OK Geometry Extensions

Reference for OK Geometry Plus Extensions (v.22)

Zlatan Magajna

February 2026

1. Activation of extensions

2. ETC_information

3. Proof – adding points manually

Fortunately, OK Geometry contains a mechanism – the generic constructions - that can help to find suitable points to be added to the construction in order to find a GDD proof. We show here how to use this mechanism for this purpose, without explaining the concept of generic constructions. Generic constructions are described in detail in Section **Napaka! Vira sklicevanja ni bilo mogoče najti.**.

Let us consider the configuration in Example 4. Suppose we want to prove that the triangles ΔBA'B' and ΔA'CC' are congruent. In the **Observe proof** form, select the property *congruent triangles* and enter the data for the two triangles. After clicking on ***Execute***, we find that the prover has not generated a GDD proof (Figure 1).

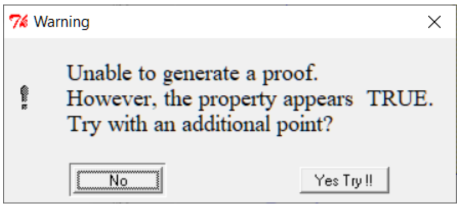


Figure 1

We can ask OK Geometry to try to position a new point so that a GDD proof of the considered property can be created. We are first informed that we will be asked to set a rule (a generic command) for the point we are looking for and that the results will be written into an archive called *gdd_xxx.arh* in the GDD archives directory.

The form for a new rule then appears (Figure 2). In the form we specify, which operations on which points to consider when generating the new point. In the form:

1. Checkmark the option *Display only proof-compliant* commands.
2. Select the operations you want to consider. We have selected: midpoint, mirror image of point in point, A-antipode in circumcircle, incentre.
3. Checkmark the points to be taken into account. We have checkmarke them all.

4. Then click on OK.
5.

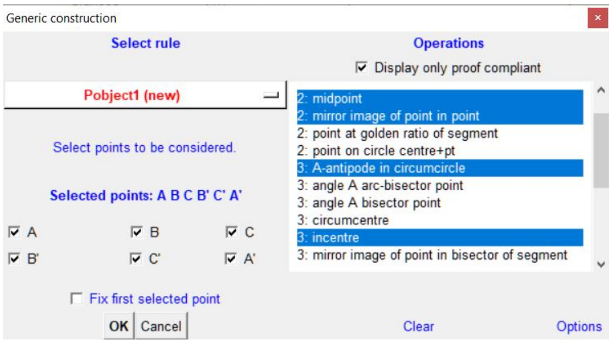


Figure 2

After a while, OK Geometry starts a search process. The search can take some time, for this reason it runs as a separate process in its own window (Figure 3). While you are doing other things in OK Geometry, you can monitor the number of solutions found. You can stop the search at any time by closing the window that contains the search process. The solutions (if any) are stored in the GDD archive *gdd_xxx.arh* in your GDD archives directory.

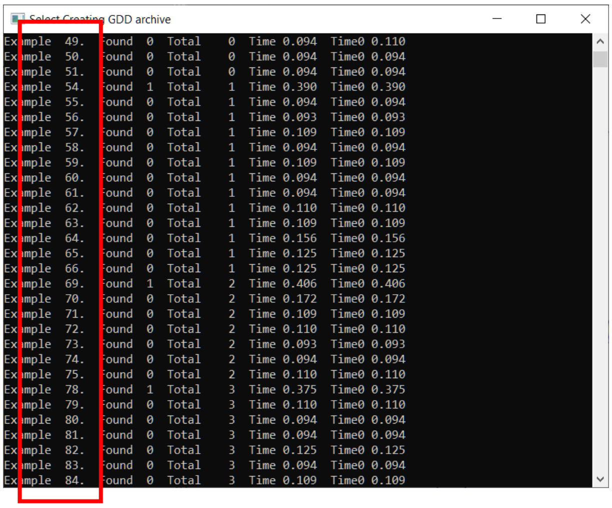


Figure 3

You can inspect the solutions with the command *File|Inspect GDD archive*. When inspecting the solutions, note that the ***Comment*** section contains the description of the added point (Figure 4). If you accept a solution, you can retry with the proof.

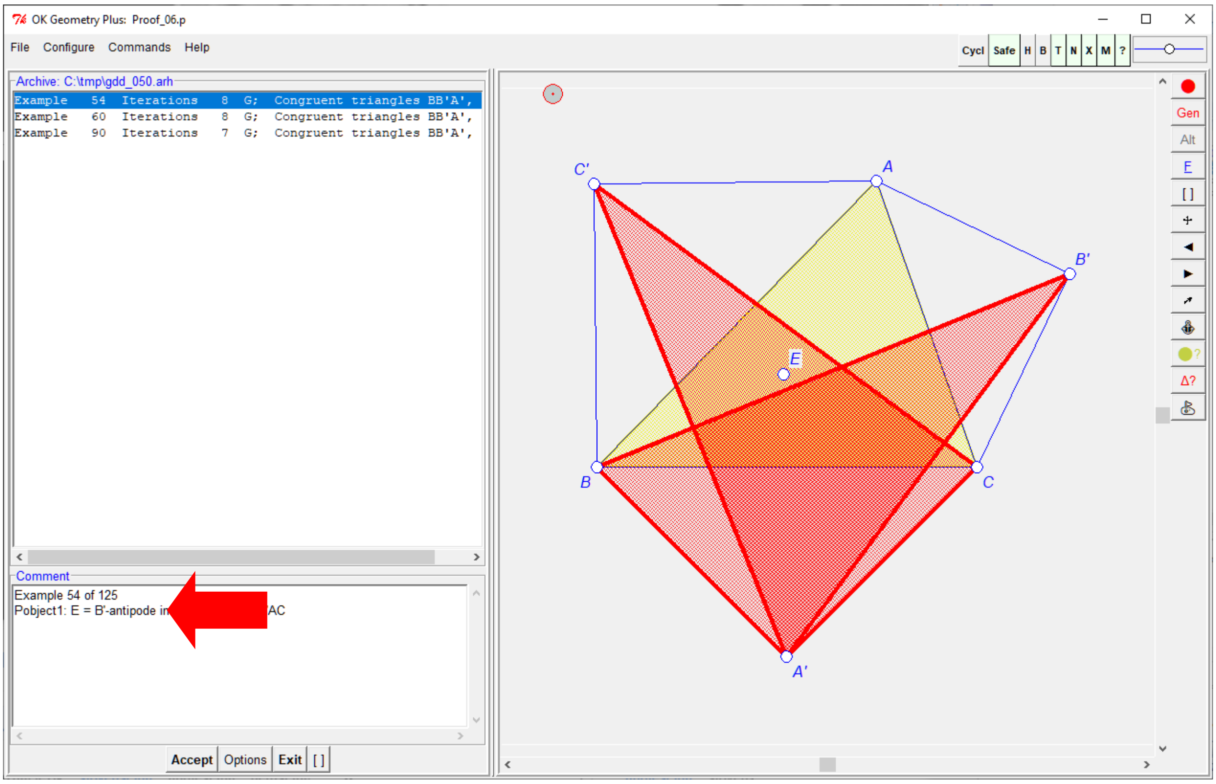


Figure 4

In our case, 3 (equivalent) solutions were found. A new point[1] E = antipode of B' in the circumcircle of AB'C enables the GDD prover to prove that the triangles ΔBA'B' and ΔA'CC' are congruent (Figure 5).

