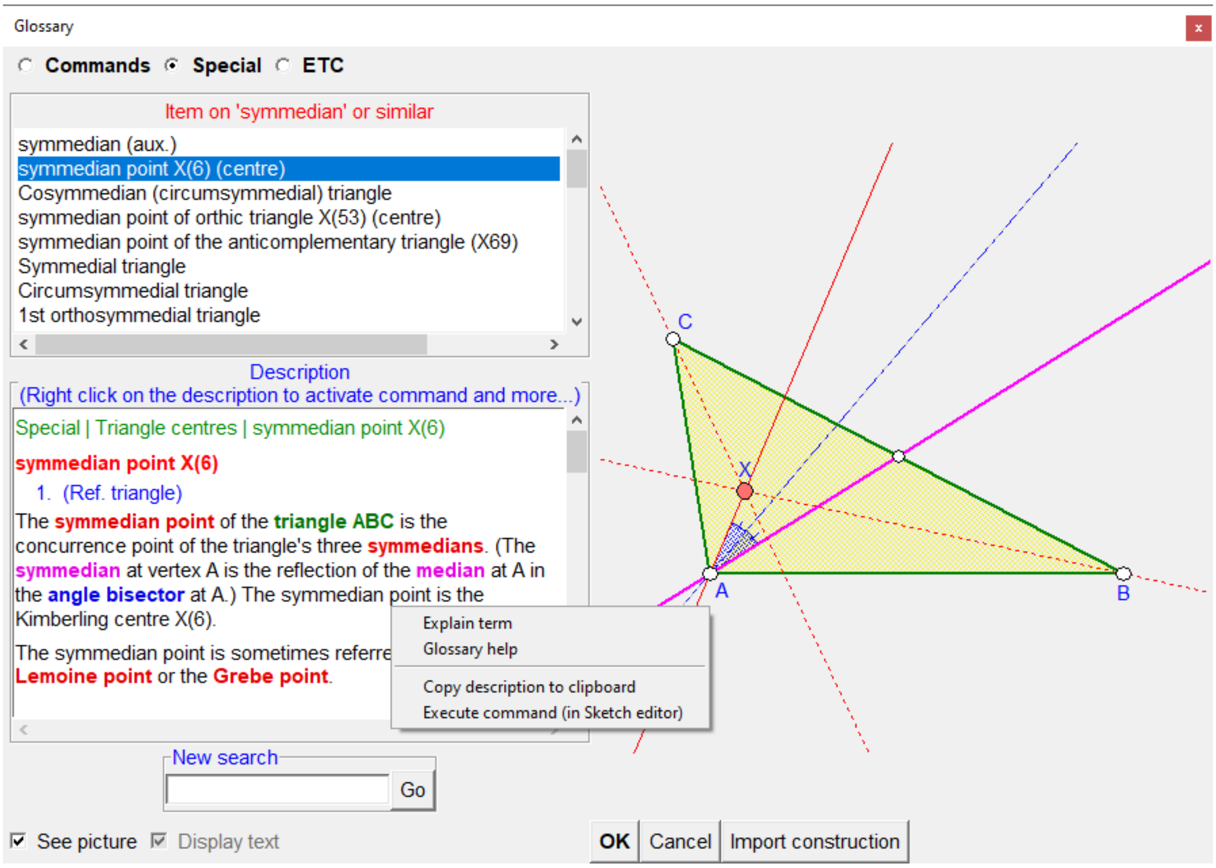


Figure 9

2.4 Special commands for triangles

The Sketch Editor in Plus mode has an additional group of commands named **Special**. The vast majority of commands in this group are specific to triangle geometry and most of them have similar syntax.

The Special commands allow you to construct various objects related to a triangle. The triangle the objects refer to is called the reference triangle. Since the same triangle is often used repeatedly as a reference triangle, it is common to declare the assumed reference triangle and use it for most commands in the group. This is done with the command *Commands|Set reference triangle* or *Special|Set reference triangle* (see Section *2.2*).

Figure 10 shows a typical group of commands (characteristic objects of a triangle).

- In the first line you select the type of object you want to create (a line, in our case).
- The commands in this group (with a few exceptions) refer to the default reference triangle, but it is always possible to specify a different reference triangle *as the first argument of the command*. Note the red line with the checkmark at the bottom, which indicates that the default reference triangle will be used unless the checkmark is removed.
- Also note the checkmark *See picture*. When this checkmark is set, a pictorial representation of the object appears on the right side.

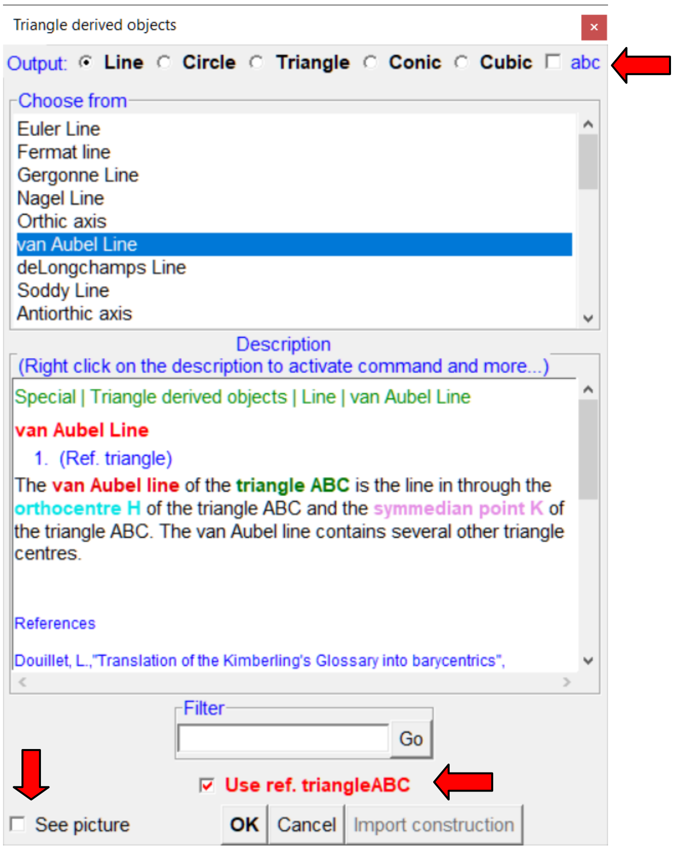


Figure 10

In many cases the form contains dozens, hundreds, or even more (sub)commands. Sometimes they are grouped by the type of object to be created. To make the selection easier, you can ***arrange*** the (sub)commands alphabetically. There is also a ***filter*** to reduce the number of entries in the list. If you enter 'malfatti' in the filter, only entries that contain the string 'malfatti' will be displayed. If you type 'malfati?', only entries containing something similar to 'malfati' will be displayed in the list.

In the Special menu the commands are grouped according to the input objects:

Triangle centres	Triangle centres are organised in five lists: - **simple centres** - the four classic triangle centres; - **advanced centres** - the most known centres, e.g. isodynamic point; - **ETC centres** - the first 16341 centres of the ETC list of triangle centres; - **ETC+ centres** - the centres from 16342 on of the ETC list of triangle centres; - **bicentric** - some well known triangle bicentres, e.g., the Brocard points. A standard notation is used for the centres from the ETC list: X_n, X*n* or X(*n*) for the *n*-th centre.
Triangle derived objects	The classic triangle objects, other than points, are organised in several lists: - **triangle lines**, e.g., the Euler line; - **triangle circles**, e.g., the Brocard circle; - **triangle triangles**, e.g., the orthic triangle; - **triangle conics**, e.g., the Kiepert hyperbola; - **triangle cubics**, e.g., the Neuberg cubic.
Triangle/Point derived objects	Various objects can be associated to a point with respect to a reference triangle. Such objects are organised in several lists, depending on the type of the objects. The considered type of resulting objects are: - **points**, e.g., the isogonic conjugation of a point in a reference triangle; - **lines**, e.g., the trilinear polar of a point in a reference triangle; - **circles**, e.g., the Cevian circle of a point; - **triangles**, e.g., the pedal triangle of a point in a reference triangle; - **conics**, e.g., the inconic of a point.

Triangle/Point/Point derived objects	The objects that are defined by 2 points with respect to the reference triangle are organised into these groups: -points, e.g., cross conjugate of two points; -conics, e.g., bicevian conic of two points.
Objects by triangle centres	This command allows an effective way to define and visualise triangle related objects passing through specified triangle centres. Simply enter the centres as indicated by the following examples: 5 ETC centre X(5), nine points centre 1-100 The first 100 centres (non all labelled) (2,14) The line through X(2) and X(14) (2,14,15) The circle through X(2),X(14),X(15) (5,6,7,8,9) The conic through X(5),X(6),X(7),X(8),X(9) (1,3,4,5,17,18) The cubic through the vertices of the reference triangle and X(1),X(3),X(4),X(5),X(17),X(18) 2,4,(3,6) The centres X(2) and X(4) and the line through X(3) and X(6)
General triangle derived point	With this command you can analytically define points, point transformations or point operations in a reference triangle. In the definition you can use barycentric, trilinear or tripolar coordinates as well as some triangle parameters. See Figure 13 for an example.
Various constructions	Here are collected various constructions of points, lines, circles, conics, and cubics. The commands are grouped into categories (see the first line in the form) according to the type of constructed object. The arguments of the commands vary from command to command. However, if the first argument of a command is a triangle then you can use the reference triangle as the first argument (by checkmarking the appropriate field, see Figure 10). In some commands you can add to the construction also auxiliary objects. Here are just some examples of commands: - **perspectivity centre** of two triangles (if it exists); - **insimilicentre** of two circles; - **polar** of a point with respect to a conic; - **eulerologic centre** of two triangles (if it exists); - **perspeconic** conic of two perspective triangle.

Set reference triangle	With this command you define the reference triangle, which is (optionally) used by default in other special commands. The vertices of the reference triangle must be labelled. To clear the reference triangle, enter in the form an empty line. OK Geometry uses standard notation for geometric quantities that refer to the reference triangle (or to the triangle that takes the place of the reference triangle). If the vertices of this triangle are PQR (in that order), then *a, b, c* denote the lengths of the sides *a* = |QR|, *b* = |RP|, *c* = |PQ|. The sizes of the angles are denoted by A (size of the angle at P), B (size of the angle at Q), C (size of the angle at R) and so on.
A triangle	This command is active only at the very beginning of the construction. It generates a 'random' triangle ABC with labelled vertices and its cyclic structure. You can set the way of labelling the vertices in *Configure|General options|Sketch|Triangle centre notation*.
Declare cyclic objects	With this command you declare three objects (of the same type) as cyclic in accordance to the current cyclic construction of the reference triangle. When the Sketch Editor is creating a 'cyclic construction', any command that involves one or more cyclic objects is repeated cyclically on the related triads of objects. Cyclic structure often significantly simplifies geometric constructions on triangles. Please, refer to Section 2.5 for more.
Cyclic construction	Turns the **Cyclic** construction flag ON and OFF. At ON, the display pointer takes the shape similar to a small circle and the commands are executed cyclically with respect to the reference triangle. Cyclic structure often significantly simplifies geometric constructions on triangles. Please, refer to Section 2.5 for more.
Detect cyclic perspectivities	Turns ON/OFF automatic detection of cyclic perspectivities. When this flag is ON, whenever in a cyclic construction a point (actually, a triplet of points) is constructed it is automatically checked whether the triangle from the constructed triplet of points is in some way perspective to the reference triangle. Please, refer to Section 2.5.1 for details.

Analyse object wrt. triangle	The command (available also as  button in tool-bar) executes a simple observation of the object you click on (see Section 2.3). If no reference triangle has been defined yet, you will be first asked to specify the reference triangle. Then click on an object (existing point, line, circle, conic, cubic). After a while, OK Geometry displays a form with a list of observed properties of the object in relation to the reference triangle. Click on the More or Extensive button for a longer or an extensive list of observations. To inspect other objects, click **Continue** and select the next object to be analysed. Here are some illustrative observations: - The selected point is the centre X(42) of the reference triangle. - The selected line is tangent to the 9-point circle of the reference triangle. - The selected point is the intersection of the lines X(6)X(9) and X(5)X(7). - The selected line passes through X(21) and the isogonal conjugate of X(35). **Note.** See Section 2.3 for an exhaustive description of the command. **Note.** Right click on the displayed results: there are several available commands to visualise and obtain information about the listed objects (place the cursor set a block in the text appropriately). **Note.** This command is intended for a quick analysis. For a more elaborated analysis, see Section 0.
Observe formulae ...	The command starts the form for the observation of algebraic relations between geometric quantities in the reference triangle. Please, refer to Section 4.4 for more.

All commands for triangle centres and other objects related to a triangle are used in a similar way. Suppose we are looking for the circumcevian triangle of a given point with respect to the reference triangle. Since our object is defined by the reference triangle and a point, choose the **Triangle/Point derived objects** command. In the form that appears (Figure 11), first select the 'Triangle' option (since the resulting object is a triangle). From the listed operations, select the option Circumcevian. ***Notice the option in the lower left corner: If a reference triangle was previously defined, specify whether you want to use the already defined reference triangle or another triangle.*** As you can see, the form contains a short description of the operation with an optional illustration (option 'See picture').

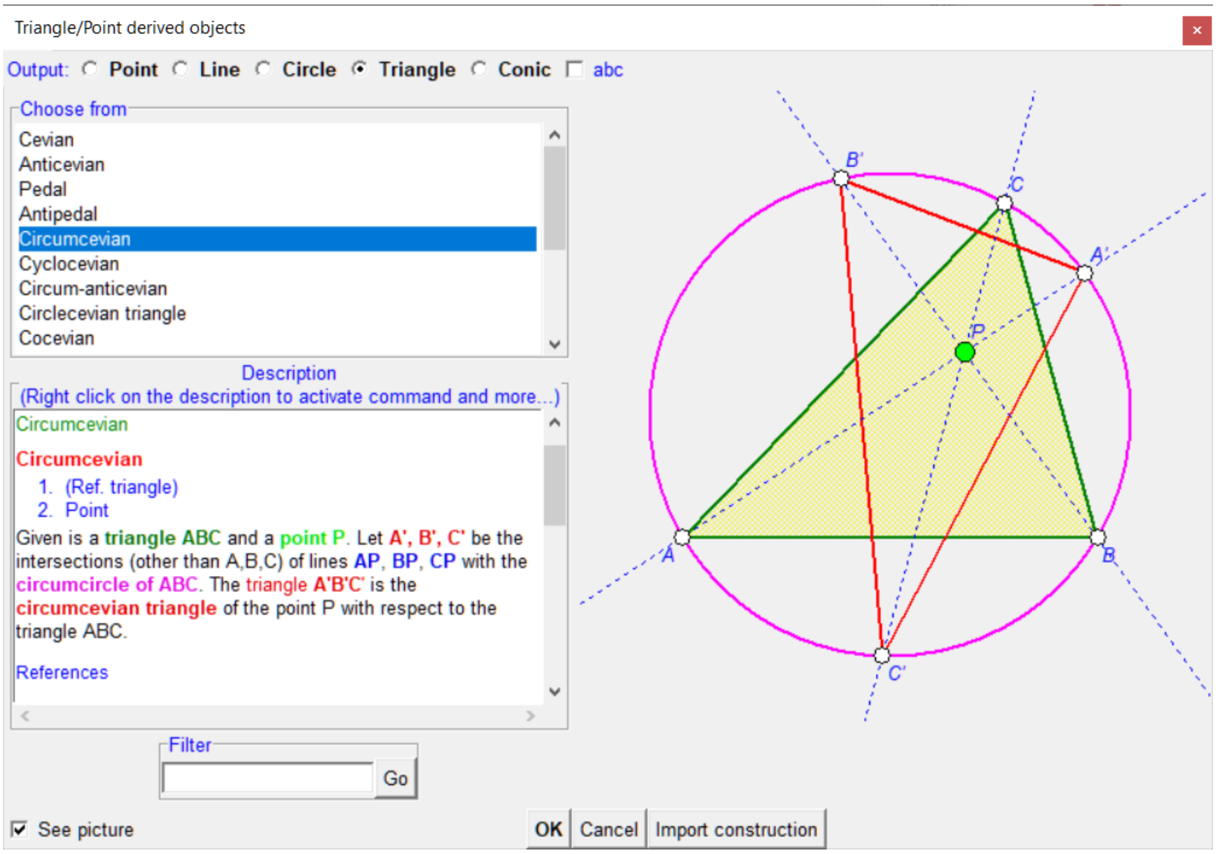


Figure 11

Here is a detailed description of the steps to follow:

<table>
<tr><td colspan="2">
1. Select the type of the resulting object (in the above case choose between Point, Line, and Triangle). A list of available objects/operations appears. Select 'abc' option if you want the list to be arranged alphabetically.
2. Select the object or operation from the displayed list. A short description of the object/operation appears. If you want an illustration of the description, select the 'See picture' option.
3. If a reference triangle has previously been defined, you can use it as a reference triangle for the operation. (You don't see this check box if the reference triangle has not been set.)
4. Click OK and select the involved objects:
</td></tr>
<tr><td>If 'Use ref. triangle' is ON</td><td>If 'Use ref. triangle' is OFF or is not present</td></tr>
<tr><td>1. Since the reference triangle already defined, you only need to select the additional arguments (depending on the command, there can be none, one point or two points).</td><td>1. ALWAYS specify the reference triangle first. Either pick its three vertices OR pick the polyline that defines the triangle.
2. Select the additional arguments (depending on the command, it can be none, one point or two points).</td></tr>
</table>

2.4.1 Triangle object defined by 0, 1 or 2 points

Triangle objects are often determined by the reference triangle (e.g. orthocentre, medial triangle), by the reference triangle and a point (e.g. isogonal conjugation of a point, Cevian triangle of a point), by the reference triangle and two points (e.g. Ceva product of two points, bicevian conic). The usual way to obtain a triangle object is as follows:

- Select the appropriate command according to the defining object:

Defining objects	**Command in group Special**
a triangle	*Triangle Centres* or *Triangle Derived objects*
a triangle and a point	*Triangle/Point objects*
a triangle and 2 points	*Triangle/Point/Point objects*
anything else	*Various constructions*

- In the form select the type of objects you want to create (Point, Line, Circle, Conic, Cubic).
- Select the triangle object from the list of objects.
- Checkmark whether you want to use the existing reference triangle or not.
- Click the button OK.
- If you decided not to use the reference triangle, set the triangle as the first argument, either by clicking its vertices or by clicking a corresponding polyline.
- Click the remaining point type arguments if necessary.

Note. The *Triangle Centres* command behaves slightly differently.

Note. The *Various constructions* command includes some often used constructions not necessarily related to reference triangle (e.g. radical centre of 3 circles).

To get to the desired command you can always use the *Help|Triangle commands* help or the **F1** button on the keyboard (glossary).

2.4.2 Triangle objects defined by triangle centres

Triangle objects are often defined by the ETC triangle centres. Such objects can, of course, be obtained by first creating first involved triangle centres of the reference triangle and then creating the object itself (e.g. a line through two points). A more elegant method is to use the command *Special|Objects by triangle centres*. With this command, you can create triangle objects (points, lines, circles, conics, cubics) by listing the triangle centres on them. You can also simultaneously create several objects. For example, the entry in Figure 1Figure 12 would create the first 20 ETC centres, the lines X2X7 and X2X8 and the circle through the centres X2, X7 and X8. For more details, see the description of the command *Objects by triangle centres*.

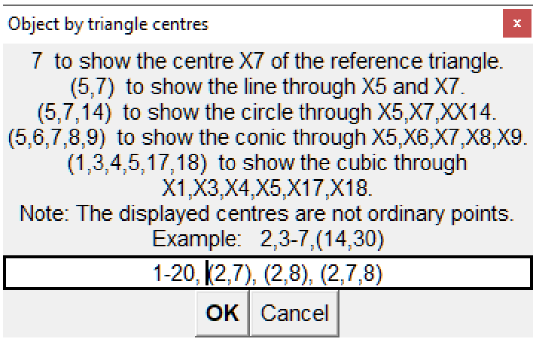


Figure 12

2.4.3 Geometric quantities of triangle

Some commands of OK Geometry use geometric quantities related to the reference triangle. The notation of these quantities considers that A, B, C are the vertices and the angles of the reference triangle. Thus, if the reference triangle is PQR, then

A .... is the size of the angle P (in degrees),

B .... is the size of the angle Q (in degrees),

a............is the length of the side QR, etc.

Here is the list of the considered quantities:

A, B, C	size of angles (in degrees) of the reference triangle
W or omega	Brocard angle
a, b, c	length of the sides of the reference triangle
s	half-perimeter of the reference triangle
sa, sb, sc	$s-a$, $s-b$, $s-c$
r	radius of the circumcircle of the reference triangle
ri	radius of the incircle of the reference triangle
Area	area of the reference triangle
Areao	oriented area of the reference triangle
S	twice the area of the reference triangle
SA, SB, SC, SW	Conway parameters for the reference triangle
cA, cB, cC	$r\cdot\cos(A)$, $r\cdot\cos(B)$, $r\cdot\cos(C)$, where r = circumradius

Note. As you may notice, the notation for quantities of dimension 1 begins with a lowercase letter. For this reason, the incircle and circumcircle radii are not denoted by r and R, as usual, but rather by r and ri.

Not only can we determine the geometric quantities of a triangle, but we can also use them in arithmetic expressions. This is the purpose of the *Number|Triangle expression* command. In this command, we first specify the reference triangle, then write an arithmetic expression that can include the geometric quantities of the triangle as well as user parameters written in square brackets. The value of the expression is set as a new parameter.

When working with geometric quantities, we must be aware that OK Geometry is based on a stochastic dynamic model of geometry. The reference triangle (or another object) displayed on the screen is merely a graphical representation of several variants of random objects. The displayed value of a geometric quantity refers to the first variant of a random object in the model and, as such, is not particularly meaningful. Parameters obtained as the value of an expression involving geometric quantities are meaningful only when the dimensions are zero (e.g. angles, the ratio of two quantities with the same dimension), or when comparing quantities with the same dimension.

2.4.4 Triangle objects defined by triangle coordinates

The position of a point relative to a reference triangle is often described using triangular coordinates. These are usually barycentric or trilinear coordinates, and less commonly tripolar coordinates. When we have this type of description for a point, we use the command *Special|General triangle derived point*.

In this command, we specify the barycentric, trilinear, or tripolar coordinates of the point as arithmetic expressions involving geometric quantities (see Section 2.4.3) relative to the reference triangle. We can also use previously defined parameters, written in square brackets.

In the example (Figure 13 left) we define a point relative to the reference triangle. The baricentric coordinates of the point are

$$b^2+c^2 : c^2+a^2 : a^2+b^2$$

where *a,b,c* are the sides of the reference triangle. If you want to define a point relative to another triangle (and not to the current reference triangle), uncheck the 'Use of reference triangle' box. After clicking on the OK button you will be asked to specify the vertices of the vertices of this triangle.

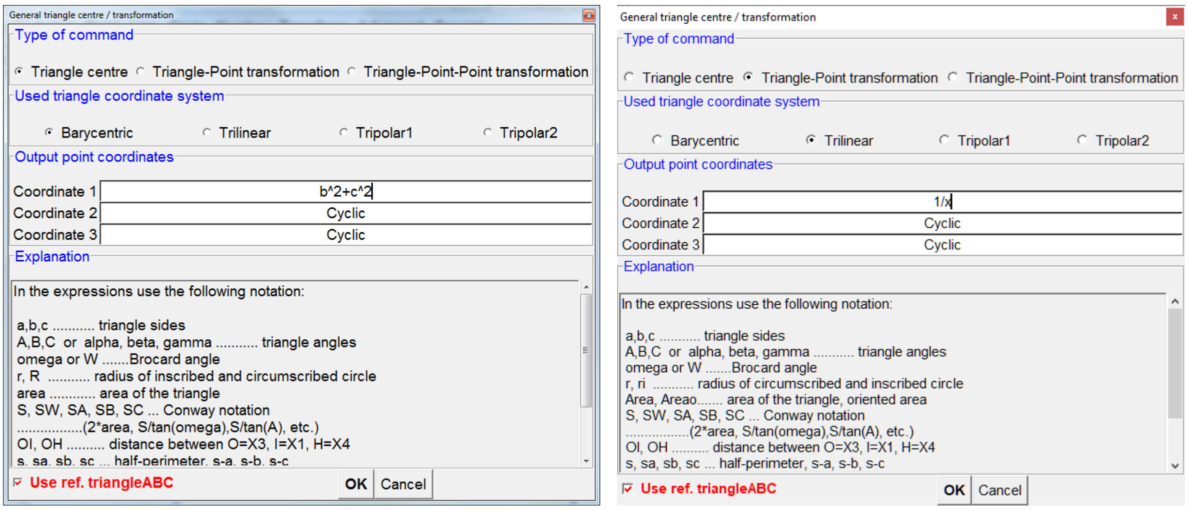


Figure 13

The Special|General triangle derived point command also allows the definition of triangular transformations of a point or a pair of points. The constructed point is described by trilinear or barycentric coordinates. The expressions for the coordinates can include, in addition to geometric quantities and previously defined parameters (in square brackets), the coordinates of the first point x : y : z and the coordinates of the second point u : v : w.

Figure 13, right, shows the transformation that maps the point x : y : z to the point 1/x : 1/y : 1/z (which, of course, is the isogonal conjugation). The command afterward asks for a point and creates the declared transformation of that point.

2.5 Cyclic constructions

Triangle constructions are often cyclic in the sense that they take into consideration the cyclic structure of the triangle. The cyclic structure consists of ordered triads of objects of a considered triangle. We say that an operation is performed cyclically if the involved arguments in the repeated operation are permuted cyclically according to the cyclic structure. Thus, a cyclic operation in a construction creates a triad of objects of the same type and the triad is added to the existing cyclic structure. It is good practice to avoid triads with identical objects.

The Sketch Editor command *Special|A triangle* creates a triangle with cyclic structure consisting of three vertices and three sides of the triangle. The **Cycl** flag indicates whether the current operation is to be executed as cyclic or normal (Figure 14). When the **Cycl** flag is ON, it is bright and the mouse cursor takes the shape of a small circle. You can change the **Cycl** flag with a click on it or using the command *Special|Cyclic construction*. When a command is executed as cyclic, the resulting triad of objects is automatically added to the cyclic structure. You can add a triad of object to the cyclic structure also with the command *Special|Declare cyclic objects*.

Figure 14

By considering the cyclic structure in triangle contructions makes the construction process much easier and faster. Also, the resulting figure is easier to understand, because the ‘cyclic copies’ of the constructed objects are bleached. (You can easily remove the bleaching with the command *Action|Emphasise|Unmark all bleach*.)

The easiest way to create cyclic constructions is to start with the command *Special|A triangle*. Then continue with other commands – but be sure to turn ON the **Cycl** flag ONLY when you want the operation to be cyclic - otherwise you will get identical copies of objects. In most cases, but not always, OK Geometry corrects such an error (i.e, ignores the cyclic flag) – in such cases the cursor turns from circle form to the square form for a fraction of a second.

Note that some command clear the **Cycl** flag automatically, e.g., the *Polyline* or *Object by triangle centres*, while some other commands ignore the **Cycl** flag, e.g. the commands for labelling or restyling objects.

We illustrate how to do a cyclic construction with a simple example[6] (Figure 15). On each side of the triangle ABC we construct outwardly an equilateral triangle. Let the added vertices be A’, B’, C’. It is

[6] OKExamples\OKG_Plus\Triangle_02.p

known that the lines AA', BB', CC' concur in a point Q (the 1st isogonic point). Moreover, let P be the centre of the circle through the points A', B', C'.

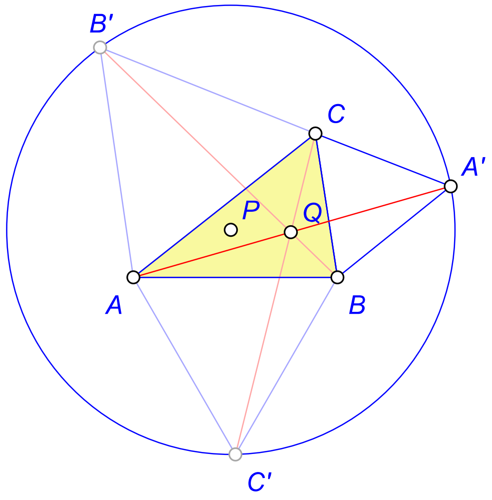


Figure 15

Here are the steps of this cyclic construction.

Step/Command	Comment
Special|A triangle	Obtain a triangle with the initial cyclic structure. Note that this command is active only if no object has yet been constructed.
Special|Cyclic construction	**Cyclic construction is turned ON**. (Note that the pointer assumes a circular form.) Alternatively, click on the **Cycl** flag.
Advanced|Shape|Equilateral triangle	Construct the equilateral triangle on the side CB (click vertex C, then B). Simultaneously, bleached equilateral triangles are constructed on the other two sides of the triangle ABC.
Actions|Label vertex	Label the new vertices A', B', C' Note. Labelling vertices is never considered a cyclic operation.
Line|Segment	Construct the segment AA'. Simultaneously two other cyclically defined segments (BB' and CC') are constructed. Note that the three segments concur in a point (1st isogonic centre P, to be constructed).

Special|Cyclic construction	**Cyclic construction is turned OFF**. This is necessary since command that follows should not be executed cyclically (creating one command per cycle)[7].
Point|Intersection	Construct the intersection point Q of segments BB’ and CC’. It was necessary to set **Cycl** OFF, otherwise three coincident intersection points would be constructed (See footnote 7).
Circle|Circle 3 obj	Construct the circle through A’, B’, C’. Note that **Cycl** is OFF.
Point|Centre of circle	Construct the point P. (The observation of triangle object P finds that P is X627 = ANTICOMPLEMENT OF X17.) Note that **Cycl** is OFF. (See footnote 7).

Example 1

Here is another nice example[8]: the Dao 6 point circle. Given is a triangle ABC with its centroid G. Let A’ be the centre of the circle through B that is tangent to the line AG at G. Let A’’ be the centre of the circle through C that is tangent to the line AG at G. Obtain the circle centres B’, B’’, C’, C’’ cyclically. The 6 points A’, A’’, B’, B’’, C’, C’’ lay on a circle, called the Dao 6-point circle (Figure 14).

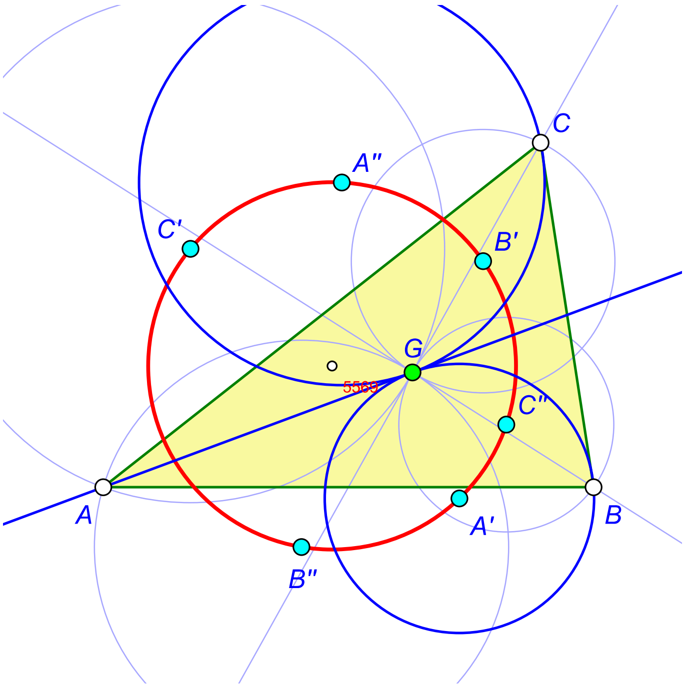


Figure 16

[7] Actually, in this case the system would detect that there is no reason to consider the command cyclically and would ignore the **Cycl** flag.

[8] OKExamples\OKG_Plus\Triangle_03.p

Step/Command	Comment
Special|A triangle	Obtain a triangle ABC with the initial cyclic structure. Note that this command is active only if no object has yet been constructed.
Special|Triangle centres|Advanced|Centroid X(2)	We do this step with the **Cycl** flag OFF. Label the centroid as G.
Special|Cyclic construction	**Cyclic construction is turned ON**. (Note that the pointer assumes a circular form.) Alternatively, click on the **Cycl** flag.
Line|Line 2 points	Pick the points A and G. The lines BG and CG are created automatically.
Circle|Circle 3 objects ***Point|Centre of circle***	Pick the point G, the line AG and the point B. Pick the created circle. At the same time you got 3 circles and then their centres. Label the centres as A', B', C'.
Circle|Circle 3 objects ***Point|Centre of circle***	Pick the point G, the line AG and the point C. Pick the created circle. At the same time you got 3 circles and then their centres. Label them as A'', B'', C''.
Special|Cyclic construction	**Cyclic construction is turned OFF**. We clear the **Cycl** flag before the Dao circle is drawn (otherwise we would get 3 copies of the circle). Alternatively, click on the **Cycl** flag.
Circle|Circle 3 objects	Construct the circle through A', B', C'.

2.5.1 Interactive detection of cyclic perspectivities

When we study cyclic constructions on a reference triangle, we are often interested in triangles that are in some way perspective to the reference triangle. For this purpose, it is possible to activate a mechanism in OK Geometry that constantly checks whether a triplet of cyclically constructed points forms a triangle that is perspective to the reference triangle. You activate this mechanism with the command ***Special|Detect cyclic perspectivities***.

Perspectivities of generated cyclic triplets of points with regard to the reference triangle are automatically detected if:

- the reference triangle and other objects in the current construction have a corresponding cyclic structure,
- the **Cycl** flag is activated (***Special|Cyclic construction***),
- the detection of perspectivities is activated (***Special|Detect cyclic perspectivities***).

The cyclic construction of a point now creates 3 cyclically related points, and the software immediately checks whether the resulting triangle is perspective, orthologically perspective, parallelogically perspective, and cyclologically persective to the reference triangle. The centres of persectivities are also displayed. The displayed centres of a perspectivity can be inspected as usual. A group of detected centres of perspectivities behave as a single object: if one of the displayed centres is coloured, hidden, deleted, all others displayed centres are coloured, hidden, deleted. If, for example, we only want to work further with one of the centres, we must to label it (or put a point on it) and hide (not delete!) the other centres.

The detection of cyclic perspectivities requires that the cyclically generated points correspond to the vertices of the reference triangle. **This means the newly generated point must cyclically correspond to the first vertex of the reference triangle**; the remaining two cyclically equivalent points will then be generated automatically.

Note. The interactive method described only detects perspectivities of the reference triangle and triangles with cyclically constructed vertices. To detect perpectivities of any triangle with any of the numerous triangles related to the reference triangle, see Sections 2.7.7.

Here is a simple example (Figure 17). Let N be the 9-point centre of triangle ABC. Let A' be the circumcentre the triangle BCN. Define B' and C' cyclically.

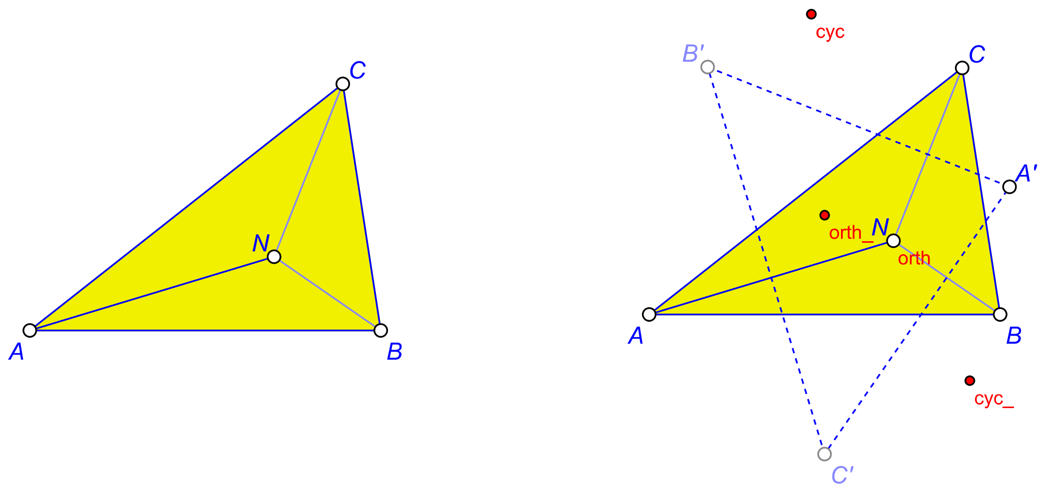


Figure 17

As soon as the point A' and its cyclic counterparts B', C' are constructed we are informed that the triangle A'B'C' is cyclically and orthologically perspective to ABC.The respective centres of perspectivities are constructed and displayed. The red points 'cyc' and 'cyc_', for example ,are the centres of cyclological perspectivity of triangles ABC and A'B'C', thus the circumcircles of triangles AB'C', A'BC', A'B'C meet in the point 'cyc' and the circumcircles of triangles A'BC, AB'C, ABC' meet in the point 'cyc_'.

Here are the steps of this cyclic construction.

Step/Command	**Comment**
Special|A triangle	Obtain a triangle ABC with the initial cyclic structure. **Note**. This command is only active if no object has been constructed yet.
Special|Triangle centres|nine-point centre X(5)	Construct the nine-point centre N of ABC. **Note.** Make sure to checkmark the entry *Use the ref. triangle ABC.* The generated point can be immediately labelled by 'N' on the keyboard.
Special|Cyclic construction	**Cyclic construction is switched ON**. (Note that the pointer takes a circular shape) Alternatively, click on the **Cycl** flag.
Special|Detect cyclic perspectivities	Switch ON the detection of perspectivities.
Special|Triangle centres|circumcentre X(3)	**Note.** Make sure to **uncheck** the entry *Use the ref. triangle ABC.* Click on points B,C,N. Simultaneously the circumcentres of triangles BCN, CAN, ABN are displayed. The perspectivity centres 'cyc', 'cyc_', 'orth', 'orth_' are also displayed.
Action|Label vertex ***Line|Segment*** ***Restyle objects***	Label properly the vertices A', B', C'. Draw (cyclically) lines AN and A'B'.
Special|Analyse object wrt. triangle	We analyse the detected perspectivity centres. The point 'cyc' and 'cyc_' , for example, are observed as X1291 and X24772.
Special|Cyclic construction	Cyclic construction flag is switched OFF.
	We would like to work further with centre 'cyc' as the point P and ignore other displayed centres of perspectivities.

Label vertex	The centre 'cyc' is labelled as P.
Action|Hide	Hide the found perspectivity centres. (Deleting the centres would also delete the point P.)

2.6 Examples

Example 2 - The Privalov conic

It is well known that in a given triangle ABC the vertices of the cevian triangles of any two different points D, E (not laying on the sidelines of ABC) are conconical (Figure 18, left)[9]. We consider the case of two particular points: the Gergonne point (D) and the Nagel point (E) of a given triangle ABC. The Gergonne point of ABC is the point of concurrence of the line segments connecting the points where the incircle of ABC touches the sides with the opposite vertices. Similarly, the Nagel point of ABC is the point of concurrence of line segments connecting the points where the excircles touch the sides of ABC with the opposite vertices (Figure 18, right)[10].

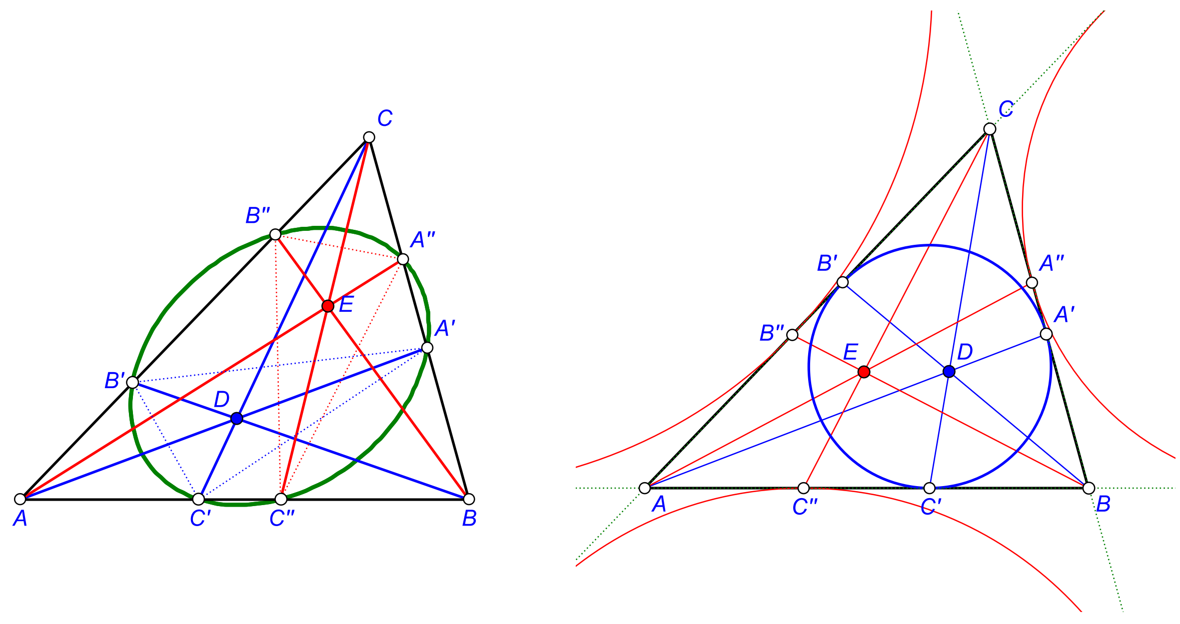


Figure 18

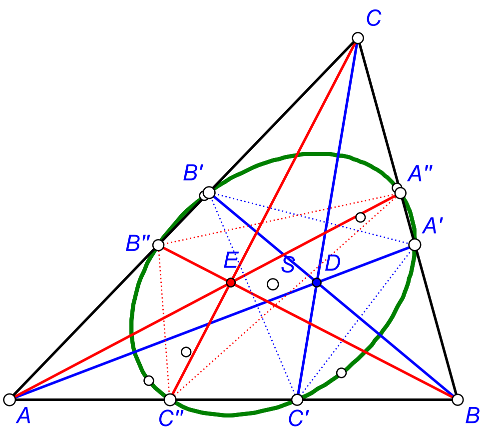
Here is a quick and efficient way of executing this construction[11]:

[9] OKExamples\OKG_Plus\Triangle_05.p
[10] OKExamples\OKG_Plus\Triangle_04.p
[11] OKExamples\OKG_Plus\Triangle_06.p

Steps	Commands	Comment
1. Draw a triangle ABC	***Point|Point*** (for A,B,C) ***Line|Polyline*** (to connect A-B-C-A) ***Special|Reference triangle***	Obtain a reference triangle.
2. Position the Gergonne point D and the Nagel point E of ABC	***Special|Triangle centres|Advanced|Gergonne point*** ***Special|Triangle centres|Advanced|Nagel point***	Keep the check-mark for using the reference triangle, since we construct the Gergonne and Nagel points of a previously defined reference triangle ABC.
3. Construct the Ceva-triangles of D and E	***Special|Triangle point derived objects|Triangle|Cevian*** (choose then points D and E)	
4. Construct the ellipse and the characteristic points of the ellipse	***Circle|Conic 5 pts*** ***Point|Conic points***	Label the centre of the ellipse as S. (Figure 19, left.)
5. Inspect the centre S of the ellipse	***Special|Analyse object wrt. triangle*** (pick S)	OK Geometry observes that S is the ETC centre X5452 . (See Figure 19, right.) A more detailed inspection with Triangle analysis suggest that, for example: S = Isogonal conjugate of the Cross conjugate of X6 and D, or S = Complement of the Isogonal conjugate of X1486, or S = Complement of the Isotomic conjugate of (X3434)
6. Inspect the points on the constructed ellipse	***Special|Analyse object wrt. triangle*** (pick the ellipse)	OK Geometry spots 3 centres along the ellipse (X3022, X3271, X4904). The extended search adds to this list 6 more transformed centres.

Here is an even more effective construction.

Steps	Commands	Comment
1a. Draw a triangle ABC	***Special|A_triangle***	Obtain a reference triangle ABC.
2a. Position the Gergonne point D and the Nagel point E of ABC	***Special|Objects by triangle centres*** In the entry write 7,8	Since X7 is the Gergonne point, X8 is the Nagel point of the triangle.
3a. Construct the Ceva-triangles of D and E	***Special|Triangle point point derived objects|Conic|Bicevian conic*** (choose then points 7 and 8)	
4a. Construct the characteristic points of the ellipse	***Point|Conic points***	Label the centre of the ellipse as S.
Proceed as in steps 5 and 6 of the previous construction.		



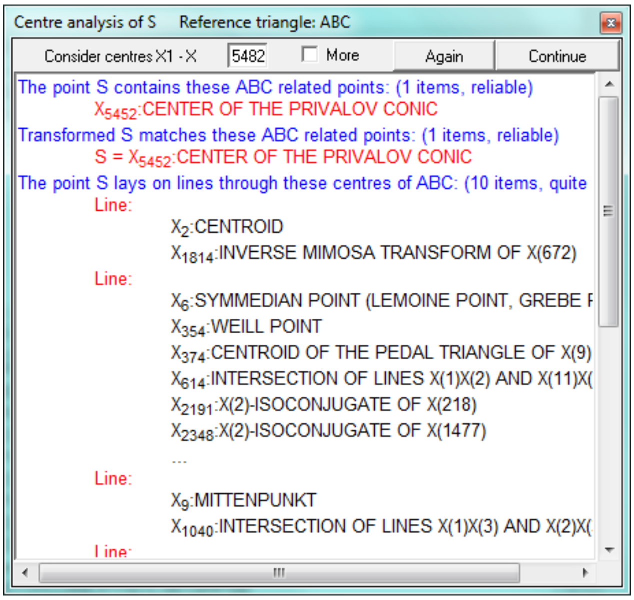


Figure 19

Example 3 - Circum-side-circle

Given is a triangle ABC. A circle that has a side of ABC as its diameter is called a side-circle of ABC. So each triangle has three side-circles. We are looking for a way to construct the circle that touches internally all three side-circles of ABC, hereby called the circum-side-circle of ABC (Figure 20)[12].

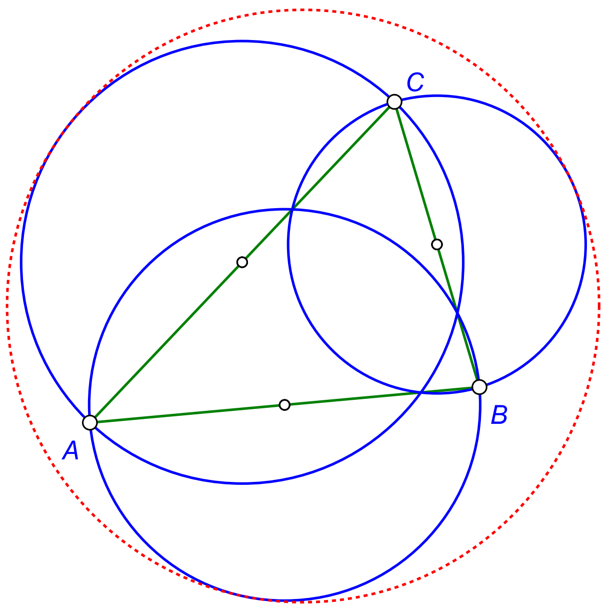


Figure 20

The strategy in the construction below is to find the three points of contact of the circum-side-circle with the three side-circles.