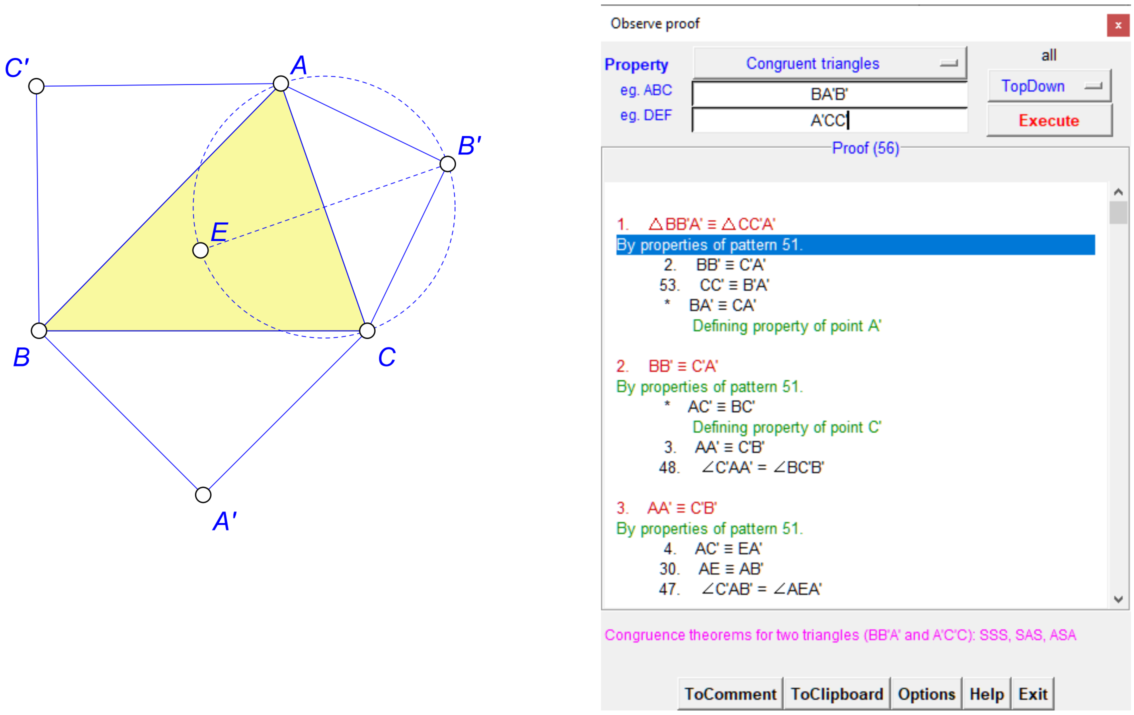


Figure 5

[1] OKExamples\OKG_Plus\Proof_04a.p

Example 5

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 (Figure 6, left).

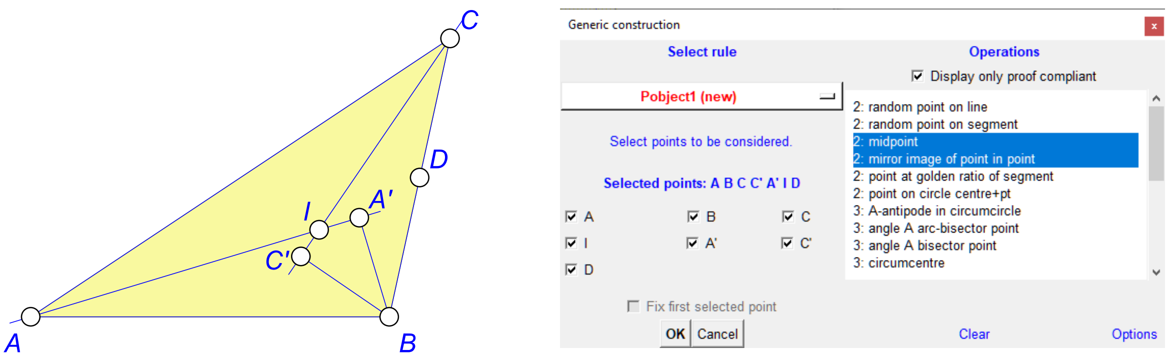


Figure 6

The configuration is easily to construct using standard commands[2]. The prover suggests to add a new point to prove the collinearity of A’, C’, D. The usual way is to first try with the midpoints or mirrored points of all labelled points (Figure 6, right). OK Geometry then finds that the collinearity of A’, C’, D can be proved by adding the midpoint E of AB (or as some other suggested points) to the construction.

Example 6[3]

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 7, left).

[2] OKExamples\OKG_Plus\Proof_05.p
[3] OKExamples\OKG_Plus\Proof_06.p

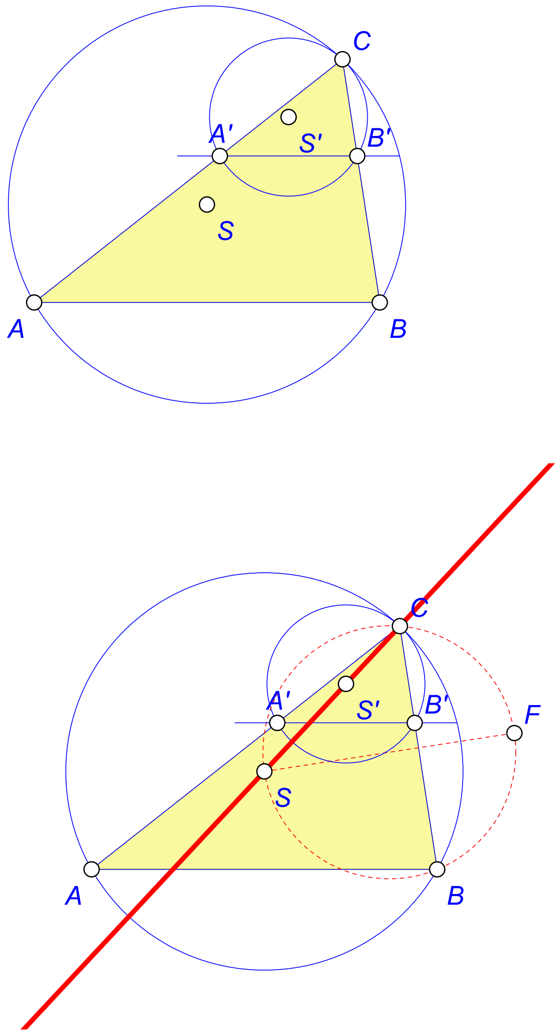


Figure 7

In this case, too, to get a proof, the prover needs an additional point, but a more sophisticated than in the previous case. If the '*A-antipode in circumcircle*' is included among the considered operations, OK Geometry finds that the antipode F of S in the circumcircle of ΔBCS leads to a proof. 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.

4. Rules (generic operations)

In Section **Napaka! Vira sklicevanja ni bilo mogoče najti.** we introduced the concept of **rule** (**generic operation**) as something analogous to a geometric operation in a construction, as a 'union' of operations. An example of geometric operation is the midpoint (of two points). We can apply this operation several times in a construction to different pairs of points. Similarly, in a generic construction a rule can be applied several times to sets of points. Only one operation of the rule can take place at time – and that same operation acts everywhere the rule is applied.

Figure 8 illustrates a rule that contains three operations:

- midpoint (of two points),

- the third vertex of a positively oriented equilateral triangle (based on two points)

- a free point on a line (through the two points).

Each operation of the rule acts on a pair of points and creates a new point. The rule was applied to the points A,B and then to B,C.

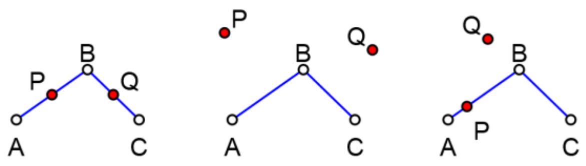


Figure 8

The rule gives rise to three examples, one for each operation in the rule. The points created with rules have a peculiarity. If they are free points, they change position but they cannot be dragged with the mouse. Thus, the red points in the third example in Figure 8 are free points on the lines AB and BC. They change position each time the example is displayed, but they cannot be dragged.

The analogy between the rules and operations is only partial. In fact, a rule consists of three components:

- The first component is the set operations that can take place when a rule is applied. The operations involved in the rule act **only on a certain number of points**. An operation, like 'construct a line through a given point and parallel to a given line' cannot be included in a rule. On the other hand, the operation 'construct a line through a given point and parallel to a line through two given points' is allowed.
- The second component is the number of points on which a rule acts. This number is specified when a rule is declared. The same rule can be applied several times in a generic construction on different sequences of points, each consisting of the same number of points.
- The third component is the role of the order of points in a sequence on which a rule operates and in particular the role of the first point (the fix-first-point attribute).
- In the form you can checkmark whether you want to restrict the operation in the rule to proof-compliant operations. Operations with no asterisk at end are proof-compliant. They contain information that is eventually used in proving properties of examples of generic constructions. You apply this restriction when you intend to observe proofs (see Section **Napaka! Vira sklicevanja ni bilo mogoče najti.**).

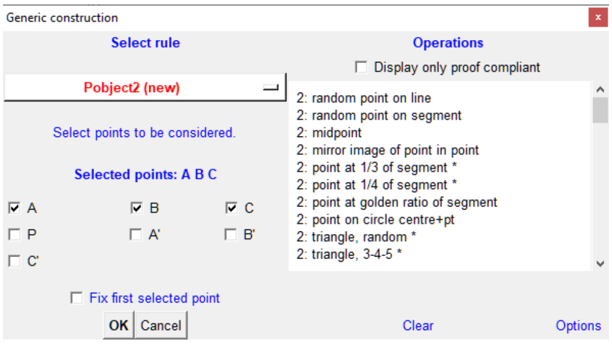


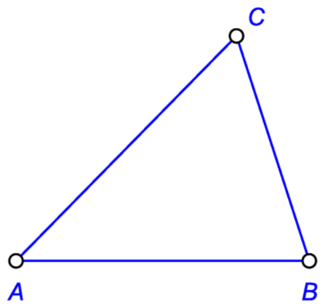
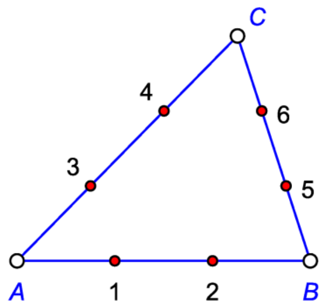
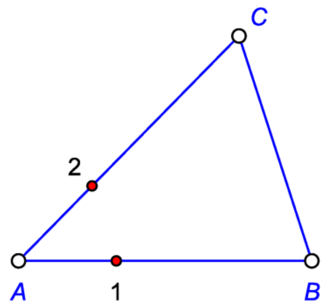
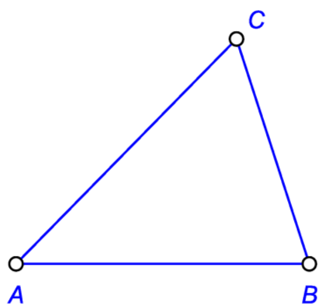
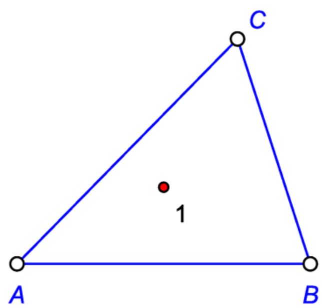
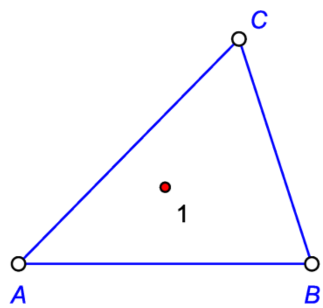
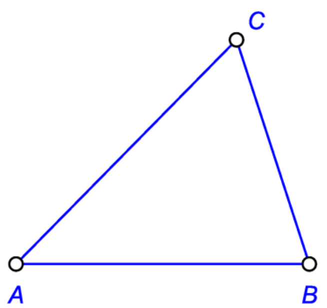
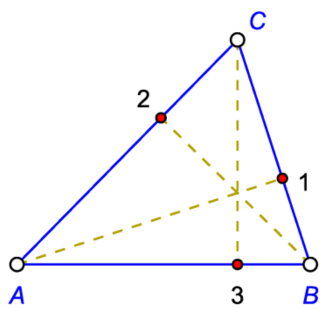
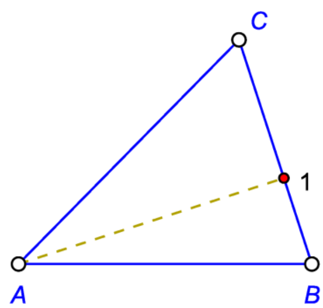
Figure 9

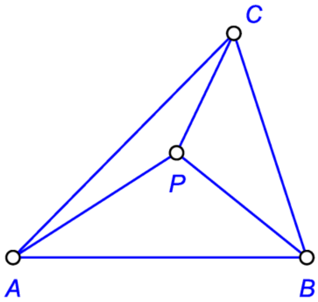
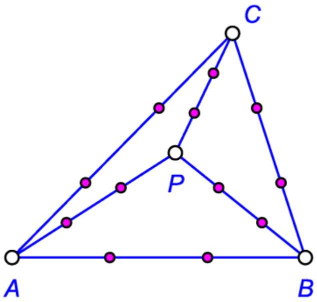
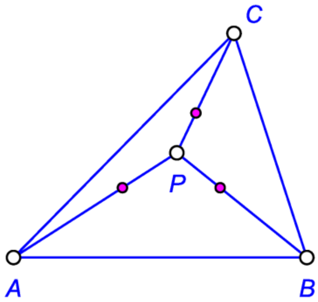
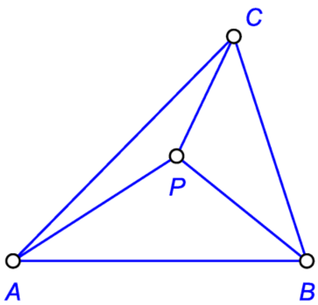
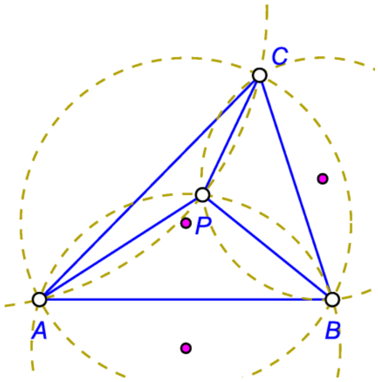
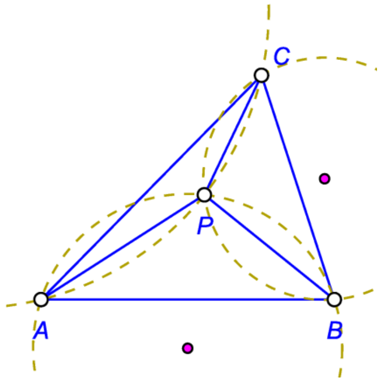
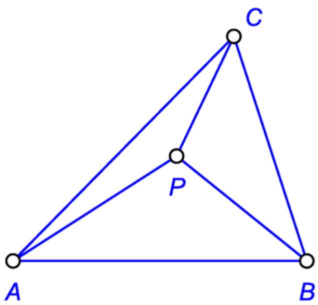
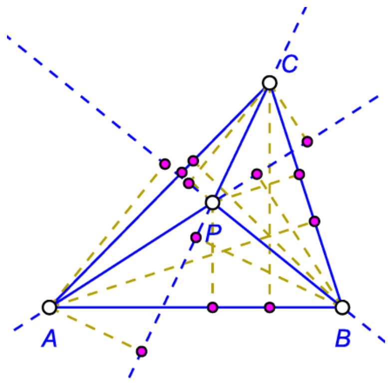
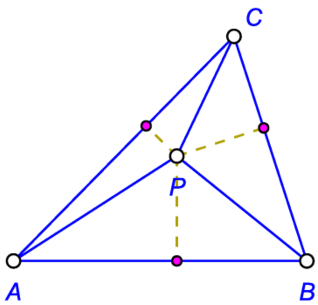
Rules are declared and called with commands in the *Generic* menu group of the Sketch Editor. The declaration of a new rule occurs simultaneously with its first application. To declare a rule, we must specify the rule name, the operations to be included in the rule, the points subject to the rule, and the fix-first-point status. 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 either. Figure 9 shows the process of declaring a new rule Pobject2. The 'project onto line' operation has been selected as one of the operations. Note the '3:' in front of the operation name – it indicates that the operation uses three points (the number of the selected points of the rule may be greater). Also note the magenta explanation of the operation – if the operation is applied to three points, then the 1st point is projected orthogonally onto the line passing through the 2nd and the 3rd point.