Steps	Commands	Comment
1. Draw a triangle ABC	***Point|Point*** (for A,B,C) ***Line|Polyline*** (to connect A-B-C-A) ***Special|Reference triangle***	Try to make ABC visibly scalene.
2. Construct the three side-circles and the circum-side-circle	***Point|Midpoint|***(pick ABC)***On each segment*** ***Circle|Circle centre+point*** ***Circle|Circle 3 objects***	Use repeatedly the Alt button (in the editor's menu bar to obtain the desired circle that is tangent to three objects.

[12] OKExamples\OKG_Plus\Triangle_07.p

3. Construct the three points of tangency A', B', C' of the circum-side-circle	***Point|Point***	The **Observe** command reveals that AA', BB', CC' concur in a point, say, P.
4. Construct the lines AA', BB', CC' and the concurrency point P.	***Line|Line 2 pts*** ***Point|Intersection***	
5. Inspect the properties of the point P in reference triangle ABC	Set ABC as the reference triangle. Enter the Triangle analysis module. Execute the Object analysis on the point P.	
6. Obtain information about the Paasche point	In the Triangle analysis module results place the cursor on X1123 and click the WhatIs button.	To obtain the basic information about the Paasche point place the cursor on this term and right-click the **WhatIs** command (Figure 21).

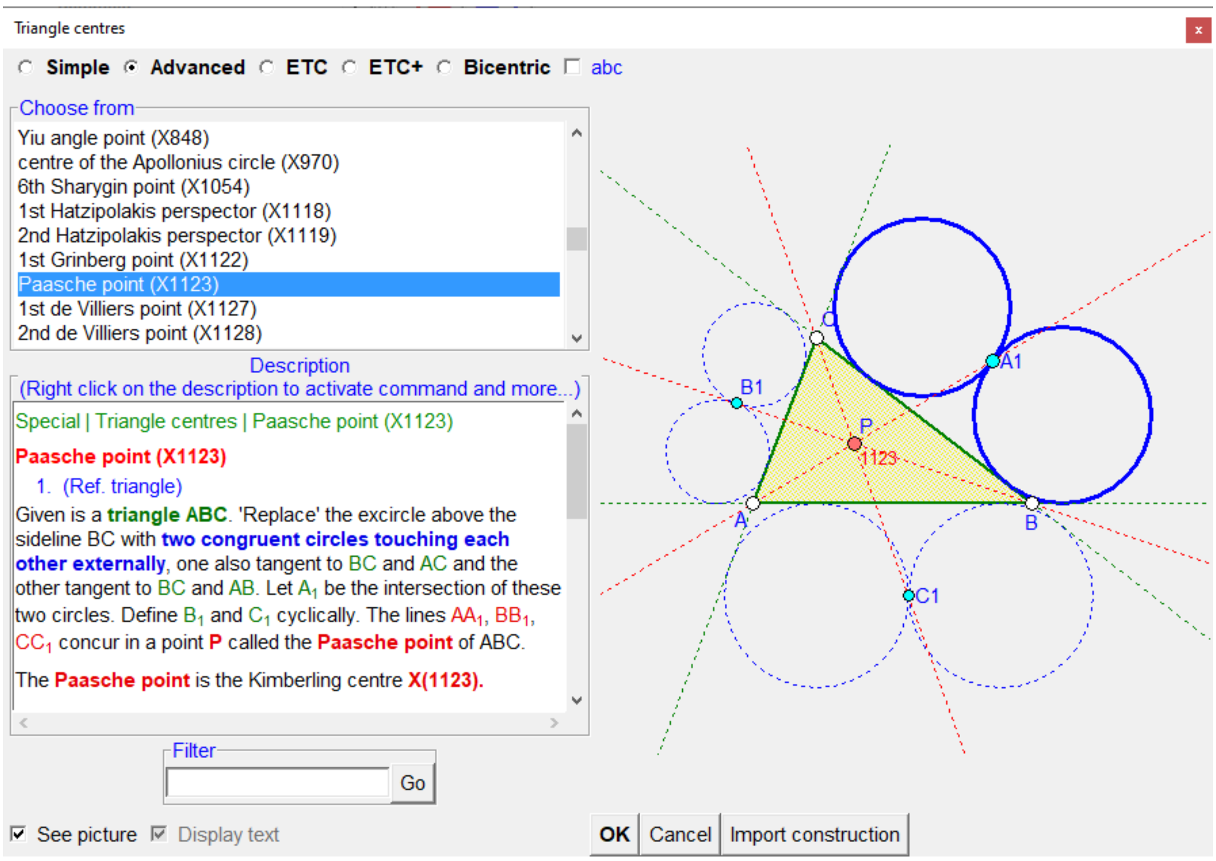
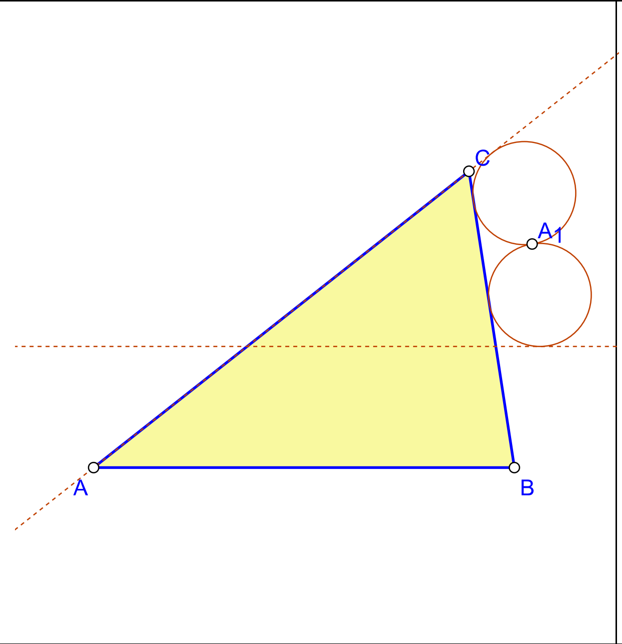


Figure 21

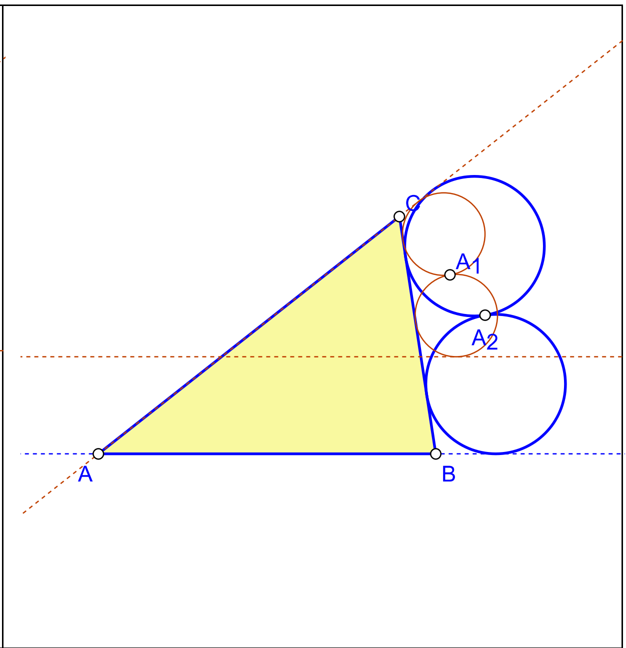
The Paasche point P of ABC can be constructed (using a homothety on a 'false' construction), and this gives way to the construction of the circum-side-circle. The construction steps are illustrated in

Figure 22. A much more elegant way can be inferred with the Advanced query method described in Section 2.8.

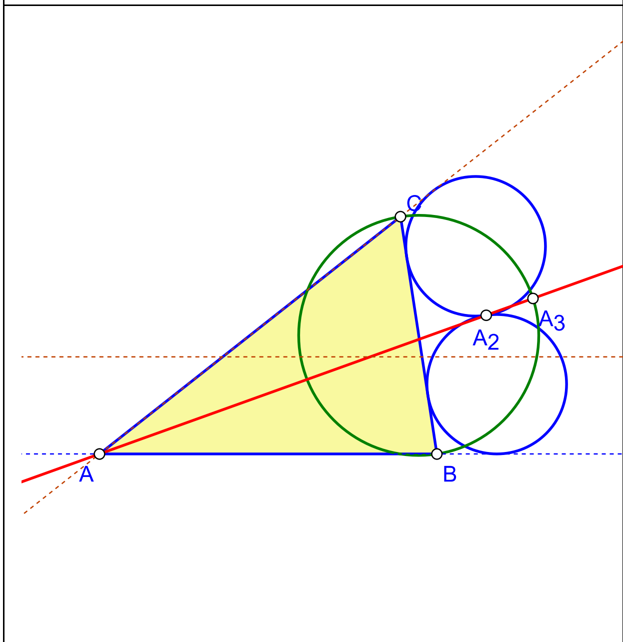
The correctness of the construction via the Paasche point can be verified with algebraic means.



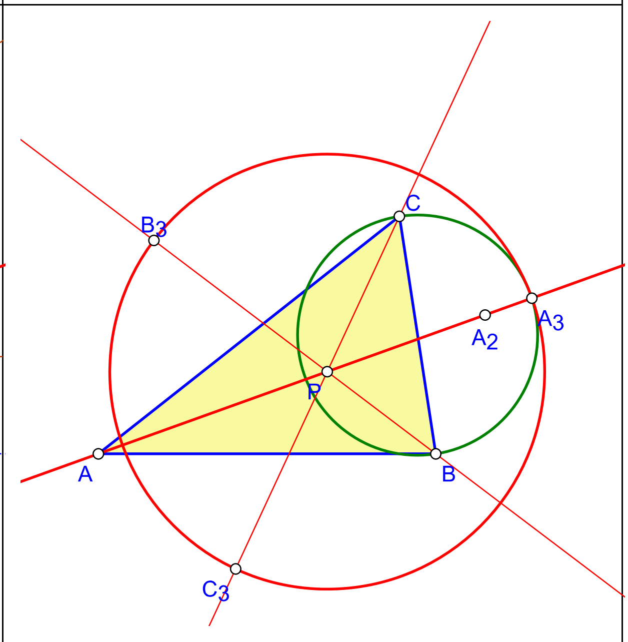
Step 1. Construct 'false' circles of equal radius (as shown), the first touching CA and BC, the second touching BC and the first circle.



Step 2. Using homothety obtain the tangency of the second circle and the line AB.



Step 3. Let A_2 be the common point of the two circles. Let A_3 be the intersection of AA_2 and the side-circle of BC. The point A_3 is the sought point of contact with the circum-side-circle of ABC.



Step 4. Obtain B_3 and C_3 cyclically. The circum-side-circle is the circle through A_3, B_3, C_3. The common intersection of AA_3, BB_3, CC_3 is the Paasche point.

Figure 22

The circum-side-circle (Figure 20) is one of the 8 circles that are tangent to the three side-circles, see Figure 23[13]. Here are some of the observed properties:

One of the 8 circles contains all three side-circles, this is the circum-side-circle with centre P_0 – the complement of the equal detour point X(176).

Three of the 8 tangent circles are contained in exactly two of the side-circles, their centres are A_2, B_2, C_2 (see Figure 23). The lines AA_2, BB_2, CC_2 concur at point P_2 – the Yiu-Pasche point X(1659).

Three of the 8 circles are contained in exactly one of the side-circles, their centres are A_1, B_1, C_1 (see Figure 23). The lines AA_1, BB_1, CC_1 concur at point P_1 - the isogonal conjugate of the 3rd Kenmotu homothetic centre X(5414).

One of the 8 circles is contained all three side-circles. Its centre is P_3 – the complement of the isoperimetric point X(175).

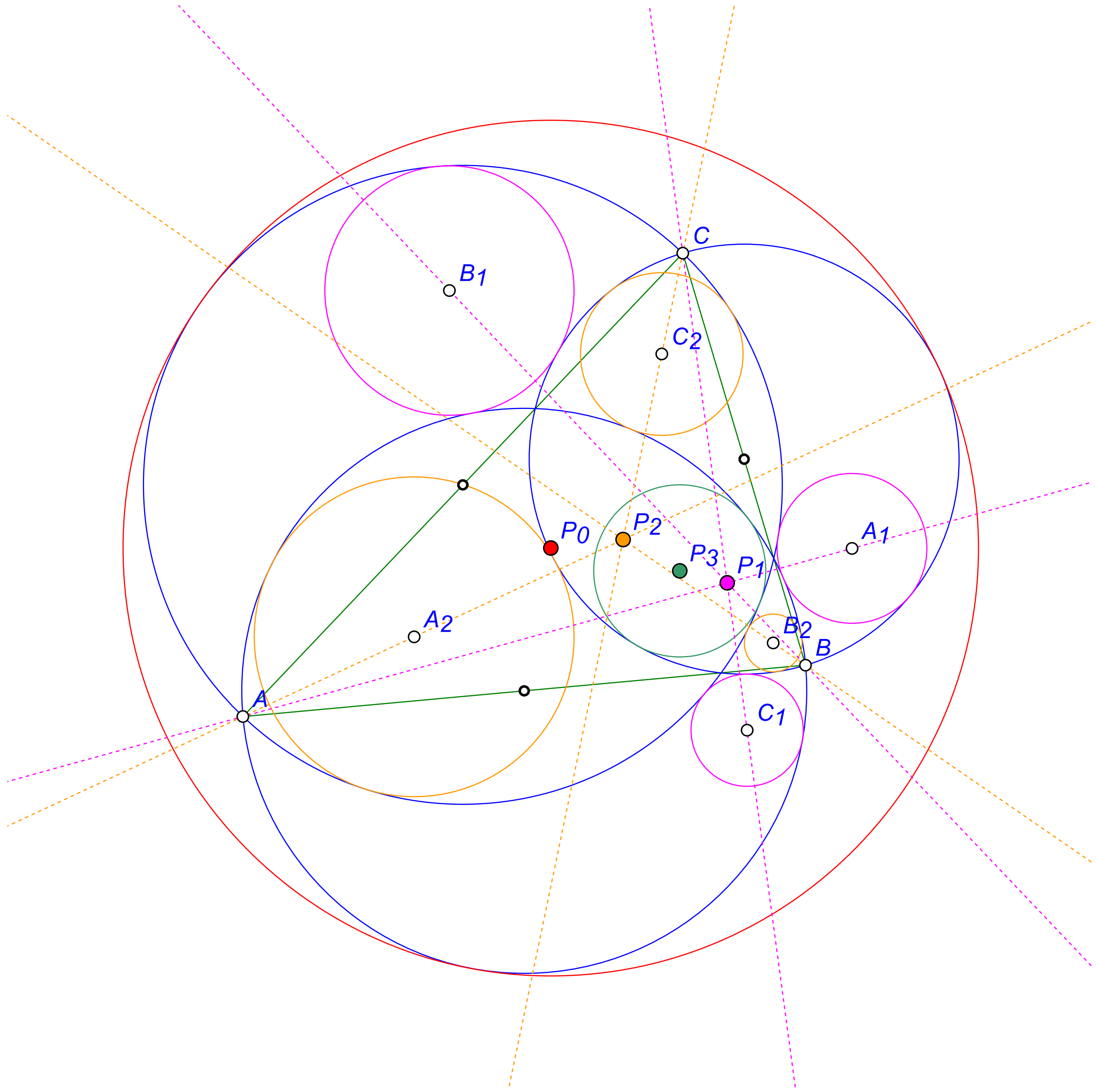


Figure 23

[13] OKExamples\OKG_Plus\Triangle_08.p

Example 4 - Trisecting triangle perimeter by pedal triangle

Given is a triangle ABC and a point D. The pedal triangle of D in ABC has as its vertices the projections A', B', C' of the point D onto the sidelines of ABC. We are looking for a point D with the property that A', B', and C' cut the perimeter of ABC into three pieces of equal length (Figure 24), so that |B'A|+|AC'| = |C'B|+|BA'| =|A'C|+|CB'|.[14]

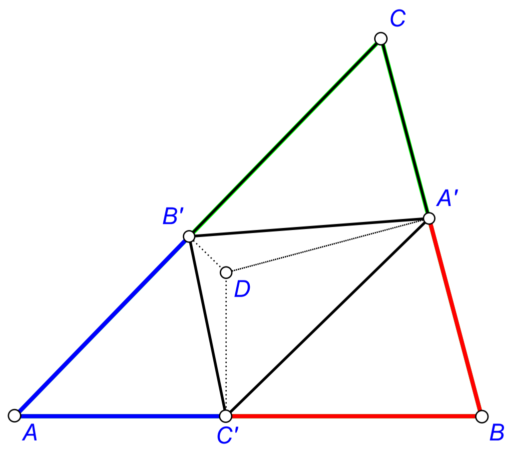


Figure 24

The following steps lead to a reliable hypothesis for the sought position of point D. The strategy is to (1) construct a general triangle, a point D and the corresponding pedal triangle, (2) redefine D with an implicit construction, (3) analyse the new position of the point D, (4) make an improved check for the proposed position.

Steps	Commands	Comment
1. Draw a triangle ABC and a point D	***Point|Point*** (for A,B,C) ***Line|Polyline*** (to connect A-B-C-A) ***Special|Reference triangle*** ***Point|Point*** (Construct a free point D.)	Try to make ABC visibly scalene.

[14] OKExamples\OKG_Plus\Triangle_09.p

2. Construct the pedal triangle of D in ABC.	***Special | Triangle Point derived objects | Triangle | Pedal*** (Pick D.) ***Action | Labels | Label vertex*** (Label the vertices A’, B’, C’.)	Note that ABC was previously set as a referrence triangle.
3. Construct the obtained ‘pieces of perimeter’ as polylines and measure their length.	***Line | Polyline*** (Pick B’,A,C’; repeat the command for C’,B,A’ and A’,C,B’.) ***Number | Length perimeter*** (Select the three constructed pieces.)	Name the lengths of the pieces as Len_A, Len_B, Len_C.
4. Check the sought condition	***Advanced | Check | Equivalence | SameValues3*** (Pick the values Len_A, Len_B, Len_C.)	Name the check as SameValues3. Obviously, it turns out as False.
5. Implicit (re)construction of D.	***Advanced | Implicit construction*** (Uncheck the restriction (OK); then pick the condition SameValues3 (OK) and the point D.)	The implicit construction should turn the condition to True.
6. Analyse the position of the point D	***Special | Analyse object wrt. triangle*** (Pick D.) OK observes that D is the centre X(165) – the centroid of the excentral triangle. To obtain a description of terms used in the result, position the cursor on it, right-click and use the command **WhatIs**.	Note that most items in the results are marked as ‘unreliable’, or even ‘great caution’. This happens when the construction is based on optimisation or implicit construction.
7. Check the construction directly	***Action | Delete*** (pick the point D) ***Special | Triangle centres | ETC | X165*** Repeat the steps 2-4.	Note that the check is now much more accurate since there are no implicit points in the construction.

2.7 Advanced triangle analysis

The **Triangle analysis** module is an advanced observation tool for triangle objects. The **Triangle analysis** module relates (by observation) examined objects to the characteristic objects of the reference triangle. In the observation analysis, the module considers the ETC centres of the reference triangle, various specific triangle transformations, characteristic lines, circles, and conics of the reference triangle. The module is an extension of the Simple observation of triangle objects (Section 2.3).

The module performs three, actually four types of analysis:

- ***Object analysis*** relates points, lines, circles, conics, cubics, and triangles to the characteristic points (centres), lines, circles, conics, cubics, and triangles of the reference triangle. Finds, for example, that the isogonal conjugation of the examined point P lays on the Kiepert hyperbola of the reference triangle.
- ***Two triangle centres*** matches the centres of the examined triangle with the centres of the reference triangle. For example, finds that the centre X_{541} of the triangle ABC coincides with the centre X_{54} of the triangle PQR.
- ***Rule-point analysis*** matches points constructed from triangle centres (of some triangle) with the centres of the reference triangle. For example, for each ETC centre Xn it finds whether the midpoint of X1 and Xn is a known centre of the reference triangle.
- ***2-rule-point analysis*** matches points constructed from pairs of triangle centres (of some triangle) with the centres of the reference triangle. For example, for each pairs of ETC centres Xm, Xn it finds whether the midpoint of Xm and Xn is a known centre of the reference triangle.
- ***Triangle perspectivity analysis*** observes whether a given triangle is perspective to to one of the characteristic triangles of the reference triangle. It considers various types of perpectivity and also analyses the centres of perspectivity.
- ***Object analysis of generic constructions*** performs an automated analysis on a family of dynamic constructions. We will explain this type of analysis in Section 6.7.

Of the four types, Object analysis is the most commonly used.

The best way to learn how to use the module is to use examples. Since all types of analysis have the same (or similar) parameters, modes of operation, and working procedures, we will first explain them. Then we will present some essential examples of the analysis.

2.7.1 Accessing the Triangle analysis module

To access the Triangle analysis module, click on the triangle **(∆) symbol on the main menu bar**. Note that for a given construction the analysis can be performed only for objects that can be identified by labelled points on them. In particular, the vertices of the reference triangle should be labelled. Furthermore, the points, lines, circles, and conics to be analysed should contain (at least) 1,2,3,5 labelled points respectively.

2.7.2 Triangle analysis scheme

The best way to learn how to use triangle analysis is by using it. For this purpose some simple and advanced examples are provided (see Section 0).

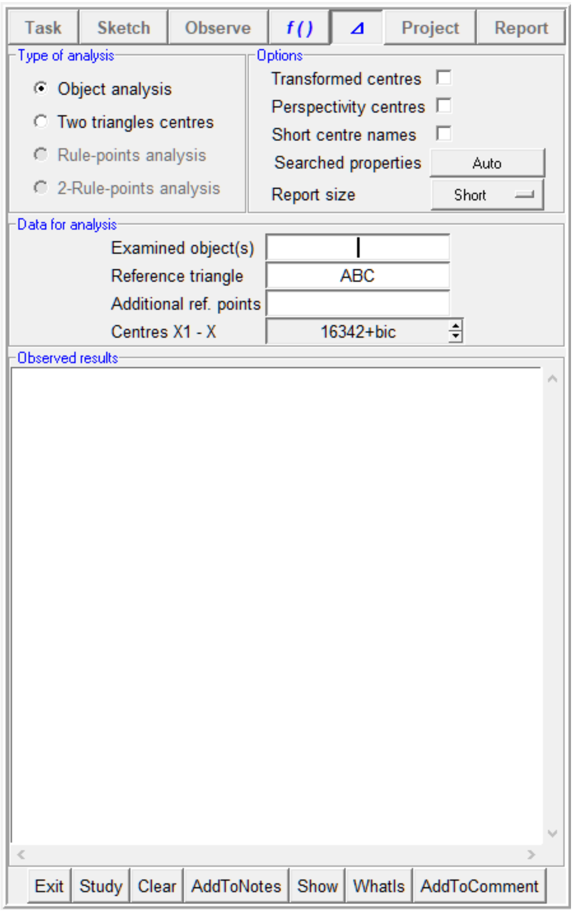


Figure 25

The following description refers to Figure 25, which shows the form that appears on the left-side pane during the Triangle analysis. Note that, depending on the construction, some of the fields may be missing or may be disabled. We provide here a detailed description of the form used in the module.

Entry	Description
Object analysis *Two triangles centres* *Rule-points analysis* *2-rule-points analysis*	Here you select the type of analysis you want to perform. The default and most often used is the Object analysis. The analysis starts after you press the **Study** button.

Examined objects	In the entry, write the object(s) you want to examine in the analysis. Refer - to points as they are labelled, e.g. A' or B or C3, - to lines by specifying two points on them, e.g. A'B, - to circles by specifying three points on their circumference, e.g. A'BC3, - to conics by specifying 5 points on them, e.g. A'BC3DE. Multi-letter labels of vertices must be put in brackets, e.g. A(PQ3)B. **Note.** You can also specify two or more objects to be examined. Separate them with space or comma, e.g., A E'. In this case, a sequence of analyses is performed, one for each Examined object.
Reference triangle	Specify the reference triangle by naming its vertices, e.g. ABC. Analysis will use the centres and characteristic objects of the reference triangle when studying the examined object(s). **Note. Avoid reference triangles that are isosceles or almost isosceles**. In such triangles, many centres are unacceptably close together, so that OK Geometry refuses to perform the analysis. **Note.** It is possible to specify two or more reference triangles (separated by spaces or commas, e.g. ABC D1E'F). In this case , a sequence of analyses is performed, one for each specified reference triangle.
Additional reference points	Here you can specify (optional) points to be considered in the analysis, in addition to the ETC centres of the reference triangle.
Condition parameter	The entry appears only if the considered construction contains one or more condition parameter. If a condition is selected in this entry, the analysis will be executed only if (or when) that condition is met, i.e. is true. Selecting a condition can be important when the studied construction 1. is based on an implicit construction, or 2. is based on the optimisation of a parameter, or 3. is a generic construction. In such situations, it may occur that the construction fails to succeed, so the analysis cannot or should not be performed. In the Sketch Editor you can define a condition (check) that is true if the analysis can be performed. Then use the condition as a condition parameter for the triangle analysis.

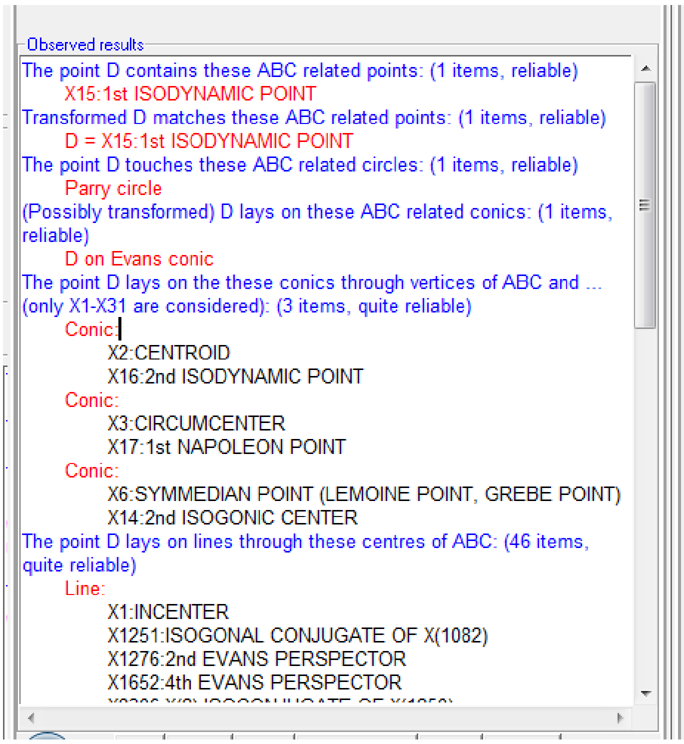
Centres X1-X	Here you specify the highest index of ETC triangle centres to be considered in the analysis. The more centres considered, the longer the computation will take and the more extensive list of results will be. For the most exhaustive analysis choose 16342+bic. Note that the when the index greater than 16342 is chosen, the analysis is less reliable and not all tests will be performed.
Transformed centres	Select this option if you want in the analysis to consider different triangle transformations of the ETC centres of the reference triangle (or the examined points).. When this option is selected, the analysis finds facts like 'an examined point is the cyclocevian of X68', or 'the complement of the examined point is the isotomic conjugate of X25'.
Perspectivity centres	This option is valid only when the examined object is a triangle. By default, the object analysis of a triangle checks whether the examined triangle is in any way perspective (perspective, orthologic, parallelogic, cyclologic) to caracteristic triangles of the reference triangle. If this option is ON, all the found perspectivity centres are calculated and stored, so that they can be retrieved and examined. For example, let us examine (with the option Perspectivity centres ON) the extouch triangle A'B'C' of the reference triangle ABC. After the Study button is pressed, OK Geometry reports the observation of 39+92 cases of perspectivity of the triangle A'B'C' with various triangles related to the reference triangle ABC. These cases give rise to 198 centres of perspectivity (not all of which differ from each other) denoted by %1 to %198. The report says, for example, that the triangle A'B'C' is related to the Garcia inner triangle in the following way: perspective (%59), orthologic (%60, %61). Thus %59 is the perspectivity centre of A'B'C' and the Garcia inner triangle of ABC, while the point %60 is the orthological centre of A'B'C' and the Garcia inner triangle of ABC, and the point %61 is the orthological centre of the Garcia inner triangle of ABC and A'B'C'. To learn more about the point %60, simply write %60 as the examined object and study it. The notation %% denotes all points %1-198 without repetition.
Short centre names	When this option is selected, the descriptive part of the triangle centre names are omitted from the report. For example, if this option is checked, in the list of results the Nagel point will be referred to as X8 instead of X8: NAGEL POINT

Searched properties	The default option is **Auto** (i.e., all properties are included). If you click the Auto button, you can specify a **selection of** properties to be included in the triangle analysis. Limiting the triangle analysis to one or a few properties is particularly useful when performing the triangle analysis for a generic construction.
Report size	For the Object analysis the report size which can be 4. Minimal (only a few facts of the very basic properties are listed), 5. Short (the default option, a limited number of basic properties are listed, only a limited number of triangle triangles and triangle circles are considered), 6. Extended (only the first 100 facts of all considered properties concerning all triangle objects are listed) 7. All (all facts of all properties are listed; may result in a very long list of facts). For the Two triangle centres analysis and for the Rule-points analysis the report can be 8. Minimal (only the first few results are listed), 9. Short (only the first few results are listed), 10. Extended (the first 1000 results are listed), 11. All (all found facts are listed). For the Rule-points analysis the report can be 12. Minimal (refers to the first 20 examples of the ETC1 rule), 13. Short (refers to the first 100 examples of the ETC1 rule), 14. Extended (refers to the first 1000 examples of the ETC1 rule), 15. All (refers to all examples of the ETC1 rule).

2.7.3 Observed results

1. Once you completed the schema in Figure 25, you can start the analysis by pressing the **Study** button. The reported results of the analysis are displayed in the **Observed results** section. Figure 26 shows an illustrative example of part of the displayed results (the analysed object is a point D).
2. The analysis can be performed several times in sequence. Simply enter the data, set the options, and press the **Study button**.
3. The reported results are always appended to the previous content of the section Observed results. At any time, e.g. before a new analysis is run, you can clear the content of the section Observed results with the **Clear** button.
4. The reports can be freely edited.

5. If you want to include a part of the Observed results in the OK Geometry report (i.e. the printable report of the project) for the studied construction, put this part of the results in a block and press the **AddToReport button**.
6. You can append a part of the Observed results to the comment of the current construction. To do this, put this part of the results in a block and press the **AddToComment button**.
7. To get an explanation of objects mentioned in the Observed results section, click on the **WhatIs button.** The cursor then takes the form of a question mark with an arrow. If you click with the arrow on a term in the Observed result section, you will get an explanation of that term (sometimes the neighbouring words are also considered). For example, to find out the definition of the Parry circle or of X_{17}, press the **WhatIs button** and click with the arrow of the question-cursor on 'Parry' or on 'X_{17}'**.**
8. To visualise on the construction one or more objects that are mentioned in the Observed results section, position the cursor on the name of the object to be shown or put in a block the objects to be shown. Then press the **Show button**. For example, to visualise X_{16}, position the cursor on X_{16} and press the **Show button**. To visualise the conic through X_2, X_{16}, and the vertices of the reference triangle, put in block the three lines from Conic to X_{16}, and press the **Show button**.
9. The **Glossary help** is accessible from the context menu when you **right-click on the Observed results** section. The command turns the cursor into a spider. A click of the spider on a term in the Observed results section activates the glossary for that term.
10. To exit from the Analysis module, press the **Exit button** or press the **(Δ)** button in the main menu bar.



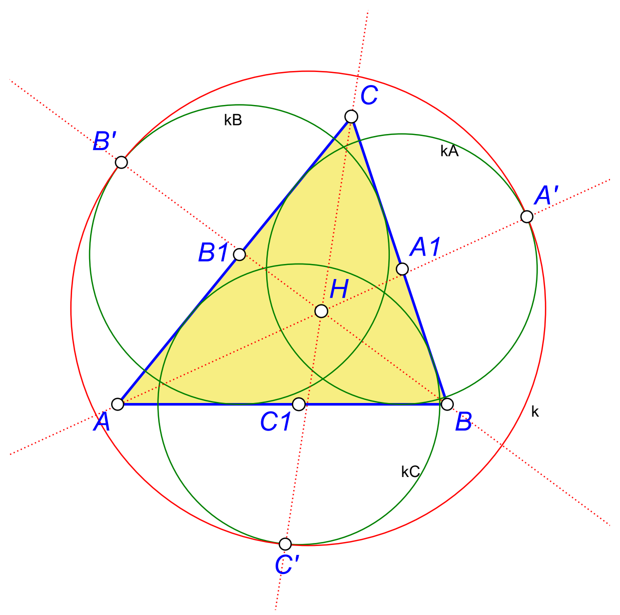
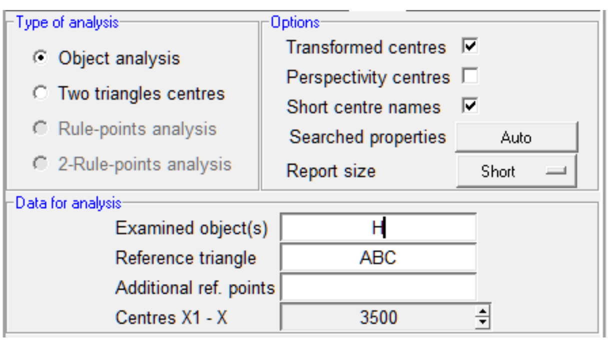
11. **Figure 26**

2.7.4 Object analysis

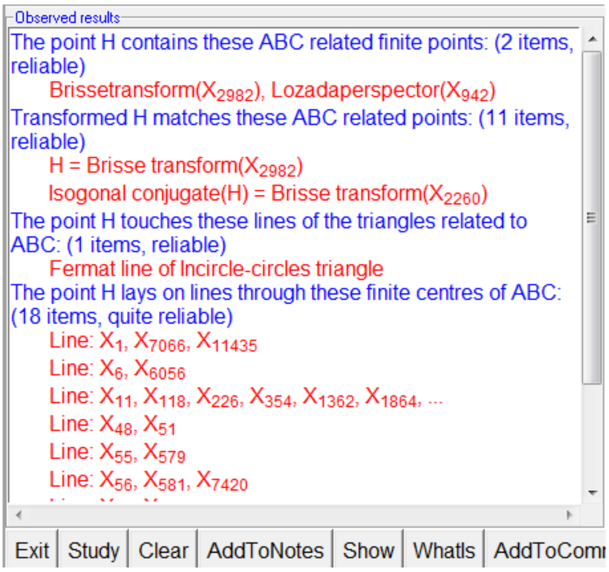
The Object analysis of the Triangle analysis module observes the relationship between the examined objects and the characteristic objects of the reference triangle (ETC centres, characteristic lines, circles, conics, cubics). Essential data for the Object analysis is the reference triangle, named after its vertices, e.g. ABC. Essential is also one or more objects to be examined. They are identified by labelled points on them: one label refers to points (e.g. P), two labels to lines (e.g. PQ), three labels to circles and triangles (e.g. PQR), five labels to conics (e.g. PQRST), six labels to cubics through the vertices of the reference triangle (e.g., PQRSTU), nine vertices to cubics through nine points (e.g., PQRSTUVZW). By filling in the **Examined objects** entry with ‘P Q PQR’ we run the analysis on the points P and Q, the circle through P,Q,R and the triangle PQR.

Example 1

Here is an illustrative example of triangle analysis[15].

	Given is a triangle ABC. Let *kA* be the circle touching the baselines AB and AC and having centre A1 on side BC. Define B1, C1 and *kB*, *kC* cyclically. Let the circle *k* touch internally the circles *kA*, *kB*, *kC* in points A’, B’, C’. OK Geometry observes that the lines AA’, BB’, CC’ meet at a common point H. We analyse the point H in relation to the reference triangle ABC.
	Make sure that the **Type of analysis** is set to **Object analysis**. Select **Short centre names** if you prefer a compact report without explanatory names of triangle centres. Select **Transformed centres** if you want the transformations of H to be included in the analysis. It is advisable to initially select a Short **Report**

[15] OKExamples\OKG_Plus\Analysis_01.p

	size with **Transformed centres** OFF. Then, if needed, increase Report size (more instances will be displayed) and/or select Transformed centres (additional properties will be considered). Press **Study** to start the analysis.
	Here is an excerpt of the results(some irrelevant lines have been deleted). The observation shows that H is, for example, the Lozada perspector of X(942), that the isogonal conjugate of H appears to be X(2260), that H is the intersection of the lines X(1)X(7066) and X(6)X(6056), etc. To get an explanation of the Lozada perspector, first click on **WhatIs** button first and then click on the term Lozada perspector. To visualise objects from the report, put them in a block and click the **Show** command. Other available functions are described in section 0.

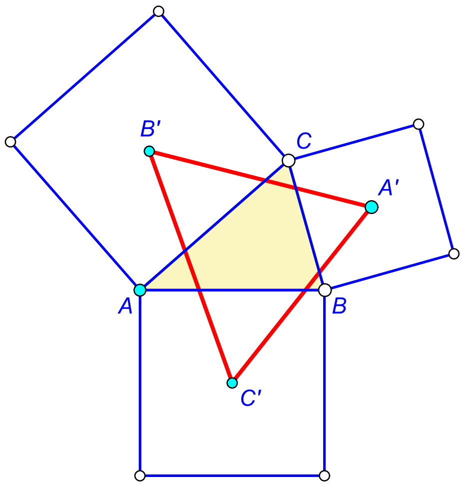
2.7.5 Two triangle centres

Two triangles centres analysis matches the centres of a triangle to the centres of the reference triangle.

Example 2

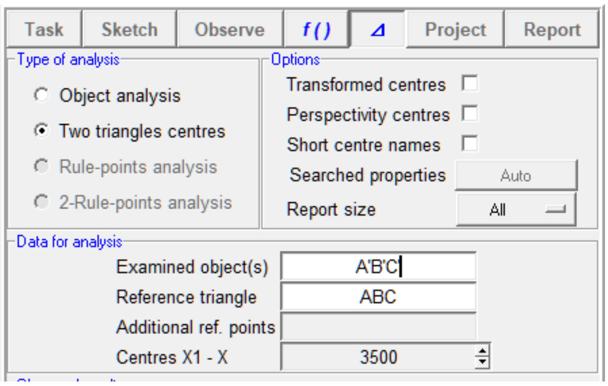
Here is an illustrative example how to perform such analysis.[16]

[16] OKExamples\OKG_Plus\Analysis_02.p



Given is a triangle ABC. Consider the outward squares on the sides BC, CA, AB of the triangle ABC. The centres A’, B’, C’ of these squares are the vertices of the outer Vecten triangle of ABC.

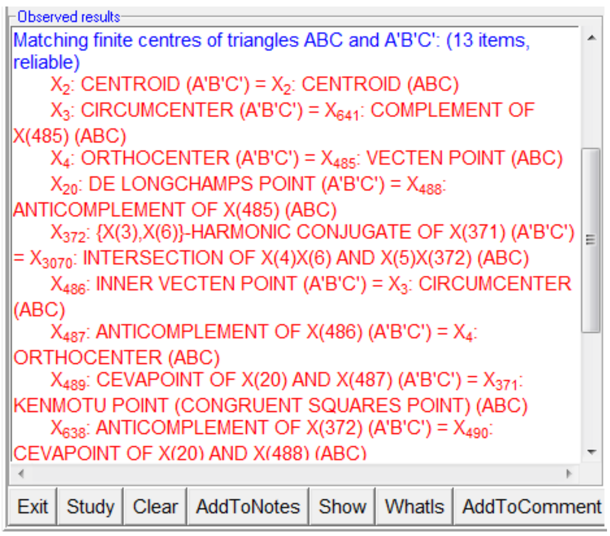
With **Two triangles centres** analysis we examine whether any centre of A’B’C’ coincides with any centre of ABC.



Select **Two triangles centres** as Type of analysis.

We choose All for **Report size**, otherwise only the first few results will be displayed.

In our analysis only the first 3500 centres are considered. If you want to consider also the transformations of the centres (e.g. isogonal conjugation, isotomic conjugation), checkmark the **Transformed centres** option.



Here is the list of observed matches. If we select the **Transformed centres** option and/or increase the number of considered centres, a much longer list will appear.

2.7.6 Rule-points and 2-rule-points analysis

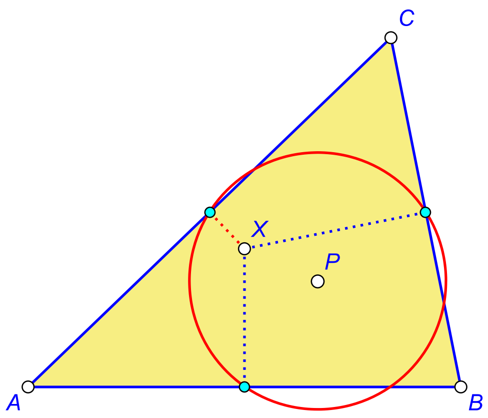
Rule-points and 2-rule-points analysis are quick methods of checking whether points constructed from the ETC centres of a triangle coincide with any ETC centre of the reference triangle. **Rule-point analysis** is performed on a generic construction, where a point, say X, is created with the **ETC1 rule** (for some triangle), and there is also a point, say P, which constructionally depends on the point X. The point X can occupy the position of various ETC centres of this triangle. The rule-centres analysis of the point P finds out whether the point P coincides with a centre of the reference triangle as the point X takes the place of the ETC centres of a triangle. It is a rather simple and fast operation.

2-rule-point analysis is likewise performed on a generic construction, where a point, say X, is created with the **ETC1 rule** (for some triangle), a point, say Y, is created with the **ETC2 rule** and there is also a point, say P, which constructionally depends on the points X and Y. The points X and Y can independently occupy the position of various ETC centres of this triangle. The 2-rule-centres analysis of the point P finds out whether the point P coincides with a centre of the reference triangle as the points X and Y take the place of the ETC centres of a triangle.

Please, refer to Section 6.8.3 for details on ETC rules and the generic constructions. Also note that the triangle analysis on generic constructions (Section 6.7) allows more complex (but not so quick) observations on generic triangles.

Example 3

Here is an illustrative example how to perform a rule-points analysis[17].



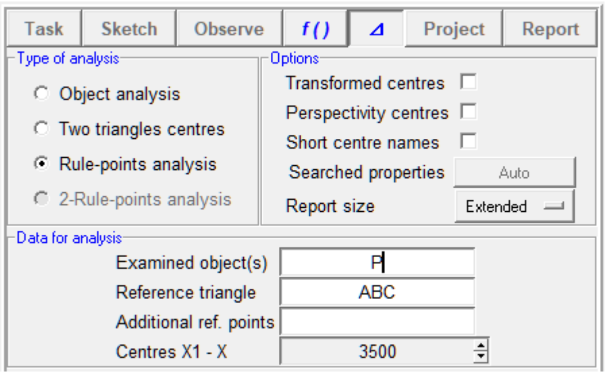
Given is a triangle ABC. For a point X, let P be the circumcentre of the pedal triangle of X in ABC. We investigate whether P is a known centre of ABC if X takes the position of the first 1000 triangle centres of ABC.

In the generic construction we define the point X by the ETC1 rule on the triangle ABC (i.e., X is a 'variable triangle centre' of ABC).

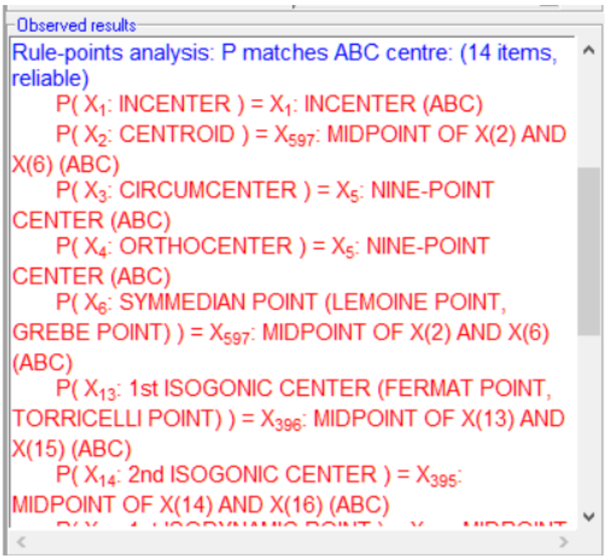
We construct the circle by projecting X onto the sidelines of ABC. (Note that A'B'C' it is also available as the pedal circle of a point in a triangle).

Now switch to the Triangle analysis module.

[17] OKExamples\OKG_Plus\Analysis_03.p

<table>
<tr>
<td>
</td>
<td>Select Rule-points analysis as a Type of analysis.

We choose Extended for Report size, so that X takes the position of the first 1000 ETC centres of ABC.

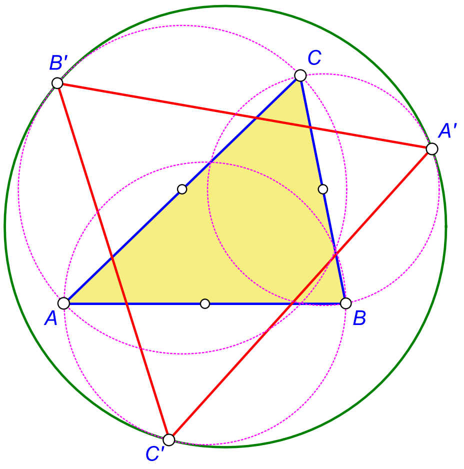
The positions of the point P are matched with the first 3500 centres of ABC. If also the transformations of these centres (e.g., isogonal conjugation, isotomic conjugation) should also be considered, select the Transformed centres option.</td>
</tr>
<tr>
<td>
</td>
<td>Here is the list of matches. For example, we find that if X is the symmedian point of ABC then the centre P of the circle is the midpoint the centroid and the symmedian point of ABC.</td>
</tr>
</table>

2.7.7 Triangle perspectivity analysis

With this analysis we investigate whether a triangle is perspective to different characteristic triangles of the reference triangle (e.g., the orthic triangle). The analysis considers up to 230 characteristic triangles of the reference triangle. For each triangle it is checked, whether it is perspective, parallelologic, orthologic or cyclologic to the examined triangle. In case of a positive result, the perspectivity centres are found – they are denoted as %1, %2,.... The set of all found perspectivity centres is denoted y %%.

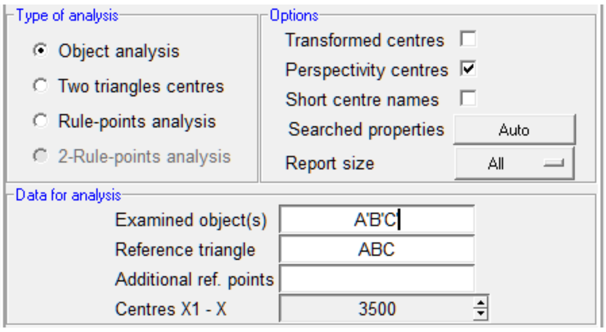
Example 4

Here is an illustrative example how to perform a triangle-perspectivity analysis[18].



Given is a triangle ABC. Consider the three circles with sides BC, CA, and AB as their respective diameters. Let A’, B’, C’ be the points of contact of these circles with the circle that touches them all internally.

We examine if the triangle A’B’C’ is perspective to any of the characteristic triangles of ABC.



Select **Object analysis** as Type of analysis.