Although one needs not to worry about the components in simple cases, we explain the role of the components with some examples.

Assume we have a rule that yields an object. The rule consists of several operations to be applied to a set of *n* points. Let us focus on one operation within the rule. Suppose the operation yields the resulting object taking *m* points ($m \leq n$). The operation within the rule acts on every *m*-tuple of *n*-points and all arrangements within the *m*-tuples (i.e. on all permutations of *m*-points out of *n*-points) yielding an object. Arrangements that are redundant, according to the meaning of the operation, are ignored. By setting the fix-first-point attribute a restriction is imposed: The first point in the list of selected points (of a rule) is also the first point of all considered permutations of points (for all operations contained in the rule).

We clarify the role of sequence of points with some examples:

Configuration, operation, and the sequence of points	**Yielded objects within rule if fix-first-point is OFF**	**Yielded objects within rule if fix-first-point is ON**
 2: Point at 1/3 of segment Points: A,B,C		
 3: Centre of circumcircle Points: A,B,C		
 3: Project onto line Points: A,B,C		

 2: Point at 1/3 of segment Points: P,A,B,C		
 3: Centre of circumcircle Points: P,A,B,C		
 3: Project onto line Points: P,A,B,C		

An operation within a rule generally results in a sequence of objects (only one created object appears in any example). To make rules manageable, it is better to use within a rule only operations that act on the same number of points. Also, a rule should require the same or a slightly larger number of points than the operations contained in the rule.

The rules in the Generic menu are divided into two or three groups. In the next subsections we give an overview of the rules. All rules create new objects from points. The names of the rules are fixed, the first letter of the name indicates the type of objects created by the rule (P – point, L – line, C – circle, T – triangle, Q – quadrilateral, S-conic). For example, the names of the rules for creating points are Pobject1, Pobject2, Pobject3, and so on. An exception are the rules ETC1, ETC2, ETC,.. for ETC triangle centres.

The commands in the first two parts of the **Generic** menu are used just like other commands for creating constructions. The commands appear like commands for creating a point, line, or other object. In fact, they create objects in the current construction, which is the base example, but at the same time they declare and/or apply a rule in the generic construction. For example Figure 8 illustrates a simple acting of a rule containing three operations. When the rule is declared or applied to a set of points, one of the operations contained in the rule also becomes a corresponding operation in the base example, which is also displayed in the construction.

Rules for creating simple objects

The commands for the rules for creating points (*Gpoint*), lines (*Gline*), and circles (*Gcircle*) are located in the top part of the **Generic** menu commands. When one of these commands is activated, a form appears (Figure 10). We will take a closer look at the form for creating a rule for points, since the forms for creating lines and circles are pretty self explanatory.

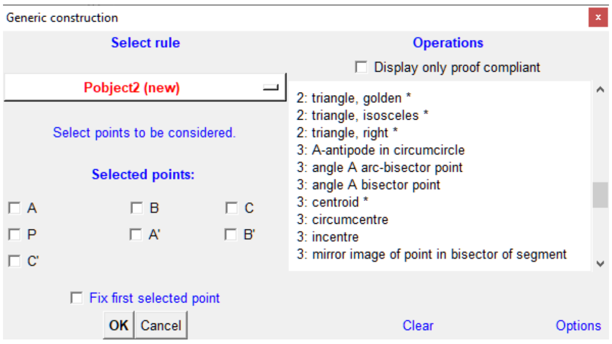


Figure 10

In the list of operations that can be included in the rule for points there are operations (e.g. midpoint) that do not require an explanation and groups of operations that do.

Point at x of segment. The values listed are 1/2 (midpoint), 1/3, 1/4, and (√5-1)/2 (golden ratio). You can add more values to the list by clicking *Options|Divide segment values* (lower right corner of the form).

Triangle 3rd vertex ... . Given two vertices (green coloured points in Figure 11), this group of commands creates the 3rd vertex (red coloured point) of a positively oriented triangle. The position of the third vertex is described (up to direct similarity) with one or two conditions in terms of parameters shown in Figure 11. Some examples: 'A=60 & B=60' positions the 3rd vertex of an equilateral triangle; 'C=90' positions the

3rd vertex on a random point on a semicircle whose endpoints are the given points; 'a:b=2 & A=72' or 'a:b:c=5:6:7' are also acceptable descriptions. Some common names are also allowed as descriptions of the resulting triangle: equilateral, 345 (i.e. a triangle similar to a right triangle with sides 3,4,5), isosceles, etc. The list of Triangle 3rd vertex operations can be modified by deleting or adding new descriptions. You can modify the list of commands by clicking *Options|Triangle 3rd vertex* (lower right corner in the form) and filling the form that appears.

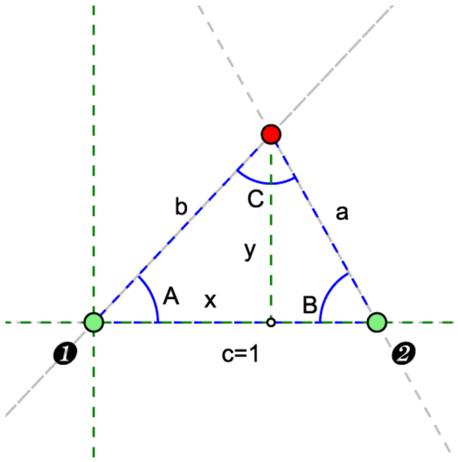


Figure 11

ETC triangle centres ... . In the list of operations, there is a group of operations that yield some ETC centres from three given points. You can modify the list using *Options|ETC triangle centres* (lower right corner of the form). In the form that appears specify the numbers of ETC centres to appear in the list. Note that a full list of ETC centres is available in another group of rules (see 0).

Quadrilateral centres ... . The list contains a few QA-quadrilateral centres (i.e. centres obtained from vertices irrespective of their order). The generic examples thus do not permute the vertices of a quadrilateral. Note that a full list of quadrilateral objects is available in another group of rules (see 0).

The rules for creating lines and circles are self-explanatory.