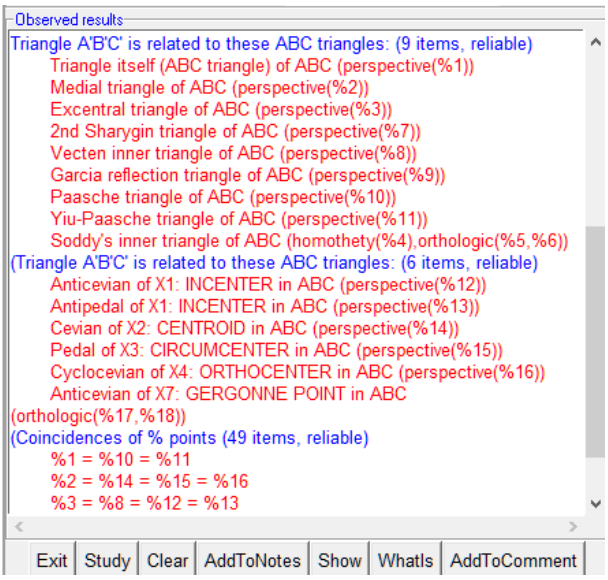
We select **All** for Report size, so that all related triangles will be considered and all the results will be displayed.

Select **Perspectivity centres,** so that the found perspectivity centres are stored as points %1, %2, ...

Our reference triangle is ABC and the examined triangle is A’B’C’. (Note that its circumcircle of A’B’C’will be examined as well).

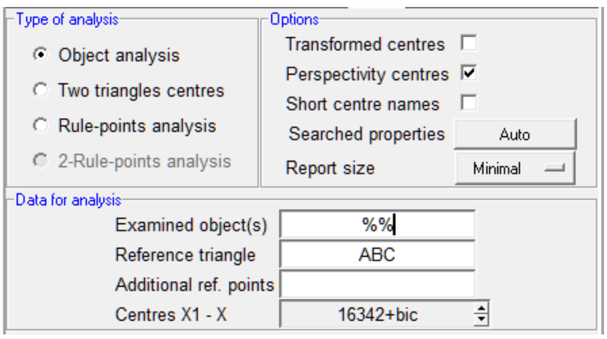
The options **Transformed centres** and **Considered centres** options will be taken into account later when analysing the %points.

[18] OKExamples\OKG_Plus\Analysis_04.p



Here is the list of observed characteristic triangles of ABC, which are homothetic, perspective, parellelologic, orthologic or cyclologic to A’B’C’. At the same time, all the corresponding perspectivity centres are computed.

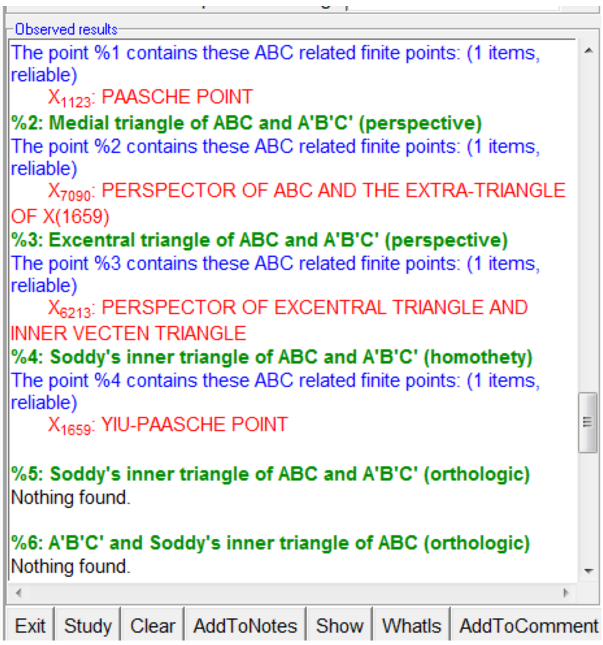
The calculated centres are listed at the end of the report. Some of the centres coincide.



The following operation must be performed **before we delete the results with the Clear button**.

We can examine the centres %1, %2, ... as ordinary points.

On the left we show how to examine all of the perspectivity centres at once (%% means the set of all found centres). Note that Report size was set to **Minimal,** otherwise the computation of detailed results will take time.



Here is a snippet of the results. We see that X(1123) is the centre of perspectivity of ABC and A’B’C’. Use **What is** to learn about the Paasche point and **Show** to display it.

Note that nothing about %5 is found. To get more observational data, enter %5 as Examined object, Increase **Report size** and perhaps checkmark **Transformed centres**.

2.8 Quadrilateral objects.

2.8.1 Basic concepts

The theory and classification of quadrilateral objects used in OK Geometry is based on vanTienhoven's text *Encyclopedia of Quadri-Figures*[19]. We use slightly different terminology, but the quadrilateral objects and their classification are the same as in his text.

We define a **quadrilateral** as a cyclic polyline with 4 vertices in general position. A quadrilateral therefore consists of 4 vertices, of which no three of them are collinear, and 4 sides. Each vertex is the end point of exactly 2 sides. There are exactly 4 pairs of contingent sides and exactly 2 pairs of non-contingent (opposite) sides. There are also exactly 4 pairs of vertices that share a common side and 2 pairs of opposite vertices (which have no common side). The lines through opposite vertices are called the diagonal lines. A quadrilateral can be self-intersecting (i.e. opposite sides can intersect) or not. In the later case, the region bounded by the sides may or may not be convex (Figure 27).

We use the terms quadrigon, quadrangle, quadrilateral as synonyms.

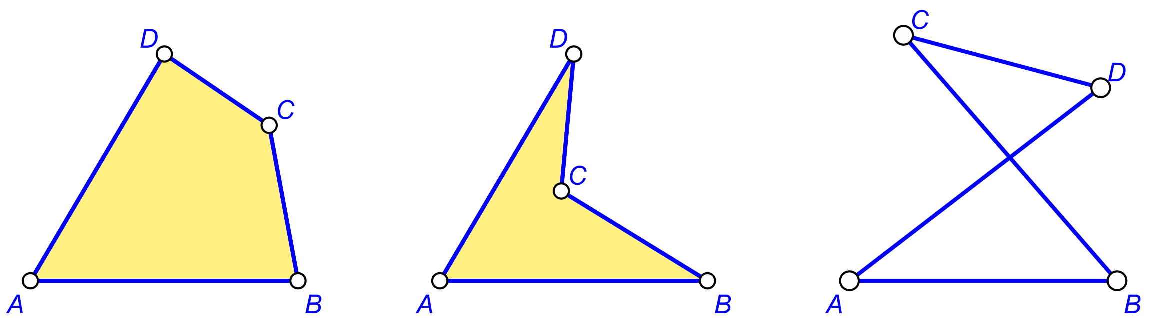


Figure 27

Quadrilaterals that have the same vertices but different sidelines are called **QA-siblings**. A quadrilateral ABCD has two QA-siblings: ABDC and ADBC (Figure 28). If the construction of an object from the vertices of a quadrilateral results in the same object for all QA-siblings, then this object is called a **QA-object**. For example, the centroid of the vertices of a quadrilateral is such an object. We refer to it as the QA-centroid (QA-P1). To avoid misunderstandings we always add to names the vanTienhoven's classification of the object, in our case QA-P1.

On the other hand, the intersection of diagonals of quadrilaterals is clearly not a QA-object.

[19] https://chrisvantienhoven.nl/mathematics/encyclopedia

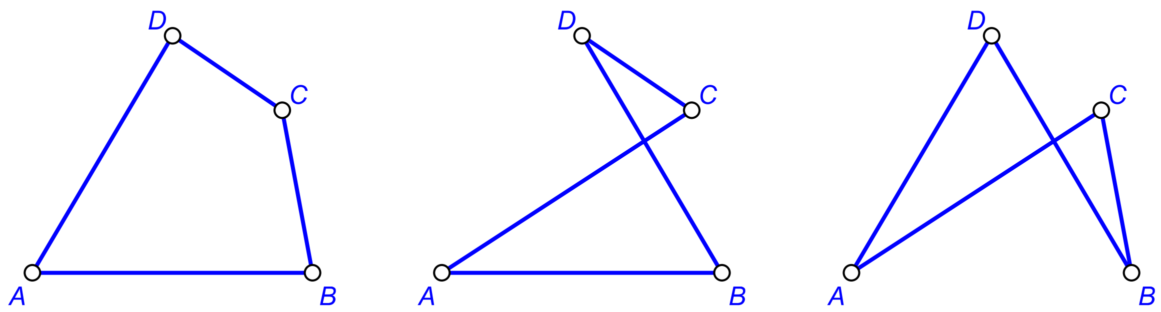


Figure 28

Quadrilaterals that have the same side-lines but different vertices are called **QL-siblings**. A quadrilateral ABCD has 2 QL-siblings (Figure 29): AECF and BEDF, where E = AB ∩ CD, F = BC ∩ DA.

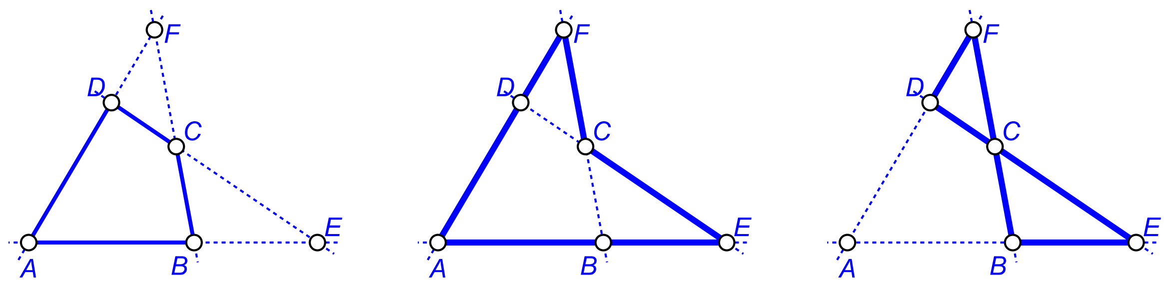


Figure 29

If the construction of an object from the side-lines of the quadrilateral results in the same object for all its QL-siblings, then this object is called a **QL-object**. For example, the Newton line of a quadrilateral (i.e. the line through the midpoints of the diagonals, Figure 30) is a QL-object. We refer to it as the QL-Newton line (QL-L1). To avoid misunderstandings we always add to names the vanTienhoven's classification, in our case QL-L1.

On the other hand, the intersection of diagonals is clearly not a QL-object.

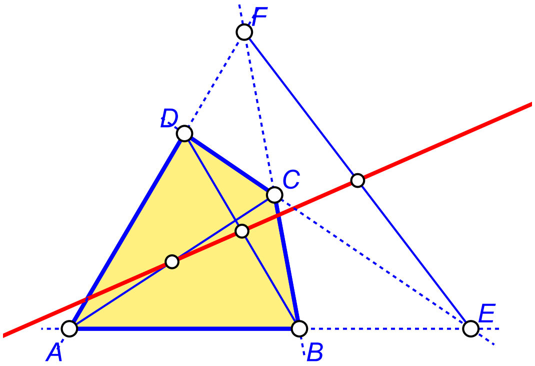


Figure 30

Finally, objects resulting from constructions that are invariant under cyclic permutation of the vertices (or sidelines) of a quadrilateral are called **QG-objects**. QG-objects are clearly also QL- and QA-objects. For example, the third diagonal of ABCD (line EF on Figure 30) is an example of a QG-object. We refer to it as QG-3rd diagonal (QG-L1). To avoid misunderstandings we always add to names the vanTienhoven's classification, in our case QG-L1.

2.8.2 Special commands for quadrilateral objects

The Sketch Editor in Plus mode has a subgroup of **Special** commands dedicated to quadrilateral objects. The commands in this group are similar to those for the triangle object and have a similar syntax.

You can use the commands in this subgroup to construct various objects that are related to a quadrilateral. In contrast to the triangle analogues, however, **there is no (default) reference quadrilateral**. In each command, you must specify the quadrilateral to be used, either by its four vertices or by the defining polyline.

The quadrilateral objects are less numerous than the triangle objects. OK Geometry can create a large part, but not all, of the objects from vanTienhoven's *Encyclopedia of quadri-objects*.

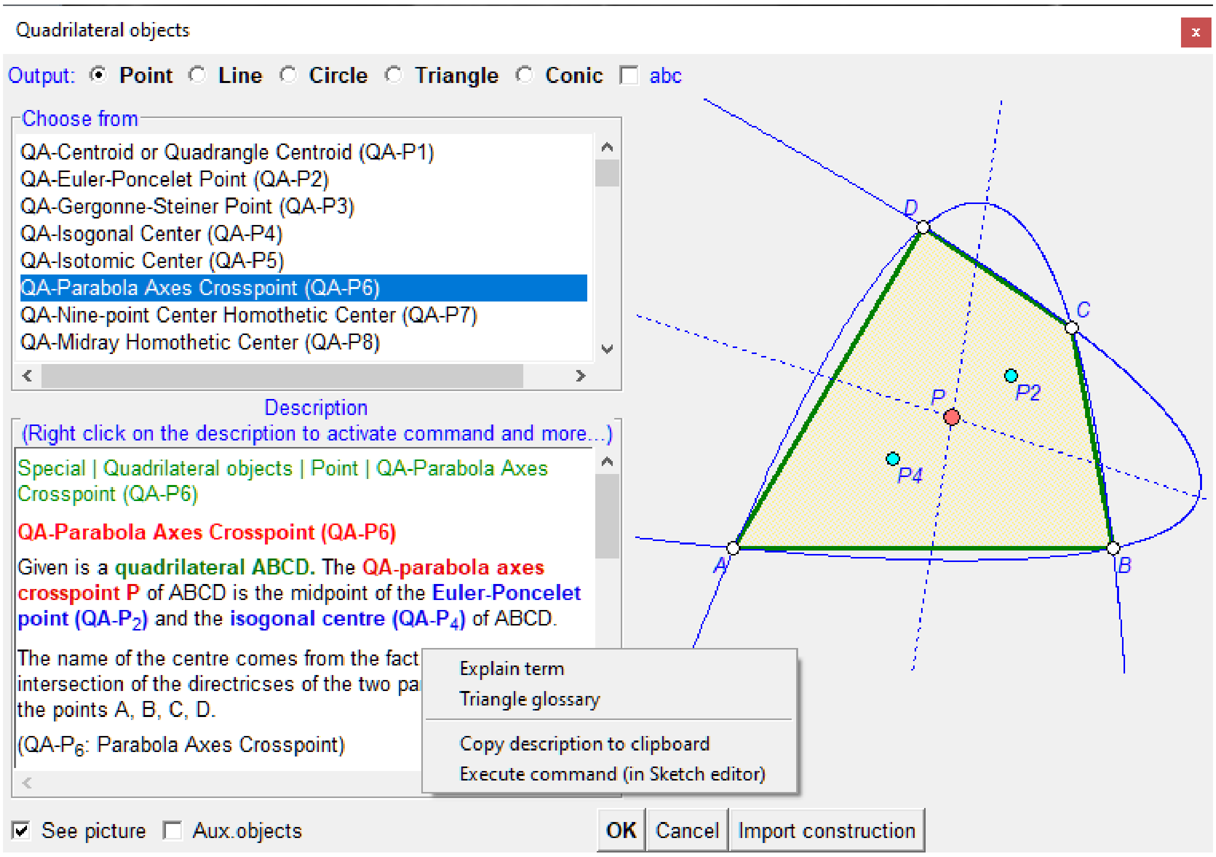


Figure 31

Figure 31 shows a typical group of commands for characteristic objects of a quadrilateral.

- In the first line, select the type of object you want to create (in our case, a point).
- The commands for objects in this group refer to the quadrilateral that you specify after clicking on OK. You can specify the quadrilateral either by its vertices or by its defining polyline.
- As with the triangle objects, a right click on the Description part of the form displays additional commands.

Here is the list of command for quadrilateral objects:

A quadrilateral	The command creates a quadrilateral in which all four vertices are free. **Note.** The quadrilateral created is not a (default) reference quadrilateral. It also has no cyclic structure attached as is the case of the reference triangle created with the command 'A triangle'. **Note.** The command is active if no objects have been created yet.

Quadrilateral objects	The quadrilateral objects are organised in several lists: - **quadrilateral points**, e.g., the QA-centroid (QA-P1); - **quadrilateral lines**, e.g., the QL-Newton line (QL-L1); - **quadrilateral circles**, e.g., the Miquel circle (QL-Ci3); - **quadrilateral triangles**, e.g.the QA-diagonal triangle (QA-Tr1); - **quadrilateral conics**, e.g., the QA-nine-point conic (QA-Co1).
Analyse objects wrt. quadrilateral	The command performs a simple observation of the object you click on (point, line, circle, conic). You are immediately asked for the quadrilateral with respect to which the object is to be observed. The output of the command is a list of the quadrilateral's objects that are (in some way) contingent to the observed object. **Note.** Several commands are available to visualise the listed objects and to obtain information about them. You can access them at the bottom of the form or with a right click on the corresponding object on the list of results. The example in Figure 32 shows the results of the observing a line with respect to the quadrilateral ABCD. It is noted that the observed line is parallel to the QL-Steiner line and to the QL-pedal line and that it passes through the QA-Harmonic centre of the quadrilateral ABCD. By a right click on the results or using the commands at the bottom right of the results, we can view the mentioned objects and obtain information about them.

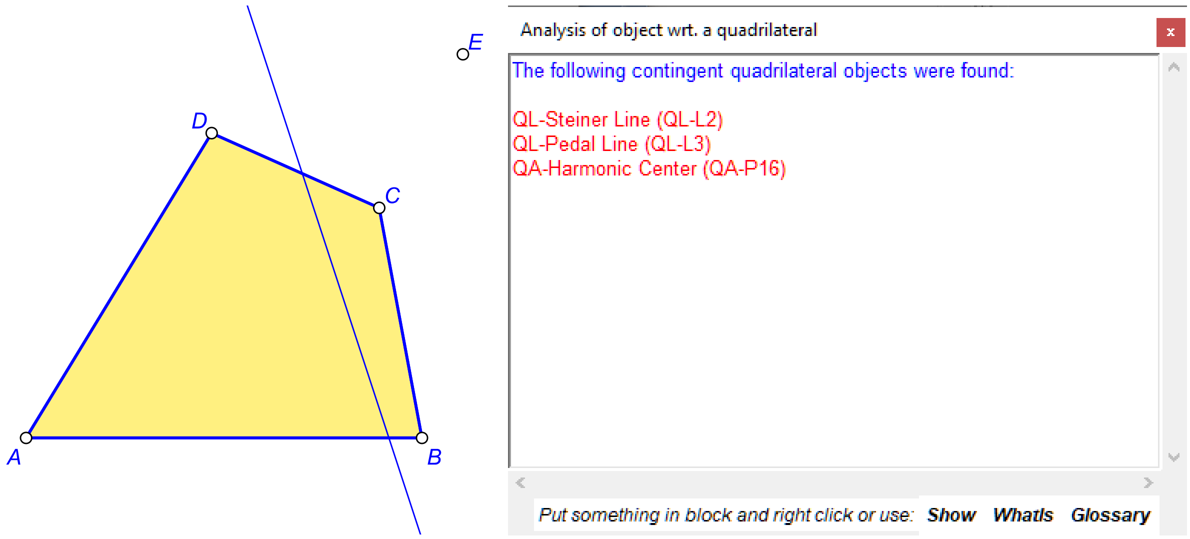


Figure 32

3 Advanced query of objects

3.1 Query commands

The observation (**Observe** button) of a configuration provides a list of observed properties. The list is sometimes very long, so it is not easy to focus on the properties that are relevant to us. If you are looking only for properties that involve a specific object, you can use the query buttons (magenta coloured buttons on the right hand side of the displayed construction, e.g., , , see Figure 33). The simple use of the query commands is explained in the OK Geometry Basic reference manual. In simple terms, the basic query operation filters the properties so that only properties pertaining to specified object are displayed.

When an object is queried, a row of dotted buttons appear above the list of properties (Figure 33, left). The number of dots on the buttons indicates the rigour of the query: the fewer the dots, the less extensive the information displayed, i.e. less relations with less objects are displayed.

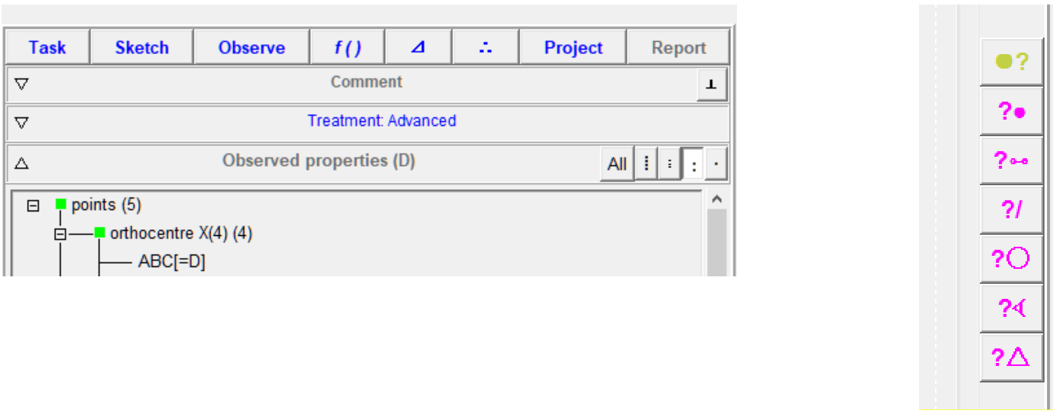


Figure 33

For the sake of completeness, here are the descriptions of the commands associated to the query buttons:

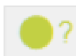
Command	Meaning
Mark/Toggle unknown status	This command (accessible also via button ) is used to mark/unmark objects that are considered unknown (see Section 3.2). Objects marked as unknown are coloured pale-green. Such objects are not considered in the query unless they are the object of the query. A typical use of this function is the study of Euclidean construction problems. In a complete construction, we mark objects that need to be constructed from known objects.

Query labelled point	Click on the point to be queried. In the left pane, only the properties related to the queried point are listed.
Query-Segment	Click on the segment to be queried or its endpoints. In the left pane, only the properties related to the queried segment are listed.
Query-Line	Click on the line to be queried or on two labelled points on it. In the left pane, only the properties related to the queried line are listed.
Query-Circle	Click on the circle to be queried or on three labelled points on it. In the left pane, only the properties related to the queried circle are listed.
Query-Angles	Click on the labelled points (ray, vertex, ray) or on the rays of the angle to be queried. In the left pane, only the properties related to the queried angle are listed.
Query-Triangle	Click on the labelled triangle vertices. In the left pane, only the properties related to the queried triangle are listed.

3.2 Known, unknown and auxiliary objects

By default, the query buttons in OK Geometry Plus have an additional functionality called **advanced query**. The advanced query operation of an object attempts to relate that object to a specified subset of objects (called **known objects**) in a configuration and also to many objects (called **auxiliary** objects) that are geometrically derived from the known objects.

The advanced query is of great help in hypothesising solutions to construction tasks. Namely, in construction tasks we start from some **known** objects and try to relate **unknown** objects to the already known ones. The advanced query can also reveal surprising relations between objects in a configuration in other situations.

Constructed objects are **known** by default. You can make an object **unknown** using the command *Action|Mark|Toggle unknown status,* which is also accessible from the green dot button  on the right side of the screen.

Advanced query is only available in OK Geometry Plus. It is enabled by default, but can be turned off in OK Geometry configuration (*Configure|General options|Observe|Advanced query operations*). Turning the Advanced query OFF will reduce the observation time.

Let us illustrate the use of advanced query with a very simple construction task:

Example 1

Inscribe a square PQRS into a given triangle ABC, as shown in Figure 34 (left)[20]?

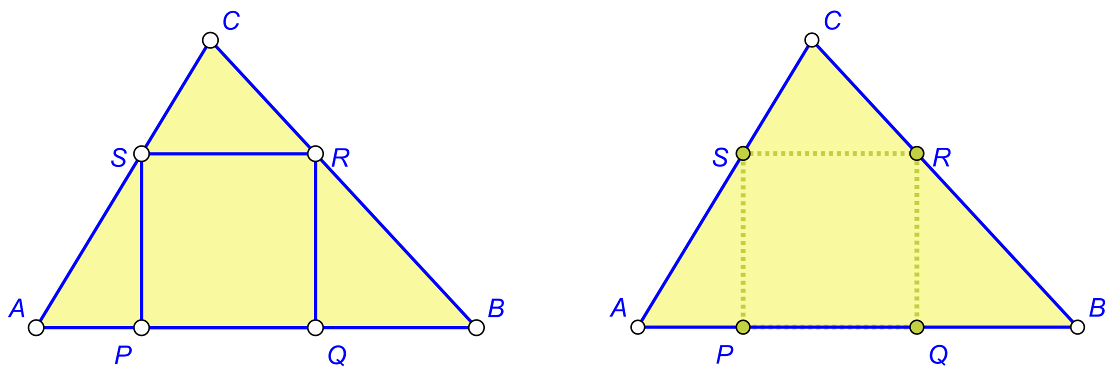
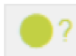


Figure 34

First, we must somehow obtain a final configuration (Figure 34, left). The easiest way is to start with a square PQRS and draw an appropriate triangle ABC around it.

The known objects in our initial construction task are the vertices A,B,C and the three sides of the triangle. The unknown objects are the vertices P,Q,R,S and the four sides of the square.

Using the command **Toggle unknown status** (accessible via the green dot button ) we mark the unknown objects. Unknown points turn pale green, and the unknown lines turn dashed and pale green (Figure 34, right).

At this point we **Observe** the configuration. We obtain the usual list of properties of the configuration. On the far right of the display appear the magenta query buttons for querying specific objects. Advanced query occurs only when we query an unknown object (otherwise a simple query is performed).

We query the point R with the Query point button  (i.e., we first click on the button and then on the point R). After a while, a short list of properties pertaining to the point R appears. At the top of the list you can find new properties that involve auxiliary points derived from known objects. A click on the property visualises this property.

[20] OKExamples\OKG_Plus\AdvQuery_01.p

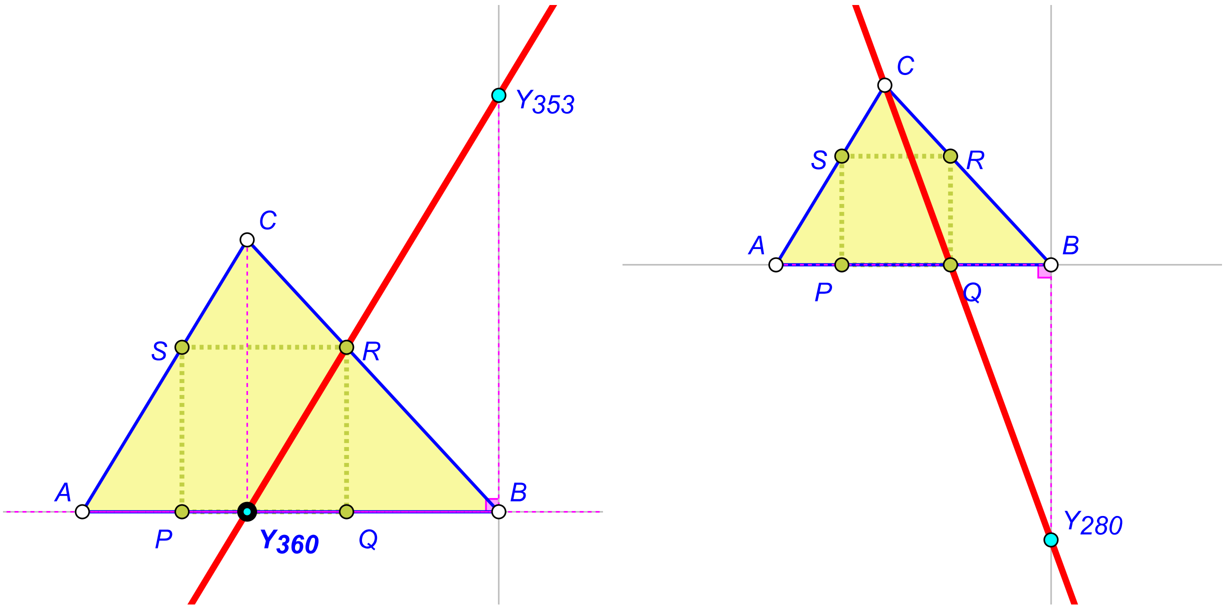


Figure 35

For example: The property *Lines invoving R* contains, among other, the relation

Lines invoving R

R, _360, _353

Note that you can get different names of auxiliary objects for the same relation, i.e. different numbers with underscore.

Clicking on the property displays the relation (Figure 35, left)[21].

If you place the cursor on an auxiliary point in the construction, an explanation of the point will appear (you can activate the visual explanation by pressing the **F8** button on the keyboard). In the example shown, _360 (or Y360) is the orthogonal projection of C onto the line AB, and the point _353 (or Y353) is point A rotated by 90 degrees clockwise around B.

As usual, right-clicking the construction area opens a context menu with various commands that allow you to export the figure or edit it, include it in a project, etc. In this case the underscores in labels of auxiliary points become Y.

To get information about point Q, apply the **Query point** command  to point Q (no need to ***Observe*** again). Among the properties you will find one (Figure 35, right) that allows you to construct the point Q (the auxiliary point _280 is the point A rotated anticlockwise around B by 90 degrees).

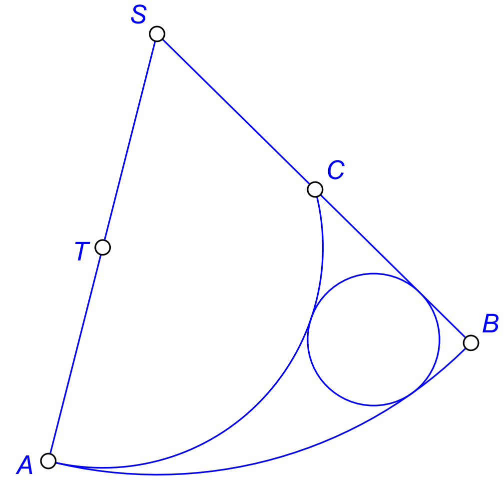
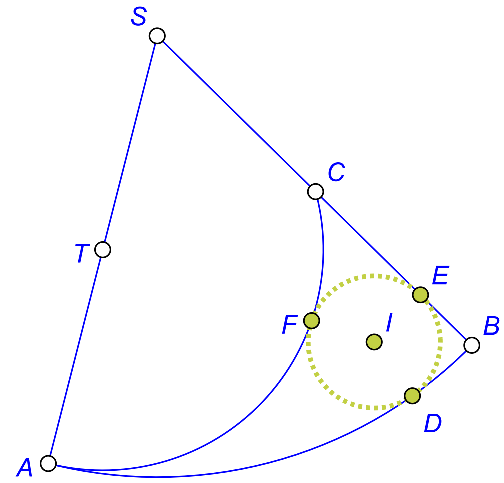
Both properties shown are easy to prove.

Note. A change in the collection of **unknown** objects requires a new observation with **Observe** command.

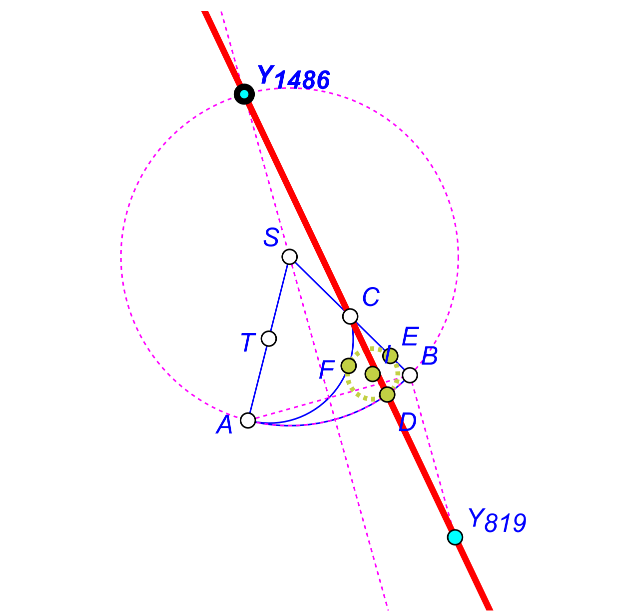
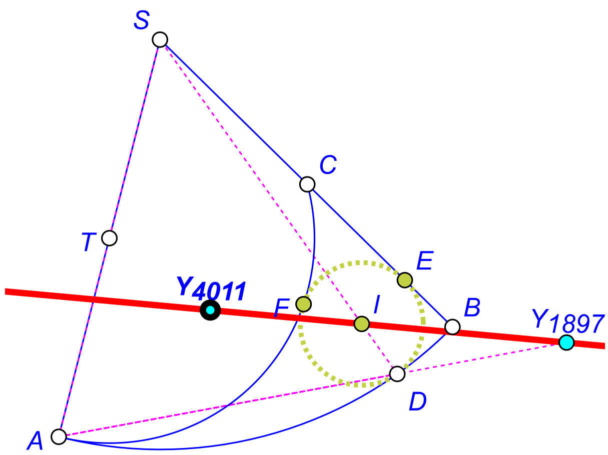
[21] OKExamples\OKG_Plus\AdvQuery_02.p

3.3 Advanced query examples

Example 2

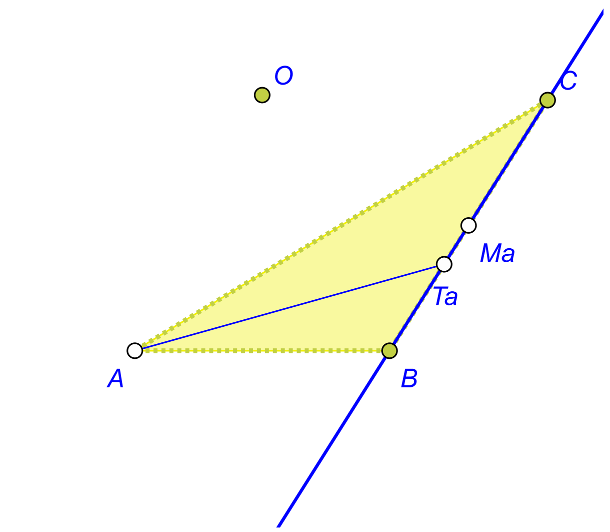
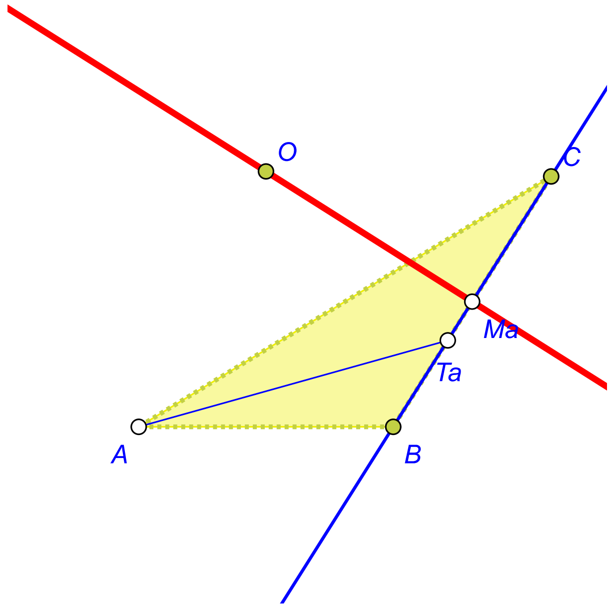
In a circle sector with centre S and A,B as arc endpoints, there is a second arc. The centre T of the second arc is the midpoint of AS, its endpoints are A and a point C on the segment BS. Construct a circle that touches the two arcs and the segment BC.[22]	
When constructing a circle that is tangent to three known objects, it is often advisable to determine the points of tangency of the circle with objects tangent to it. The inscribed circle is easily drawn with OK Geometry, using the command *Circle 3 objects*. We also draw the points of contact D, E, F and the centre I of the circle. To get clues for a construction, we mark the inscribed circle, the touch points D,E,F, and the centre I of the inscribed circle as unknown objects.[23] Then we proceed with the command **Observe.** Then we query the unknown points.	

[22] OKExamples\OKG_Plus\AdvQuery_03.p
[23] OKExamples\OKG_Plus\AdvQuery_04.p

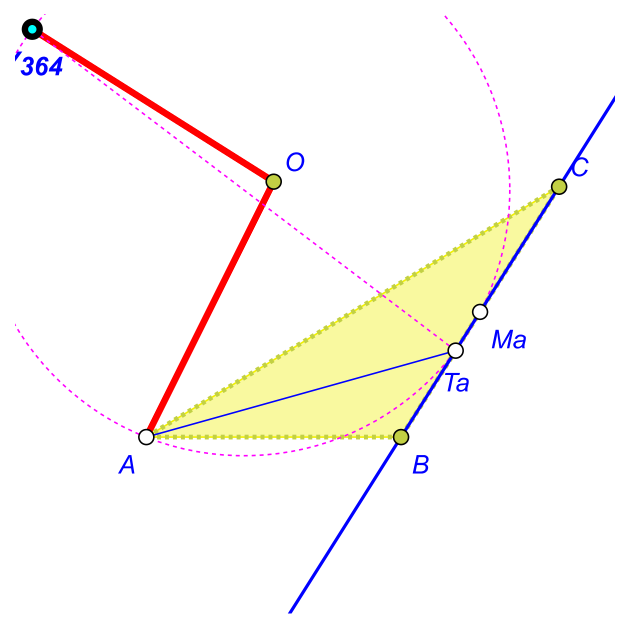
Here is one of several suggestions to obtaining the point D: The point D lays on the line through the auxiliary points Y1486 and Y819, where 16. Y1486 is the intersection of the circle k(S,A) with the bisector of AB; 17. Y819 ia the image of A rotated by 90 degrees counterclockwise around B. Note that the displayed line does not contain the point C (as it might appear in the figure).	
Once the point D is known, it is recommended to mark it as known, and repeat the **Observe** procedure. The advanced query of the centre I gives several suggestions for the construction of I (obviously I lays on the segment SD). According to one of the suggestions, the point I lays on the line through Y4011 and Y1897, where 18. Y4011 is the symmedian point of the triangle ADS, and 19. Y1897 is the mirror image of the midpoint of AD in point D.	

Example 3

The following example is the 25th problem in the famous Wernick list of construction problems.

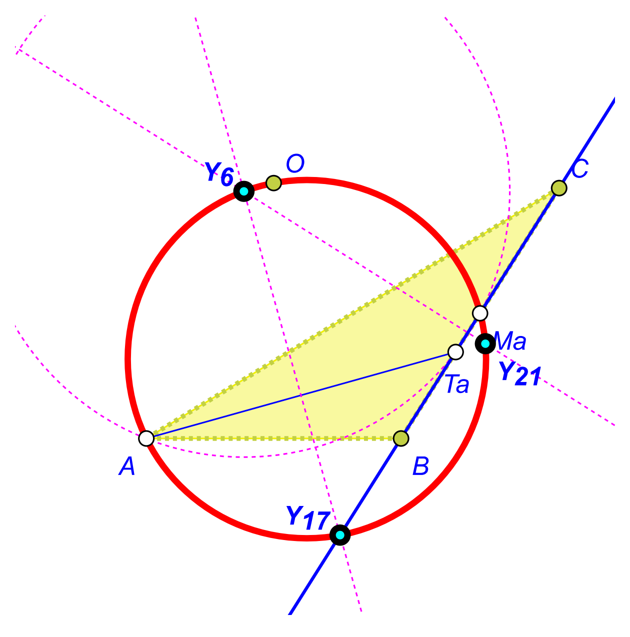
	Construct a triangle ABC if the location of the following points of the triangle ABC is known: 20. the vertex A, 21. the midpoint Ma of the side BC, 22. the intersection Ta of the angle bisector at A and the side BC. In a triangle ABC we construct the points Ma and Ta. Note that A, Ma, Ta, and the line BC are known (as is the location of the centroid). We mark everything else as unknown.[24]
	Since observation command and query of points B, C yield nothing useful, we try to determine some other important points of the triangle. The unknown circumcentre O of the triangle ABC is a good choice. An observation and query of the circumcentre O yields several hypotheses that can be quite easily proved. Once we know the position of O, the construction problem will be easily solved. Here is the first (trivial) property: the line O(Ma) is perpendicular to BC in Ma.

[24] OKExamples\OKG_Plus\AdvQuery_05.p



Furthermore, O lays on the bisector of the segment A(Y364), where

23. Y364 is the antipode of Ta on the circle through A,Ta,Ma.



The circumcentre O also lays on the circle through the points A, Ma, Y17, where

24. Y17 is the intersection of the line BC with the bisector of A(Ta).

Note that O is the antipode of Y17 in the mentioned circle, which contains also the points

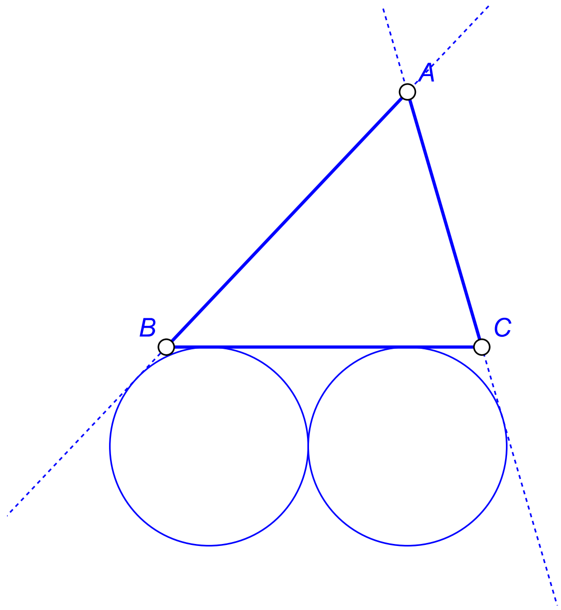
25. Y21 - the intersection of A(Ta) with the bisector of (Ma)(Ta),
26. Y6 - the intersection of the line BC with the bisector of (Ma)(Ta).

Example 4

The following task is related to the Paasche point of triangle (Figure 21).

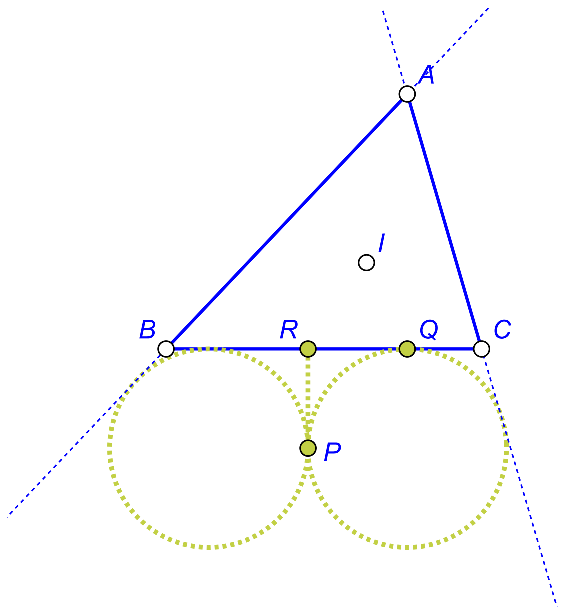
<table>
<tr><td>Given is a triangle ABC. In its exterior we want to construct two congruent circles touching each other externally, both touching the side BC, one touching the line AB and the other touching the line AC.[25]

We are looking for a solution that is nicer than the solution via homothety of a false solution.

To study the construction, we create a described configuration, starting from the congruent circles.</td><td>
</td></tr>
<tr><td>If the vertices of a triangle are known in a construction task, it is sometimes useful to add one or more of its simple centres. A little investigation (and common sense) shows that the incentre I of the triangle ABC is a good choice.

We also add some points as unknown points. If we manage to construct one of them, the task is solved.

We mark the two circles and the added points P, Q, R as unknown.[26]

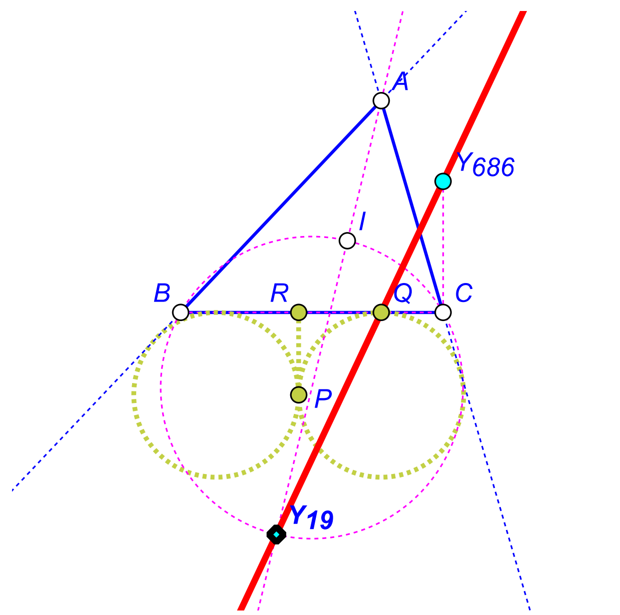
Then we start the observation.</td><td>
</td></tr>
</table>

[25] OKExamples\OKG_Plus\AdvQuery_06.p
[26] OKExamples\OKG_Plus\AdvQuery_07.p

First we query the point Q. OK Geometry suggests that Q is the intersection of the line BC and the line $Y_{19}Y_{686}$, where

27. Y_{19} is one of the intersections of the circle through B, C, I and the line AI; it is also the antipode of I in this circle;
28. Y_{686} is defined by the condition: the angle BCY_{686} is a right angle; $|BC| : |CY_{686}| = 2 : 1$.

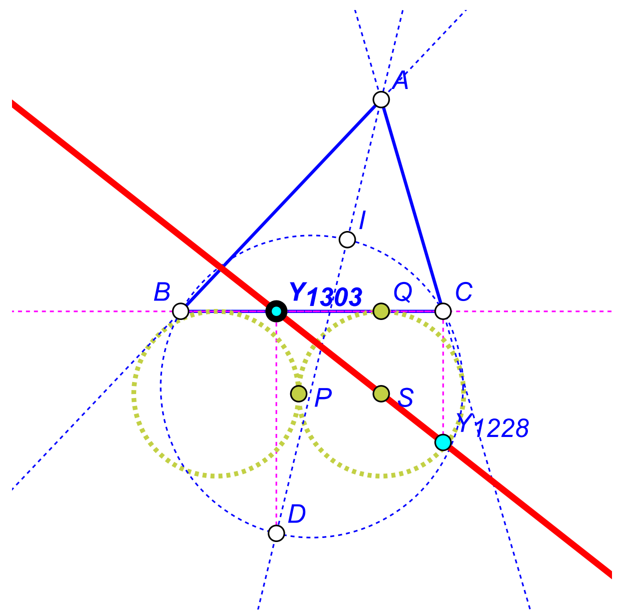
A solution can be found by querying the point R, but not P.



Here is a nice solution for the centre S of one of the two circles. We have added the known point D, the antipode of I.

The point S is the intersection of the line CD (i.e., the angle bisector at C) and the line $Y_{1228}Y_{1303}$, where

29. Y_{1228} is defined by the condition: the angle BCY_{1228} is a right angle; $|BC| : |CY_{1228}| = 2 : 1$.
30. Y_{1303} is the projection of D onto the line BC.



4 Observing algebraic relations (formulae)

4.1 Introduction

OK Geometry observes geometric properties as well as simple algebraic relations in dynamic constructions. When detecting simple algebraic relations with the **Observe** command in the main menu, OK Geometry considers a whole lot of geometric quantities: all distances between labelled points, all angles between labelled points, and areas of triangles with vertices on labelled points. OK Geometry, for example, detects whether the area of a triangle is the arithmetic mean of the areas of two other triangles in a configuration, or whether some angle between labelled points is twice as large as some other angle in the configuration.