It is also possible to create rules for numeric parameters. These rules are simply lists of numbers or numeric expressions.

Rules for transforming simple objects

The **GTransformation** rule creates new points (lines, circles) from existing originals using a set of selected transformations. The rule leaves the original untouched, merely adding a new object of the same type as the original to the construction. The operations (i.e. transformations) in the rule are always defined in terms of some points.

To declare a transformation rule

1. Activate the *Generic|GTransformation* command.

2. Select the object (point, line, circle) to be transformed.
3. A form appears (Figure 12), the name of the declared rule is Ptransf1, Ptransf2,… for points, Ltransf1, Ltransf2,… for lines, Ctransf1, Ctransf2,…for circles.
4. Fill in the form and press OK. A new object appears in the base construction.

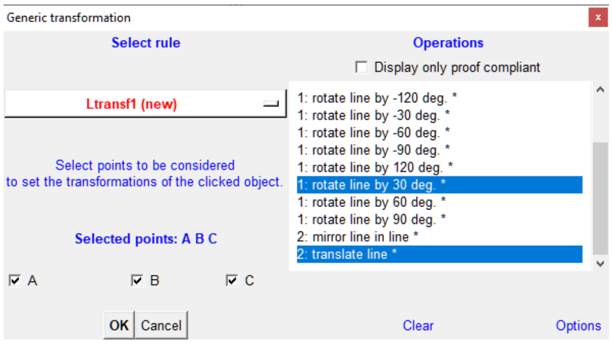


Figure 12

We first comment on the form for declaring rules for transforming lines. Recall that the form appears after a line was selected in the base construction. In our case (Figure 12), the base example clearly contains points A,B,C. You need to select the operations and the points that define the operations. In Figure 12, three points have been selected and two operations (rotation by 30 degrees and translation). Rotation by 30 degrees is defined by 1 point (see '1:' in front of the name of the operation). Thus the generated line in the examples is the original line rotated by 30 degrees around A, around B and around C. In further examples (resulting from the same rule) the generated line is the translation of the original line by vectors AB, BA, AC, CA, BC, CB. Altogheter, the rule produces 9 lines, which are transformations of the original line.

In the list of operations for transformation rules, besides self-explanatory entries, there are two groups of transformations: 'rotate by x degree' and 'dilatate by factor x'. The values x in the list can be modified by setting the desired values with the commands *Options|Rotation angles and Options|Dilatation factors* (lower right corner of the form).

The rules for transforming points have a peculiarity. Suppose the base construction contains a point P and points A and B. We declare a rule for transforming of a point and choose the point P. The point to be created will be thus a transformation of P. Consider now the form that appears.

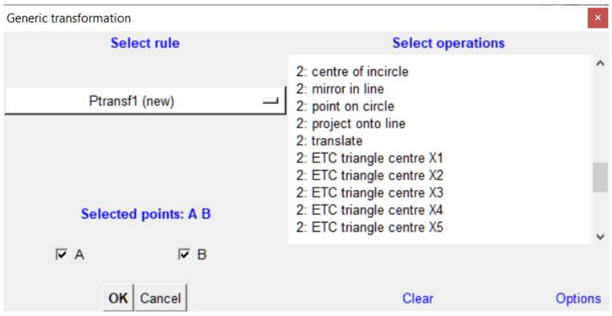


Figure 13

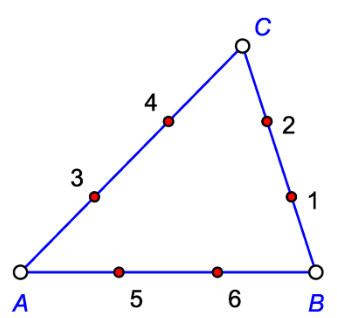
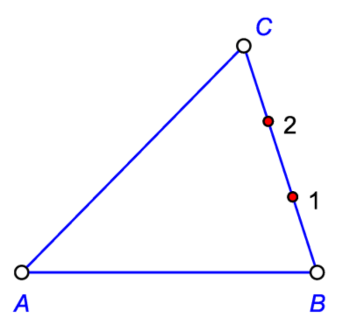
Among the listed operations you can see some 'standard' transformations. For example, the entry '2: translate' in our case would yield 2 examples: the point P translated by AB and P translated by BA. On the other hand, observe the entry '2: centre of incircle'. Centre of incircle is usually not considered a transformation, it is rather considered an operation on 3 (not 2) points. In such cases the selected point (to be transformed) is considered the first argument of the operation. The transformed point P in this case is thus the incentre of the triangle PAB. Mixing geometric operations and transformation may appear confusing, but on the other hand enables the declaration of richer rules.

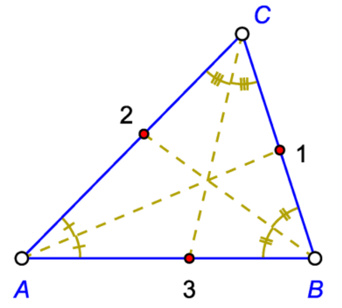
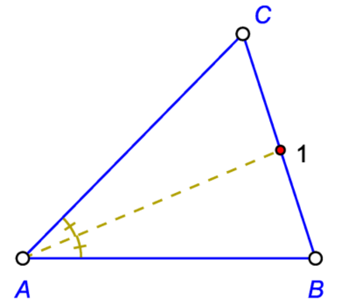
The next group of rules are all defined on a fixed number of points.

Triangle A-points

The rules named Apoint1, Apoint2,... contain operations that create points in a triangle, often with respect to only one of the vertices of the triangle. These rules can only be applied to triplets of points. As with any rule, the operations in the rule are applied to all permutations of the points in the triplets. So a single operation in the rule will result in up to 6 different points in respective examples. If the first point of the triplet is fixed, a single operation gives rise to 1 or 2 points.

Here are two examples of A-operations:

Configuration, operation, sequence of points	Yielded objects within rule if fix-first-point is OFF	Yielded objects within rule if fix-first-point is ON
Points: A, B, C Operation: Point at 1/3 of A-side		

Points: A, B, C Operation: Angle A bisector point		

The list of operations for triangle A-points contains common operations for points, perhaps modified with respect to a specific vertex. Some operations are rather specific to triangles. In addition to triangle centres (the list can be modified with *Options|ETC triangle centres*) the list of operations contains, relative to a vertex, the pedal point, the Cevian point, the circumcevian point, and the circlecevian point of the included triangle centres (see Figure 14). In this way, a rich variety of operation can be contained in a single rule.

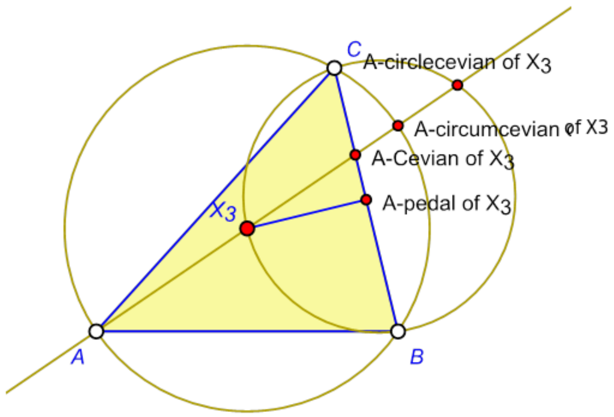


Figure 14

Rules for triangles and quadrilaterals on a segment

The *Generic|Triangle on segment* command creates a rule that yields various types of positively oriented triangles with two given vertices. The rules for triangles are named Tobject1, Tobject2,... The entries in the list of operations for triangles are the same as for *Triangle 3rd vertex,* except that the sides of the triangle are also displayed. The types of triangles in the list of operations, more precisely the triangle's 3rd vertices, can be set and changed by the user with a click on *Options|Triangle 3rd vertex.*

The command *Generic|Quadrilateral on segment* creates a rule that yields various types of positively oriented quadrilaterals with two given adjacent vertices. The rules for quadrilaterals are named Qobject1, Qobject2,...