OK Geometry is also capable of observing more complex algebraic relationships between selected geometric quantities in a configuration. This is accomplished with the **Observe formulae** commands that we describe in this section.

The **Observe formulae** command consists of three variants: **simple**, **advanced**, and **triangle**. The variants differ only in the way the geometric quantities are conveyed to the **Observe formulae** command.

- The **simple** variant searches for algebraic relations involving specified quantities, which were measured beforehand in the **Sketch Editor**.
- The **advanced** variant looks for algebraic relations that involve specified geometric quantities that need not to be measured beforehand.
- The **triangle** variant searches for algebraic relations between a specified geometric quantity or ratio of two quantities of a triangle and various unspecified geometric quantities of this triangle.

You can access the *Observe formulae* command via the main menu *Commands|Observe formulae* or via the ***f()*** button in the main toolbar (Figure 36, left). Then select the desired variant In the form that appears (Figure 36,right). Note that the command is available also in Sketch Editor menu as: *Special|Observe formula.* You can switch between the variants (modes) while using the command.

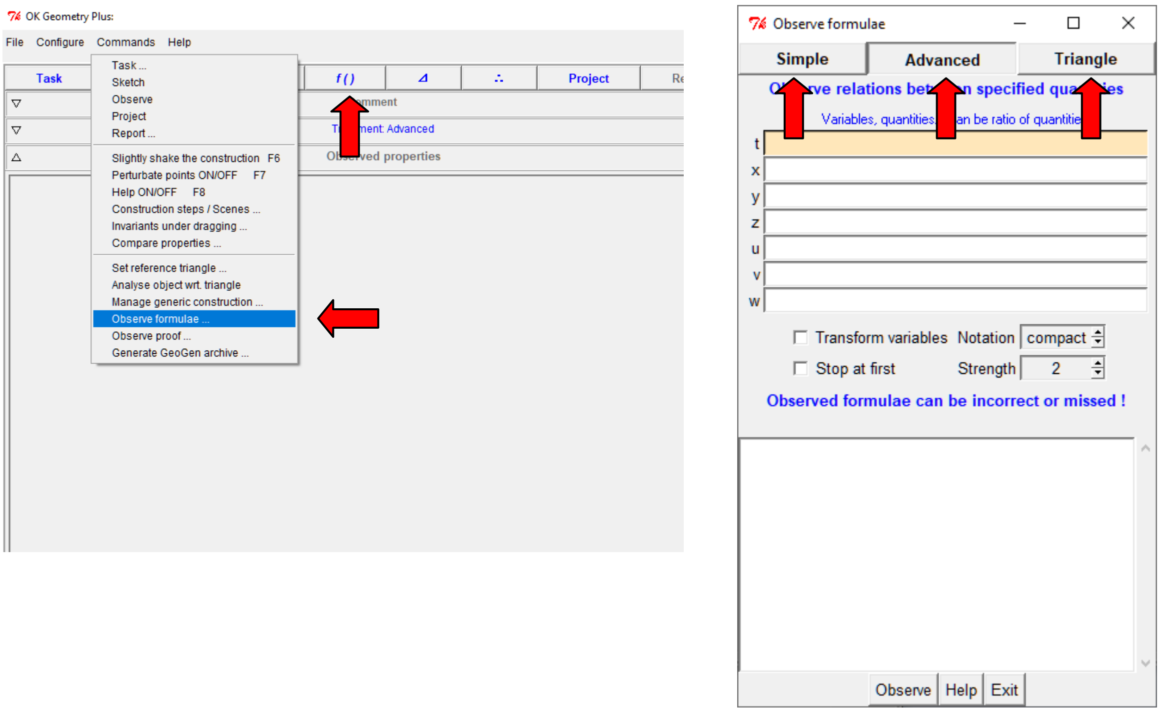


Figure 36

Figure 37 shows a simple illustration of the use of the **Observe formulae** commands. Given is a triangle ABC. Let ri be the radius of its incircle. Furthermore, let ra be the radius of the circle that touches the sides AB, AC and the incircle and lays between A and the incircle. Define rb and rc cyclically. We wonder if there is a nice relationship between the radii ra, rb, rc, ri.[27]

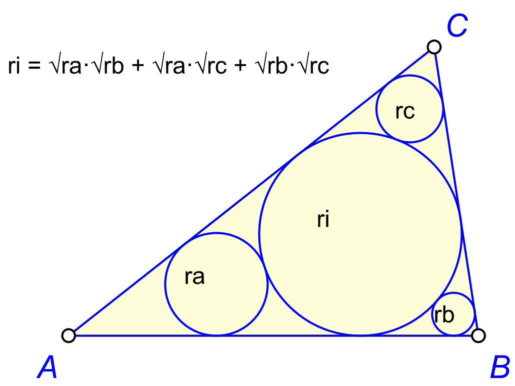


Figure 37

[27] OKExamples\OKG_Plus\Formula_01.p

The **simple** variant of the command **Observe formulae** requires that in the dynamic construction we first measure the radii of the four circles. **Observe formulae** then looks for algebraic relationships between the four radii and detects that:

$$ri = \sqrt{ra \cdot rb} + \sqrt{rb \cdot rc} + \sqrt{rc \cdot ra}$$

The **advanced** variant of the command **Observe formulae** performs a similar task, except that the considered quantities do not have to be measured beforehand in the Sketch Editor.

The **triangle** variant of the command **Observe formulae** examines only a single geometric quantity (length or area) or a ratio of two geometric quantities and tries to relate it algebraically to the reference triangle. The triangle variant, for example, detects the following relationships (rewritten by algebraic manipulation) for ra:

$$ra = r \cdot \sin\frac{A}{2} \cdot \sin\frac{B}{2} \cdot \sin\frac{C}{2} \cdot \frac{1 - \sin\frac{A}{2}}{1 + \sin\frac{A}{2}}$$

where r is the circumradius of the triangle. It also detects the following expression for the ratio of $ra{:}ri$

$$\frac{ra}{ri} = \frac{1 - \sin\frac{A}{2}}{1 + \sin\frac{A}{2}}$$

It is important to keep in mind that the command **Observe formulae** merely observes whether geometric quantities satisfy some relations that can be (in some way) reduced to polynomial expressions. It does not prove the relations and it may also overlook existing relations. Thus, it is a useful heuristic tool for obtaining hypotheses that should be proved in a mathematically decent way. To prove and to geometrically interpret the mentioned algebraic observations related to Figure 37, for example, is a nice recreational task.

The process of observing formulae may take time, especially for the **triangle** variant. For this reason you can monitor the fraction of work done and you can also interrupt the observation process.

In the following sections we explain how to use the three variants of the **Observe formulae** command.

4.2 Observe formulae – simple

In this section we show how to use of the **simple** variant of the command.

We first explain the procedure on a very trivial example.

Example 1

How does the area of an isosceles trapezium relate to its sides *a,b,c,d* ? [28]

1. Provide a dynamic construction of the object(s) under consideration. The construction can be imported from some dynamic geometry software or created in OK Geometry Sketch Editor.

[28] OKExamples\OKG_Plus\Formula_02.p

2. Measure the potentially relevant geometric quantities in the Sketch Editor. You can use the commands *Number|Distance, Number|Radius, Number|Length* or *Number|Area.* In our case we have measured the sides *a, b, c, d* and the area *Ar* (Figure 38).

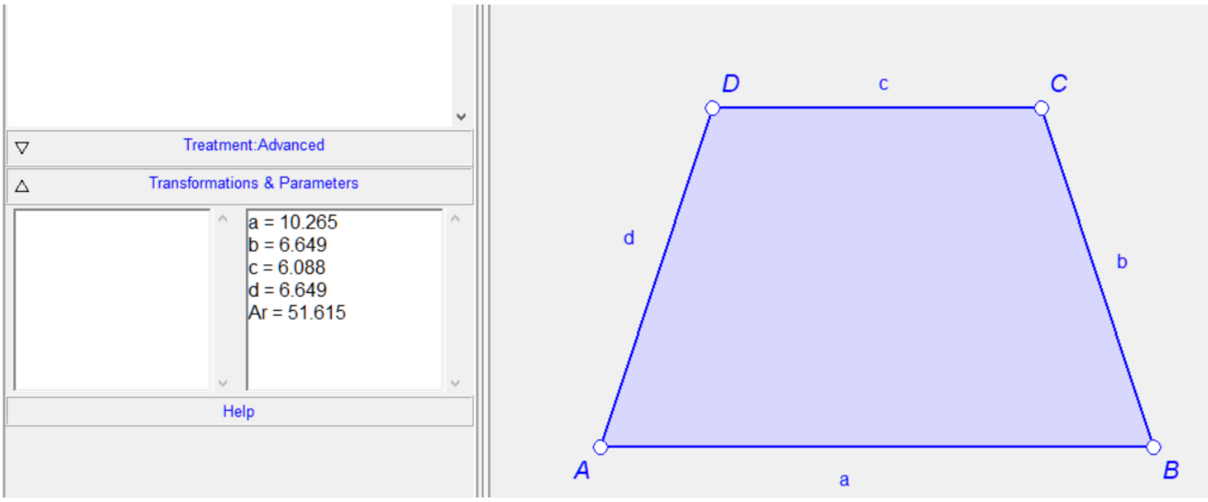


Figure 38

3. Apply the command *Command|Observe formulae|Simple*. A form appears. In the list of measured quantities, select the ones you think are relevant. In our case we have selected all of them. Then click the **Observe** button (Figure 38, left).

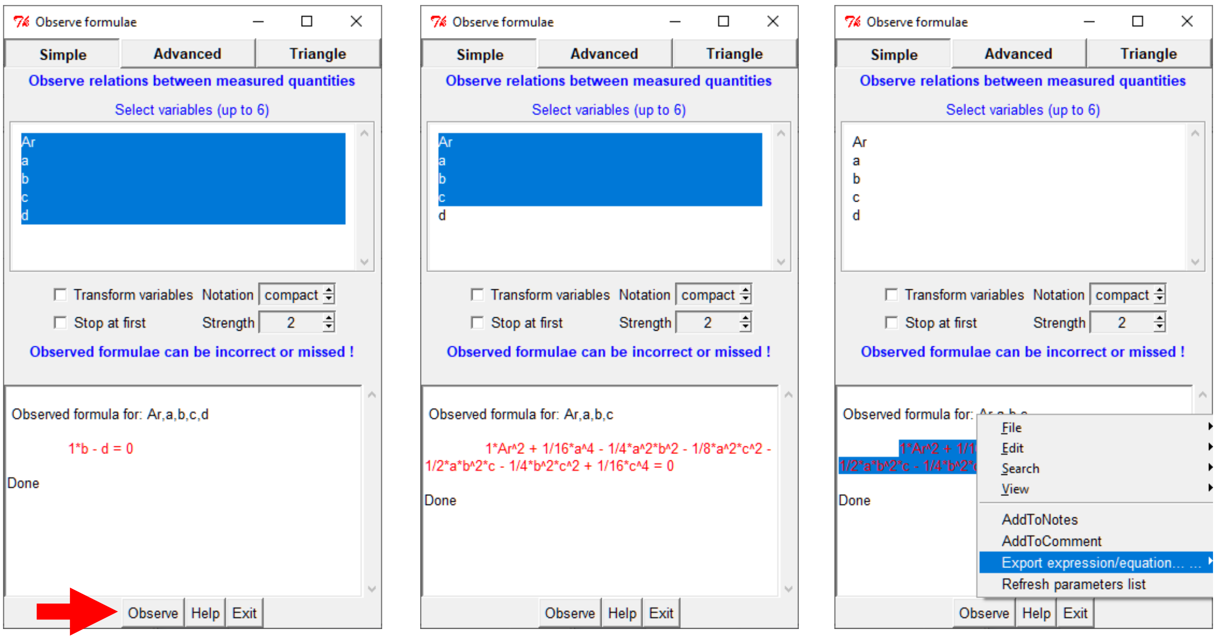


Figure 39

4. OK Geometry looks for a polynomial relation of the lowest degree containing the selected variables. In our case there is a trivial relation *b = d,* which is certainly not of our interest. Therefore we unselect a redundant variable *b* or *d* and repeat the observation (Figure 39, centre).

5. The obtained polynomial relation usually requires additional algebraic manipulation, depending on your aesthetic standards and mathematical goals.
6. To add a further dynamic numerical test for the obtained relation, proceed as follows (Figure 39, right):
    a. Put the considered algebraic relation in a block.
    b. Right click on the relation and select the command *Export expression|Add as parameter in construction.*
    c. A parameter *Fcondx* is added to the dynamic construction. The parameter numerically checks the correctness of the equation in the block. You can further dynamically modify the construction and observe if *Fcondx* has the value 'True'. Note that you can declare as a parameter only a part of the relation by putting in block only this part of the relation.

4.2.1 Comments on observed formulae

It is important to be aware of the status of the relations found regardless of the variant used (**simple**, **advanced** or **triangle**).

1. **Observing formulae** command considers specified geometric quantities of a dynamic construction in order to detect polynomial relationships between them or relations that can be reduced to a polynomial relationship of degree up to 4. If several variables are involved, the maximal degree can be 3.
2. If one or more relations of a certain degree are found, **Observe formulae** will not search for relations of higher degree.
3. **The observed algebraic relations are the result of numerical observations of the dynamic construction. The relations found are not proven and there is no guarantee they are correct. A relationship can also be overlooked.** In general, overlooking a relation occurs more likely than finding a relation that is not valid at least locally.

4.2.2 Setting the parameters for observing formulae

Observing formulae commands have a few options:

Notation	The **compact** option uses some idiosyncratic notation that is explained in the advanced and the triangle mode of Observe formulae. For example, sA2 stands for for sin(A/2). The **extended** option, when necessary, adds to compact formulae a readable version of them. The **xyz** option writes the formulae in terms of the variables *t,x,y,z,u,v,w*. This way the expressions are easier to read and can be more easily recognised by computer algebra systems.
Transform variables	If checkmarked, OK Geometry will look for polynomial relations that involve selected variables and also transformed variables (e.g. the squares of variables). In this case the calculation takes more time.

Stop at first	In many cases **Observe formulae** searches for several results. For example, if 'Transform variables' is checkmarked, results are searched for each type of transformation of variables. The **triangle** variant of **Observing formulae** considers even more types of results, and all this can be time-consuming. In order to reduce the calculation time, checkmark the 'Stop at first' option. In this case the computation will stop as soon as the first relation is found.
Strength	A higher strength means a higher probability of finding a relation (at the cost of longer computation time). You can set the strength from 1 to 4, the default value is 2. The usual strategy is to work with strength 2. If the observation does not find a relationship then try again with a higher strength.

4.2.3 Managing the results of Observe formulae

The results of the command **Observe formulae** are written in an ordinary text editor. In addition to the common text editor commands, available by a right-clicking on the results area, there are some specific commands (see Figure 39, right). All specific commands refer to the text in block. **So, to use the specific commands, you must first put in block the text on which to apply the command**.

Here is a brief description of the commands:

Add to Notes. The text in the block will be added to the Notes section of the project report.

Add to Comment. The text in the block will be added to the Comment section of the current task.

Export expression/equation. A **single relation** or **part of a single relation** in **compact** or **extended notation** can be put in a block and exported in different ways for different purposes.

- **As triangle expression.** The relation in the block is converted so that it can be used in the command *Number|Triangle expression* of the Sketch Editor.
- **As Xcas/Giac input**. The relation in block is converted so that it can be used as input for algebraic manipulation in Xcas/Giac software.
- **As Derive input**. The relation in block is converted so that it can be used as input for algebraic manipulation in Derive software. The Derive input options should be set to: Input mode = Word, Case sensitivity = Sensitive.
- **As Maxima input.** The relation in block is converted so that it can be used as input for algebraic manipulation in Maxima/wxMaxima software.
- **As Mathematica input.** The relation in block is converted so that it can be used as input for algebraic manipulation in Mathematica software.
- **As parameter in construction.** This operation can be used as a check of the relation under a manual dragging of the dynamic construction. The expression in the block is converted to a parameter of the current construction. If the block contains the = symbol, the generated parameter is a condition *Fcondx*, otherwise it is an ordinary numeric parameter called *Fexprx*.

Refresh parameter list. Use this command when, while observing formulae in the **Simple mode** you define new parameters as geometric quantities. So, during the process of formulae observation you can define new parameters as geometric quantities in the Sketch Editor and refresh the parameter list.

You may find useful also the ordinary **Copy** command for transferring **single relations** or **part of relations** to computer algebra systems (CAS). The xyz notation is probably the best option for this operation.

4.2.4 More examples

Example 2

Given is a triangle ABC. Let A', B', C' be the points of contact of the incircle of ABC with the sides BC, CA, AB of the triangle ABC (Figure 40). Let ra, rb, rc, rr be the radii of respective incircles of the triangles AC'B', BA'C', CB'A', A'B'C'.[29]

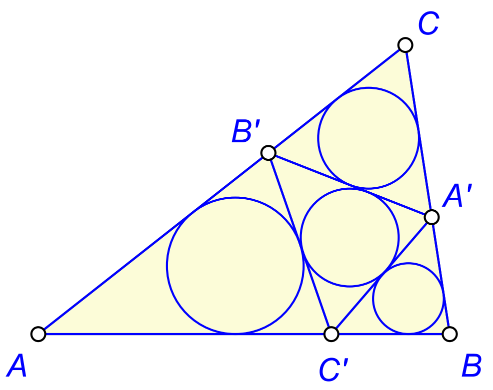


Figure 40

The **Observe formulae** command detects the relationship

$$4 \cdot ra \cdot rb \cdot rc = (ra + rb + rc + rr) \cdot rr^2$$

Can you prove this relation?

Example 3

Given a triangle ABC, let rA, rB, rC be the radii of the mutually externally tangent circles with centres A, B, C. Furthermore, let rI and rO be the radii of the circles externally and internally tangent to the three mentioned circles (i.e. the inner and outer Soddy circles). (Figure 41)[30]

[29] OKExamples\OKG_Plus\Formula_03.p

[30] OKExamples\OKG_Plus\Formula_04.p

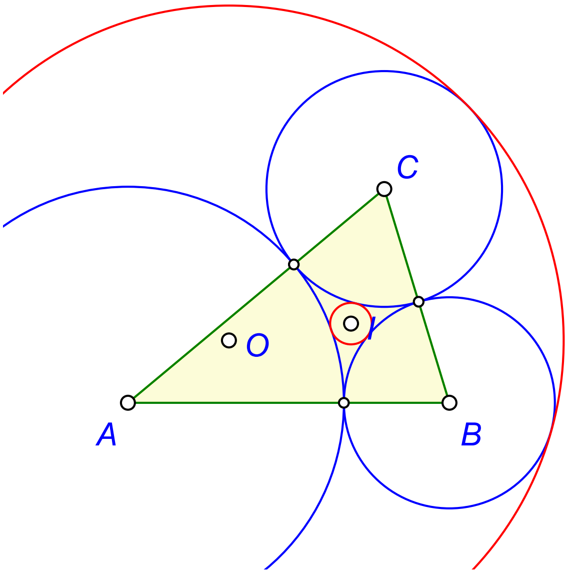


Figure 41

Observe formulae suggests nice relationships between variables

- rA, rB, rC, rI
- $rA, rB,\ rC, rO$
- $rA,\ rB, rC, rI, rO$

Note. Checkmark the **Transform variables** option.

Example 4

Let H be the orthocentre of the acute triangle ABC. Let ra, rb, rc be the radii of incircles of the triangles HBC, HCA, and HAB respectivery. Furthermore, let ri and r be the radii of the incircle and the circumcircle of the triangle ABC. (See Figure 42.)[31]

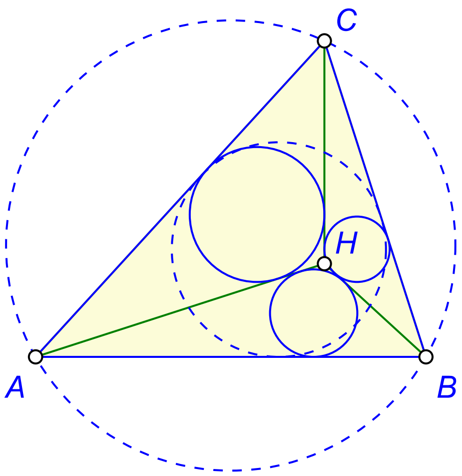


Figure 42

[31] OKExamples\OKG_Plus\Formula_05.p

Observe formulae suggest this relationship between the 5 radii:

$$4 * (ra^2 + rb^2 + rc^2 + r^2) =$$
$$= (ra + rb + rc + ri - 2r)^2 + 8 \cdot r \cdot ri$$

Does the relation also hold for obtuse triangles?

4.3 Observe formulae – advanced

The **advanced** variant of **Observe formulae** is similar to the **simple** one, except that you can also directly specify the involved geometric quantities of the dynamic construction. Thus, the geometric quantities to be considered in a relation do not have to be measured beforehand in the Sketch Editor.

In this section we will show how to use of the **advanced** variant of the command.

The first entry of the form that appears (Figure 44) is lightly coloured to emphasise its special role:

- In the first entry (t) you can write a geometric quantity or a ratio of two quantities (of the same dimension). Note that the remaining entries can only contain geometric quantities (or are empty).
- If the first entry is not empty, OK Geometry searches only for formulae that contain this entry.

We explain the **advanced** variant of the command with a simple example:

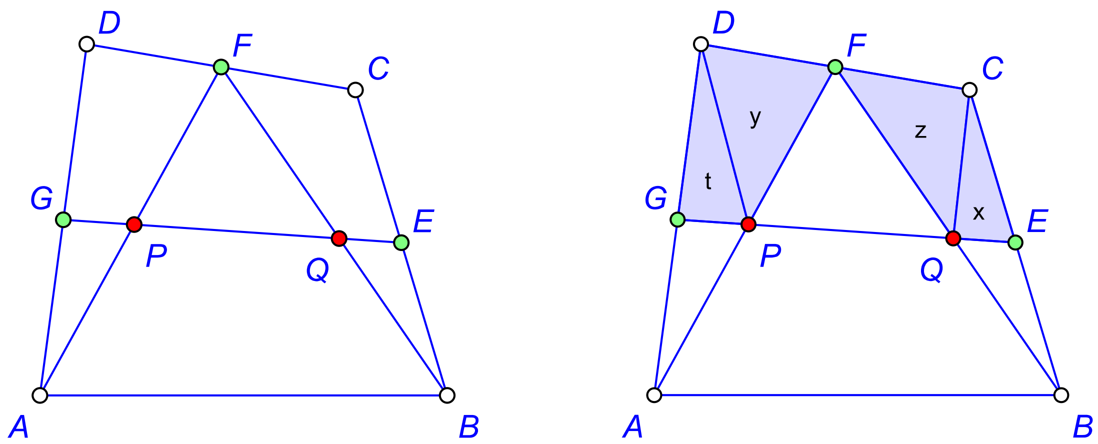


Figure 43

In a convex quadrilateral ABCD let E, F, and G be the midpoints of sides BC, CD, and DA. Furthermore, let P and Q be the intersection of line GE with the lines AF and BF (Figure 43, left).

How are related the lengths of segments AP, PF, BQ, and BF? (See Figure 43, left.)[32]

What is the relationship between the areas of triangles labelled as t, x, y, z? (See Figure 43, right.)[33]

Proceed as follows:

1. Provide a dynamic construction of the object(s) under consideration. The construction can be imported from dynamic geometry software or created in the OK Geometry Sketch Editor.
2. Apply the command *Command|Observe formulae|Advanced*. A form appears. In the entries t, x, ... write the quantities to be considered. In our case we write d(AP) for the length of the segment AP and similarly for the other three segments. Then click on the **Observe** button (Figure 44, left).
3. The polynomial relation obtained usually requires additional algebraic manipulation, depending on your aesthetic standards and mathematical goals. The found relation can be rewritten as

$$\frac{|AP|}{|PF|} + \frac{|BQ|}{|QF|} = 2$$

   Remember that we are dealing with a result of an observation. It is a nice exercise in geometry to actually prove the correctness of the formula.
4. To add a further dynamic numerical test for the obtained relation proceed as follows:
   a. Insert the considered relation into a block. The expression must be in **compact** or **extended notation**.
   b. Right click on the relation and select the command *Export expression/equation|as parameter in construction.*
   c. A parameter *Fcondx* is added to the dynamic construction. The parameter numerically checks the correctness of the equation in the block. You can dynamically modify the construction and observe if *Fcondx* has the value 'True'. In our case, we can dynamically check if convexity of the quadrilateral ABCD is a necessary condition for the relation to be true. Note that you can as well declare as a parameter only a part of the relation by putting only that part of the relation in a block.
5. The problem with areas is handled in a similar way. In order to consider the area of triangle PGD we fill the entry t as Area(PGD). Note that the result of the observation is written differently because we have chosen the notation option 'yxz'.

Variables: t=Area(PGD) , x=Area(CEQ) , y=Area(PFD) , z=Area(CFQ)
1*t*z + x*y - y*z = 0

[32] OKExamples\OKG_Plus\Formula_06.p

[33] OKExamples\OKG_Plus\Formula_07.p

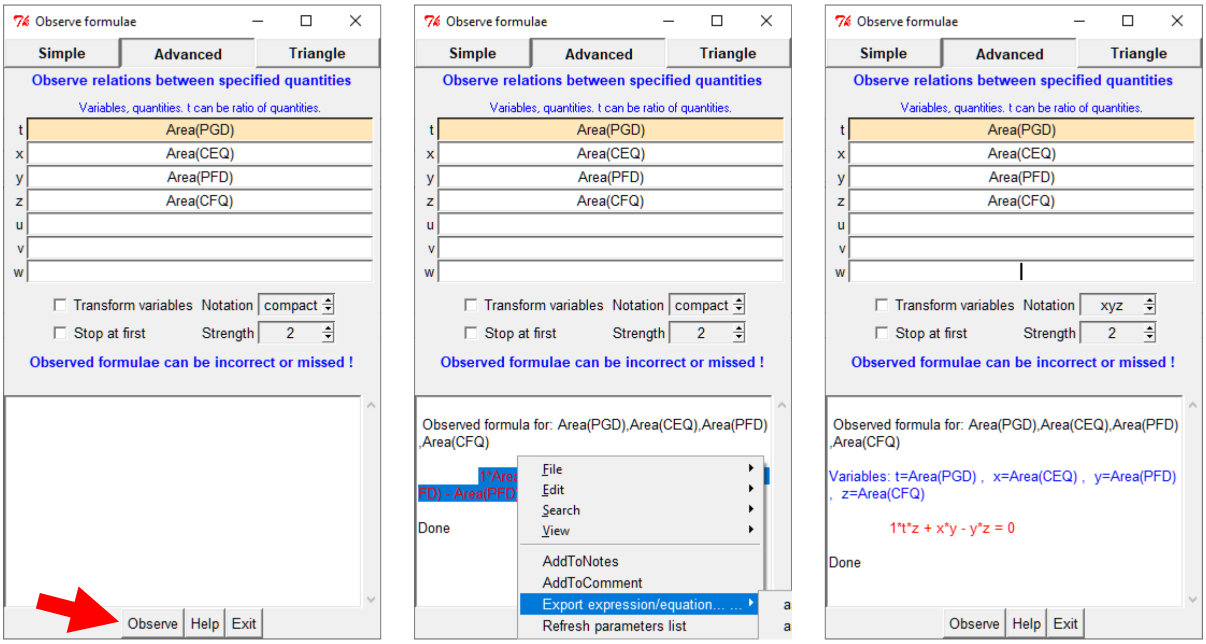


Figure 44

4.3.1 Comments on observed formulae

Many problems can be solved with the above scheme, i.e. by considering the distance between the points P and Q as d(PQ) and the area of the triangle PQR as Area(PQR). However, the **advanced** variant can handle more complex situations. Here are the conventions to be followed in describing the geometric quantities in the entries t, x, y, ...:

- The entries can also be quantities measured in the Sketch Editor (as parameters) with the measuring commands (Distance to object, Length, Perimeter, Area). The parameters from the Sketch Editor can be written in brackets, e.g., [*par*], or without brackets, e.g., *par*.
- The entries can also be parameters obtained by computation in the Sketch Editor. In this case **if the parameter is 1-dimensional, its name should start with a lowercase letter, and if the parameter is 2-dimensional quantity, its name should start with an uppercase letter.**
- The first entry (t) may contain a geometric quantity, e.g. d(AB), or **a ratio of two quantities** of the same dimension, e.g. d(AB):d(CD) or Area(ABP):Area(CAP).
- In addition to d(PQ) and Area(PQR) you can use a whole range of geometric quantities that are specified within the entry. The list of all allowed geometric quantities is in Section 4.3.2.
- It is possible to fill an entry with more than one quantity, separated by a space. In this case, each of the space-separated quantity is considered separately.
- The options of the command are the same as described in Section 4.2.2. The treatment of the results is the same as described in Section 4.2.3.

4.3.2 Symbolising geometric quantities

Two symbols are used for distance:

d(AB)	distance between the points A and B
o(ABC)	oriented distance from the point A to the oriented line BC

The quantities listed below refer to triangles. **If the triangle argument is missing, the quantity refers to the (assumed) reference triangle**. For example:

Area(PQR) .... denotes the area of the triangle PQR,

Area .... denotes the area of the reference triangle (if it exists),

a(PQR) .... denoted the length of the side QR in triangle PQR,

a .... denotes the length of the side *a* of the reference triangle (if it exists).

a(ABC) b(ABC) c(ABC)	length of the side BC length of the side CA length of the side AB
s(ABC)	half-perimeter of the triangle ABC
r(ABC)	radius of the circumcircle of the triangle ABC
ri(ABC)	radius of the incircle of the triangle ABC
Area(ABC)	Area of the triangle ABC
S(ABC)	2·Area(ABC)
SA(ABC) SB(ABC) SC(ABC) SW(ABC)	Conway parameters for triangle ABC

The following quantities also refer to triangles. Each quantity in the list consists of three cases related to the three angles of the triangle argument. We list only the cases related to the first angle of the triangle. If the triangle argument is missing the quantity refers to the reference triangle (if it exists).

For example:

cA(PQR) = r·cos(P),
cB(PQR) = r·cos(Q),
cC(PQR) = r·cos(R), where r is the circumradius of PQR.

<table>
<tr><td>cA(ABC)</td><td>r·cos(A)</td><td rowspan="3">r .... circuradius of the triangle ABC
A,B,C .... angles of the triangle ABC
W .... Brocard angle of the triangle ABC</td></tr>
<tr><td>tA(ABC)</td><td>r·tan(A)</td></tr>
<tr><td>cA2(ABC)</td><td>r·cos(A/2)</td></tr>
</table>

sA2(ABC)	r·sin(A/2)	
tA2(ABC)	r·tan(A/2)	
cA3(ABC)	r·cos(A/3)	
sA3(ABC)	r·sin(A/3)	
tA3(ABC)	r·tan(A/3)	
cW(ABC) sW(ABC) tW(ABC)	r·cos(W) r·sin(W) r·tan(W)	

4.3.3 More examples

Example 5

Let G be the centroid of the triangle ABC. Let A’ be the intersection, other than A, of the line AG with the circumcircle of ABC. Define B’ and C’ cyclically (Figure 45)[34].

Consider the following areas:

> t = Area(ABC)
> x = Area(A’BC)
> y = Area(AB’C)
> z = Area(ABC’)

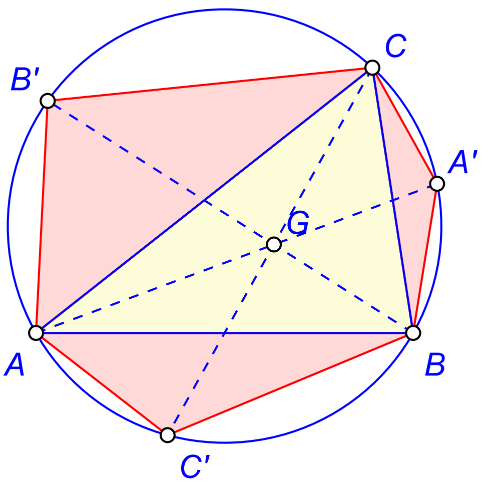


Figure 45

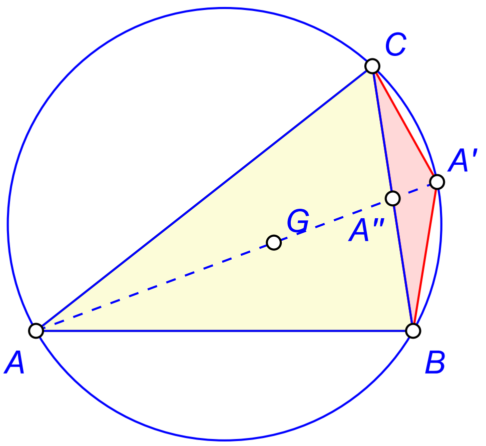
[34] OKExamples\OKG_Plus\Formula_08.p

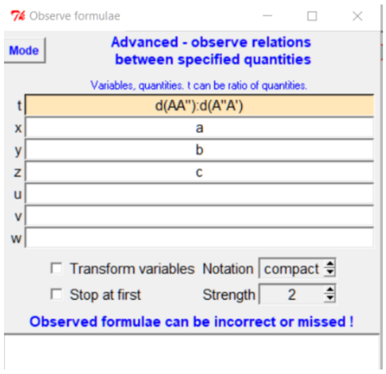
The advanced variant of **Observe formulae** command suggests that the following (reordered) relation holds:

$$\left(\frac{x}{t}+\frac{y}{t}+\frac{z}{t}\right)+3\cdot\frac{x}{t}\cdot\frac{y}{t}\cdot\frac{z}{t}\cdot\left(\frac{t}{x}+\frac{t}{y}+\frac{t}{z}+9\right)=1$$

Example 6

To shed more light on the configuration in Figure 45, we consider the ratio |AA’’| : |A’’A’| in the configuration on Figure 46 (left). We try to express this ratio in terms of the sides of the triangle ABC. Note that if ABC is declared as a reference triangle we can use the symbols *a, b, c* for the length of the sides of the triangle ABC (Figure 46, right)[35].





$$\frac{d(AA'')}{d(A''A')}=\frac{-a^2+2b^2+2c^2}{a^2}$$

Figure 46

4.4 Observing formulae - triangle

The **triangle** variant of **Observe formulae** is a heuristic tool that suggests how to express a geometric quantity or a ratio of two geometric quantities in terms of some ‘commonly used triangle quantities’ such as length of its sides. This variant is quite easy to use, unfortunately it is sometimes time consuming. We explain the use of the triangle variant with a trivial example.

Example 7

Given a triangle ABC, let A’ be the centre of its A-excircle, i.e. A’ is the centre of the circle that lays outside the triangle ABC and touches the extension of the side AB, the extension of the side AC, and

[35] OKExamples\OKG_Plus\Formula_09.p

the side BC. Furthermore, let A'' be the point of contact of the A-excircle with the side BC (Figure 47).[36]

We want to express somehow the radius of the A-excircle, i.e. |A'A''|, as well as the ratio |BA''|:|A''C|.

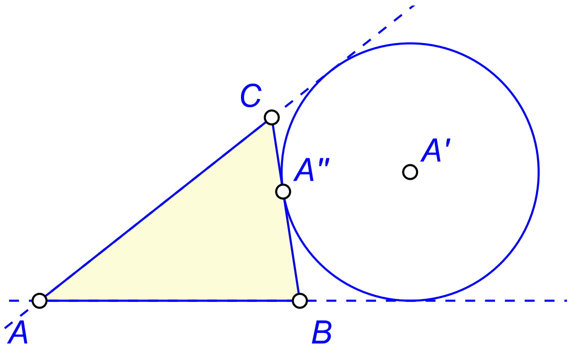


Figure 47

We proceed as follows:

- Provide a dynamic construction of the object(s) under consideration. The construction can be imported from dynamic geometry system or created in the OK Geometry Sketch Editor. **Declare the triangle ABC as the reference triangle.**
- Activate the command *Command|Observe formulae|Triangle*. A form will appear. In the first entry write the considered (Figure 48, left) quantity or the ratio of two quantities (Figure 48, centre): in our case, d(A'A'') for the first observation, and d(BA''):d(A''C) for the second observation. Then click on the **Observe** button.

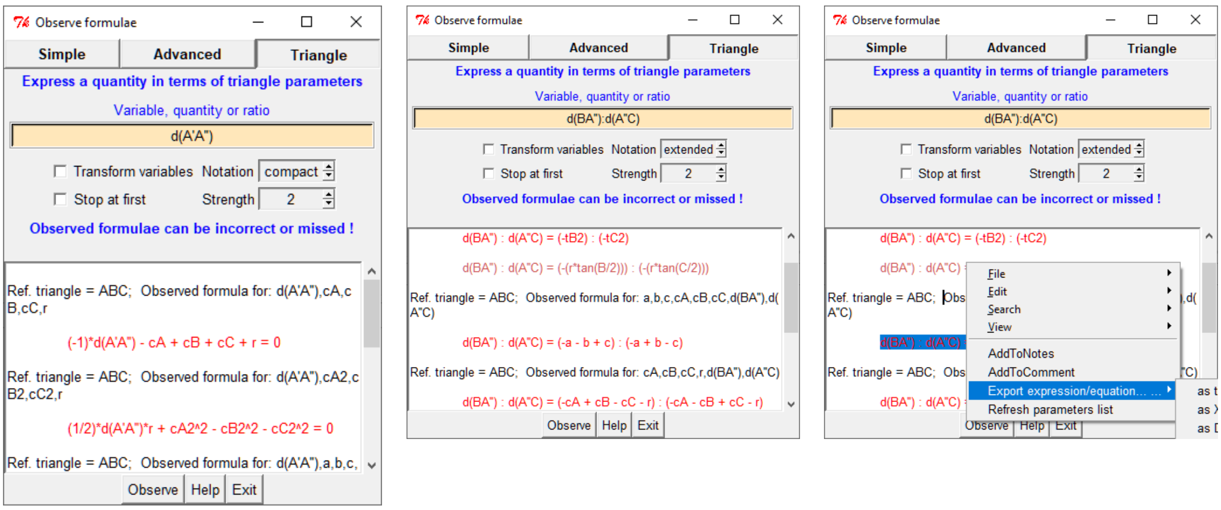


Figure 48

[36] OKExamples\OKG_Plus\Formula_10.p

- OK Geometry looks for various ways of expressing the considered quantity or ratio in terms common triangle elements. Some of the suggestions require an interpretation as explained in Section 4.3.2. The highlighted part of the results in Figure 48 (left) and Figure 48 (right), for example, can be rewritten as

$$\frac{d(A'A'')}{2} - r \cdot \cos\left(\frac{A}{2}\right)^2 + r \cdot \cos\left(\frac{B}{2}\right)^2 + r \cdot \cos\left(\frac{C}{2}\right)^2 = 0$$

$$d(A'A'') \cdot ri = (a - b + c)(a + b - c)$$

$$\frac{d(BA'')}{d(A''C)} = \frac{\tan\frac{B}{2}}{\tan\frac{C}{2}}$$

$$\frac{d(BA'')}{d(A''C)} = \frac{a + b - c}{a - b + c}$$

$$\frac{d(BA'')}{d(A''C)} = \frac{\cos A - \cos B + \cos C + 1}{\cos A + \cos B - \cos C + 1}$$

  In the above expressions, according to Section 4.3.2, *r* is the radius of the circumcircle of ABC, *ri* is the radius of the inscribed circle, *a, b, c* are the side lengths, and *A, B, C* are the angles of the triangle ABC.
  We emphasize again that the **Observe formulae** command does not provide a proof of the detected relations. In the above case one can easily provide geometric proofs for the proposed relations.
  In general, the list of results can contain several items or can be empty.

- To add a further dynamic numerical test for the obtained relation proceed as follows:
  a. Insert the considered relation into a block. The expression must be in **compact** or **extended notation**.
  b. Right click on the relation and select the command *Export expression/equation|as parameter in construction.*
  c. A parameter *Fcondx* is added to the dynamic construction. The parameter numerically checks the correctness of the equation in the block. You can modify the construction dynamically and observe if *Fcondx* has the value 'True'. Note that you can declare only a part of the relation as parameter by putting in block only this part of the relation.

4.4.1 Comments on observed formulae

- **Observed algebraic relations are a result of numerical observations of the dynamic construction. The found relations are not proved and there is no guarantee they are correct. A relation can also be missed. In general, a missed relation occurs more likely than a found relation that does not hold at least locally.**
- The considered entry can be also a measured or calculated parameter of the dynamic construction.
- The entry can be a parameter that was computed in the Sketch Editor (e.g., a product of two distances). **If the parameter is 1-dimensional, its name should begin with a lowercase**

letter. If the parameter is 2-dimensional quantity, its name should begin with an uppercase letter.

- The considered entry can contain a geometric quantity, e.g. d(AB), or **a ratio of two quantities** of the same dimension, e.g. d(AB):d(CD) or Area(ABP):Area(CAP).
- Besides d(PQ) and Area(PQR) you can use a whole range of geometric quantities. The list of all admissible geometric quantities is in Section 4.4.2.
- The options of the command are the same as described in Section 4.2.2. The treatment of the results is the same as described in Section 4.2.3.
  The triangle variant of **Observing formulae** is time consuming. Increasing the **strength** and allowing **transformation of variables** significantly increases the computational time. You may consider in such cases to checkmark the **Stop at first** option.

4.4.2 Geometric quantities of the reference triangle

Two symbols are used for distance:

d(AB)	distance between the points A and B
o(ABC)	oriented distance from the point A to the oriented line BC

The results of the **triangle** variant of **Observe formulae** expressed in terms of certain parameters of the reference triangle. Below is the list of the used parameters. Note that the list is essentially the same as the list in Section 4.3.2. The list is self-explanatory. For example, if PQR if the reference triangle, then cA means *r*·cos(P), where *r* is the circumradius of the triangle PQR.

<table>
<tr><td>a, b, c</td><td colspan="2">length of the sides of the reference triangle</td></tr>
<tr><td>s</td><td colspan="2">half-perimeter of the reference triangle</td></tr>
<tr><td>r</td><td colspan="2">radius of the circumcircle of the reference triangle</td></tr>
<tr><td>ri</td><td colspan="2">radius of the incircle of the reference triangle</td></tr>
<tr><td>Area</td><td colspan="2">area of the reference triangle</td></tr>
<tr><td>S</td><td colspan="2">twice the area of the reference triangle</td></tr>
<tr><td>SA, SB, SC, SW</td><td colspan="2">Conway parameters for the reference triangle</td></tr>
<tr><td>xt(P),yt(P),zt(P)</td><td colspan="2">trilinears of point P</td></tr>
<tr><td>Xb(P),Yb(P),Zb(P)</td><td colspan="2">barycentrics of point P</td></tr>
<tr><td>a, b, c</td><td>2r·sin(A), 2r·sin(B), 2r·sin(C)
(no need for sA, sB, sC)</td><td rowspan="7">r .... circumradius of the reference triangle
A,B,C .... angles of the reference triangle
W .... Brocard angle of the reference triangle</td></tr>
<tr><td>cA, cB, cC</td><td>r·cos(A), r·cos(B), r·cos(C)</td></tr>
<tr><td>tA, tB, tC</td><td>r·tan(A), r·tan(B), r·tan(C)</td></tr>
<tr><td>cA2, cB2, cC2</td><td>r·cos(A/2), r·cos(B/2), r·cos(C/2)</td></tr>
<tr><td>sA2, sB2, sC2</td><td>r·sin(A/2), r·sin(B/2), r·sin(C/2)</td></tr>
<tr><td>tA2, tB2, tC2</td><td>r·tan(A/2), r·tan(B/2), r·tan(C/2)</td></tr>
<tr><td>cA3, cB3, cC3</td><td>r·cos(A/3), r·cos(B/3), r·cos(C/3)</td></tr>
</table>

sA3, sB3, sC3	r·sin(A/3), r·sin(B/3), r·sin(C/3)	
tA3, tB3. tC3	r·tan(A/3), r·tan(B/3), r·tan(C/3)	
cW, sW, tW	r·cos(W), r·sin(W), r·tan(W)	

4.4.3 More examples

Example 8

Given is a triangle ABC. Let A', B', C' be the points of contact of the incircle of ABC with the sides of ABC (Figure 49). The lines AA', BB', CC' concur in a point P (the Gergonne point).[37]

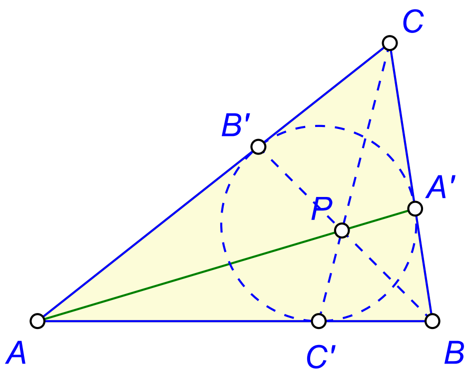


Figure 49

The **triangle** variant of **Observe formulae** suggest several expressions for |AP|:|PA'|as well as |AP| or |AA'|. (The more the strength the more relations are detected.) Here are just some of them:

$$\frac{|AP|}{|PA'|}=\frac{\tan\frac{B}{2}+\tan\frac{C}{2}}{\tan\frac{A}{2}}=\frac{2a(-a+b+c)}{(a-b+c)(a+b-c)}.$$

Example 9

Given is an **acute** triangle ABC. Consider the circles kA, kB, kC that have as diameters the sides BC, CA, AB of the triangle ABC. What is the radius *rr* of the circle k that contains these circles and is tangent to all of them (Figure 50, left).[38]

[37] OKExamples\OKG_Plus\Formula_12.p

[38] OKExamples\OKG_Plus\Formula_13.p

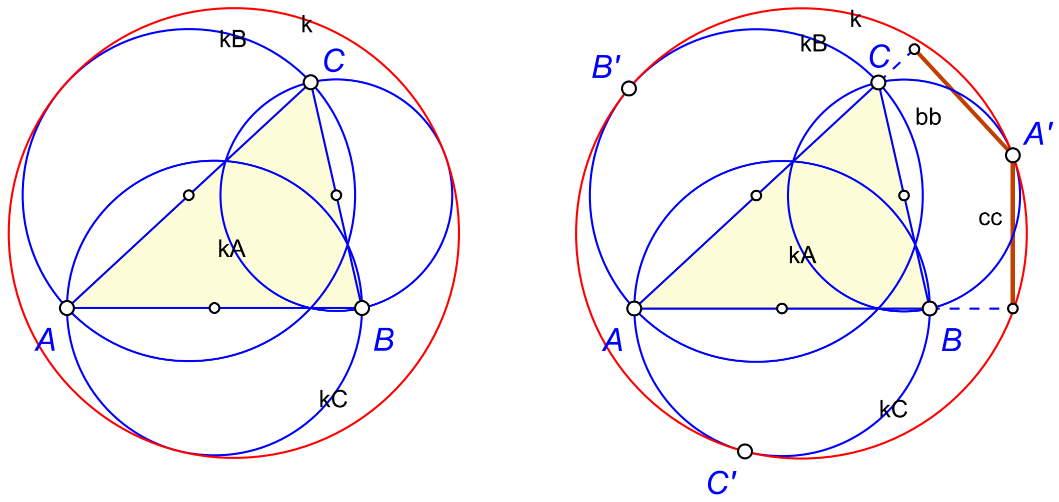


Figure 50

Observe formulae for triangles at strength 4 suggests that

$$rr = \frac{S \cdot (2r + ri) + 2 \cdot ri \cdot \big(SW + ri \cdot (4r + ri)\big)}{2 \cdot (S + ri \cdot (4r + ri))}$$

Continuing the previous construction, let A', B', C' be the points of contact of the circles kA, kB, kC and the circle k. Furthermore, let *bb* and *cc* be the distance from A' to the orthogonal projections of A' onto AC and onto AB (Figure 50, right).[39]

Observe formulae suggests that

$$\frac{bb}{cc} = \frac{1 + \sin C}{1 + \sin B}$$

Under this assumption we deduce that AA', BB',and CC' meet at a common point with trilinear coordinates

$$\frac{1}{1 + \sin A} : \frac{1}{1 + \sin B} : \frac{1}{1 + \sin C}$$

The point of concurence of AA', BB', CC' is the Paasche point (X1123) of triangle ABC.