The rules for triangles and quadrilaterals on a segment are ordinary rules and can be used in constructions just like other rules. However, they are primarily used to create the object at the very beginning of the construction process. When we study the properties of various triangles or quadrilaterals, it is useful to initiate the generic construction with 2 points and a rule for triangle or quadrilateral.

An example with detailed description of the use of rules for triangles and quadrilaterals on a segment can be found in Section **Napaka! Vira sklicevanja ni bilo mogoče najti.**.

Rules for triangle objects

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

All rules in this group require exactly three points as vertices of a triangle, and possibly one or more as an argument (for example, the Cevian triangle of a point). All rules of this group are also obtained in a very similar fashion. We illustrate the process for the case of rule for triangle circles. We activate the command *Generic|Triangle objects|Triangle circles.* In the form that appears (Figure 15, left), we checkmark exactly three triangle vertices. We can use a previously declared rule or decide to declare a new rule. In the latter case, after clicking OK, a second form appears (Figure 15, right). In this form, we select the type of circles to be included in the rule and click OK.

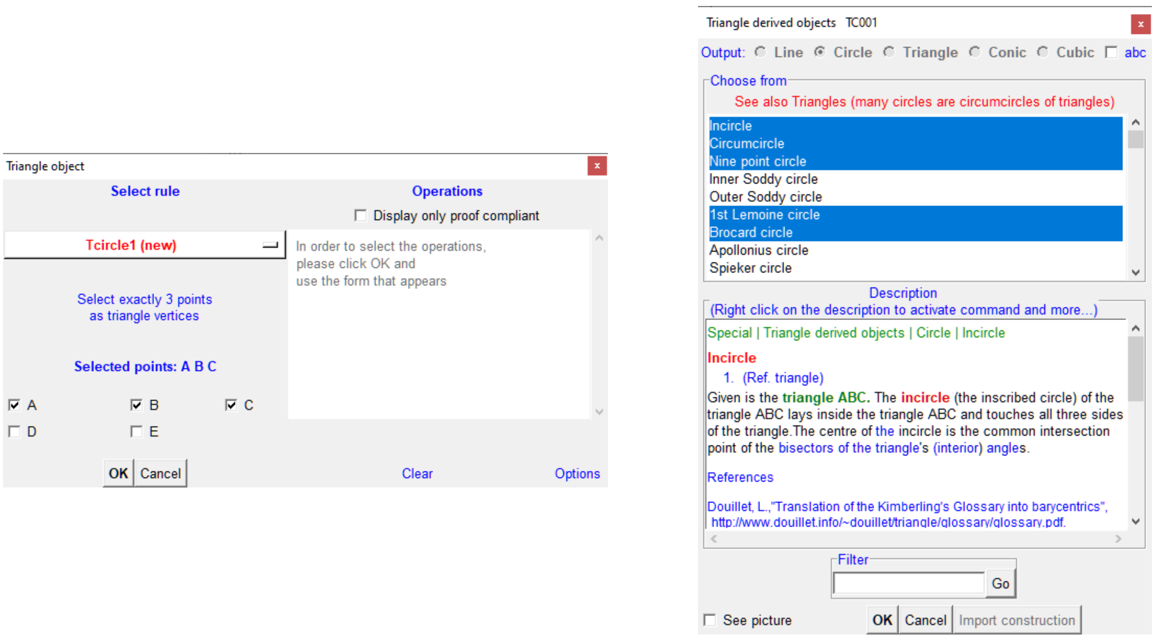


Figure 15

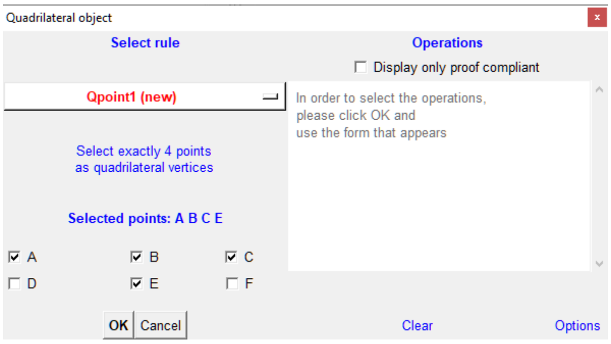
Rules for quadrilateral objects

It is also possible to create rules for all kinds of quadrilateral objects. The rules are generic versions of most of the commands for creating Quadrilateral objects in

Sketch menu *Special.* The rules for creating centres of quadrilateral are named Qpoint1, Qpoint2,..., the rules for quadrilateral lines can take the names Qline1, Qline2,..., for circles Qcircle1, Qcircle2,... and so on.

All rules in this group require exactly four points as vertices of a quadrilateral. All rules in this group are obtained in a very similar fashion. We illustrate the process for the case of rule for quadrilateral centres (points). We activate the command *Generic|Quadrilateral objects|Quadrilateral points.* In the form that now appears (Figure 15, left), we checkmark exactly four vertices. We can use a previously declared rule or decide to declare a new rule. In the latter case a second form appears after clicking OK (Figure 16, right). In this form, we select the quadrilateral points to be included in the rule and click OK.

Quadrilateral objects are only partially analogous to triangle objects in generic constructions. Triangle objects tare declared with three vertices, e.g., the Euler line of the triangle ABC, and are independent of the order in which the vertices are selected (the Euler line of the triangle ABC is the same as the Euler line of the triangle CBA). This is not the case with quadrilateral objects. A selected object, e.g. QL-Miguel point (QL-P1), on the quadrilateral with vertices A, B, C, D (in this order) will yield the QL-Miquel point of the quadrilateral ABCD only. The QL-Miquel points of the QA-siblings or QL-siblings or even QG-siblings of the quadrilateral ABCD may differ from that of the quadrilateral ABCD and they are not taken into account in the generic construction.



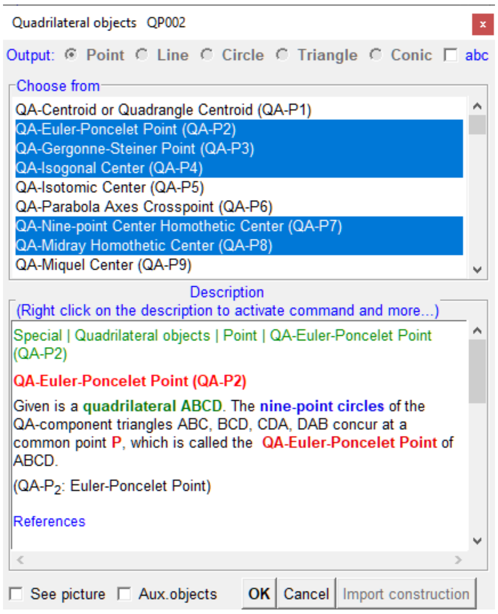


Figure 16

5. GeoGen archives

Introduction

GeoGen is a software program developed by **Patrik Bak**. Its aim is to generate non-trivial proving problems in planar geometry, i.e. planar geometric configurations together with associated properties that are not easy to prove. You can access the complete development version of GeoGen on GitHub[4]. If you are not familiar with development versions, you can find an easy-to-install 'publish' version of the parts of GeoGen that are called up from OK Geometry on the download page of OK Geometry. To run GeoGen from OK Geometry, unzip the files from GeoGenPublish.zip into a directory of your choice. Both installationions of GeoGen require a .NET 7.0 runtime environment[5].