[39] OKExamples\OKG_Plus\Formula_14.p

5 Observing proofs

5.1 Introduction

OK Geometry is primarily intended for the observation of facts in plane geometry and the discovery of conjectures, which may be interesting in themselves or serve to prove something. But since proofs are the essence of mathematics, we have equipped OK Geometry with a mechanism for creating (less sophisticated) proofs of observed facts. The proving mechanism is based on the Geometry Deductive Database method[40] of Chou, Gao and Zhang, which is why we refer to it as the GDD prover or as prover.

Strategies for observing dynamic constructions are different than strategies for deductive proving. An observation tool requires commands that create, in some way (not necessarily with Euclidean construction steps), multiple instances of precise configurations. The GDD proving method, on the other hand, requires construction steps with clearly stated assumed and derived properties that can be used in the proof of a fact. We hereby refer to such construction steps as **proof-compliant operations (commands)** (see Section 5.3.5)**.** For example, the midpoint of a segment or the incentre of a given triangle are proof-compliant operations (and thus acceptable for proving a property of a construction), since ‘everybody’ knows and understands how the midpoint/incentre is related to the endpoints of the segment/vertices of the triangle. On the other hand, creating a circle that touches three given circles or creating an implicitly defined object are not proof-compliant and can be an obstacle in the GDD proving process.

The GDD proving method requires that the facts being proved refer to a construction created using proof-compliant operations (commands). Fortunately, most OK Geometry construction commands are proof-compliant. However, as we will see, OK Geometry includes some mechanisms that may allow a fact about a construction to be proved even if there are some non-compliant steps (see Example 4 in Section 5.2, Section 5.3.5).

There are three possible outcomes of GDD-proving process:

1. The prover provides a proof.
2. The prover observes that the claim is false and refuses to prove it (Figure 51, left).
3. The prover observes that the claim is true, but is not able to prove it (Figure 51, center) . In this case, the prover offers the option of proving by adding a point (see Section 5.5). If we choose this option, which may take some time, the proving process ends either with a successful result or a message of failure (Figure 51, right).

[40] S.C. Chou, X.S. Gao, and J.Z. Zhang, A Deductive Database Approach To Automated Geometry Theorem Proving and Discovering, *Journal of Automated Reasoning*, 25(3), 219-246, 2000.

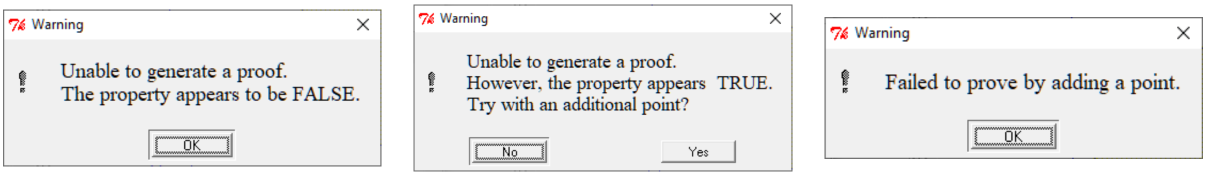


Figure 51

We now illustrate the process of proving with OK Geometry on this very simple task (Figure 52)[41]:

Example 1

Let H be the orthocentre of a given triangle ABC. Let B’ be the centre of the circle through C, A, H, and let C’ be the centre of the circle through B, A, H. Prove that the line B’C’ is parallel to the line BC.

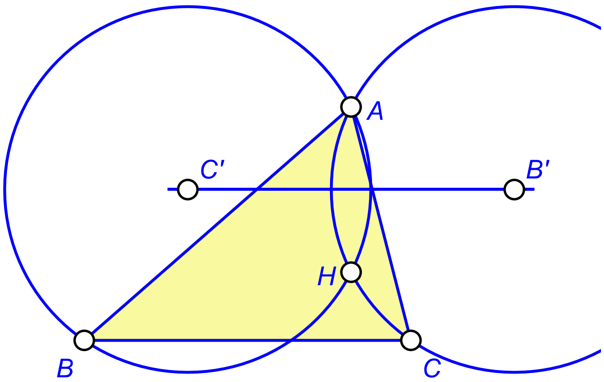


Figure 52

The first step in the proof is to create the described configuration with proof-compliant construction commands:

- Create a triangle ΔABC.

  You can use the command *Special|A triangle* or you can draw the points A, B, C and connect them with a polyline or separate segments.

- Create the orthocentre H of ΔABC.

  This can be done with the command *Point|Triangle 4 centres|Orthocentre* or as the intersections of the perpendiculars from two vertices to the opposite sides of the triangle.

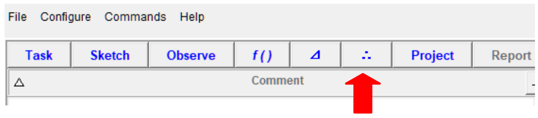
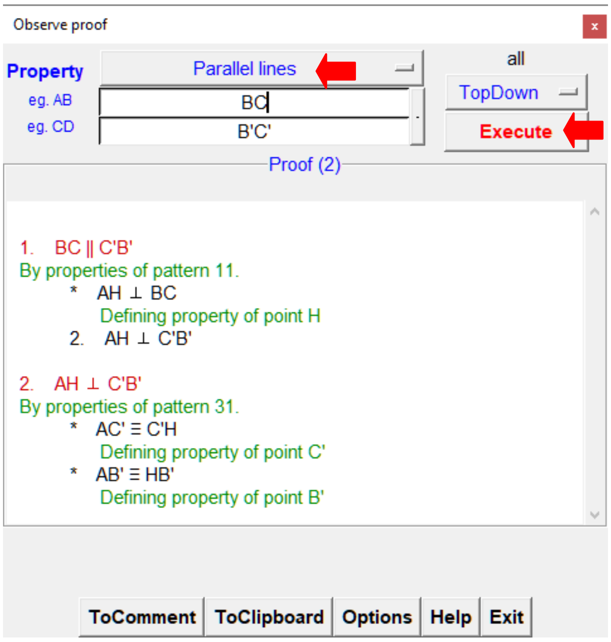
- Create the centres C’, B’ of the circles through A, B, H and through A, C, H.

[41] OKExamples\OKG_Plus\Proof_01.p

You can do this directly with the command *Point|Triangle 4 centres|Circumcentre* or by creating circles with *Circle|Circle 3 objects* and their centres with *Point|Centre of circle.*

- Create the line segment B'C'. Create the two circles through A,H,B and A,H,C.

  We have now constructed the essential points of our task. Note that the line B'C' and the two circles shown do not play a role in the proof.

Now with the button (or with the menu ***Commands|Observe proof***) activate the *Observe proof* form**.**	
Enter the required data in the form: • Select the ***Parallel lines*** property**.** • Fill in the names of the lines BC and B'C'. (**Note**. Instead of writing "B" you can also click on the point B in the figure.) Leave the other data unchanged. Then click on ***Execute***. After a while, a proof will appear.	

By default, the proof is top-down (i.e. backwards from the final claim to the hypotheses). The red assertions are immediately followed by the supporting arguments (in black). The arguments that require further elaboration are preceded by numbers. **A right click on such a line moves the cursor to the proof of that property**. The steps for which there is an obvious explanation are marked with an asterisk. Explanatory comments are in green.

Most lines of the proof are commented and can be visualised. For example, a click on the line immediately after step 2 (Figure 53) explains the deduction in step 2. A short explanation appears below the *Proof* section and on the construction pane. Additional information is available in the Help section (you can activate or deactivate this section by pressing **F8** key).

Further technical details of the **Observe proof** form will not be discussed here (see Section 5.3). Note only the explanation of why, for example, AH is perpendicular to BC (AH ⊥ BC), namely 'Defining property of point H'. It means that the property (AH ⊥ BC) is true because of the operation used in constructing the point H (orthocentre of ΔABC). Also, AH ⊥ B’C’ because AC’HB’ is a deltoid (‘pattern 31’) and it is a well known property of the deltoid that its diagonals are mutually perpendicular.

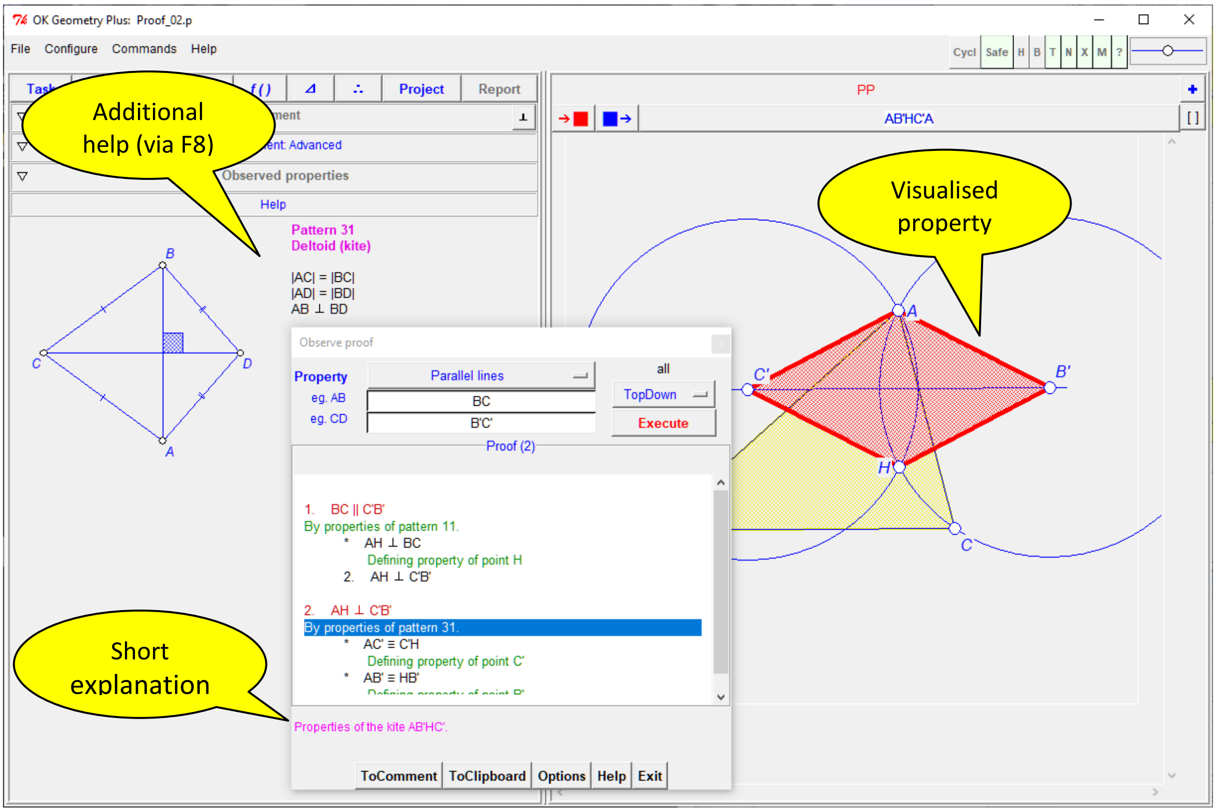


Figure 53

In this simple case one can directly transform the proof to a paragraph form: B’A and B’H are the radii of the same circle and so are C’A and C’H. Thus AC’HB’ is a kite (deltoid) and it is well known that the diagonal lines of any kite are perpendicular. The altitude line AH is thus perpendicular to B’C’. By definition, the altitude line AH is also perpendicular to the baseline BC. Thus the lines BC and B’C’ are parallel.

5.2 Examples

In this section, we present three examples of proving a proposition to illustrate some features of the proof procedure. The details of these features will be described in the sections that follow.

Example 2[42]

Given is a triangle ABC. Let O be the midpoint of A and the projection D of A onto BC. Furthermore, let E and F be the intersections (other than A) of the circle *k*(O,D) with the lines AB and AC (Figure 54, left). Prove that the points B, C, E and F are concyclic.

[42] OKExamples\OKG_Plus\Proof_02.p

The GDD proof requires a construction, which can be achieved using standard construction commands. For example, the triangle ABC can be created with the command *Special|A triangle* or by placing three points A, B, C and connecting them with segment or lines, or a polyline. Point D can be constructed with the command *Line|Perpendicular connector,* or as the intersection of line BC and the perpendicular to BC through A. Point O is obtained with the *Point|Midpoint* command, and E and F are the usual intersection points.

To generate a proof that B, C, E and F are concyclic, press the ∴ button, select the property *Cocircular points,* and enter the points B, C, E, and F. Note that you can click on these points in the figure instead of typing them. Then click the **Execute** button, a proof will appear.

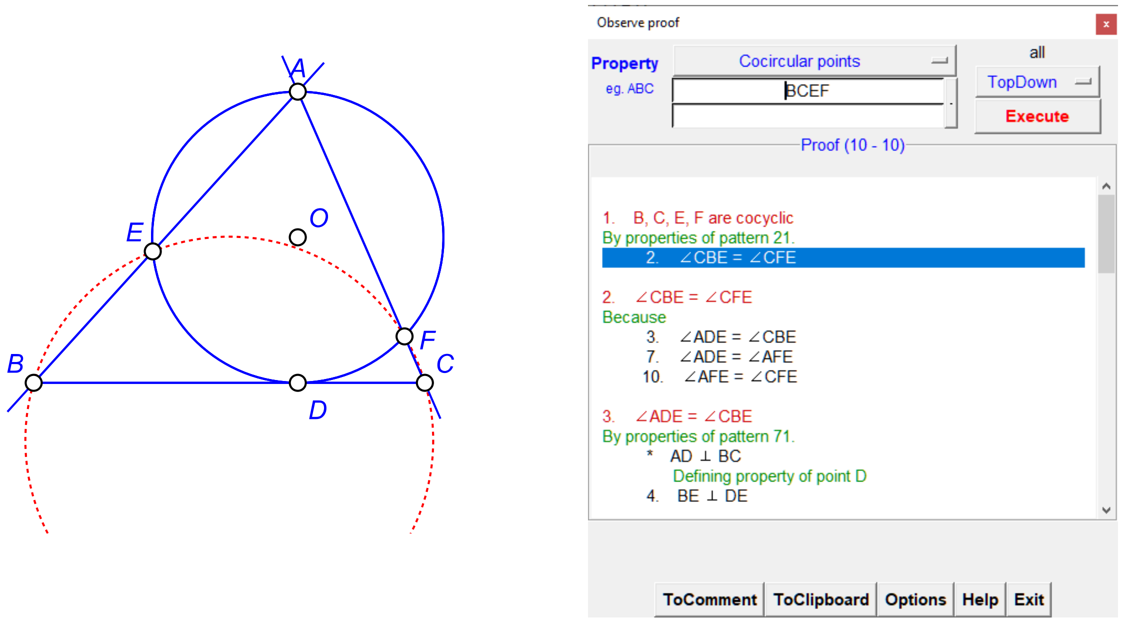


Figure 54

- The initial proof will (probably) consist of 19 steps. By repeating the **Execute** command, you may sometimes obtain a shorter proof with fewer steps (see Section 5.3.4). In our case, you can probably find a proof in 10 steps.
- Interpreting the proof can sometimes be tedious, despite the comments and illustrative figures. However, grasping the main ideas of the generated proof is often easy. In our case: B, C, E, F are concyclic because ∠CBE = ∠AFE (i.e., ∠CBE and ∠EFC are supplementary). And this is true because both, ∠CBE and ∠AFE, are congruent to ∠ADE. The proof often contains technical details that are easier to work out independently than to interpret directly from the presented proof.

Example 3

Given is a triangle ΔABC with incentre I. Let A' be the orthogonal projection of B onto AI. Let C' be the orthogonal projection of B onto CI. If D is the midpoint of BC, prove that A', C' and D are collinear.

The configuration is ealy to construct using standard commands[43]. Note that the command *Point|Triangle 4 centres|Incentre* or *Special|Triangle centres|Incentre (X1)* is proof-compliant, so is acceptable to use them for creating the incentre I of ABC.

To generate a proof that A’, C’, and D are collinear, press the ∴ button, select the property *Collinear points,* and enter the points A', C', and D. Note that you can click on these points in the figure instead of typing them. Then click the **Execute** button. Instead of the proof, a message appears (Figure 51, center). If you accept the suggestion to allow OK Geometry to attempt a proof by adding a point to the construction, you will receive a proof and a note that a proof has been found by adding a point. In our case, the midpoint of A and B has been added to the construction (Figure 55). Note that you may get a different proof and a different added point.

See Section 5.5 for more information about proofs obtained by adding points.

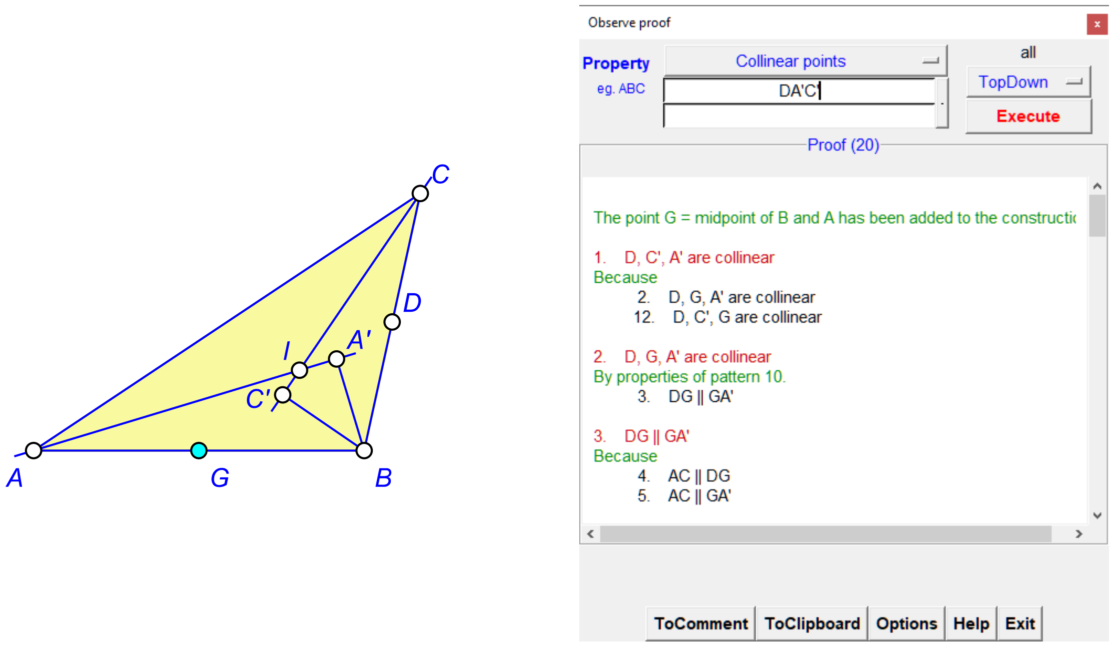


Figure 55

The next example illustrates what happens when the configuration used for a GDD proof contains proof-noncompliant operations.

Example 4[44]

Given is a right triangle ABC with BC as hypotenuse. Let F be the projection of the incentre D of ABC onto AB and let G be the projection of the C-excentre E of ABC onto AB (Figure 57). Prove that the segment EG is congruent to the segment BF.

---

[43] OKExamples\OKG_Plus\Proof_03.p
[44] OKExamples\OKG_Plus\Proof_04.p

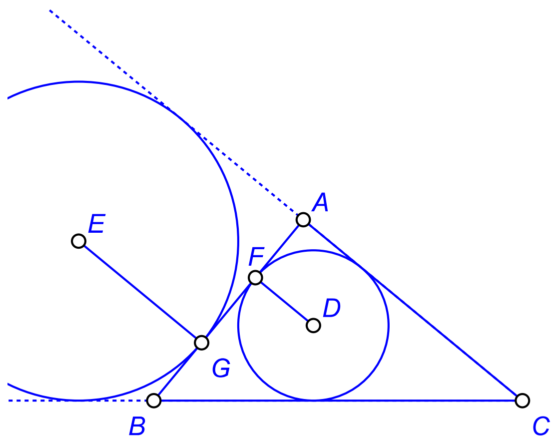


Figure 56

Solution 1. The above construction can be carried out using only proof-compliant commands. We can create a right triangle with *Advanced|Shapes|Right triangle*, or by other means, the incentre D with the command *Point|Triangle 4 centres|Incentre*, and obtain E as the *intersection* of the external angle bisectors at A and C using *Line|Angle bisector – Alt*. The points F and G are orthogonal projections onto a line, so we use *Line|Perpendicular connector*. All the mentioned commands are proof-compliant, so we can obtain a proof as described in the previous examples.

Solution 2. For illustrative purposes, we create some objects using proof-noncompliant operations. We create the incircle and the C-excircle of ABC with the command *Circle|Circle 3 objects*, as circles touching the three sidelines of ABC. The command Circle 3 objects is proof-compliant only when the three arguments are points, which is not the case here. We then create the centres D and E of the created circles. The projected points F and G are created with the command *Line|Perpendicular connector* (perpendicular lines could also be used).

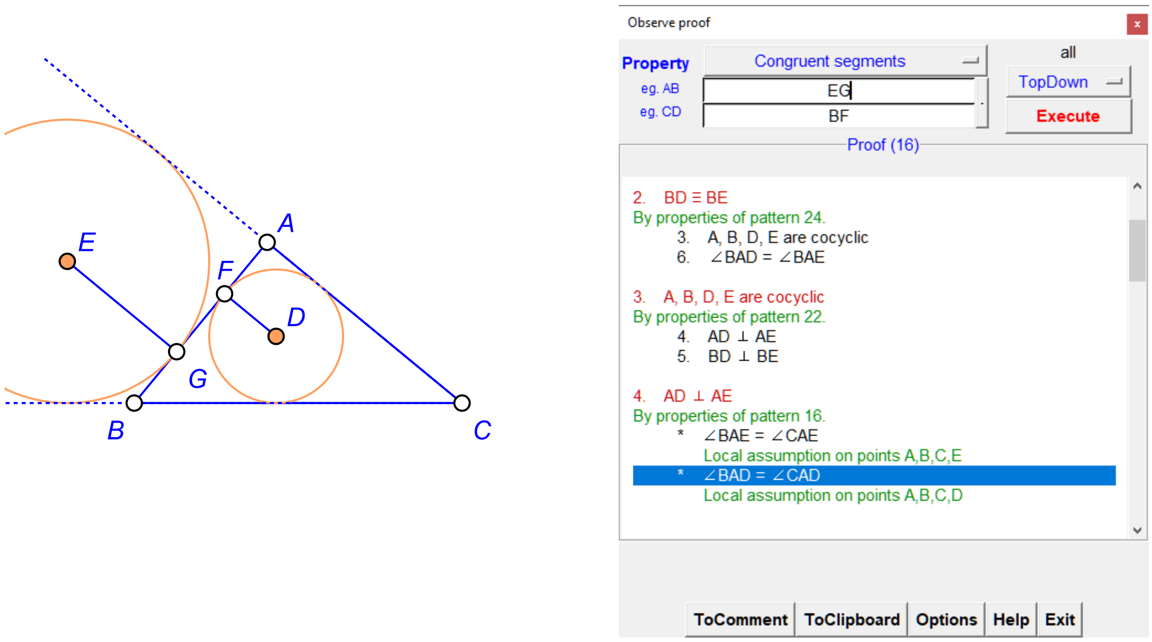


Figure 57

To generate a proof that EG is congruent to BG, press the  button, select the property *Congruent segments,* and enter the points E, G, B, and F. Note that you can click on these points in the figure instead of typing them. Then click the **Execute** button. A proof appears (Figure 57, right).

At the very beginning of the proving process, objects created with noncompliant commands turn light red. For such objects, the prover does not have information about their properties, which could eventually be used in the proof. In these cases, the prover resorts to observation and uses the observed properties as local assumptions. To obtain a valid proof, the user must prove these assumptions. In the presented example, the prover created a proof under the assumptions that the lines BD and BE are angle bisectors at B, and the lines AD and AE are angle bisectors at A. In this case, the local assumptions are easy to prove, but this is not always the case.

Solution 3. We start from the construction in Solution 2. Since noncompliant operations have been used, the prover lacks essential properties of points D and E. We can provide this information separately in the comment to the construction. In our case we declare that ∠BAD = ∠DAC, ∠CBD = ∠DBA, ∠BAE = ∠EAC, and ∠CBE = ∠EBA – note that in such equalities we do not distinguish between an angle and its supplementary angle (Figure 58, left). The prover considers the provided properties, if confirmed by observation, as (declared) facts (Figure 58, right).

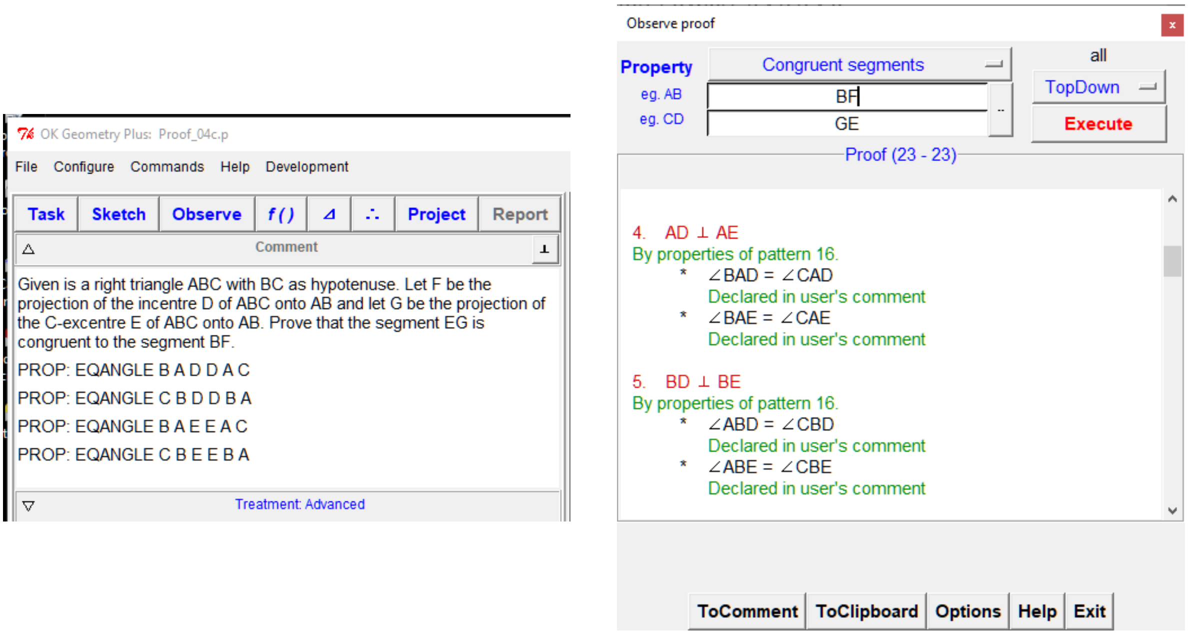


Figure 58

See Section 5.3.5 for more information about proof (non)compliant operations and proofs.

5.3 The proving method

5.3.1 Points used in proofs

The prover requires that all points of the construction are labelled and visible. When the prover is activated, all unlabelled points are automatically labelled and all hidden points become visible. Double points (e.g. an unlabelled point that is hidden behind a displayed point) are not allowed when proving.

As the processing time for producing a proof increases with the number of geometric properties of the construction, it is advisable to avoid irrelevant points in the analysed configurations.

5.3.2 The considered properties

The GDD prover only considers certain kinds of geometric properties among points (Figure 59). Proofs may also involve equality of angles between lines (e.g. AB#CD = EF#GH), but overall, only the listed properties are used in proofs.

Unless your construction contains noncompliant commands, all construction-derived properties are obtained automatically during the construction, in the **Observe proof form** you specify only the property to be proved.

Information in **Observe proof form**	'SHOW:' or 'PROP:' information in **Comment section**
points A, B, C are collinear	COLLINEAR A B C

points A, B, C, D are cocircular (cyclic)	CYCLIC A B C D
lines AB and CD are parallel	PARALLEL A B C D
lines AB and CD are perpendicular	PERPENDICULAR A B C D
angles ∠CAB and ∠QPR are congruent	EQANGLE C A B Q P R
line segments AB and CD are congruent	EQDISTANCE A B C D
segment ratio AB : CD is equal to the ratio PQ : RS	EQRATIO A B C D P Q R S
triangles ΔABC and ΔPQR are congruent	CON_TRIANGLE A B C P Q R
triangles ΔABC and ΔPQR are similar	SIM_TRIANGLE A B C P Q R
lines AB, CD and EF concur at a point	CONCURRENT A B C D E F
(*Non proven facts to archive*).	

Figure 59

To prove a property, first select the type of property in the **Observe proof** form (Figure 60) and then enter the involved objects, as required by the form. When entering the point labels, use the OK Geometry established notation. For example, to prove the collinearity of the points P, Q and R, enter PQR; to prove the collinearity of points A, B1’ and CD3, enter AB1’(CD3).

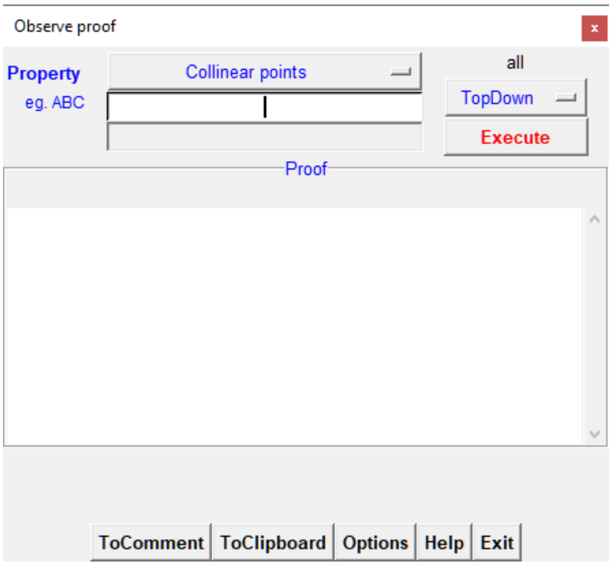


Figure 60

It should also be noted that:

- Point labels can be entered in the **Observe proof** form via keyboard or **by clicking on the point on the construction**.
- For clarity, properties can be written on two lines, but **only the order of the points typed (or clicked on) is important**. For example, the arguments for the perpendicularity of lines AB and CD can be written on two lines or on just one line as AB CD or AB,CD or just ABCD.
- The angle ∠ATB can be entered as ATB or AT#TB. Note that the prover only in some cases considers angles with unspecified angle vertex (e.g. AB#CD).

- **Angles** in notations must be **interpreted in context**. Equality ∠A = ∠B can mean the equality of angles A and B or the equality of the angle A and the supplementary angle of B.
- **The order of labels in the designations of lines, segments, triangles, and quadrilaterals in the proof is not relevant**. For example, if the prover says that the triangles ABC and PQR are congruent, this **does not mean** that in this congruence the angle ∠A corresponds to the angle ∠P, etc.

In the Comment section of the construction you can specify the property to be proved with a line SHOW: *property argument*-points. For example

SHOW: COLLINEAR A B1 C

instructs the prover to prove that A, B1, and C are collinear.

When noncompliant commands are used in the construction, you can declare in the Comment section the properties you want to be used in the proof. You do this with lines with syntax PROP: *property argument-points.*For example

PROP: PERPENDICULAR A B1 C D’

declares that the line AB1 si perpendicular to the line CD’. The spaces between points are for clarity only, and are not required.

Figure 59 contains the list of properties that can be declared or proved. When writing a comment, you can access these properties with a right-click on the Comment section (Figure 61).

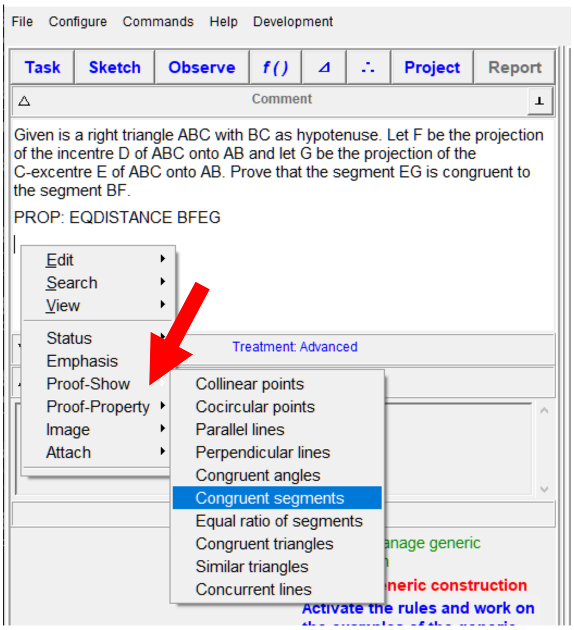


Figure 61

The last entry in the list of properties in **Observe form** (*Non proven facts to archive*), is obviously not a property. It creates a list of observed properties that the GDD prover was not able to prove. Details can be found in Section 5.4.1.

5.3.3 Patterns and GDD method

There are several established methods for proving facts in plane geometry. They range from the algebraic approach with Cartesian or trilinear coordinates to sophisticated methods of commutative algebra (e.g. the use of Groebner bases, the Wu method), from methods based on specific geometric quantities (area method, angle method) to classical deductive argumentation. In OK Geometry, we have implemented a variant of the **Geometric deduction database (GDD)** method by Chou, Gao and Zhang. This method, if successful, leads to a human-readable proof (somehow different from the common school-type deductive argumentation). Besides generating proofs, we use the method to distinguish between trivial and relevant observations (see Section 5.4).

The implemented method considers a limited number (about 45) of geometric configurations called GDD patterns. Figure 62 shows one of them (Pattern 30): a right triangle with the midpoint of its hypotenuse. The following properties of this pattern are taken into consideration:

(1) $AD \perp BD$ (2) A, B, C are collinear (3) $|AC| = |BC|$ (4) $|AC| = |CD|$ (5) $|BC| = |CD|$.

If a sufficient subset of these properties is true, then the remaining necessary properties are also true. For example:

- if (1), (2), and (3) are true, then also (4) and (5) are also true;
- if (1), (2), and (4) are true, then also (3) and (5) are also true;
- if (1), (2), and (5) are true, then also (3) and (4) are also true;
- if (1), (3), (4), and (5) are true, then (2) is also true;
- if (2), (3), (4), and (5) are true, then (1) is also true.

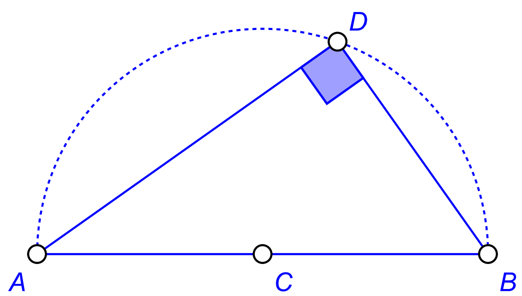


Figure 62

Each GDD pattern is similarly associated with a set of deductions. In the proof, properties are justified either by obvious deductions (e.g. transitivity of congruence) or by properties of a pattern. In the latter case, the property is a necessary condition for the specified pattern, which is proved by the listed sufficient conditions. It is assumed that "it is well known" or "can be easily inferred" by the user that the necessary condition is a consequence of the specified sufficient conditions (Figure 63). For the sake of clarity, identical relations are not listed among the sufficient conditions.

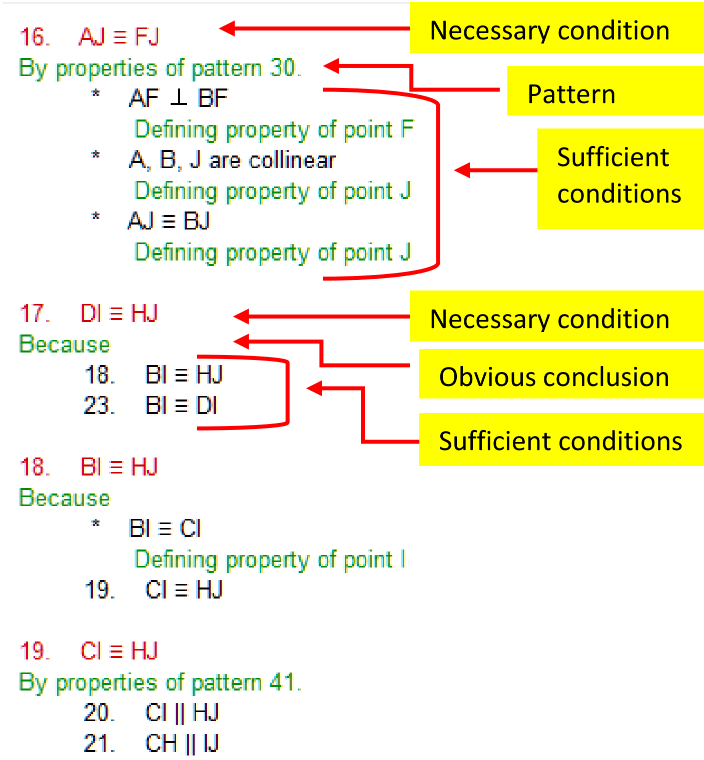


Figure 63

The patterns and the respective implications of necessary conditions from sufficient conditions are the core of the GDD proving method. The implications include: obvious facts, some well known theorems, some less known theorems, and also some useful situations that occasionally occur in proofs.

To a certain extent it is possible to control, which GDD patterns are available for a proof. You can do this via the **Options** button in the ***Observe proof*** form: in the entry *Used patterns* you can choose between levels:

- **simple** (elementary facts, not including proportionality),
- **advanced (**advanced facts, including proportionality),
- **all** (all patterns, including homothety, Ceva and Menelaus' theorem).

Section 5.7 contains the list of GDD patterns used (patterns marked with * and ** are used in *advanced* and *all* level only). The default setting uses all patterns.

The GDD method starts with a (finite) set of points and a set of assumed ("proven") facts about these points. The proven facts are a subset of the finite set of all potential relations between the points of the set. Initially, the "proven" facts consist of the defining properties of the constructed objects (points). If the prover finds sufficient conditions for a pattern among the proven facts, it adds the necessary conditions of this pattern to the set of proven facts. Note that deducing that a fact is true does not increase the set of points (thus the set of potential facts of the construction remains unchanged). An iteration in the proving process consists of the systematic check of the sufficient conditions of all patterns. The number of proven facts may increase after iteration, but the number of points in the construction and the number of potential facts remains the same. If no new facts are proved during iteration, the process is stopped. There may still be true facts that have not been reached by this process – the GDD method is simply not able to prove them. It is clear from this description that the method not only proves properties, but also generates readable proofs for them.

As we have just mentioned, not all facts can be proved in this way. Also, the proofs obtained are sometimes unnecessarily complicated or even unusual, atypical. In most cases, however, it is possible to make sense of the proof and modify it into a decent proof.

5.3.4 Improving the proof

The proof obtained may depend on the order in which the patterns were checked during the iteration. A repeated proving of the same property, but with changed order of patterns in iterations, may lead to a different proof, perhaps shorter or longer.

The first time you try to prove a property by clicking the ***Execute*** button, the prover uses an initial order of patterns. Subsequent clicks on ***Execute*** for the same property will use a random order of properties. A new proof is only displayed if it is shorter than the displayed one. In other words, repeated clicks on the ***Execute*** button sometimes lead to a shorter proof.

A click on ***Alt+Execute*** button starts automatic repetitions with randomised arrangements of the patterns. To stop the repetitions and display the shortest proof found, press the ***Escape*** key on the keyboard.

5.3.5 Proof-compliant construction commands

In the context of proofs, we use the term '**proof-compliant construction command'** for commands that create objects whose defining properties can be directly expressed with argument-objects and the properties used in proving (see Section 5.3.1). All commonly used construction commands are proof-compliant.

Here is the list of proof-compliant and proof-noncompliant construction operations in the Sketch Editor:

Group	**Proof-noncompliant commands**	**Proof-compliant commands**
Points	*Divide ratio, (Point) At length, Conic points, Grids*	All other commands
Lines	All constructions that use angle or circle as arguments (e.g. *Line 2 objects* when one or both objects are circles)	All constructions that use only points and lines as arguments
Circles	All constructions that use lines, circles or angles as arguments (e.g. *Circle 3 objects* when one or more arguments are lines, circles or angles)	All constructions that use only points as arguments

Transform	*Projectivity, Compose transformations*	All other transformations when applied to points.
Advanced	*Locus, Implicit locus, Implicit construction*, optimisation based constructions, some triangular shapes	Most triangle and quadrilateral shapes
Special	Almost all special triangle objects constructions.	Some of the basic triangle centres (X1, X3, X4, X13, X14, X15, X16)

When non-compliant operations are used in construction, some objects may lack information about their properties relevant to the proof. Such objects are displayed in light red.

Here is a general suggestion regarding the construction to use in a proof:

- If possible, create the construction using only proof-compliant commands.
- For points that are (directly or indirectly) created with non-compliant operations, declare their defining properties in the Comment section (as described in Section 5.3.2), if possible.
- If, for any reason, you do not declare the required defining properties of points created via non-compliant commands, the prover will attempt to obtain them via observation as local assumptions. Sometimes this may lead to a proof, but it is your responability to substantiate these assumptions.
  Local assumptions are not a panacea, as it often happens that essential properties in the proof turn out to be a 'local assumptions'.

Example 5

Given is a circle *k*=*k*(O,A) and a point D outside the circle. Let *p* be a tangent to *k* through the point D. Let B be the intersection point (other than A) of the circle *k* and the line DA. If C is the point of contact of the line *p* and the circle *k*, prove that the triangles DAC and DBC are similar. (Figure 64)[45]

- Clearly, the construction can be created with proof-compliant commands.

But, for our purposes, let us create the configuration exactly as described above: first the circle *k*, then the point D and the line *p*, then the point B as the intersection of *k* and the line AB, and finally the point of contact C of *p* and *k*. Here, we have used here a proof-noncompliant command *Line 2 objects*.

When the **Observe proof** form is activated, objects whose defining properties are not set are displayed in light red (Figure 65, left). This is the case for the point of contact C, which depends on points D, O, and A.

[45] OKExamples\OKG_Plus\Proof_05.p

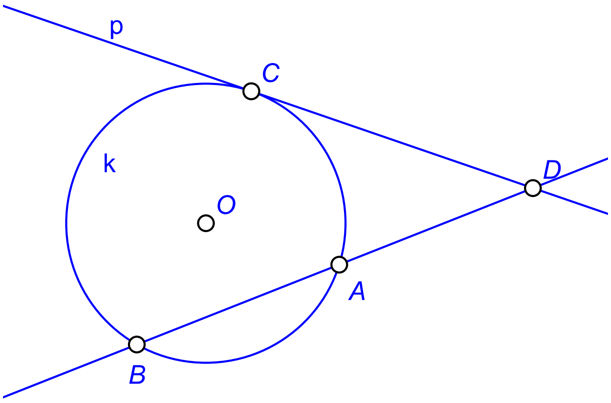


Figure 64

- At this stage we can declare (in the Comment section) the defining properties of the point C:

  PROP: EQUIDISTANCE O A O C
  PROP: PERPENDICULAR O C D C

  The prover then obtains a proof.

- If we do not declare the defining properties of point C, the prover treats the observed properties that relate C to D, O, and A (e.g. OA ≡ OC and OC ⊥ CD) as true local assumptions that should be verified by the user (Figure 65, right).

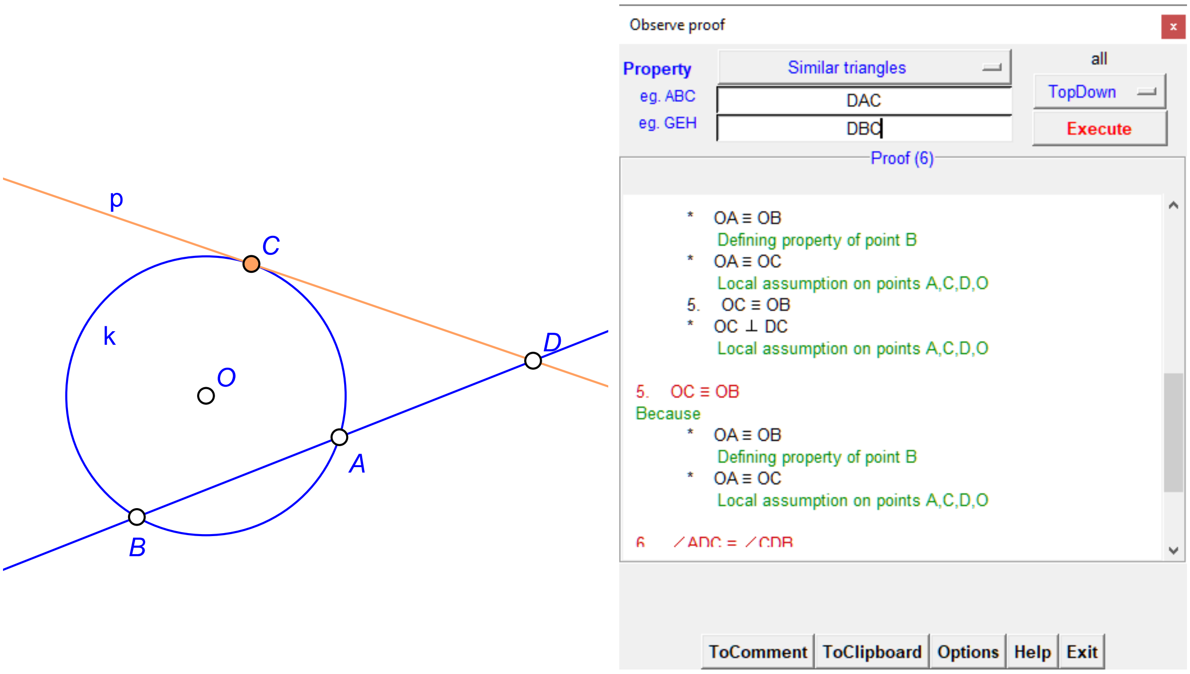


Figure 65

Example 6

Given is a square ABCD. Let *k*(E,A) be the circle with diameter AD. Let F be the point of contact (other than D) of the circle k(E,A) and its tangent through C (Figure 66). Prove that the line DF passes through the midpoint G of AB.