To use GeoGen in combination with OK Geometry you need to specify in the OK Geometry configuration (*Configure|General options*|Archives) the directory that contains the GeoGen execution file(s) and the directory of your working files related to GeoGen.

Before we explain the concept on not-trivial tasks, we will introduce the workflow with GeoGen. The process of creating non-trivial proving problems (and the associated configurations) consists of three steps:

1. Designing the family of considered configurations	You can perform this phase within OK Geometry with the command *Commands|Generate GeoGen archive.* The resulting designs are normally saved in files with prefix 'input_' and extension '.txt', e.g. input_test.txt .
2. Generating the set of non-trivial problems	For this ***essential phase***, OK Geometry calls the programme GeoGen by Patrik Bak as an external programme. GeoGen is executed separately and independently of OK Geometry. You start the generation from one or more input design files at once with the '*CreateGeoGen archive*' button. The generation process can take from a few seconds to several hours, depending on the complexity and the number of configurations taken into account. However, you can stop the process at any time by simply closing the window in which the generation is running. As a result, you get the corresponding output files (archives) with prefix 'output_' and the extension '.json' or '.txt'. For example, from the files *input_test1.txt*, *input_test2.txt*,... you get the *files output_test1.json, output_test2.json*,...
3. Inspecting the generated set of	This phase can be carried out within OK Geometry using the command *Files|Inspect GeoGen archives*. You can inspect one or

[4] https://github.com/PatrikBak/GeoGen
[5] https://dotnet.microsoft.com/en-us/download/dotnet/7.0

problems	more output files (archives) at once. Each task in the archives can be visualised, set as an icon in the current project, saved as a OK Geometry construction, transferred to the work archive. An inspected task can also be transferred to OK Geometry as the current construction so that it can be analysed immediately with the tools available in OK Geometry.

GeoGen's'reasoning' is based on variant of the deductive database method (GDD) method of Chou, Gao and Zhang. Important elements of this method are the list of **allowed operations** (construction steps) and the list of **inference rules** (generally known facts about properties).

Three steps are required to create the design for the family of considered configurations:

1. Start from the **initial configuration** (a generic triangle, a generic quadrilateral or a previously saved design to be possibly modified).
2. Create the **base configuration** by adding new objects by applying allowed operations to previously defined objects.
3. Select a subset of the allowed operations that will be used in all meaningful combinations (of objects and operations) in **further construction steps**. Here it is also necessary to specify the number of steps and some other parameters. Depending on the number of objects in the base configuration and the number of operations selected and the number of subsequent steps, the choices can easily lead to excessively large number of configurations to be considered. Moderation and good intuition are therefore required here.

GeoGen takes several steps to identify configurations with interesting properties from among those under consideration. First, irrelevant configurations are sorted out (e.g., only geometrically non-isomorphic configurations are considered). Then, GeoGen identifies some basic properties of the considered configuration by numerical observation (e.g. it finds that three points are collinear). The crucial step is to find out whether the observed property is of interest or not. To this end, GeoGen attempts to prove the property using a time-efficient, specially developed GDD method, i.e., to check whether the observed property can be deduced (in a given time interval) by applying inference rules from the constructional properties of the considered configuration. If this check fails, the property is declared as **non-trivial**. A non-trivial property can be:

- (very very unlikely) the result of an observation error, i.e. numerically true in the observed cases, but geometrically not true;
- (very likely) geometrically true but the property cannot be inferred from the constructional properties of configuration using the available inference rules – at least not in the required time interval;
- **ambiguously true**, i.e. the found property is geometrically true in some cases, but not generally true (e.g. is true only for non-obtuse initial triangle).

The observed properties of the base configuration are assumed to be true – and never treated as a problem. A property is considered a potentially non-trivial property only if it is dependent on all objects of the configuration by construction.

Finally, the author, Patrik Bak, provides criteria for evaluating the 'complexity' and 'beauty' of the found properties of configurations, according to his expertise in solving and designing geometry problems.

GeoGen only considers the following properties of configurations:

collineation of three points,

concurrency of three lines,

concyclicity of four points,

congruence of two segments,

tangentiality of lines to circles,

parallelism of lines,

perpendicularity of lines,

tangentiality of circles,

incidence of points and circles/lines.

GeoGen assigns to non-trivial problems respective relevance factors. OK Geometry displays four of them in a slightly modified form with respect to GeoGen. In the list of found problems the listed factors from left to right are (see Figure 19):

- **Relevance of the task**. In the OK Geometry display, the relevance ranges from 10 000 upwards. The higher the relevance, the more "difficult and beautiful" the problem is. You can ask OK Geometry to also display the so-called simple tasks, i.e. the properties of configurations that GeoGen was able to show in a very short time (see Section 0). The simple problems have relevance 9200 – 10000. In general, although not as a rule, a higher relevance indicates a more difficult problem.
- **Symmetry of the property**. Symmetric properties are more appealing than non symmetric ones. The level of symmetry is expressed in a normalised fashion in the range 0-100, where 0 means complete asymmetry and 100 means complete symmetry.
- **Level of the configuration**. Configurations are considered to be (more) appealing, if the constructed objects are obtained (more) directly from the initial configuration using one of the available operations. On the other hand, the presence of objects that are obtained sequentially with several operations diminishes the 'beauty' of the configuration. In OK geometry the level of the configuration is expressed in a normalised fashion in the range 0-100. A higher number means overall shorter sequences of operations for creating objects from the initial configuration.

The details of the whole process can be found in the Patrik Bak master's thesis[6].

[6] https://drive.google.com/file/d/1dsaxDCMzIAPfB3e4rd8ut2RuZ_sn2Zm5/view

Using GeoGen from OK Geometry

We explain the use of GeoGen from OK Geometry with an example[7]. Consider a simple base configuration (Figure 17):

In a given triangle ABC, let D be the midpoint of BC. Let the bisector l of AD intersect the sideline AB in E and the sideline AC in F.

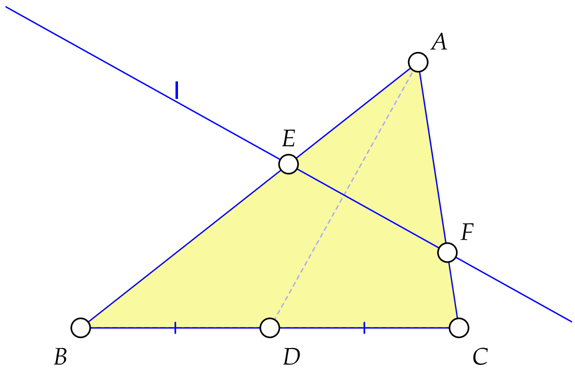


Figure 17

To create new configurations from this rather simple configuration, we add two new objects by applying two out of these three operations:

- a new point is the midpoint of two existing points,
- a new point is the circumcentre of the triangle formed by three existing points,
- a new point is the antipode of an existing point in the circumcircle of a triangle formed by that point and two other existing points.