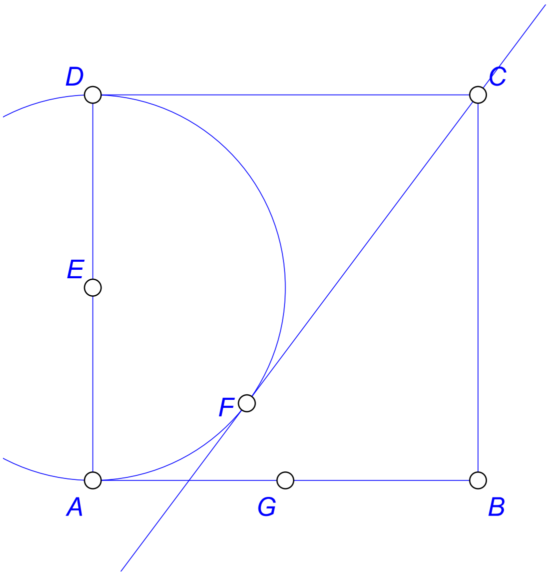
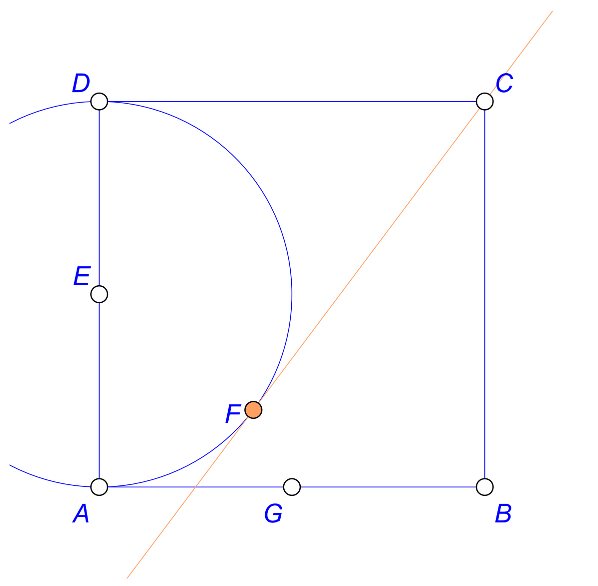
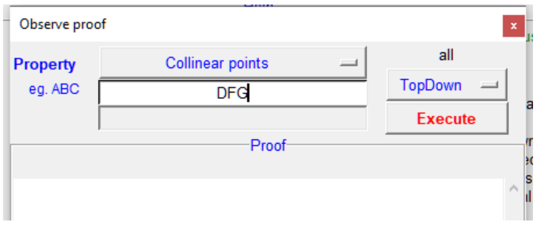
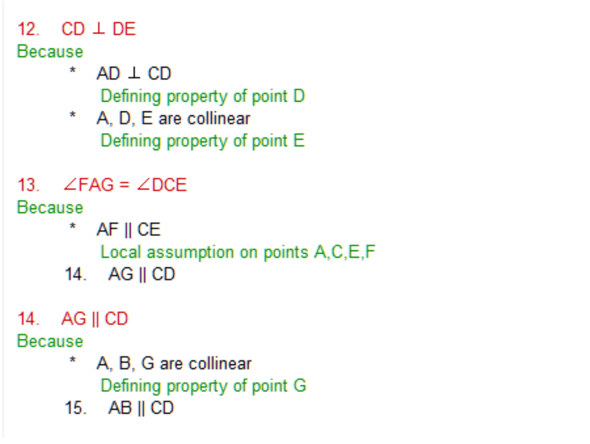


Figure 66

The construction can be carried out as described in the example using noncompliant command for the tangent line[46]. (Note that it is better to use only proof-compliant commands, i.e. to create F as the intersection of the circles *k*(C,D) and *k*(E,A).)

Create a square ABCD with the command *Advanced|Shapes|Quadrilateral|Square.* Label the vertices. Create the midpoints E and G of AD and AB. Create the circle *k*(E,A).	

[46] OKExamples\OKG_Plus\Proof_06.p

Create the tangent line from C to *k*(E,A) with *Line|Line 2 objects.* A noncompliant command is used here. Create the point of contact F. Note that F depends on E, A, and C. Therefore the geometric properties that relate F to E, A, and C are found by observation and assumed to be true.	
Now click on the ***Observe proof*** button . Some objects are coloured light red. In the form that appears, select the property *Collinear points*. Then click on the ***Execute*** button.	
In the resulting proof, there are some local assumptions regarding the points A, E, C, and F, such as the not immediately obvious fact that AF is parallel to CE. A more reasonable proof can be obtained by declaring the defining properties of point F: PROP: EQUIDISTANCE E A E F PROP: PERPENDICULAR E F C F	12. CD ⊥ DE Because * AD ⊥ CD Defining property of point D * A, D, E are collinear Defining property of point E 13. ∠FAG = ∠DCE Because * AF || CE Local assumption on points A,C,E,F 14. AG || CD 14. AG || CD Because * A, B, G are collinear Defining property of point G 15. AB || CD

5.3.6 Proofs and imported constructions

Constructions imported from other dynamic systems (Geogebra, Cynderella, JGEX, Sketchometry, Cabri, Z.u.L, Cabri Geometre, Cabri Express, Geometry Expressions), in most cases, cannot be used for proving. If you want to do proofs on imported constructions from Geogebra, Cynderella, JGEX or Z.u.L , we recommend that you

- construct all points using points, lines and circles only (i.e. without using polylines);
- before the import, raise the OK Geometry flag **Safe intersections**.

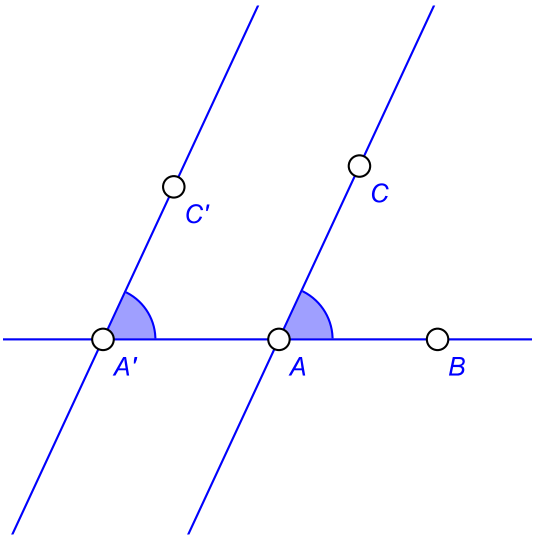
5.3.7 GDD method and observation of dynamic constructions

In this section, we explain how OK Geometry uses observation of dynamic constructions to assist in creating deductive proofs. Observation is indeed used in GDD proving for two purposes. The first purpose is to identify GDD patterns in the configuration. Although these patterns are found through observation, their existence is always established deductively. In this respect, observation does not undermine the validity of the proof. The second purpose of using observation is to make decisions in certain situations, which we will explain in the following examples:

Let A, A' and B be collinear points. Let C and C' be points, so that ∠BA'C = ∠BAC (Figure 67, left). Strictly speaking, we should not deduce from this that A'C' is parallel to AC. In fact, this is true in the situation shown, but not in general – just imagine the point B between A and A'. In cases like this, OK Geometry uses observation to check if the deduction is correct in the particular configuration and, according to the observation concludes that A'C' is parallel to AC or not. The deduction is correct under certain assumptions, but not generally.

An analogous situation is shown in Figure 67, right. From the fact that A, B, B' and C are concyclic, it is not generally true that ∠ABC = ∠AB'C. Also in this case, OK Geometry would observe the construction to decide whether to make the deduction about angles or not.

It follows that sometimes the obtained GDD proof hides certain assumptions that are implicit in the instance of the studied construction.



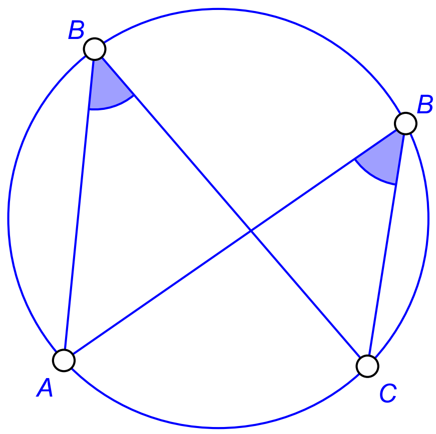


Figure 67

5.4 Archive of unproven facts

5.4.1 Archive of unproven facts

The GDD proofs are obtained by iterating deductions through the set of GDD patterns. Initially, the proven properties consist of the defining properties of the constructed objects. Each iteration augments the set of proven properties (until the process stops). As a measure of the difficulty of proving a property, we take the number of iterations after which the property is proved. The properties that the GDD prover does not prove are assigned the highest difficulty.

For a given construction, OK Geometry can list the properties that are difficult or impossible to prove with the GDD method (without adding a point to the construction). The list is available as an archive and can be inspected like other archives (see Section 7). This section only describes how to create such an archive:

1. Create a suitable configuration and activate the **Observe proof** form.
2. For the *Properties* entry select the item *Non-proven facts to archive*. Some options appear (Figure 68).
3. Fill in the data for the archive (Figure 68, see comments below).
4. Click on ***Execute***.

After a short elaboration (in a separate window), an archive is created in the Archives directory (this directory must be declared in *Configure|General options|General options*). The generated archive is automatically named xxxxxxxxxxxxxxx*.arh* (where x..x are digits reflecting the current time). You can view the generated archive with the command *File|Latest archive*.

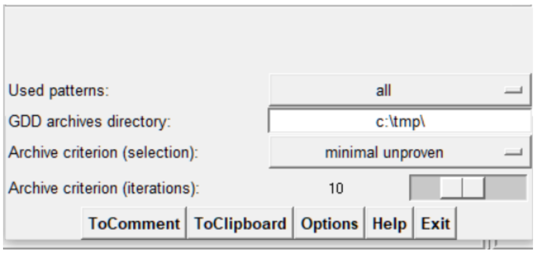


Figure 68

The archive entries to be filled (Figure 68) require some explanation:

Used patterns specifies the available set of patterns for the proof. Choose between **simple** (elementary facts, without proportionality), **advanced (**advanced facts, including proportionality), **all** (all patterns, including homothety, Ceva and Menelaus' theorems)**.**

GDD archives directory is included in the configuration of OK Geometry (it can also be set in *Configure|General options|General options*). This is the directory where the prover creates archives. The created archives are named as xxxxxxxxxxxxxxx*.arh* (where x..x are digits reflecting the current time). The latest created archive can be accessed with the command *File|Latest archive*.

Archive criterion (iterations). A property fulfils this criterion if the GDD prover does not prove it in fewer than the specified number of iterations . Note that <u>iterations are not the same as steps in the proof</u>. An iteration can result in one or several steps in the proof. The default value for this criterion is 10. Very simple proofs require 3 or 4 iterations, most high school geometry problems are usually solved in 5 or 6 iterations.

Archive criterion (selection). This entry specifies which entries among those that fulfil the iteration criterion are to be included in the archive. The options are:

1. **All unproven**. The archive will contain all properties that require at least the specified number of GDD iterations to be proved. The resulting list of properties can be overwhelming and difficult to use.

2. **Different unproven**. The list of 'all unproven' properties is reduced by avoiding properties that can be derived from the previously listed properties using equivalence of relations. For example, if P, Q, R are collinear and the list contains the property 'PQ is parallel to UV', then the list would not contain the property 'PR is parallel to UV 'and 'QR is parallel to UV'. If the configuration has a declared cyclic structure (see *Invariant cycle* entries in the *Observed proprieties* section of the left-hand pane), this option also avoids repetitions of cyclically equivalent properties. All this can significantly reduce the number of archived properties.
3. **Minimal unproven**. In this case, the prover adds new properties to the archive sequentially. At each step, a property is added only if it cannot be GDD-derived from the previously proven or previously added unproven properties. If the configuration has a declared cyclic structure (see *Invariant cycle* entries in the *Observed proprieties* section of the left-hand pane), this option also avoids repetitions of cyclically equivalent properties. This method significantly reduces the number of properties in the archive. This is the default mode for selecting unproven properties.

Example 7

Given is a triangle ΔABC. The sides of this triangle are the hypotenuses of the outwardly constructed isosceles right triangles ΔA'CB, ΔB'AC, C'BA (Figure 69). In this configuration we are looking for properties that are "difficult to prove".

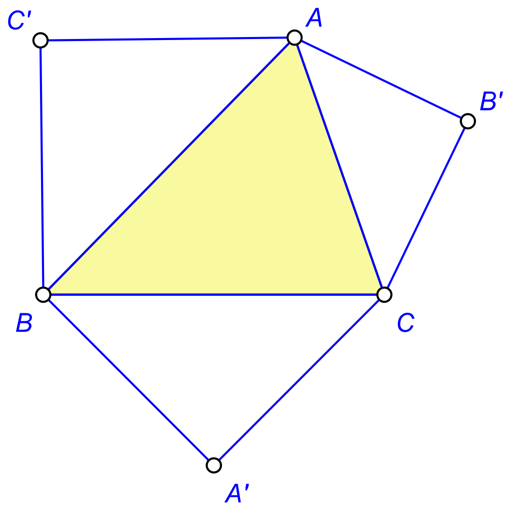


Figure 69

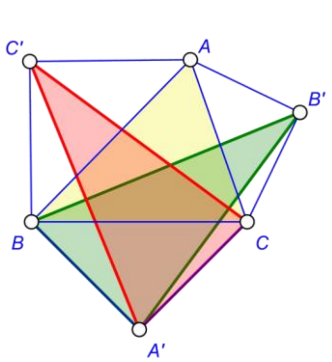
We look for the properties of this configuration that the GDD prover cannot prove. First of all, note that the described configuration is indeed easy to create[47]: simply draw the triangle ΔABC with the command *Special|A triangle* and then the triangles on the sides with *Advanced|Shapes|Triangle,45-45-90.* There is also evident cyclic structure, so we declare the *Invariant cycle1* to be: ABC A'B'C'.

[47] OKExamples\OKG_Plus\Proof_07.p

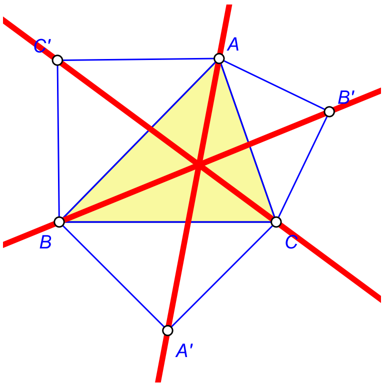
In the **Observe proof** form, select the property '*Non-proven facts to archive*'. Among the options of this form, choose *all* used patterns, 10 iterations and *all unproven* as *Archive criterion*. Then click on ***Execute*** button. After a while, an archive will be created in your Archive directory. You can inspect the properties found with the *File|Latest archive* command.

The size of the archive depends on the selection criterion. If *all unproven* properties was chosen, the archive contains 43 cases (i.e. observed properties that were not proved by the GDD method). With the *different unproven* option the number of archived properties shrinks to 20 cases, while the *minimal unproven option* still reduces the number to only 2 properties, as shown in Figure 70.

Let us reiterate how to interpret the two unproven properties obtained the Archive criterion minimal unproven. In the configuration under consideration, the GDD method proved (without adding points) all but 43 properties. One of these is the congruence of triangles A'CC' an BA'B'. Assuming this property is true, the GDD method can prove many other properties, but not the congruence of lines AA', BB', CC'. Assuming this property is also true, the GDD method can prove all the properties of the configuration under consideration.



Congruent triangles A'CC', BA'B'



Concurrent lines AA', BB', CC'

Figure 70

5.5 Adding points for proof

5.5.1 Adding a point automatically

When OK Geometry cannot prove a given property of a construction, it offers the option to attempt the proof by adding a point to the construction (Figure 51, center). This attempt may take some time, but it might succeed. The process can be stopped at any time by closing the progress display. Alternatively, you can add one or more points yourself in the Sketch Editor and repeat the proving process.

In Example 7 (Figure 69), we found some properties of the configuration under consideration that the GDD method did not prove (Figure 70). We can attempt to prove these by adding a point. If we attempt to prove that the triangles ΔBA'B' and ΔA'CC' are congruent (Figure 70, left), the prover proves the property by adding the point E = mirror image of C' in AB (Figure 71). Note that you may obtain a different solution due to some randomness in choosing the point to be added to the construction.

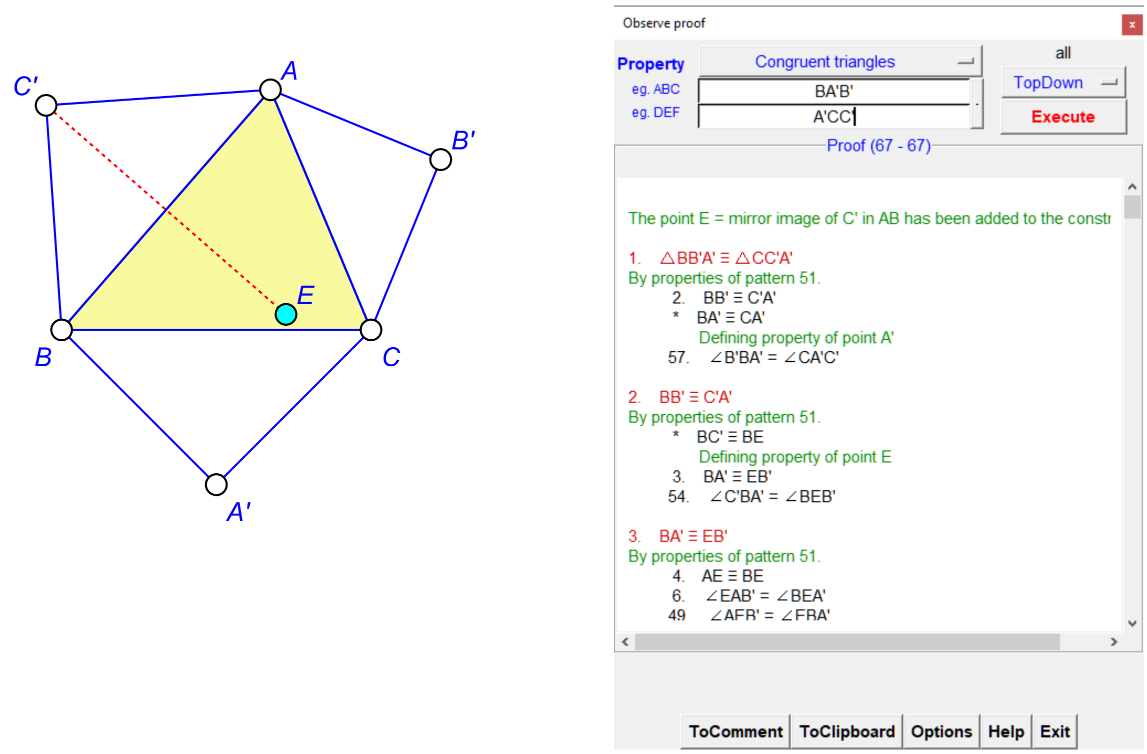


Figure 71

Example 8[48]

Given is a triangle ΔABC. On its sidelines AC and BC there are points A’ and B’ so that A’B’ is parallel to AB. If S and S’ are the circumcentres of ΔABC and ΔA’B’C, prove that C, S and S’ are collinear (Figure 72, left).

[48] OKExamples\OKG_Plus\Proof_08.p

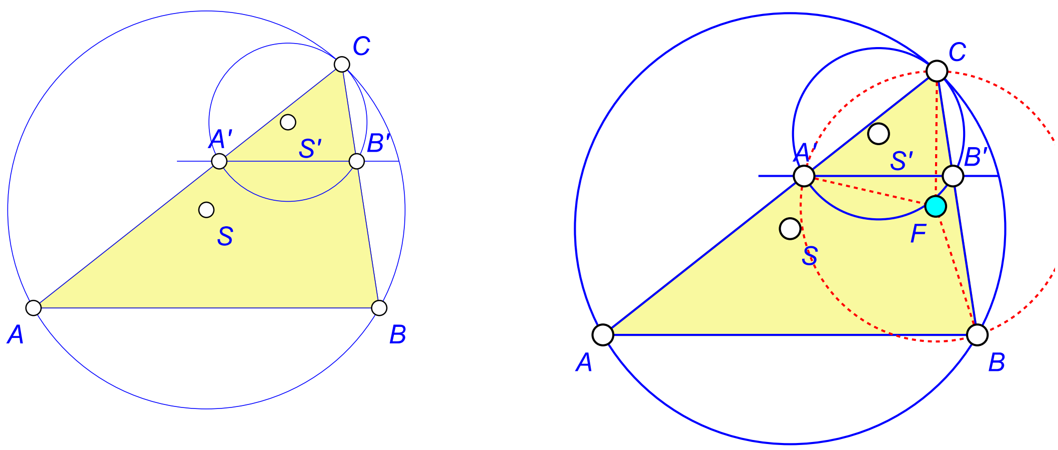


Figure 72

In this case, too, to get a proof, the prover needs an additional point. The added point is the circumcentre of the triangle A'BC (you may get something else). This example shows that sometimes the GDD prover finds a rather sophisticated proof for facts that are almost trivial to prove if a suitable theory is applied, in this case the homothety of triangles ΔABC and ΔA'B'C.

5.5.2 A background detail

Even in simple configurations, there are many ways of placing a new point to potencially enable the creation of a proof. As testing all possibilities would take a long time, OK Geometry tests only a subset of them. Without going into detail, let us say that many options for the added point are derived from the first 3 defined points, and somewhat fewer possibilities are derived from the first 6 defined points of the construction.

If we wish to intervene in the selection of points, we can specify which 6 points should be used to generate the selected points. We do this immediately after the appearance of the **Observe proof** form by using the **Toggle Unknown status** button  to select the 6 points in the desired order.

5.5.3 Adding points manually

If the GDD prover is unable to prove a property, it is often worth retrying to work out a proof, but with a slightly modified construction. The usual approach is to add one (or more) points to the configuration. This sometimes helps! The big question is where and how to add a new point – this is where good intuition and well-structured knowledge of the topic come into play. Nevertheless, there are a few typical ways of adding points that are worth trying out to get a proof (Figure 73). You can add one or more point to the considered construction and retry with the GDD prover.

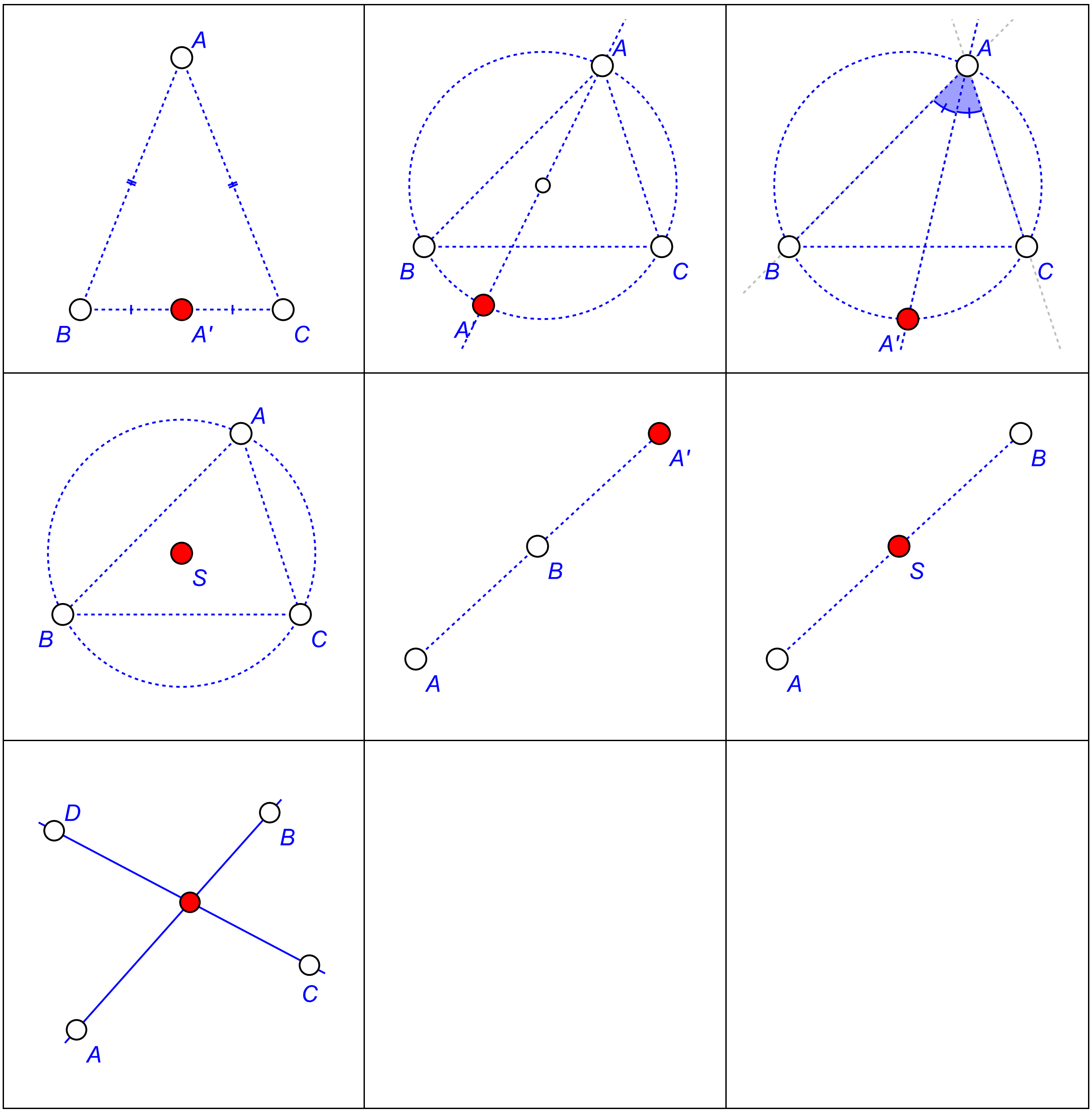


Figure 73

5.6 Additional comments

We present here some less important options, procedures and comments on GDD proving.

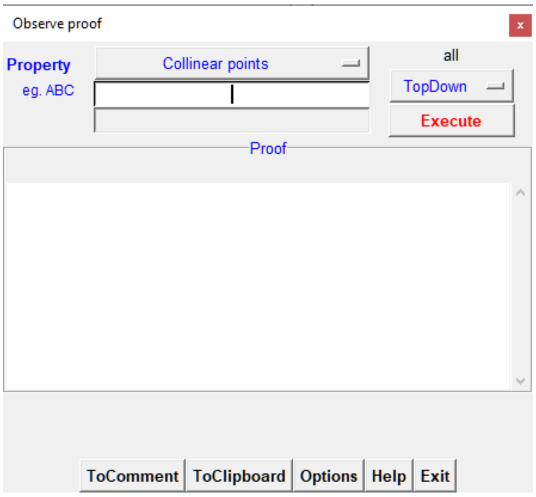
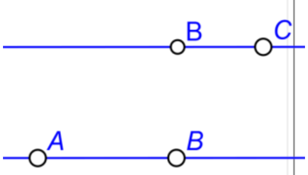


Figure 74

- **Type of proof.** The default type of proof is ***TopDown**, i.e.* the proof starts with the claim to be proved and recursively breaks down the arguments to known facts (hypotheses). This is not the usual way of presenting proofs (in schools), where the bottom-up presentation is the standard. The prover supports the ***BottomUp*** method to some extent, but currently it is an experimental feature. A third way of presenting a proof is ***Just tips*.** This kind of proof presentation is only a lists of patterns used in the proof – so it only gives hints for the proof.
- The found proof can be copied as a comment to the studied construction (i.e. pasted into the section *Comment* in the Sketch or Observation module). You achieve this with the ***ToComment*** button in the **Observe proof** form (Figure 74). As part of the comment, it can be then converted to an icon, exported to a report, etc.
  Another way to transfer the proof is via the clipboard (the ***ToClipboard*** button**) . I**n this case, the symbols in the proof are converted to plain text so that they can be easily transferred to other software.
- The **Options** button displays the settings for creating the archive of unproven properties.
- The **Help** button is not a comprehensive help for the prover. It only reminds of some features of the **Observe proof** form.

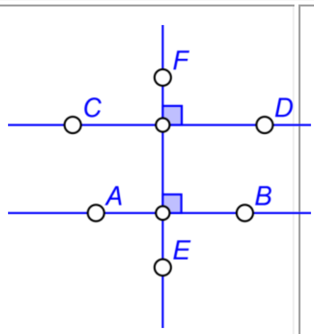
Finally, it should be noted that the prover described here also deals with **generic constructions** (Section 6). Generic constructions are families of constructions. Investigating the properties of a family of dozens or thousands of constructions is a challenge, since most properties of most constructions of the family are trivial. The prover can serve as a filter that eliminates obvious (i.e. easy to prove) properties of the constructions. Details can be found in the Section 6.10.

5.7 Used GDD patterns



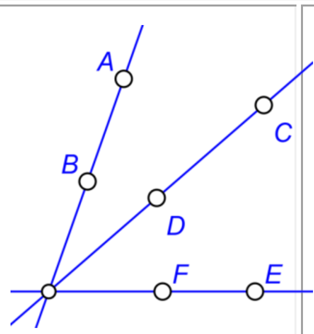
10 Parallel lines through a common point

AB || BC
A, B, C are collinear.



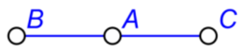
11 Lines that are perpendicular to a common line

AB ⊥ EF
CD ⊥ EF
AB || CD



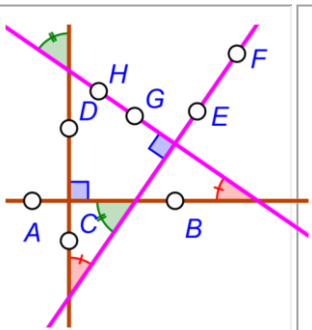
12* Three different and concurrent lines

AB, CD, EF meet at a common point.



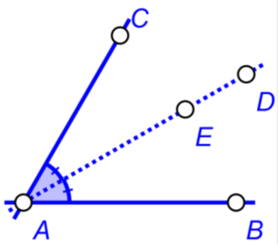
13* Midpoint of a segment

A,B,C are collinear.
|AB| = |AC|



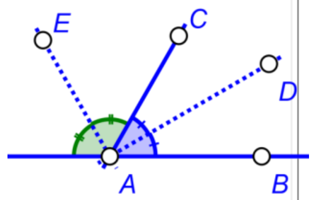
14 Angles with orthogonal respective rays

AB ⊥ CD
EF ⊥ GH
∠(AB,EF) = ∠(CD,GH)
∠(AB,GH) = ∠(CD,EF)



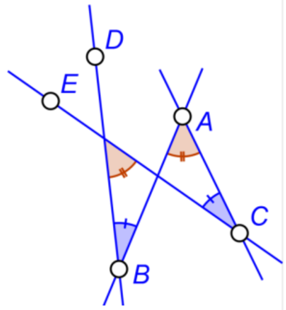
15 Points on the bisector of angle

∠BAD = ∠CAD
∠BAE = ∠CAE
A, D, E are collinear



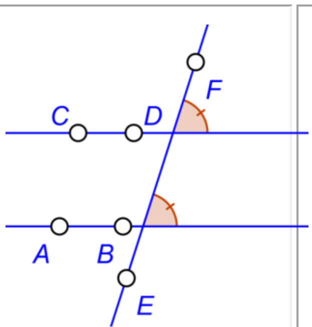
16 Points on the bisectors of angle and its supplement

∠BAD = ∠CAD
∠(AB,AE) = ∠(AC,AE)
AD ⊥AE



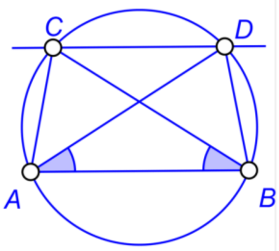
17 Rotated rays of angle

∠ABD = ∠ACE
∠BAC = ∠(AB,CE)



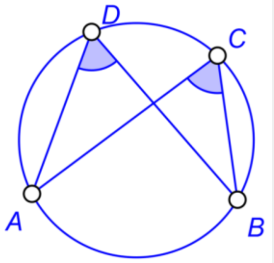
18 Transversal of parallel lines.

∠(AB,EF) = ∠(CD,EF)
AB || CD



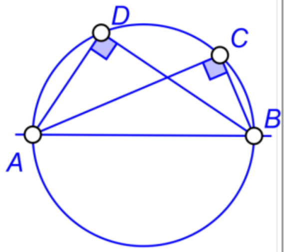
20 Parallel chords of a circle

A, B, C, D are concyclic.
AB || CD
∠ABC = ∠DAB



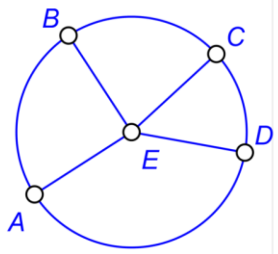
21 Four concyclic points

A, B, C, D are concyclic.
∠ADB = ∠ACB



22 Concyclic points over a circle diameter

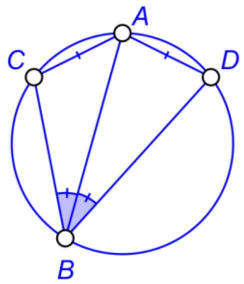
AC ⊥ BC
AD ⊥ BD
A, B, C, D are concyclic



23 *Four points on a circle with given centre*

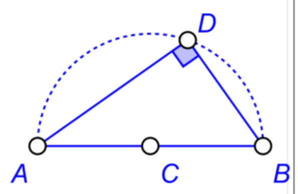
A,B.C.D lay on a circle with centre E.

|AE| = |BE| = |CE| = |DE|



24 *Chords and peripheral angles*

Congruent chords correspond to congruent (or supplementary) peripheral angles.



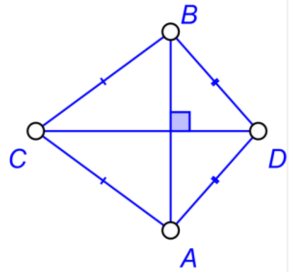
30 *Circumcentre of a right triangle*

△ABD

AD ⊥ BD

|AC| = |BC| = |DC|

A, B, C are collinear

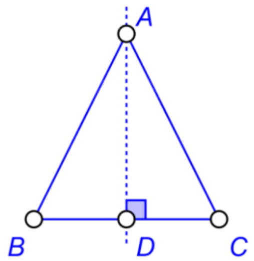


31 *Deltoid (kite)*

|AC| = |BC|

|AD| = |BD|

AB ⊥ BD



32 *Isosceles triangle with base altitude*

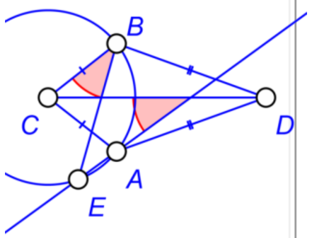
△ABC

|AB| = |AC|

B, C, D are collinear

BD ⊥ AD

|BD| = [CD|

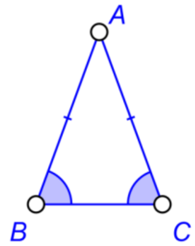


33 *A variant of central and peripheral angle theorem in circle*

|CA| = |CB| = |CE|

|DA| = |DB|

∠CBE = ∠(AE,CD)

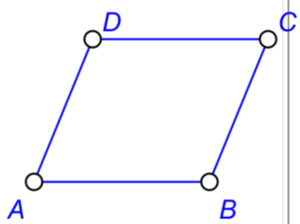


40 *Isosceles triangle*

△ABC

|AB| = |AC|

∠B = ∠C



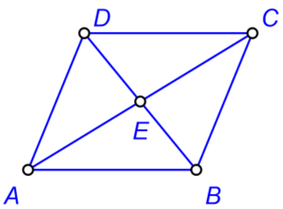
41 *Parallelogram*

AB || CD

AD || BC

|AB| = |CD|

|AD| = |BC|



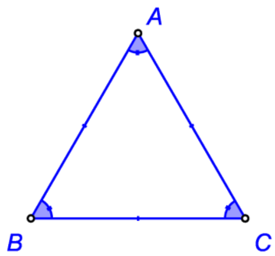
42 *Parallelogram with its centre*

A, E, C are collinear

B, E, D are collinear

|AE| = |CE|

|BE| = |DE|

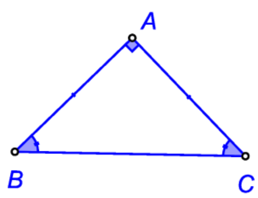


43 *Equilateral triangle*

△ABC

|AB| = |BC| = |CA|

∠A = ∠B = ∠C



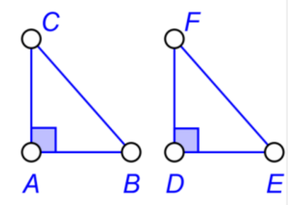
43 *Equilateral isosceles triangle*

△ABC

AB ⊥ AC

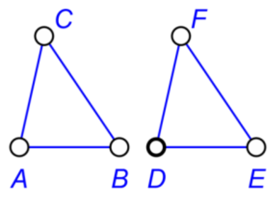
|AB| = |AC|

∠B = ∠C



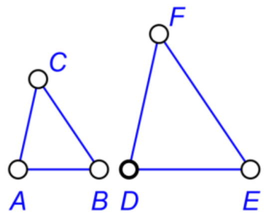
50 Congruent right triangles

△ABC ≡ △DEF
∠A = ∠D = right angle
SSS congruence
SAS congruence
ASA congruence
SSA congruence



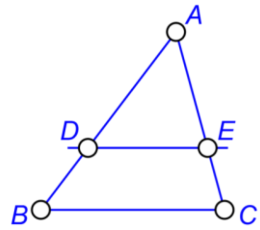
51 Congruent triangles

△ABC ≡ △DEF
SSS congruence
ASA congruence
SAS congruence



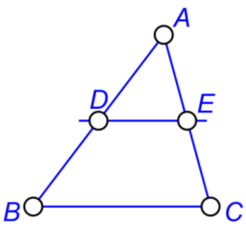
60 Similar(scalene) triangles

△ABC ~ △DEF
SSS similarity
SAS similarity
ASA similarity



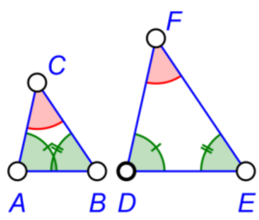
61 Thales theorem (Similar triangles)

A, B, D are collinear.
A, C, E are collinear.
BC || DE
|AD| : |AB| = |AE| : |AC|



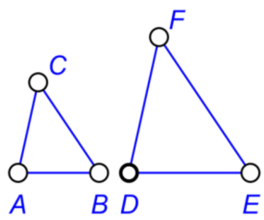
62 Midpoint Thales theorem

A, B, D are collinear.
A, C, E are collinear.
|AD| = |BD|
|AE| = |CE|
BC || DE



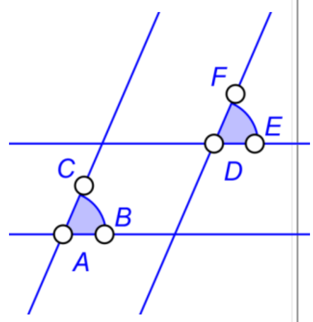
63 Similar triangles

If two pairs of the corresponding angles of triangles are congruent, then so is the third pair of angles.



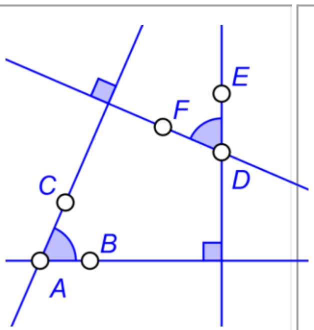
60 Similar triangles

△ABC ~ △DEF
SSS similarity
SAS similarity
ASA similarity



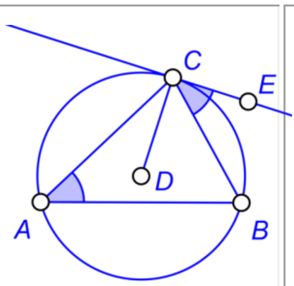
70 Angles between pairs of parallel lines

AB || DE
AC || DF
∠BAC = ∠EDF



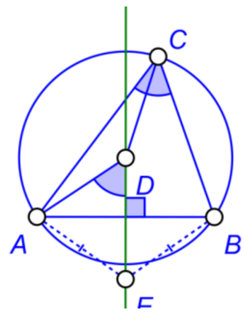
71 Angles having their respective rays orthogonal

AB ⊥ DE
AC ⊥ DF
∠BAC = ∠EDF



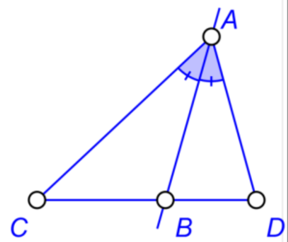
72 Chord tangent angle

|AD| = |BD| = |CD|
CD ⊥ CE
∠BAC = ∠BCE



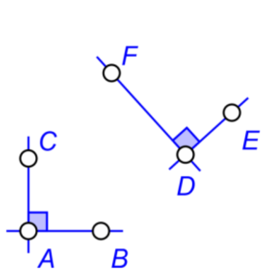
73 Central and peripheral angle

|AD| = |BD| = |CD|
|AE| = |BE|
AB ⊥ DE
∠ACB = ∠ADE



74 Angle bisector in triangle

△ABC
B, C, D are collinear
∠CAB = ∠BAD
|BC| : |BD| = |AC| : |AD|

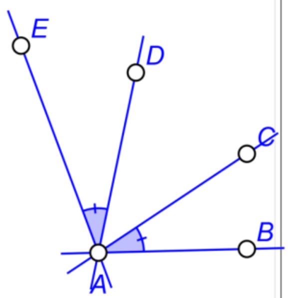


75 Right angles are congruent

AB ⊥ AC

DE ⊥ FF

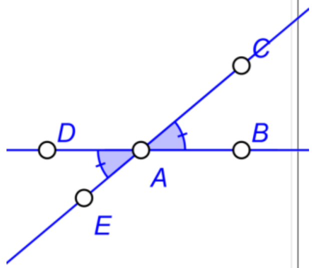
∠BAC = ∠EDF



76 Congruent angles with common vertex

∠BAC = ∠DAE

∠BAD = ∠CAE

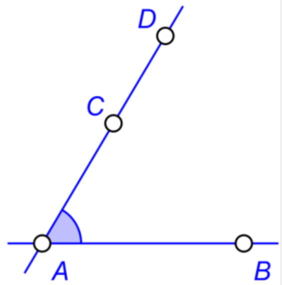


77 Alternate angles with a common vertex

∠BAC = ∠DAE

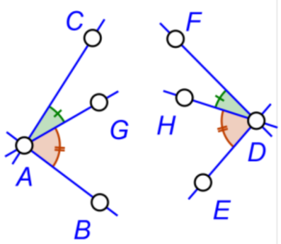
A, B, D are collinear.

A, C, E are collinear



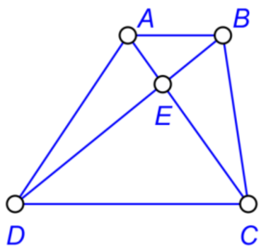
78 Same angle

Collinear points on rays give rise to congruent (or supplementary) angles



79 Sum/difference of adjacent angles

The angle sum/ difference of pairwise congruent adjacent angles are congruent angles

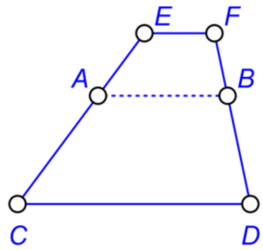


80 Ratio and parallel lines

AB || CD

E = AC ∩ BD

|AE| : |EC| = |BE| : |BD|



81 Parallels in trapezium

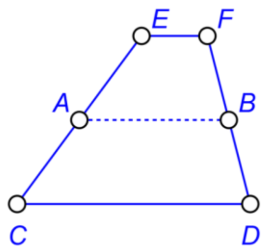
CD || EF

A, C, E are collinear.

B, D, F are collinear.

AB || CD

|AC| : |AE| = |BD| : |BF|



82 Midpoint parallels in trapezium

CD || EF

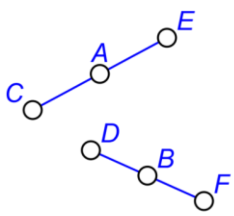
A, C, E are collinear.

B, D, F are collinear.

|AC| = |AD|

|BD| = |BF|

AB || CD



83 Midpoints and ratio

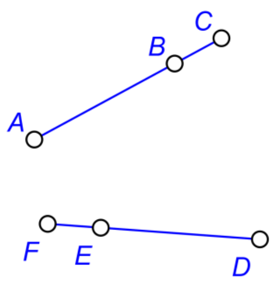
A, C, E are collinear.

B, D, F are collinear.

|AC| = |AE|

|BD| = |BF|

|AC| : |AE| = |BD| : |BF|



84 Sum/difference of segments

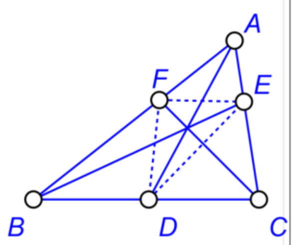
A, B, C are collinear.

D, E, F are collinear.

|AB| = |DE|

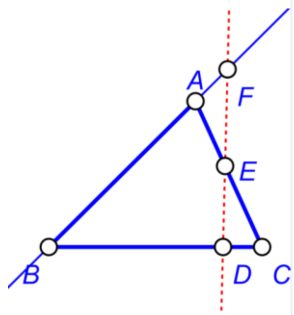
|BC| = |EF|

|AC| = |DF|



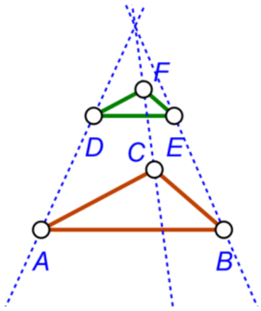
91 Inscribed perspective triangle in a triangle

Ceva theorem

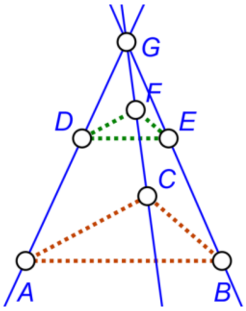


92 Collinear points on the sidelines of a triangle

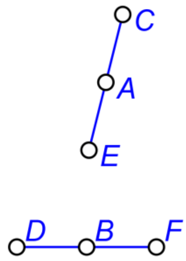
Menelaus' theorem



93 Homothetic triangles



94 Homothetic triangles



98 Midpoints of congruent segments

98 Algebraic manipulation on fractions

6 Generic constructions

6.1 Introduction

Consider the following construction (Figure 75): let P be the orthocentre of a given triangle ABC. Consider the following three circles: the circle passing through B, C, P, the circle passing through C, A, P, and the circle through A, B, P. Label the centres of these circles A', B' and C' respectively[49]. This well known configuration is blessed with several nice properties that can be easily observed with OK Geometry. For example, the lines AA', BB', and CC' concur at a point, a fact that we check with the condition named *Concu_xxx* (concurrence of three lines). Moreover, the three circles are mutually congruent and the triangle A'B'C' is homothetic and congruent to the triangle ABC.

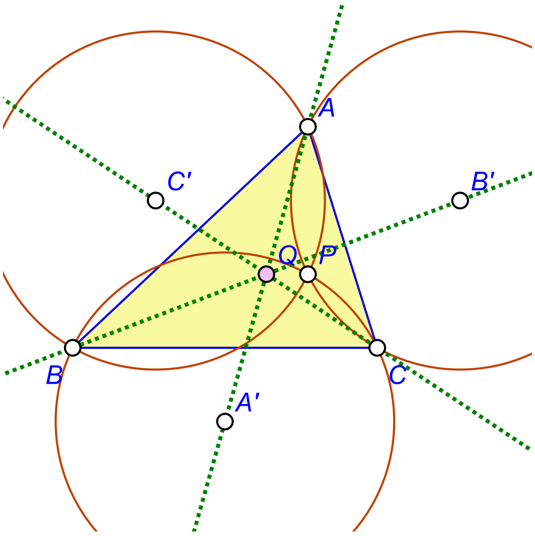


Figure 75

It is natural to examine constructions similar to the one presented here. The point P could be the centroid of ABC, the circumcentre of ABC, or some other centre of the triangle ABC. Instead of the circumcircles of triangles BCP, CAP, ABP, one can consider incircles, 9-point circles, or other circles related to these triangles. Such modified constructions may have interesting properties, but constructing each one individually would be tedious. The generic constructions have been designed to make life easier in such cases.

Note that the initial configuration was obtained as a sequence of operations, starting from the three vertices A, B, and C, the lines connecting each pair of them, and so on. What we wish to do is to vary two operations:

- Instead of the operation 'declare the point P to be the orthocentre of a given triangle' we want something like 'declare the point P to be (for the triangle) each of the following: incentre (ETC centre X1), centroid (ETC centre X2), circumcentre (ETC centre X3), orthocentre (ETC centre X4), 9-point centre (ETC centre X5)'.

[49] OKExamples\OKG_Plus\Generic_01.p

- Instead of the operation ‘construct a circle as the circumcircle of a given triangle’, we want something like ‘declare a circle as each of the following circles of a given triangle: circumcircle, incircle, 9-point circle’.

By combining these options we could consider 5×3 = 15 constructions (Figure 85) at once as a generic construction.

Put simply, a **generic construction** is a construction in which one or more operations have variants called **rules**. A generic construction can contain dozens, hundreds or even thousands constructions, called **examples**, which are obtained by applying variants of the operations in rules. The concept of rule (generic operation) is explained in detail in Section6.2.

Generic constructions allow you to:

1. ***simultaneously generate a large number of examples*** (constructions);
2. ***easily access any example of a generic construction***; examples can be studied in the same way as ordinary constructions: they can be observed, exported, subjected to triangle analysis, formulae observation, proof observation, etc.;
3. ***automatically search for examples with specified properties****, e.g.* find examples in which tangency of circles occurs or there are similar triangles;
4. ***automatically search for examples with specific instances of relations****, e.g.* find examples in which the lines AA’, BB’, CC’ concur or in which the given points A, B, C, D are cocircular;
5. ***automatically analyse all examples*** of a generic construction, e.g. in all examples analyse the position of point P with respect to the reference triangle A’B’C, or in all examples observe the formulae for the distance between points P and Q with respect to the reference triangle ABC;
6. **among all considered properties of all examples in the generic constructions, look for those that are difficult to prove**;
7. ***automatically create projects or archives from examples or a selection of examples*** of a generic construction.

Generic constructions as files are stored like ordinary constructions (files with the extension .p) or projects (files with the extension .pro).

6.2 Rules and generic constructions

Generic constructions differ from ordinary constructions primarily in that they contain one or more **rules** (or **generic operations**). You can think of a rule as a ‘union of operations’ on a set of points of a given cardinality. A rule is described by the following characteristics:

- **Type of generated object.** The naming of rules is predefined and reflects the type of generated object and the nature of the rule.

- **List of operations in the rule.** All operations in the list must generate an object of the same type (point, line, or circle) from points as arguments. The number of arguments (points) may vary between operations.
- **Cardinality and order of the set of points** to which a rule is applied. A given rule can be applied several times to different sets of points, all of the same cardinality.
- **Specific requirements**. Additional or optional requirements may be imposed to a rule. Some rules have a predefined cardinality of arguments. In a rule, we can distinguish (fix) the first argument. If required, a rule may contain only proof-compliant operations.

For an overview of the types of rules and available operations for rules see Section 6.8.

We illustrate the relationship between a list of operations and a set of points with a simple example of a rule. Assume a generic construction has only one rule, called Pobject1, that creates a point. The rule uses 2 operations: ‘point at 1/3 of segment 12’ and ETC triangle centre ‘x(1) of 123’. The rule will be applied twice on a triplet of points, first on point A, B, C to create point G, and then on points D, E, F to create point H. This rule gives rise to 7 examples:

	Instance with arguments A, B, C	Instance with arguments D, E, F
Example 1	G = point at 1/3 of AB	H = point at 1/3 of DE
Example 2	G = point at 1/3 of BA	H = point at 1/3 of ED
Example 3	G = point at 1/3 of BC	H = point at 1/3 of EF
Example 4	G = point at 1/3 of CB	H = point at 1/3 of FE
Example 5	G = point at 1/3 of AC	H = point at 1/3 of DF
Example 6	G = point at 1/3 of CA	H = point at 1/3 of FD
Example 7	G = incentre of ABC	H = incentre of DEF

In the examples, the same variant is applied in all instances of the rule. If the variant operation requires *m* arguments, all arrangements of *m* points among the arguments are considered, avoiding redundant repetitions. In our case, all ordered pairs of points among 3 arguments were considered for ‘point at 1/3 of segment’. Since the incentre is the same for all permutations of the vertices of a triangle, the triplet in the argument are not permuted. If the first argument was declared as fixed, only examples with A or D as the first argument of the operation would appear (Examples 1, 5, and 7).

Rules are declared and called with commands in the *Generic* menu group of the Sketch Editor. For the above situation we used the command *Generic|GPoint*. The declaration of a new rule occurs simultaneously with its first application. To declare a rule, we must specify the rule name, the operations to be included in the rule, and the points subject to the rule (Figure 76, left). Each subsequent occurrence of the rule will contain the same operations, the rule must be applied to the same number of points, and the fix-first-point status cannot be changed (Figure 76, right).

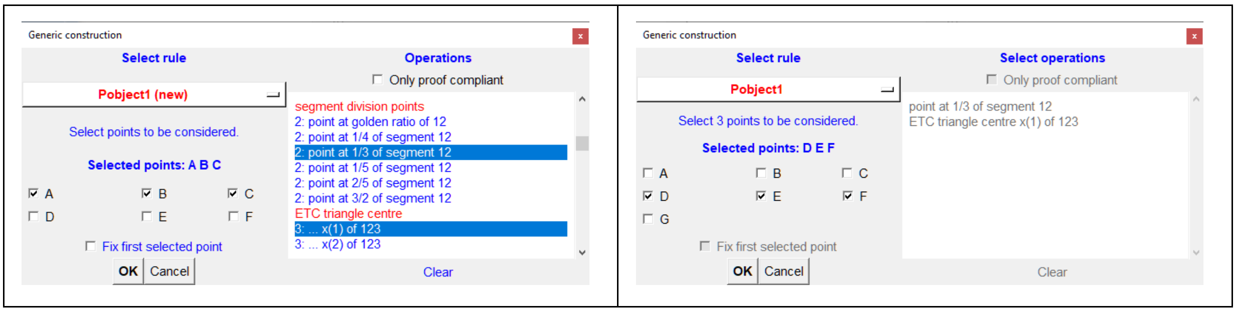


Figure 76

In each rule, one of the operation is designated as the **initial** operation (of the rule). By default, the initial operation is the first operation in the list of operations of the rule, but the user can change this (see Section 6.4.3). A generic construction, using the initial operations of its rules, is called the **initial example** of the generic construction. In ordinary work in OK Geometry, a generic construction appears as its initial example and behaves like a usual construction. The examples within the generic construction are activated when the **Examples form** is activated (Section 6.4).

It is expected that examples of a generic construction do not contain coincident objects (e.g, overlapping points). Examples in which such unwanted events occur are considered **degenerate** and are often ignored.

Generic constructions as files are stored like ordinary constructions (files with the extension .p) or projects (files with the extension .pro).

6.3 Creating generic constructions

All commands for creating generic constructions are located in the group *Generic* of the Sketch Editor. There are three subgroups of commands: 1. generic operations for creating new objects (GPoint, GLine, etc.), 2. generic operations for non-standard objects and triangle objects, 3. commands for handling generic constructions.

You create a generic construction in the same way as ordinary constructions, except that you create rules (and only rules) using commands in the *Generic* menu. We illustrate the process using the example described in Section 6.1. In this process we create two rules Pobject1 and Cobject1. The first rule creates point P as various centres of the triangle ABC, the second rule creates various triangle related circles and is applied to triangles BCP, CAP, and ABP (Figure 75).

1. Create the triangle ABC somehow. For example, you can use the command *Special|A triangle*.
2. Create the generic point P with the command *Generic|GPoint*. Fill the form that appears exactly as shown in Figure 77. Press OK to create the rule Pobject1 and apply it to points A, B, C. The first operation in the list (incentre) is, by default, the initial operation, the point P is thus displayed as

the incentre of ABC.

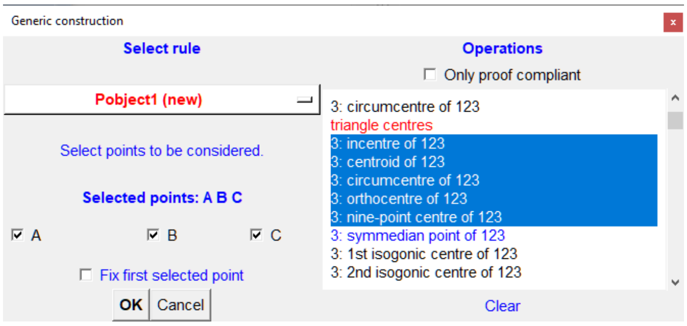


Figure 77

3. Create the generic circle with the command *Generic|GCircle*. Name the rule as Cobject1. Apply this rule to points B, C, and P (see Figure 78). Click OK to create the rule Cobject1 and apply it to points B, C, and P. The first operation in the list (circumcircle) is, by default, the initial operation, so the circumcircle of triangle BCP is displayed.

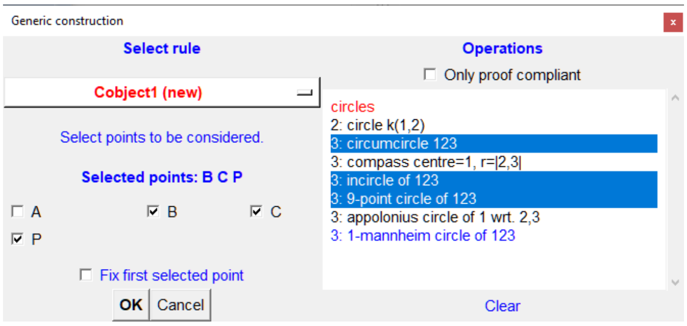


Figure 78

4. Create a new generic circle with the command *Generic|GCircle*. Be sure to apply the previously defined rule *Cobject1* on points C, A, and P (Figure 79).

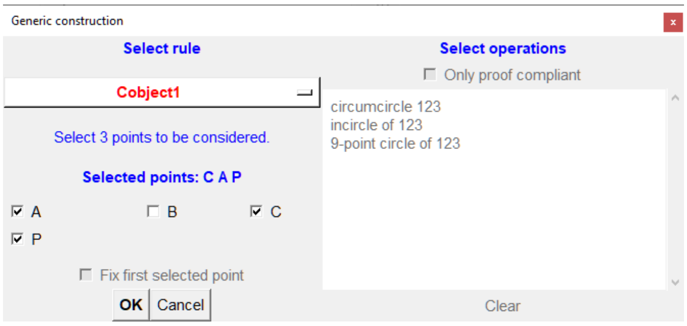


Figure 79

5. Create another generic circle with the command *Generic|GCircle*. Be sure to apply the previously defined rule *Cobject1* on points A, B, and P.
6. Use the standard commands (e.g. *Point|Centre of circle*) to obtain the centres A', B', and C' of the three circles generated with the rule *Cobject1.*
7. Add a check named *Concu_xxx* for the concurrence of the lines AA', BB', and CC'. We do this with the standard command *Advanced|Check property|ConcurrentL*.
8. Note that, in general, the lines AA', BB', and CC' do not necessarily concur, and if they do, the intersection may be different from P (see Figure 75). At this stage, it may not be possible to create the common intersection point Q, as it coincides with the point P (see Figure 80)

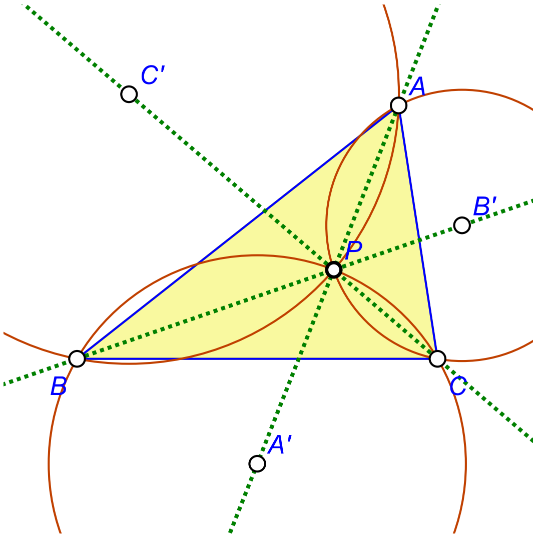


Figure 80

The obtained generic construction can be treated and saved as an ordinary construction in a file with the .p extension[50].

Note. An existing ordinary object of a construction can be converted to a generic object. You achieve this with the command *Generic|Attach to object.* Then click on the object to modify. In the form that appears, define the generic object by setting a new rule or using an existing rule. Such modification will preserve the apparent initial example of the generic construction, but everything else will be correct.

6.4 Accessing and studying examples of generic construction

6.4.1 Accessing the Examples form

All commands related to generic constructions are shown in Figure 81 and Figure 82.

The generic construction from the introductory example can be found in the file G*eneric_02.p*. When you open a generic construction, you see the initial example as an ordinary construction. The only indication that it is a generic construction is the **Gen** button**,** which turns red (otherwise it is grey and inactive), as shown in Figure 81. The **Gen** button indicates whether the current construction is generic.

[50] OKExamples\OKG_Plus\Generic_02.p

Generic constructions can be accessed at any time (except when icons of a project are displayed).

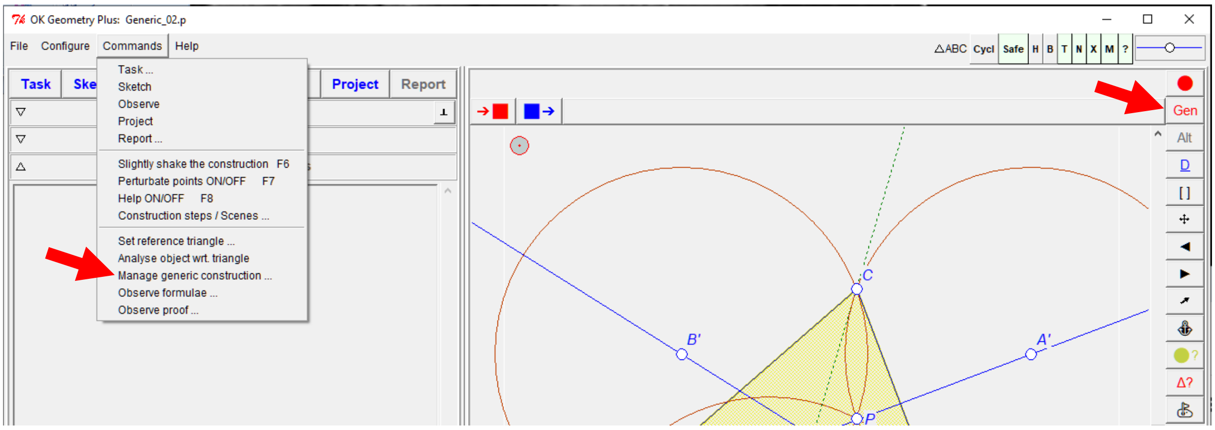


Figure 81

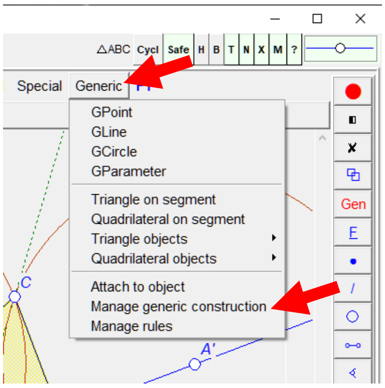


Figure 82

There are three ways to activate the examples in a generic construction (see Figure 81 and Figure 82):

- Press the **Gen** button on the toolbar to the right. This is the usual way and is always accessible.
- Use the command *Commands|Manage generic constructions*.
- In Sketch Editor, click the command *Generic|Manage generic constructions*.

A form for managing examples appears (Figure 84). For convenience, we refer to this form as the **Examples form.** The form is used for:

- Viewing and studying examples of a generic construction
- Scanning examples to create a selection of examples of a generic construction
- Observing a selection or all examples of a generic construction
- Exporting a selection of examples or all examples of a generic construction to the current project or to an archive.

Important note. The Examples form precludes any modification of the considered generic construction. Once you exit the Examples form, the generic construction is exactly as it was when you accessed the form. When the Examples form is active, changes to a displayed example do not modify the generic construction.

We shall explain and illustrate how to handle and study generic constructions. We will use the generic construction *Generic_02.p*[51] (Generic_ 03.p[52]), see Figure 83.

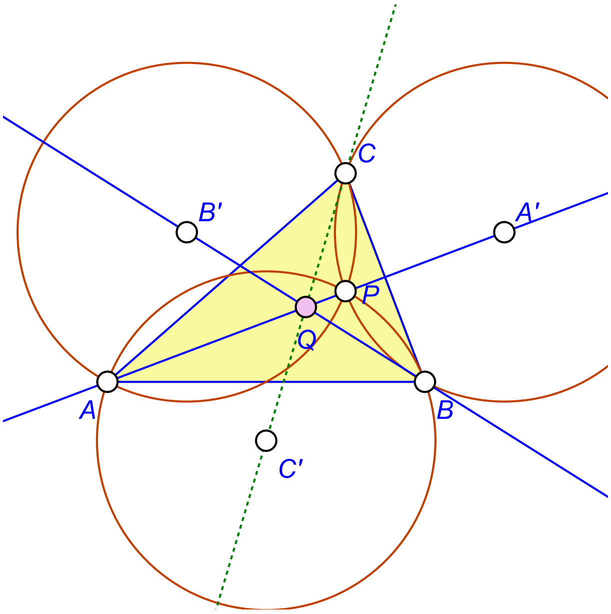


Figure 83

6.4.2 Viewing and studying examples individually

Please, open the construction *Generic_02.p*[51] and activate the **Examples form** by pressing the **Gen** button.

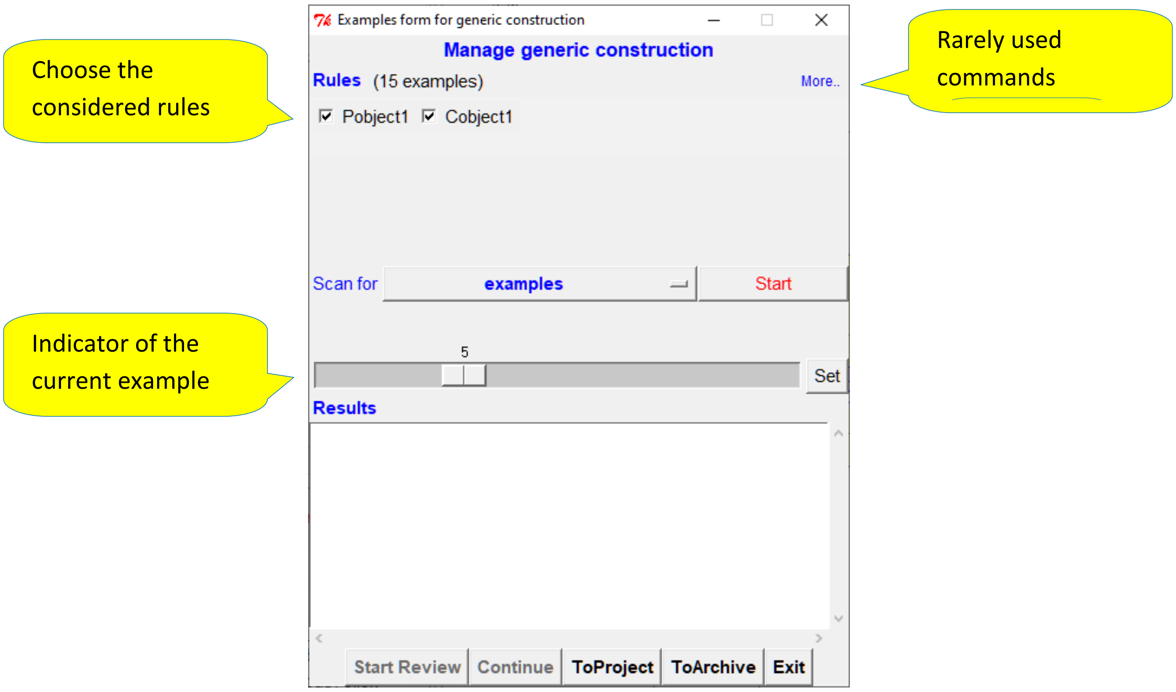


Figure 84

[51] OKExamples\OKG_Plus\Generic_02.p
[52] OKExamples\OKG_Plus\Generic_03.p

In the **Examples form** (Figure 84), checkmark the rules you wish to activate. The order in which the rules are marked determines the sequence order of the examples. If a rule is not activated, the initial operation of the rule is used in all considered examples. We activated both rules, which resulted in a total of 15 examples.

The slider in the middle of the form indicates the serial number of the displayed example. To display, say, Example 14 of the generic construction, set the slider to 14 and press the **Set** button on the right hand side of the slider. In this way you can access to each example of a generic form (Figure 84).

If the current (displayed) example is degenerate, the slider indicator turns reddish. This occurs at Example 1 and Example 12, since some created objects coincide. The degenerate examples are usually avoided (see Section 16.4.4).

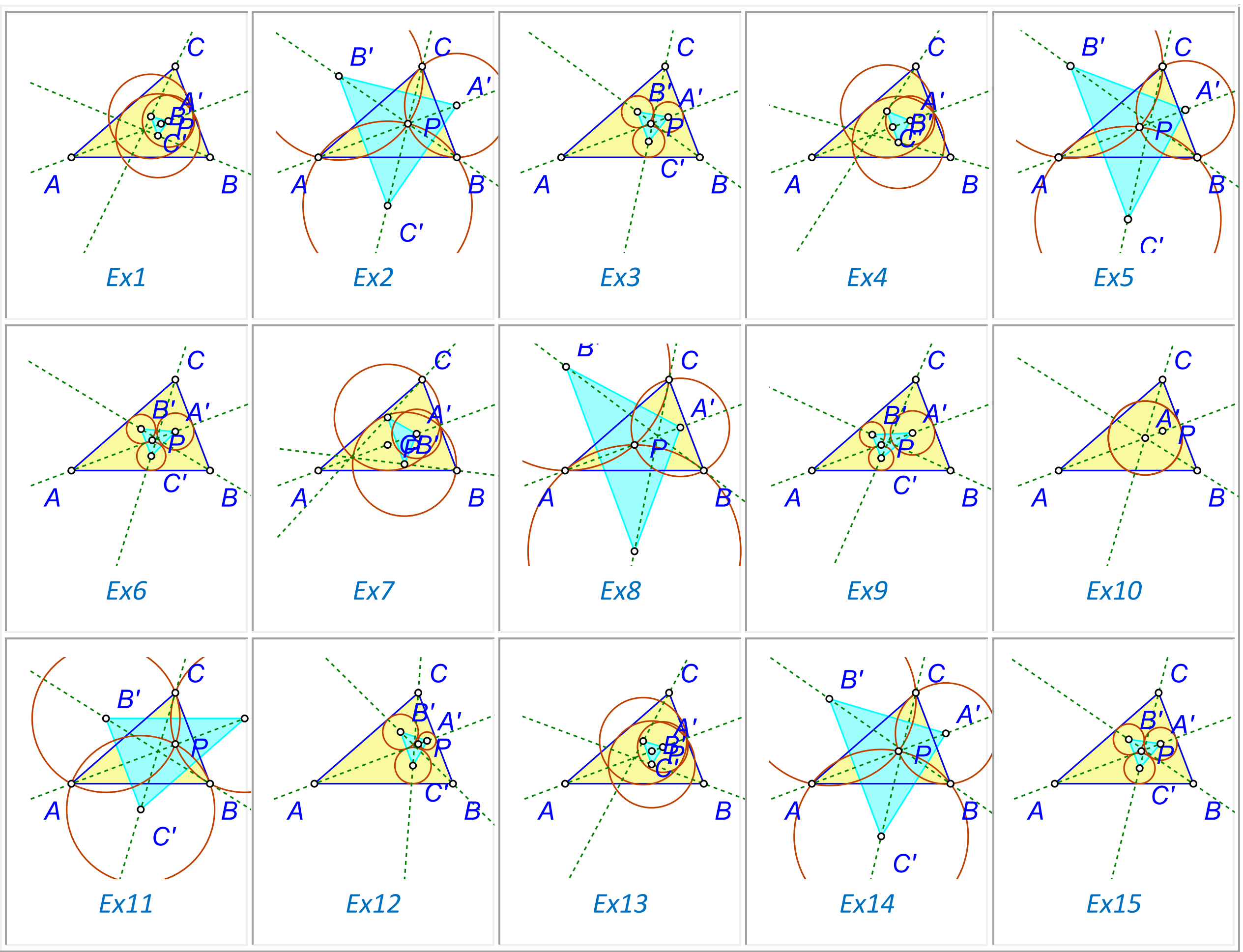


Figure 85

6.4.3 Setting the initial example.

Usually, there is no reason to modify the default initial example. However, if the initial example is degenerate or if we wish to change it for some reason, it is easy to adjust this.

Recall that we are not satisfied with Example 1 as the initial example, since it does not allow the construction of the point Q. A good choice for the initial construction is example 10 (i.e. the construction described in Section 6.1). Here is how to do this.

1. Display the desired example. In our case we display example 10.
2. (Optional) At this point you can add objects and modify their appearance. In our case, we add the point Q as the intersection of lines AA' and BB'.
3. In the **Examples form,** apply the command *More|Set as initial example.*
4. Exit the **Examples form**. At this point we save the generic construction as *Generic_03.p*.

6.4.4 More commands

In the **Examples form**, there is a menu **More...** containing some less frequently used commands and options:
Show aux. objects – refers to auxiliary objects associated with generic operations (in the *Generic* menu). For example, when the circumcentre of a triangle is created, the triangle is displayed as an auxiliary object. By default, this option is Off.
SkipDegenerate (examples) – is an optional setting to avoid degenerate examples, i.e. examples where some constructed objects coincide. Degenerate examples often cause annoying warnings. For this reason, this option is On by default.

6.5 Working on individual examples

When an example is displayed (and the **Examples form** is active), you can treat the example as an ordinary construction. For example, you can:

- save the construction as a construction *.p* file;
- export the displayed image;
- observe the example;
- observe formulae related to the example;
- perform triangle analysis on the example;
- observe proofs of properties of the example (if proof-compliant operations are used in rules);
- etc.

You can also modify the construction, move, add or delete objects and then perform the above operations on the modified construction. **However, changes to the example do not become part of the considered generic construction.** Any redisplay of the example (e.g. a click on the ***Set*** button) shows the original example.

Consider again the generic construction *Generic_03.p*[53]. In the **Examples form** set the example 7 as the current example (with P as the circumcentre of ABC and the circles as circumcircles of the triangles BCP, CAP, ABP). You can analyse the point Q wrt. triangle ABC (OK Geometry observes that Q is the Kosnita point X54 of ABC).

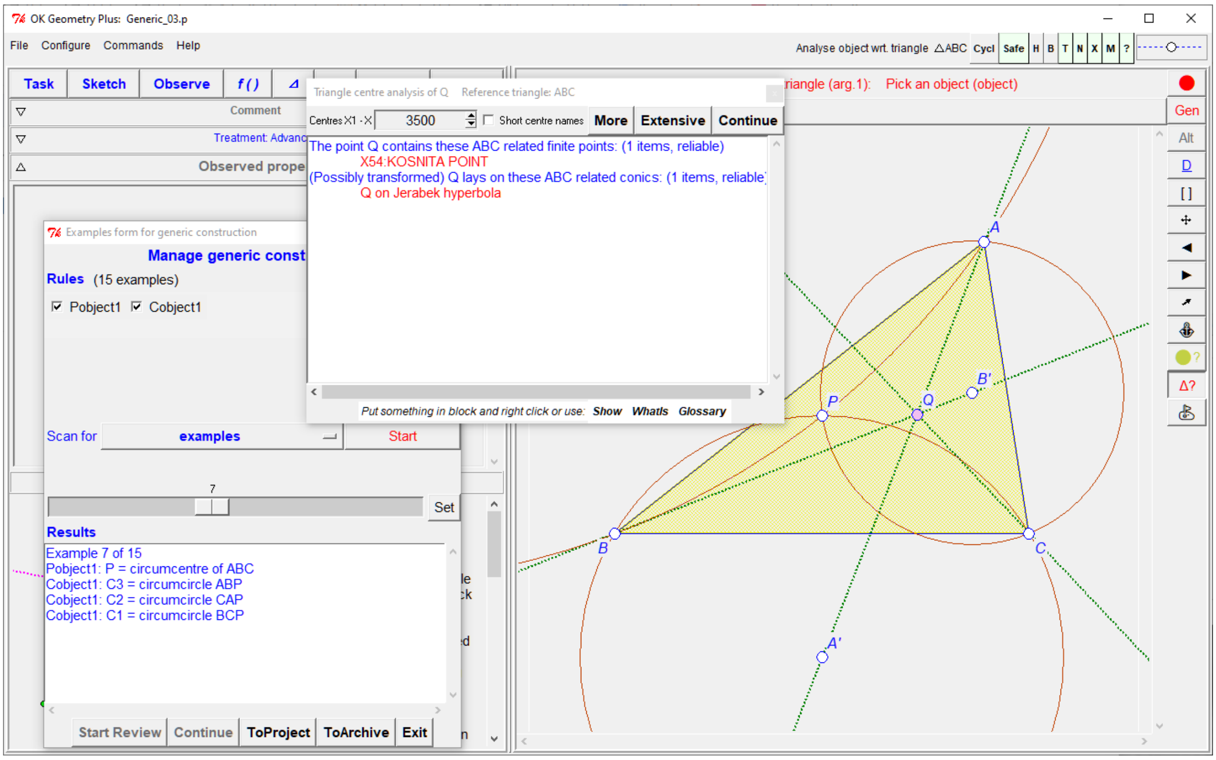


Figure 86

6.6 Scanning examples in generic construction

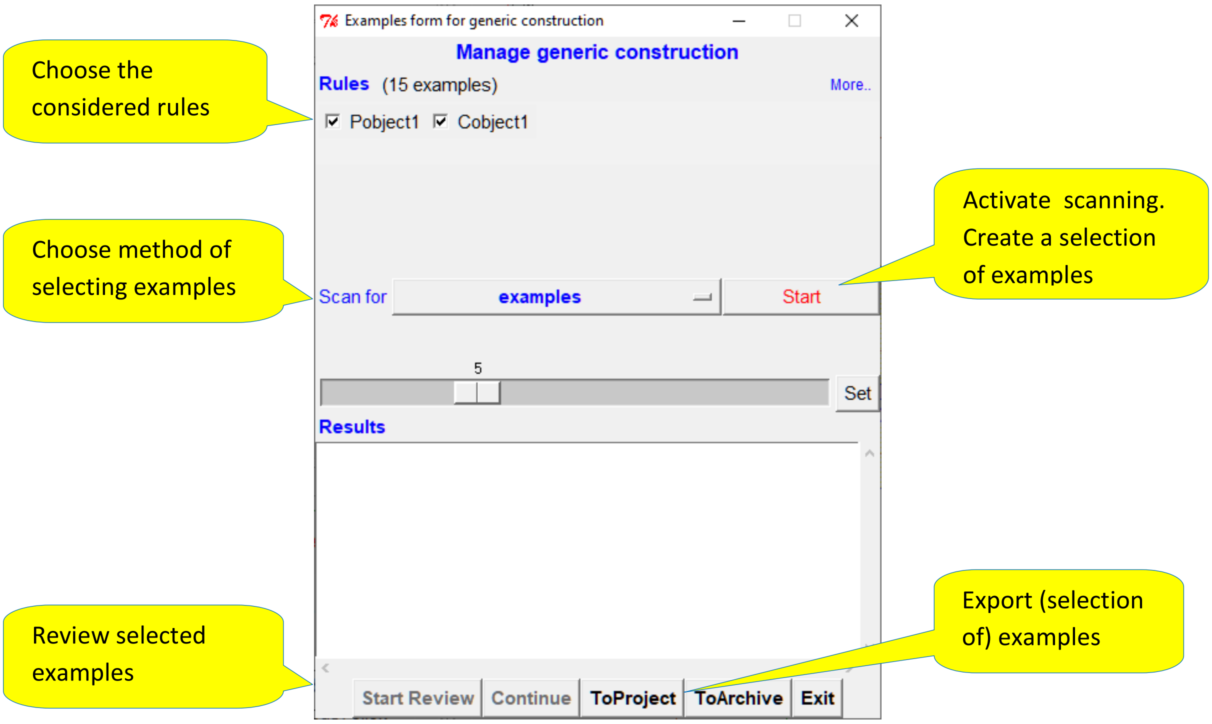


Figure 87

With the **Examples form,** we scan examples and create a selection from them. The available methods of scanning are:

- Scan for (all) examples,
- Scan for conditions,
- Scan for properties,
- Scan for GDD unproven facts.

The workflow is very similar for all methods, with a slight exception for the last one.

1. Open the construction and activate the **Examples form** by pressing the **Gen** button.
2. Choose the rules to consider in the set of examples.
3. Choose the desired scanning method.
   Fill in the optional entries.
   Then press the Start button.
   After a while (depending on the method), a selection of examples is created.
4. Access to the examples in the selection using the **Start Review** button followed repeatedly by the **Continue** button. (Reviewing examples differs when the GDD method is used.)
5. The selection of examples can be exported to the current project or to an archive.

The rules used and the indices of the examples in the selection can be checked. This is done with the ***More...|Rules and selection data*** command.

While reviewing examples, keep in mind that the displayed examples are de facto ordinary constructions, so you can work with them (perform observations, analyse them, export them, etc.) as described in Section 6.5. You can also modify the example, move, add or delete objects. **However, the changes to the example do not become part of the considered generic construction.** Any redisplay of the example (e.g. a click on the ***Set*** button) shows the original example.

We describe the scanning methods using the generic construction *Generic_03.p*[53] . In the following subsections it is assumed that this generic construction has been opened and the **Examples form** has been activated with the **Gen** button.

6.6.1 Scanning for examples

If the 'examples' method of scanning is applied, the resulting selection consists of all examples of the generic construction. The review displays examples from first to last. By default, degenerate examples are not considered, but you can include them by setting the ***More|Skip degenerate examples*** flag to Off.

6.6.2 Scanning for conditions

By condition, we mean a property relating to specific objects. In the considered construction (*Generic_03.p*), the requirement that the lines AA', BB', and CC' concur at a point, or that the line AB is parallel to the line A'B', is an example of a condition.

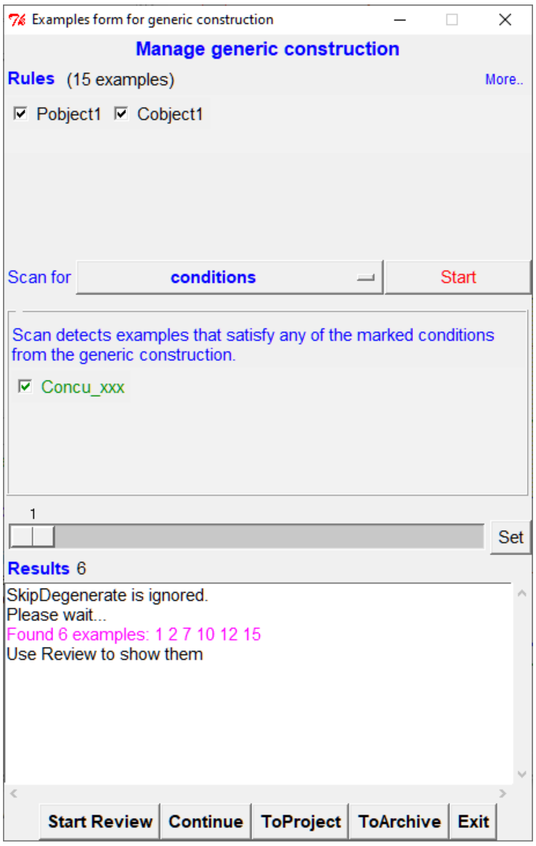


Figure 88

[53] OKExamples\OKG_Plus\Generic_03.p

The conditions to be scanned must be included in the generic construction, that is, they must be set in Sketch Editor using the command *Advanced|Check property.* In the considered generic construction we have defined the condition *Concu_xxx* as the condition of the concurrence of lines AA’, BB’, and CC’.

The ‘*Scan for conditions’* method displays all conditions set in the generic construction. (Figure 88). The obtained selection contains examples that satisfy any of the marked condition (even if they are degenerate). The obtained selection of examples can be reviewed in the usual way (see Section 6.6). You can delete the degenerate examples with the command ***More...|Rules and Selection data.***

6.6.3 Scan for properties

The method ‘*Scan for a properties*’ does not restrict itself to specific objects. It considers whether a given property is present anywhere in the example construction. If we want to know, which examples contain, anywhere in the example, four cocircular points, concurrent lines, etc., then the method to use is ‘*Scan for properties’*.

We detect examples with given properties as follows:

1. Choose ‘*Scan for properties’ as the scanning*. Some options and entries will appear.
2. Click on the ***Manage properties*** button.

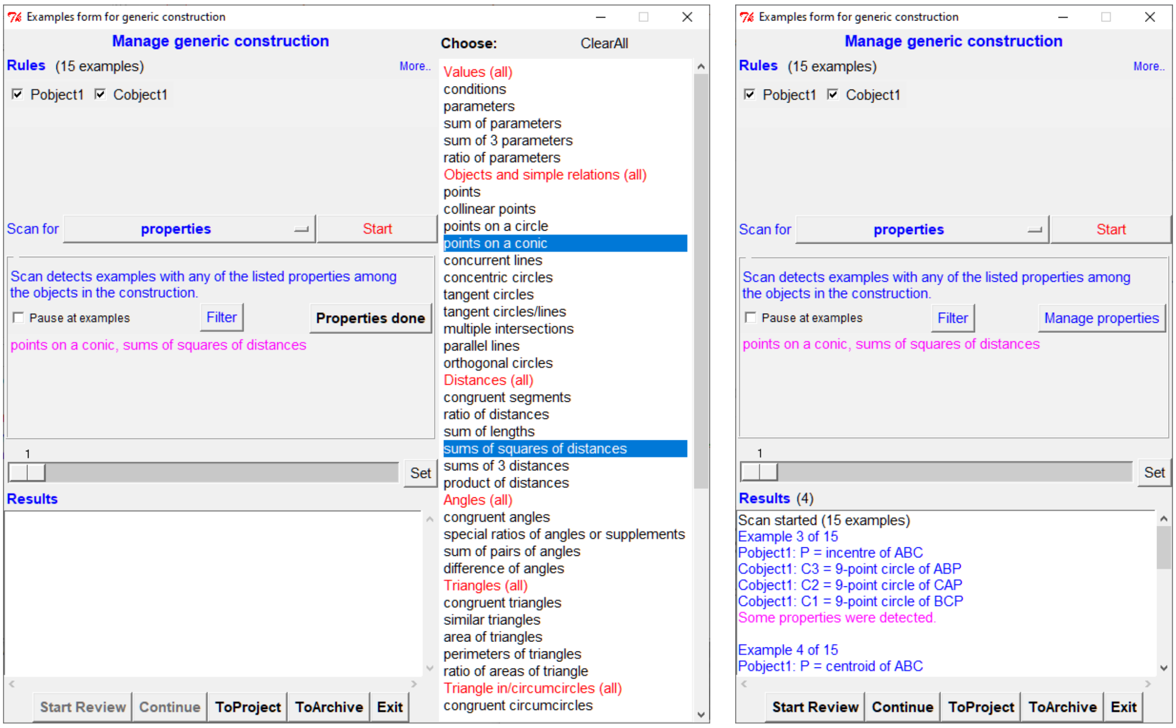


Figure 89

3. In the list of properties (Figure 89, left), select the properties you wish to observe. In our case we have selected 2 properties: if any 6-tuple of points lay on a conic, and whether any two pairs of squares of the distances between points in the example add up to same sum. (Figure 89, left). As you can see, the properties are divided into groups. If you select a property coloured red, you select all the properties in the group below it.

4. Once you have selected the properties, click the ***Properties done*** button (<u>otherwise you will not be able to continue</u>).
5. Then press the Start button.
6. Review the selected examples as described in Section 6.6. During the review, expand the section *Observed properties*. In this section you can see the fulfilled properties. A click on a property displays the respective property (Figure 90). By default, degenerate examples are not considered, but you can include them by setting the ***More|Skip degenerate examples*** flag.

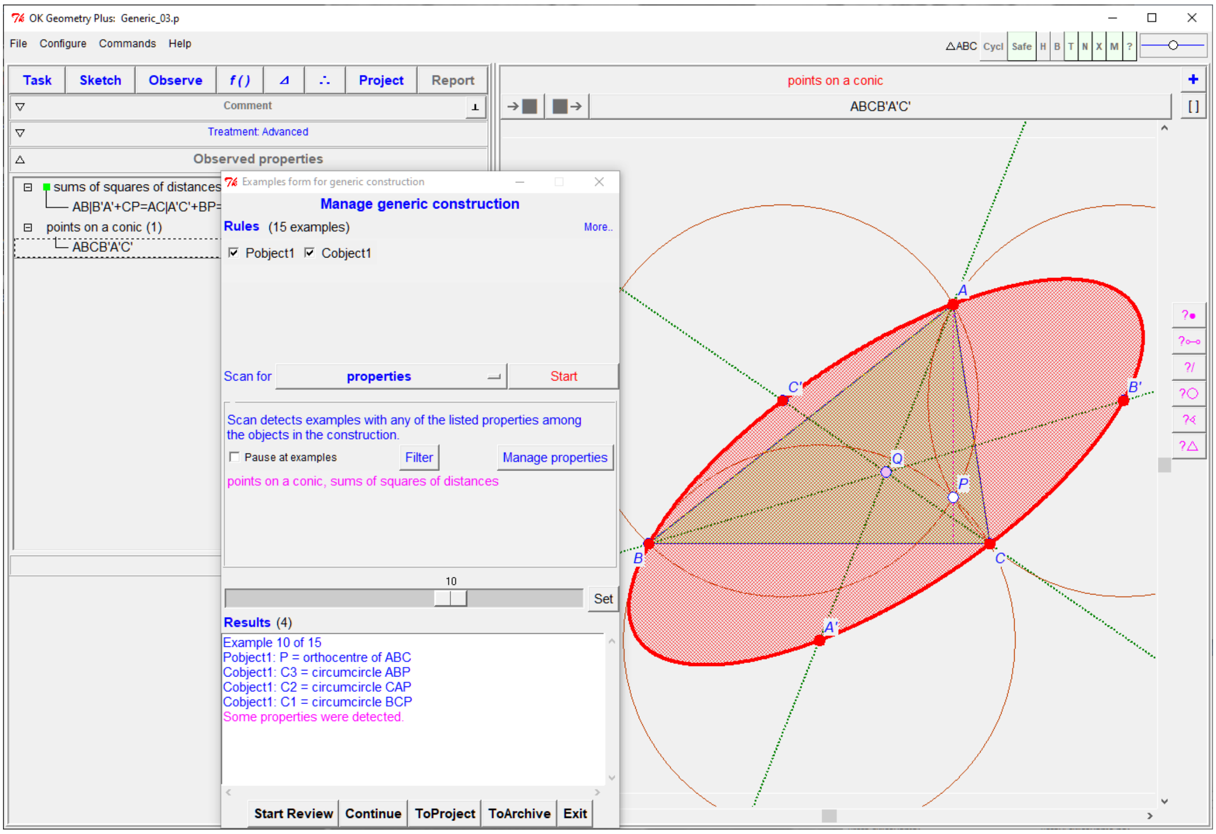


Figure 90

- While reviewing examples, keep in mind that you can work on them (perform observations, analyse them, export them, etc.) as described in Section 6.5. You can also modify the reviewed example, move, add or delete objects. **However, changes to the example do not become part of the considered generic construction.**
- The method '*Scan for properties*' has an option to pause at each found example.
- You can also set a filter, so that the resulting selection contains only properties in which given vertices are present.
- The method can take considerable time. You may wish to restrict the observation to a subset of points in the *Observe properties* section.
- The properties in the list of observations (Figure 89, left) are to some extent self-explanatory. Nevertheless, there are some peculiarities that are not self-evident:
    - Properties based on geometric quantities or numerical values (e.g., parameters, congruent segments, congruent angles, area of triangle, congruent circumcircles,

congruent incircle) take into account not only equality of the quantities but also many relationships between triplets of quantities (e.g. the radius of a circle is equal to the sum of radii of two other circles)

- Properties based on the ratio of two geometric properties (e.g., ratio of parameters, ratio of distances, ratio of angle sizes, ratio of areas of triangles, ratio of radii of circumcircles, ratio of radii of incircles) consider not only the equality of two ratios, but also whether a ratio is observed to be constant, as well as some relationships between triplets of ratios.
- The property 'points' observes whether a point is a 'simple' centre of any triplet of labelled points in the configuration.

6.6.4 Scan for GDD unproven properties

This method of scanning examples is intended to identify in all examples th properties that are difficult-to-prove. Because it is handled differently, we discuss this method in a separate section (see Section 6.10).

6.7 Group analysis of examples in generic construction

A generic construction may contain many examples. Analyzing each example individually (see Section 6.4.2), or only part of them, requires a great deal of patience. With group analysis, we can perform a triangle analysis or a formulae observation on all selected cases, or on all examples of a generic construction, at once.

Group analysis is carried out in the following steps:

1. Create (or open) the generic construction.
2. Activate the **Examples form** (click the **Gen** button on the toolbar to the right).
3. Unless you intend to analyse all examples of the generic construction, create the selection of object you intend to analyse. If possible, choose the '*Scan for conditions*' method, since it is fast.
4. Activate the **Triangle analysis** module (the button with the triangle in the main menu in the upper left corner of the display) or the **Formulae observation** module, depending on the required type of analysis.
5. Carefully enter the necessary data in the form that appears. Pay attention to the entry '**Examples**' in either analysis. Set this entry to 'all' or 'selection', according to your intention.
6. Execute the analysis.

The analysis may take considerable time, and the output can be very long. Therefore it is advisable to avoid exhaustive output.

We illustrate group analysis using the generic construction *Generic_03.p* (Figure 101). Recall that it contains a rule for point P (5 different centres of the triangle ABC) and a rule for the three circles (circumcircle, incircle, 9-point circle) applied to triangles BCP, CAP, and ABP. Altogether, it contains 15 examples.

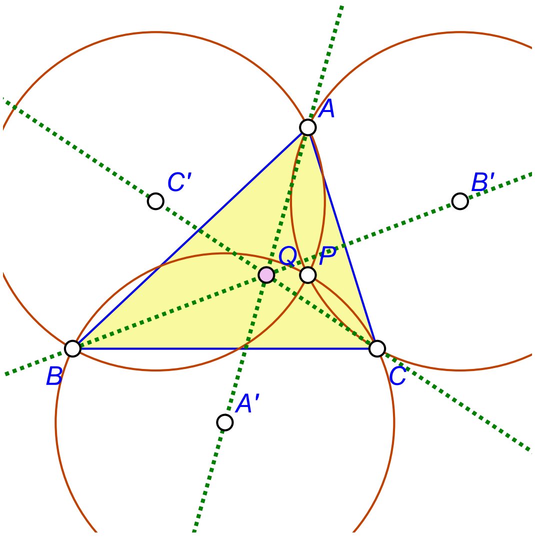


Figure 91

We have found that in some examples the lines AA', BB', and CC' concur at the point Q. We investigate the position Q wrt. ABC in these examples. We proceed as described above.

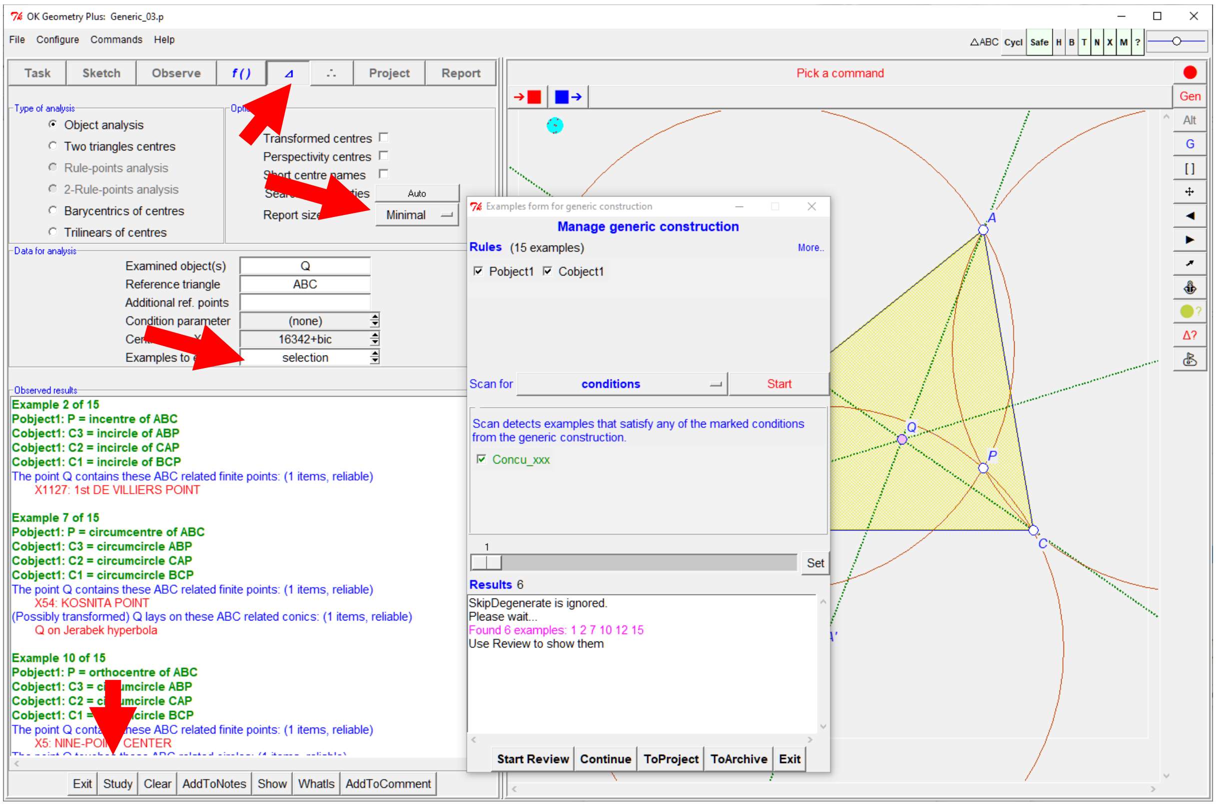


Figure 92

a. Open the file *Generic_03.p* and activate the **Gen** button.
b. In the **Examples form** checkmark the Pobject1 and Cobject1 rules.
c. Choose the '*Scan for condition*' option, checkmark the *Concu_xxx* condition, and press Start to generate the selection of examples where AA', BB', and CC' concur at Q.

d. (Review the examples in the selection (**Start Review** + **Continue**) to check for degenerate examples in the selection. It turns that Example 1 and Example 12 are degenerate.)

e. Open the triangle analysis module. Ente the data for Object analysis. Choose 'selection' for the Examples option. To avoid a long report, choose the *Minimal* or *Short* report size.(Figure 92).

f. Press Study to generate the report about Q in ABC for all examples in the selection.

We proceed analogously for the ***Observe formulae*** tool***.*** Suppose we are investigating the radius of the incircle of the triangle A'B'C' for the examples where AA', BB', CC' concur.

e. Open the ***Observe formulae*** module. Enter the data for Object analysis. Note that ABC is the reference triangle and that we are studying the quantity *ri(A'B'C')*. Choose 'selection' for the Examples option.(Figure 92).

f. Press Observe in the ***Observe formulae*** form to generate the report on *ri(A'B'C')* in ABC for all examples in the selection. Reasonable algebraic expressions are found in case of examples 7 and 10, but not in case of examples 2 and 15.

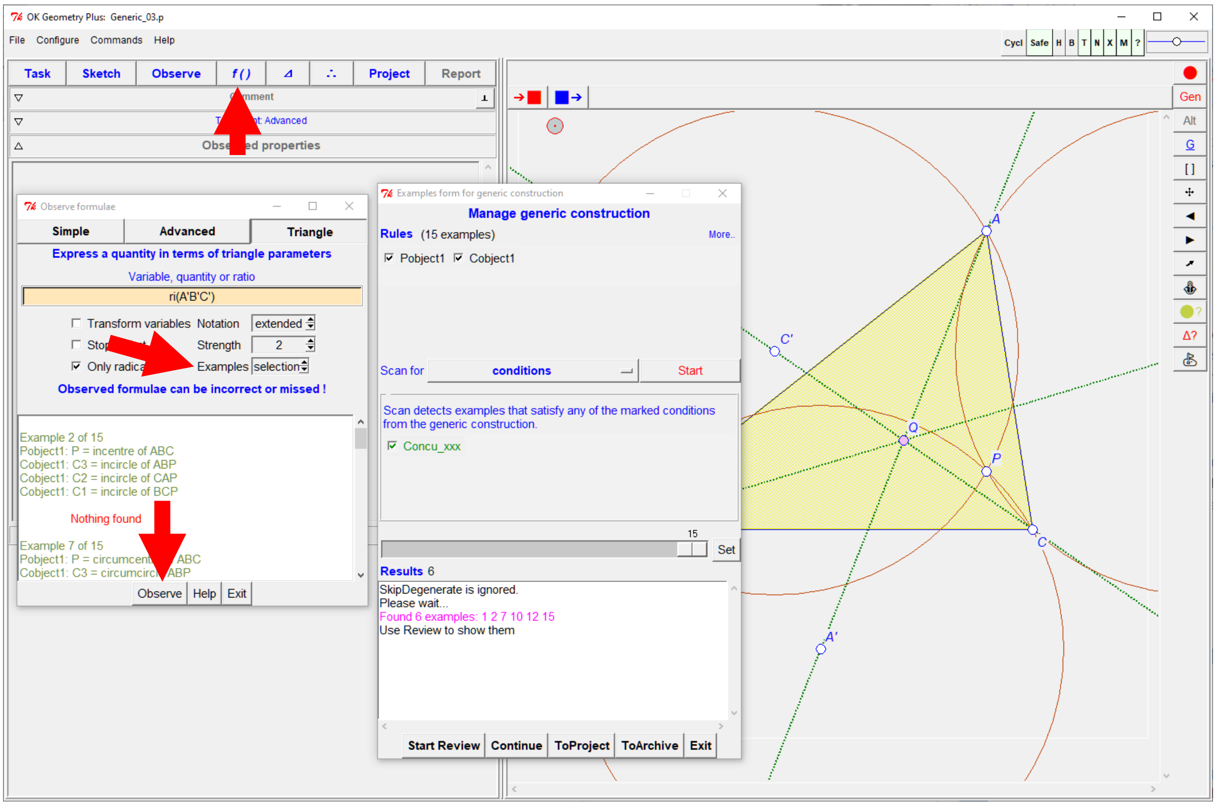


Figure 93

6.8 Overview of the groups of rules

The rules in the Generic menu are organised into groups. All rules create new objects from points. The names of the rules in each group are fixed; the first letter of a name indicates the type of object created by the rule (P – point, L – line, C – circle, S – conic, T – triangle (and triangle objects), Q – quadrilateral (amd quadrilateral objects)). For example, the rules for creating points are Pobject1, Pobject2, Pobject3, and so on. An exception are the rules ETC1, ETC2, ETC3,.. for ETC triangle centres.

6.8.1 Rules for creating simple objects

The commands for creating generic points (*GPoint*), lines (*GLine*), and circles (*GCircle*) are found at the top of the **Generic** menu commands. When one of these commands is selected, a form appears (Figure 94). We will examine the form for creating a rule for points in detail, since the forms for creating lines and circles are self explanatory.

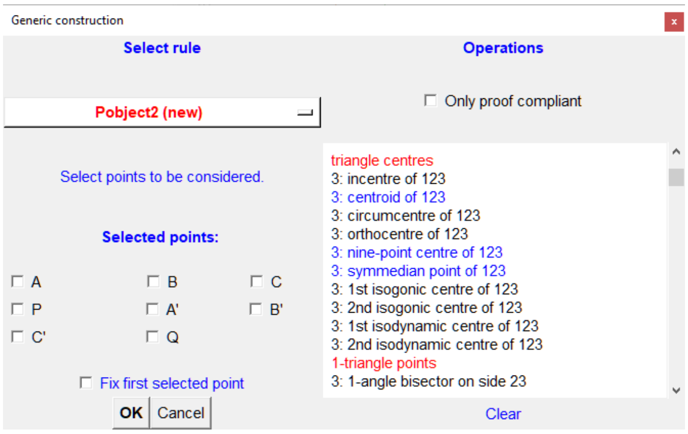


Figure 94

The list of operations contains red items (section names), black items (proof-compliant operations), and blue items (proof-noncompliant operations). The number before each operation indicates the number of point-arguments, denoted by the digits 1, 2, 3, etc.

Some sections in the list of GPoint command require further explanation.

3rd vertex. The arguments (2 green coloured points in Figure 95) are considered the base vertices A and B of a triangle. This group of commands creates the third vertex C (red coloured point) of a positively oriented triangle. The position of the third vertex is described (up to direct similarity) in terms of angles A, B, C and side lengths *a, b, c*. Some common names are also allowed as descriptions of the resulting triangle: equilateral, 3-4-5 (i.e. a triangle similar to a right triangle with sides 3,4,5), isosceles, etc.

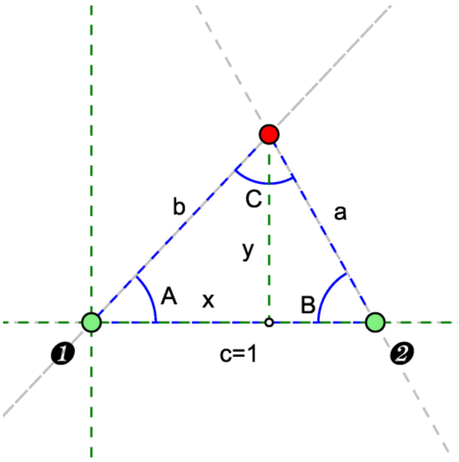


Figure 95

ETC triangle centres and related points. In the list of operations for *GPoints*, there is a group of operations that yield the first 19 ETC centres from three given points as triangle vertices 1, 2, 3. The 1-cevian point of a centre, say, X3, is the central projection of X3 from vertex 1 onto the line 23. Figure 96 explains some similar constructions: 1- pedal, 1-circumcevian, 1-circlecevian.

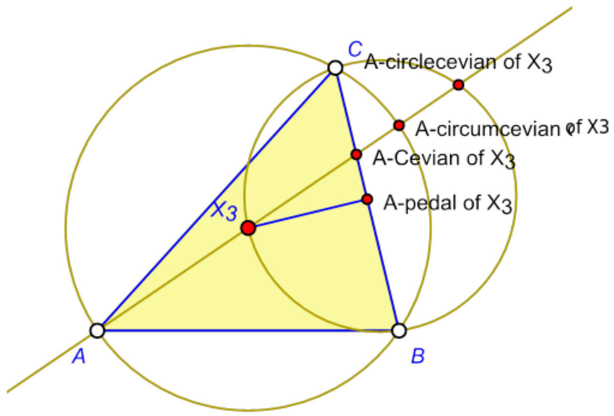


Figure 96

6.8.2 Rules for triangle and quadrilateral on a segment

The *Generic|Triangle on segment* command creates a rule that generates various types of positively oriented triangles with two given vertices. The rules for triangles are named Tobject1, Tobject2, and so on. The entries in the list of operations for triangles are the same as for *3rd vertex,* except that the sides of the triangle are also displayed. The command *Generic|Quadrilateral on segment* creates a rule that generates various types of positively oriented quadrilaterals with two given adjacent vertices. The rules for quadrilaterals are named Qobject1, Qobject2, and so on.

The rules for triangles and quadrilaterals on a segment are ordinary rules and can be used in constructions just like other rules. When studying the properties of various kinds of triangles or quadrilaterals, it is useful to begin the generic construction with two points and a rule for a triangle or a quadrilateral.

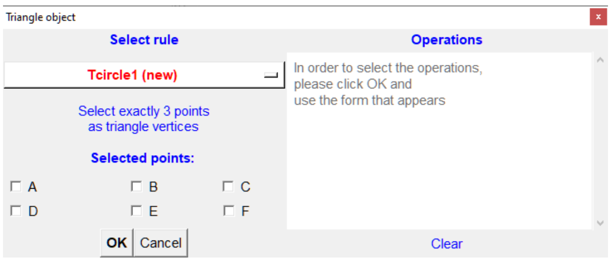
An example with a detailed description of the use of rules for triangles and quadrilaterals on a segment can be found in Section 6.9.

6.8.3 Rules for triangle and quadrilateral objects

It is possible to create rules for all kinds of triangle objects. These rules are generic versions of most commands for creating triangle objects in the Sketch menu *Special.* The rules for creating ETC centres are named ETC1, ETC2, and so on, the rules for triangle lines are named Tline1, Tline2, and so on, for circles Tcircle1, Tcircle2, and so on.

All rules in this group require exactly three points as the vertices of a triangle, and possibly one or more additional point as arguments (for example, the Cevian triangle of a point). All rules in this group are obtained in a very similar manner. We illustrate the process using the rule for triangle circles. We activate the command *Generic|Triangle objects|Triangle circles.* In the form that appears (Figure 97, left), we checkmark exactly three triangle vertices. We can use a previously declared rule or choose to declare a new rule. In the latter case, after clicking OK, a second form appears (Figure 97, right). In this form, we select the type of circles to be included in the rule and click OK.

We proceed similarly in the case of quadrilateral objects, taking into account that a quadrilateral has four vertices and that some quadrilateral objects depend on the order of the selected vertices.



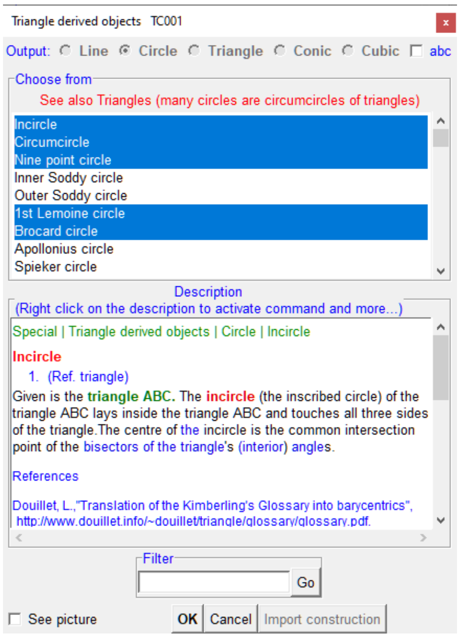


Figure 97

6.8.4 Editing rules in generic construction

Creating a generic construction requires some preparation, since the user has limited control over the rule once it is declared. We present some notes on this here.

It is important to bear in mind that a generic construction cannot be modified when the Examples-form is displayed (see Section 6.4). The rules of a generic construction can only be changed when the Examples-form is not displayed.

The command *Generic|Manage rules* allows modification of the selected operations for a rule. It is possible to delete a rule, but note that deleting a rule does not delete the object(s) created by the rule in the initial example. Similarly deleting an object, created by a rule, does not delete the rule, only the object.

6.8.5 Editing rules locally

On the **Examples form** you may have noticed the ***More...*** button in the upper right corner, which provides access to a few commands.

The command **Manage rules (locally)** allows you to manage rules. This appears to be the same command as in the *Generic* menu, but there is one important difference: with this command you modify rules only locally. When the **Examples form** is closed, the rules revert to their previous stare , i.e. when the **Examples form** was opened. This makes possible to change the rules locally while working on examples, without altering the generic construction.

The third command **Set examples to Review** allows you to overview or set the examples to be reviewed with the StartReview and Continue buttons.

6.9 Examples of generic constructions

Example 1

Consider in a quadrilateral ABCD the incentres A', B', C', D' of the triangles BCD, CDA, DAB, ABC. Is A'B'C'D' a cyclic quadrilateral (see Figure 98)? Is it cyclic for certain special quadrilaterals? What if A', B', C', D' are other triangle centres?

A generic construction can help to answer these questions at an observation level. For this purpose, we create a generic construction that considers different types of quadrilaterals and different centres of the four triangles, for example X1 to X5.

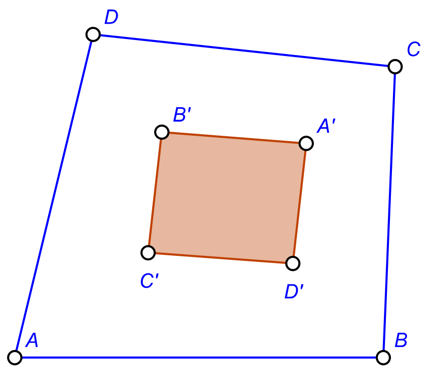


Figure 98

There are several ways to approach this task. Here we show the method based on using *Scan for properties* for selecting examples. Below is a detailed description of the entire process, including the creation of the generic construction. The resulting generic construction is in the file *Generic_04.p*[54].

1. We begin by declaring a rule *Qobject1* for quadrilaterals.
   - Draw points A and B.
   - Activate the command *Generic|Quadrilateral on segment.*
   - Complete the form: checkmark points A and B and choose the types of quadrilateral. In our case we selected the following quadrilaterals: random, bicentric, cyclic, diametric, and tangential.
   - Then click OK.
2. Next, we declare the rule *Pobject1* for the triangle centres X1 to X5.
   - Activate the command *Generic|GPoint.*
   - In the list of operations select *ETC triangle centre X1*, ... to *ETC triangle centre X5.*
   - Checkmark the points B, C, D. (In this case, the order is irrelevant, as is the FixFirst status.)
   - The rule *Pobject1 (new)* is declared after you press the button OK.
   - The incentre of triangle BCD appears. Label it A'. (A' is the 'variable' point created by the rule *Pobject1* in triangle BCD.)
3. We apply the rule *Pobject1* to the triangles CDA, DAB, and ABC.
   - Activate the *Generic|GPoint* command*.*
   - Select the rule *Pobject1.* A list of previously selected operations appears.
   - Checkmark the points C, D, A. (Again, the order is irrelevant, as is the FixFirst status.)
   - Press OK. The incentre of triangle CDA appears. Label it B'.
   - Repeat these steps for the triangles DAB and ABC to obtain points C' and D'.
4. The generic construction is now ready. To begin the observations, click the **Gen** button on the toolbar to the right of the display.
   - In the form that appears, checkmark both rules.
   - Choose the option *Scan for properties.* Click on the Manage properties button. Select the red coloured Shapes (all) entry**.** This entry includes the observations of the shapes of triangles and quadrilaterals. Complete the form as shown in Figure 99.
   - Click the **Properties done** button (otherwise you will not be able to continue).
5. We are interested only in the shape of quadrilateral A'B'C'D', so it is important to restrict the observations to these points. Therefore, in the **Treatment** section of the observation pane fill in the entry **Consider only points** with A'B'C'D'.
6. Press Start button for scanning all examples.
   - Press **StartReview,** and then repeatedly press **Continue** to review the results of the observations. You can see the observed properties of each example found in the **Observed properties** pane.

[54] OKExamples\OKG_Plus\Generic_04.p

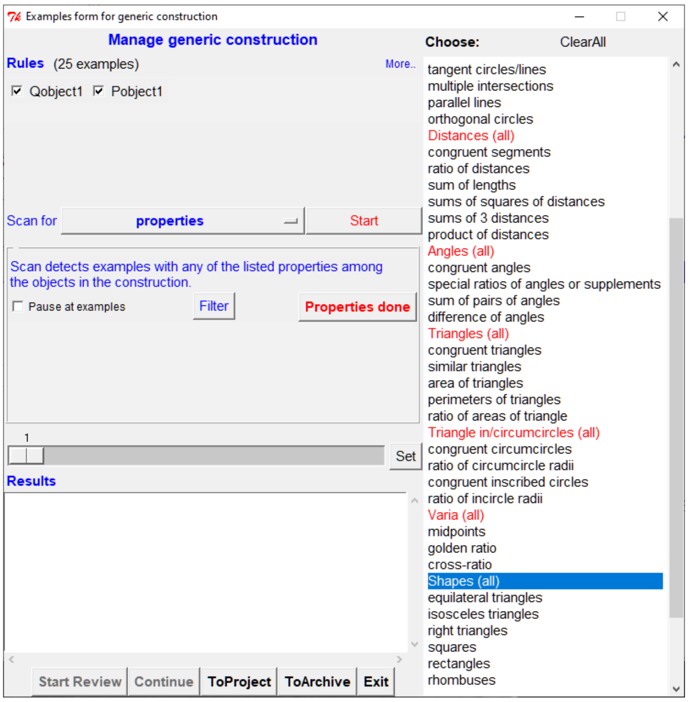


Figure 99

Example 2

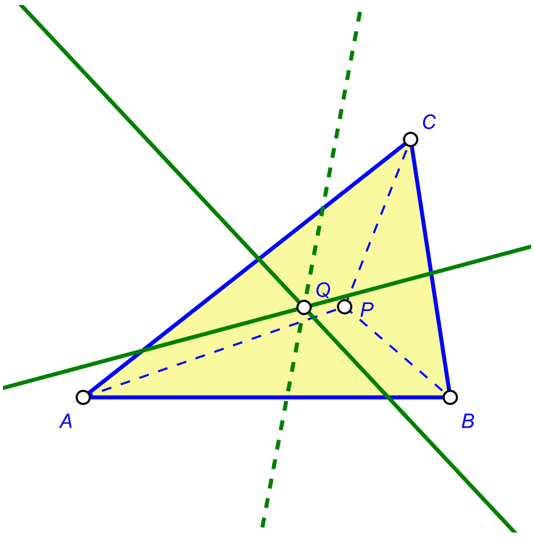


Figure 100

Let P be the incentre of a given triangle ABC. It is well known that the Euler lines of the triangles PBC, PCA, and PAB concur in a point Q. Let us explore whether a similar concurrence occurs when P is a different triangle centre and when a different type of triangle lines is considered.

We present a brief description of how to create and analyse the generic construction for this investigation. You can download it as the file *Generic_05.p*[55].

1. Construct a triangle ABC. (You can use the command *Special|A triangle*)
2. Use the *Generic|GPoint* command to declare the rule for the ETC centres X1 to X5 of the triangle ABC. On the displayed base example the point P (as incentre X1) appears.
3. On the base example create the line segments PA, PB, and PC.
4. Use the *Generic|Triangle objects|Triangle lines* command to create the rule Tline1 for triangle lines of the triangle with vertices P, B, and C. Select all available lines as operations. When you are done the Euler line of the triangle PBC appears.
5. Using the *Generic|Triangle objects|Triangle lines* command, apply the Tline1 rule to the triangle PCA and then to PAB. The Euler lines of these triangles should appear on the screen.
6. Construct a point Q as the intersection of two of the constructed (Euler) lines. If the three (Euler) lines concur then Q is the point of concurrence.
7. Construct a condition *concur_xxx* for concurrency of the three lines. Use the command *Advanced|Check property|ConcurrentL.*
8. Press the **Gen** button. The **Examples form** appears. Checkmark both rules (Tpoint1 and Tline1). Also checkmark the *Concu_xxx* condition. For Scan mode, choose the option *Scan for conditions*. Checkmark the *Concu_xxx* condition.
9. Press the button Start to scan examples**.** Press the **StartReview,** and then the **Continue** button to view and analyse each case in the result list. Note that in some of the found examples the points P and Q coincide, so they appear as degenerate cases

Example 3

Let D and E be two ETC centres of a given triangle ABC (Figure 101). Does it happen for some centres D and E that the lines AD and CE are perpendicular? If not for general triangles, does it occur for some special types of triangles?

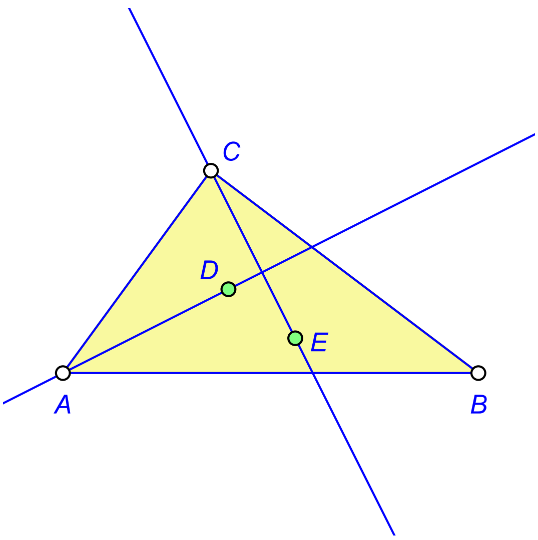


Figure 101

[55] OKExamples\OKG_Plus\Generic_05.p

To investigate this we make a generic construction[56] (file *Generic_06.p*). We consider the centres X1 to X29 for D and E. We examine the following types of triangles: a random triangle, a triangle with angle C=60 degrees, a triangle proportional to the 6,9,13 triangle, and a triangle where a=2b.

Here is an overview of the construction steps:

1. Draw the points A and B.
2. Use the *Generic|Triangle on segment* command to create the rule for a triangle with A and B as two vertices. In the form choose the following 3rd vertices operations: random, C=60, a:b:c=6:9:13,a:b=2.
3. Use the command *Generic|Triangle objects|More ETC centres* to create the rule ETC1 for point D. Checkmark the points A,B,C and select the centres X1 to X29. The point D appears as the X1 centre of ABC.
4. Use the *Generic|Triangle objects|More ETC centres* command to create the ETC2 rule for point E. Checkmark points A,B,C and select the centres X2 to X29. The point E appears as the X2 centre of ABC. Note: Two separate rules are needed for the points D and E. The selected operations start from X2 to avoid overlapping of points D and E in the base example.
5. Construct lines AD and CE. Use the *Advanced|Check property|PerpendicularL* command to create the condition parameter Perpe_XX for the perpendicularity of AD and CE.
6. The generic construction is ready. It contains almost 3500 examples. To check the condition of perpendicularity among them, press the button **Gen**.
7. In the **Examples form** checkmark all the rules as well as the *Perpe_xx* condition. For scanning mode, choose the *Scan for condition* option. Then press Start button. OK Geometry observes that 3 examples satisfy the condition.
8. Use **Start Review** and **Continue** to identify them.

[56] OKExamples\OKG_Plus\Generic_06.p

6.10 Generic constructions and observing proofs

A generic construction allows GDD proving only if the construction steps are proof-compliant, including the operations in rules. When creating generic constructions where proofs are to be used, it is therefore important to restrict the rules to proof-compliant operations. You do this by check marking the *Only proof-compliant* entry in the rule creation form (Figure 102). As you may have noticed, the proof-compliant operations are in black, while noncompliant ones are in blue.

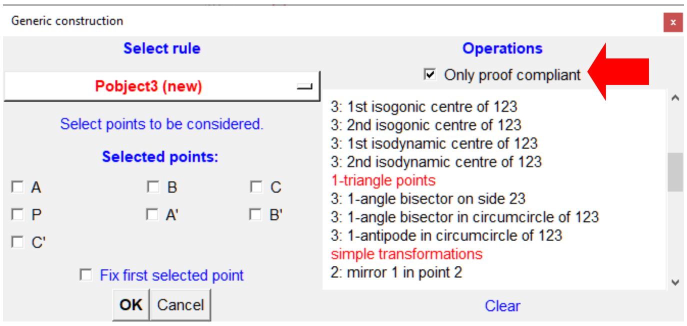


Figure 102

When the **Examples form** is active, we can prove the properties of the current example as with ordinary constructions. We simply activate the prover with the  button and prove the considered property. The prover works exactly as described in Section 5, with the only exception that it does not suggest adding a point to the construction if the proof is not found.

OK Geometry also uses the prover as a means of searching for non-trivial properties in examples.

Suppose we want to find interesting cases of collinearity of points (or some other property) in examples of a generic construction. If we scan the examples for the existence of ‘collinear points’, we almost certainly encounter difficulties, since collinearity is likely to occur multiple times in many examples. Moreover, most cases of collinearity found in the examples turn out to be trivial and uninteresting. This is where the prover comes into play. With the prover, we eliminate trivial cases of collinearity (or any other GDD property) by selecting cases of property that the prover cannot prove or cannot prove 'quickly'. We perform such a search for all properties (considered by the prover) on all non-degenerate examples in the generic construction. This can take considerable time, which is why the proving process runs separately from the ongoing work in OK Geometry. In the end, we are rewarded with an archive containing a small set of interesting, non-trivial properties in individual examples of the studied generic construction.

For reasons of time efficiency, the prover assumes that all properties observed before the first rule are true.

Example 4

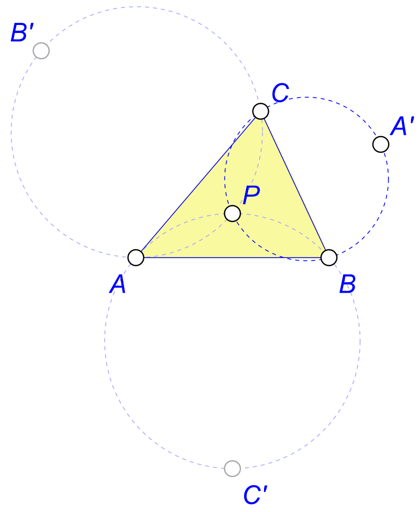


Figure 103

Consider the following generic construction:

Given is a triangle ABC. Let P be one of the following centres: incentre, circumcentre, orthocentre. For each case, let A' be obtained from P, B, and C in one of the following ways:

A' is the antipode of P in the circumcircle of PBC;

A' is the angle arc-bisector of P in PBC;

A' is the circumcentre of PBC;

A' is the incentre of PBC;

A' is the mirror image of point P in the bisector of segment BC;

A' is the mirror image of P in line BC;

A' is the mirror image of P in the midpoint of segment BC;

A' is the ortho-translation of P by vector BC;

A' is the orthocentre of triangle PBC.

For each of these cases, define B' and C' cyclically.

Let us investigate the non-trivial properties of these configurations.

The description clearly corresponds to a generic construction[57] with two rules. The first rule defines the point P with regard to ABC; the second rule defines A' with respect to PBC, B' with respect to PCA, and C' with respect to PAB. In declaring the rules we need to pay attention to some details, as shown in Figure 104. In both rules we used only proof-compliant commands. In the second rule we fixed the first vertex in the triplets of vertices PBC, PCA, and PAB.

[57] OKExamples\OKG_Plus\Generic_07.p

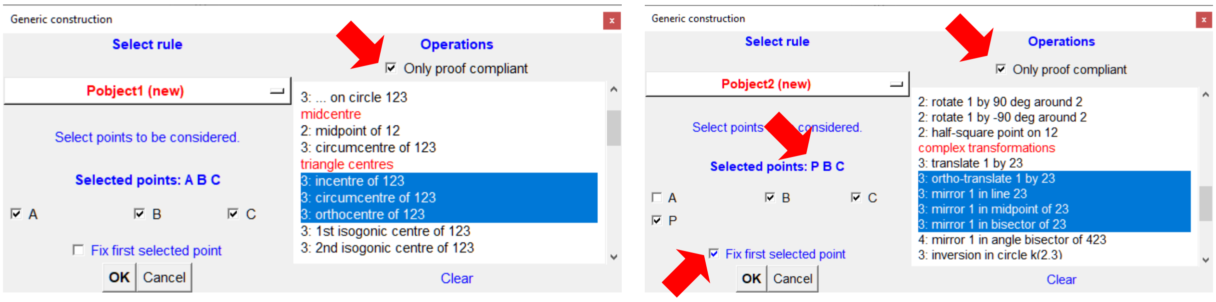


Figure 104

The construction has an evident cyclic structure, therefore we declare the *Invariant cycle 1* as ABC A'B'C'. To detect non-trivial properties in the examples in the resulting generic construction[58], proceed as follows:

1. Activate the **Examples form** by clicking the **Gen** button in the toolbar to the right. This is the usual way to start this form.
2. In the **Examples form,** checkmark the rules you wish to activate. In our case, we have selected both rules, *Pobject1* and *Pobject2*. We observe that the rules give rise to 33 examples.
3. Select the option *Scan for GDD unproven facts* as method of scanning (see Figure 105).

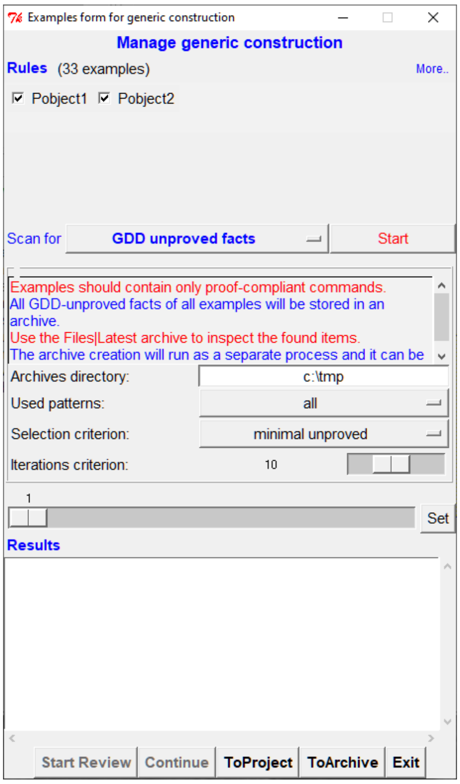


Figure 105

Bear in mind that the prover will attempt to prove all properties in all examples of the generic construction and will keep track of properties in examples that are too difficult to prove. In the next step, we therefore set the parameters for this process (see also Section 5.4):

[58] OKExamples\OKG_Plus\Generic_07.p

Used patterns refers to the available set of patterns for the proof. Choose between **simple** (elementary facts, without proportionality), **advanced (**advanced facts, including proportionality), **all** (all patterns, including homothety, Ceva and Menelaus' theorem)**.**

GDD archives directory is included in the configuration of OK Geometry (it can also be set in *Configure|General options|General options*). This is the directory where the prover creates archives. The created archives are named as xxxxxxxxxxxxxxx.*arh* (where x..x are digits reflecting the current time). The latest created archive can be accessed with the command *File|Latest archive*.

Archive criterion (iterations). A property meets this criterion if the GDD prover does not prove it in fewer than the specified number of iterations . Note that iterations are not the same as steps in the proof. An iteration can result in one or several steps in the proof. The default value for this criterion is 10. Very simple proofs require 3 or 4 iterations, most high school geometry problems are usually solved in 5 or 6 iterations.

Archive criterion (selection). This entry specifies which entries among those that meet the iteration criterion are to be included in the archive. The options are:

1. **All unproven**. The archive will contain all properties that require at least the specified number of GDD iterations to be proved. The resulting list of properties can be overwhelming and difficult to use.
2. **Different unproven**. The list of 'all unproven' properties is reduced by omitting properties that can be derived from previously listed properties using equivalence of relations. For example, if P, Q, R are collinear and the list contains the property 'PQ is parallel to UV', then the list would not contain the properties 'PR is parallel to UV 'and 'QR is parallel to UV'. If the configuration has a declared cyclic structure (see *Invariant cycle* entries in the *Observed properties* section of the left-hand pane), this option also avoids repetitions of cyclically equivalent properties. This can significantly reduce the number of archived properties.
3. **Minimal unproven**. In this case, the prover adds new properties to the archive sequentially. At each step, a property is added only if it cannot be GDD-derived from the previously proven or previously added properties. If the configuration has a declared cyclic structure (see *Invariant cycle* entries in the *Observed properties* section of the left-hand pane), this option also avoids repetitions of cyclically equivalent properties. This method significantly reduces the number of properties in the archive. This is the default mode for selecting unproven properties.

Note. To avoid an excessive number of properties in the archive, **only properties that contain the last created point in the construction** (in this case, the point C') are stored in the archive. This makes sense, as properties without the last added point can be produced with a simpler generic construction. In addition, this option avoids repeating properties in declared cyclic constructions.

Now click on the *Start* button. After some time, the selection is started in a separate window (Figure 106). As this process can take a considerable amount of time (depending on the complexity of the generic construction and the number of examples it contains), it runs independently in a separate window. You can stop the process at any time by simply closing the window.

```
Example  1 /  30  Found  1  AllFound   1  Time 0.203
Example  2 /  30  Found  1  AllFound   2  Time 0.281
Example  3 /  30  Found  3  AllFound   5  Time 0.219
Example  4 /  30  Found  1  AllFound   6  Time 0.078
Example  6 /  30  Found 11  AllFound  17  Time 5.110
Example  7 /  30  Found  0  AllFound  17  Time 0.250
Example  8 /  30  Found  0  AllFound  17  Time 0.063
Example  9 /  30  Found  0  AllFound  17  Time 0.062
Example 10 /  30  Found  0  AllFound  17  Time 0.141
Example 11 /  30  Found  0  AllFound  17  Time 1.031
Example 12 /  30  Found  3  AllFound  20  Time 0.063
Example 13 /  30  Found  0  AllFound  20  Time 0.500
Example 14 /  30  Found  8  AllFound  28  Time 0.109
Example 15 /  30  Found  3  AllFound  31  Time 0.547
```

Figure 106

All examples of non-trivial properties found are stored in the newly created file x...x.*arh* in the archives directory. The archive can be inspected with the command *File|Latest archive*. The examples found can be sorted in various ways, and each can also be visualised (Figure 107). When viewing individual examples, the *Comment* section contains descriptions of the rules used in the example. Please, see Section 7.3 for more details.

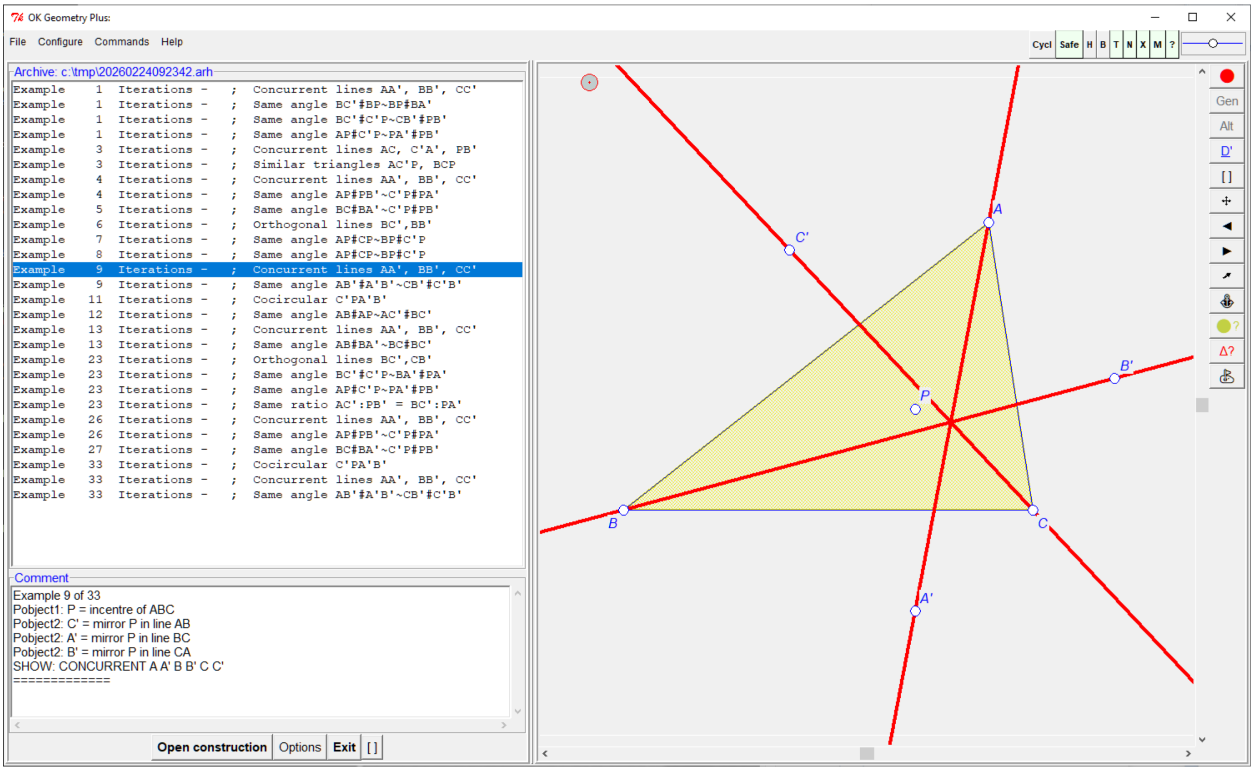


Figure 107

7 Archives of constructions

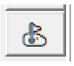
7.1 Introduction

Archives are special files that store a large number of constructions in a compact way. Archives are created and possibly updated automatically; the user can only view the constructions in the archive, visualise them, and export them as a regular construction file or as an icon in the current project. Archives become relevant when dealing with a large number of constructions, which are often automatically generated. Such situations arise when working with generic constructions (e.g. lists of constructions that satisfy given conditions), when proving (e.g. lists of constructions and properties that are not GDD provable), but also in ordinary work, where is a need to access to constructions previously worked on and perhaps not stored. The archive past constructions (often not saved in files) is called the **Work archive** and somewhat special. All the other archives are often created automatically and are, by default, contained in the **Archives directory.**

7.2 Work archive – the archive of recent constructions

During work, constructions created in OK Geometry can be archived automatically and also manually in a Work archive. You specify the path to your **Work archive** file in the *Configure|General options|General options|Work Archive* form. The recommended name for the Work archive is WorkArchive.arh or WorkArchive.txt. If you wish (for example if the Work archive becomes too large), you set a new file as theWork archive. If no Work archive file is not specified, constructions that created during work are not archived.

Each time you use the command *New construction*, *New Project* or exit from OK Geometry, the **current construction** (not the current project!) is archived in the Work archive.

You can also archive the current state or variant of a construction while you working on it by clicking the *Snapshot to archive* button  in the toolbox on the right-hand side of the display. In this case, you can add a tag and comments to the archived construction.

You can view and restore the archived constructions with the command *File|Work archive*. Inspecting the work archive displays the list of the constructions in it. You can see

- the date of archiving;
- an optional indicator (before the semicolon): **G** stands for a generic construction, **O** means that the construction contains an optimisation procedure;
- an optional tag, such as Work, etc.;
- anoptional archiving title.

You can perform various operations with the list of archived files by right-clicking on the list. In the archive, you can, for example, find previous geometrically similar constructions to the current construction or to a selected archived construction. The selected archived construction becomes the current construction when you click the ***Open construction*** button.

The archive is useful for restoring constructions from previous work and for quickly saving ideas that arise during work.

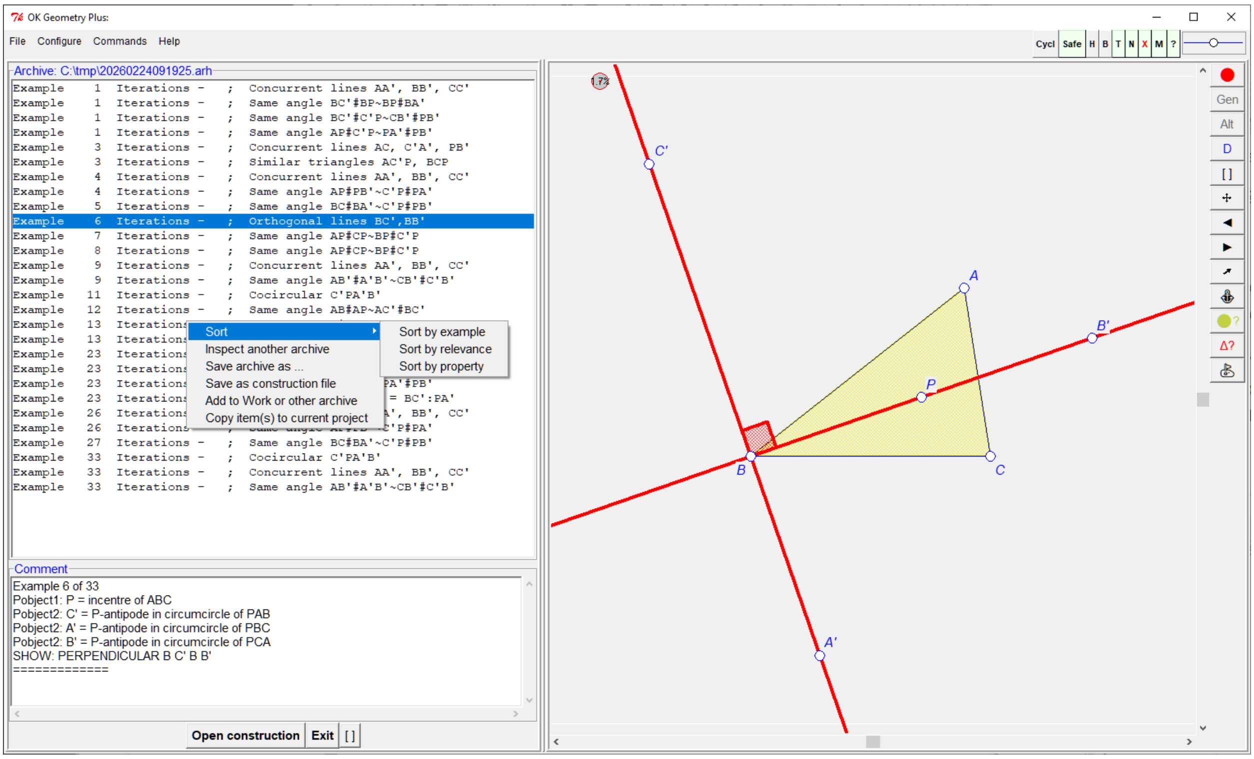


Figure 108

7.3 Other archives

When using the GDD prover, you can create an archive of GDD unproven facts for the studied construction. A similar, but more extensive, archive is created when you scan a generic construction for GDD unproven properties across all examples of the studied generic construction. Archives can also be created to store a selection of examples of a generic construction, obtained by any scanning method.

By default, archives are stored in the **Archives directory**. You specify this directory in the *Configure|General options|General options|Archives directory* form. The name of an archive reflects the time of its creation and is set automatically. For example, the archive named 20260205085314.arh was created on 2026.02.05 at 08:53:14. You can access to the most recently created archive with the command *Files|Latest archive*. Of course, you can rename or copy the archive, especially if it contains relevant data.

Inspecting the work archive displays a list of the constructions it contains. Depending on how the archive was created, you may see:

- the Example number of the example in the generic construction;
- the required number of iterations in the GDD proof;
- the date of archiving;
- the property related to the archived construction.

You can perform various operations with the list of archived files by right-clicking on the list (Figure 108). In the archive you can, for example, find previous geometrically similar constructions to the current construction or to a selected archived construction. The selected archived construction becomes the current construction when you click the ***Open construction*** button.