[7] OkExamples\GeoGen_examples\input_mydesign.txt
OkExamples\GeoGen_examples\output_mydesign.json

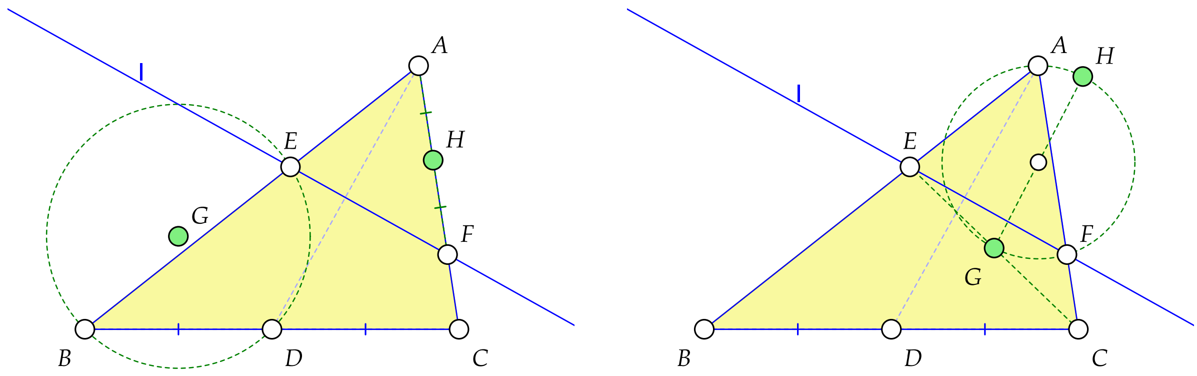


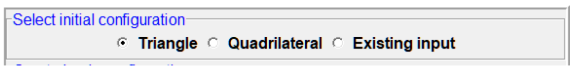
Figure 18

Figure 18 shows only 2 examples (out of 15295 possible cases, including the degenerate ones, but GeoGen reduces the family to about 4000 significantly different cases) of configurations obtained by adding the points G and H to the base configuration.

To create the design for the family of configurations described above, proceed as follows:

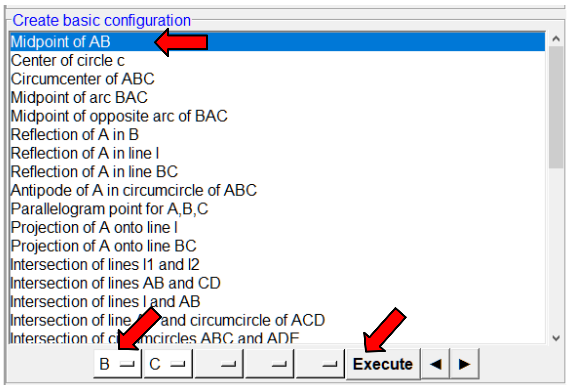
<table>
<tr><td>Select the command Commands|Generate GeoGen archive in the main menu.

A form dedicated for this purpose will appear. Proceed 'from top to bottom' In the form.</td><td></td></tr>
<tr><td>Select the initial configuration, in our case it is a triangle.

A triangle ABC appears on the right-hand side.</td><td>
</td></tr>
<tr><td>Click on the operation 'Midpoint of AB'.

Select points B and C.

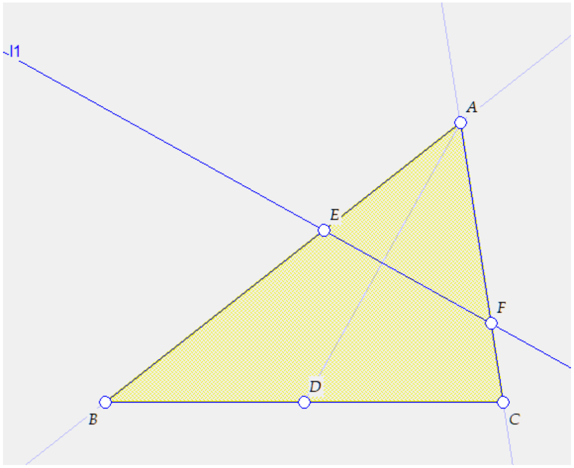
Then click on Execute.

The midpoint D appears on the right-hand side.</td><td>
</td></tr>
</table>

Proceed in the same way as above for the commands:

- *Perpendicular bisector of AB* for the points A and D to obtain line l1,
- *Intersection of lines l and AB* for the line l1 and the points A, B to obtain point E,
- *Intersection of lines l and AB* for the line l1 and the points A, C to obtain point F.

The base configuration is now complete. Note that the resulting figure on the right also contains auxiliary objects for easier interpretation. The auxiliary lines are bleached, you can make them more or less bleached with Alt+MouseScroll.



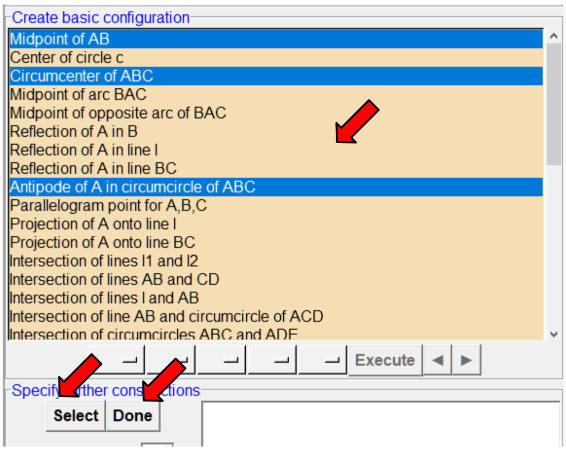
We proceed to select the list of further operations:

Click on **Select**. The list of constructions will now be coloured.

Select the operations to be included in the list.

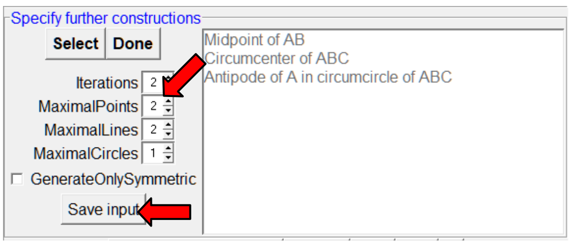
Then click on **Done**.

The selected operations are now displayed in a separate list.

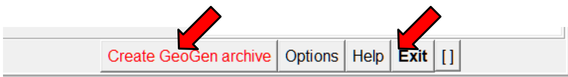


Now set the parameters for further operations. In our case, we want 2 additional operations (2 *Iterations*) that will generate 2 points. The entries *MaximalLines* and *MaximalCircles* are irrelevant in our case.

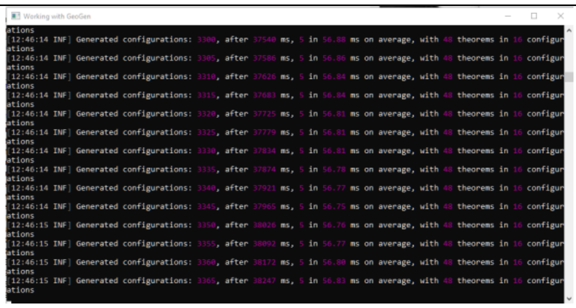
It is important that you save your design as input file, otherwise you will not be able to generate the GeoGen archive of non-trivial problems. If you write '*mydesign*' as name of file, the design will be saved as file named '*input_mydesign.txt*'.



Once the design has been saved, click on **Create GeoGen archive.** You can select one or more input files to be processed simultaneously. The command calls up GeoGen as external programme that is executed in a separate window. The execution may take some time, but you can stop it by simply closing the window. In the meantime you can do other things in OK Geometry.



<table>
<tr><td>For each input file,e.g. input_mydesign.txt, GeoGen generates a corresponding output file, e.g.output_mydesign.json. The output files are always located in the directory for GeoGen working files specified in the configuration.

To end the design of the GeoGen input files, click on Exit in the design form.</td><td></td></tr>
<tr><td>You can create several design files to be processed later or to be used as a starting point for new design files.</td><td></td></tr>
</table>

Inspecting GeoGen archives

The archive files created with GeoGen can be viewed in OK Geometry with the command *Files|Inspect GeoGen archives.* You can select several GeoGen archive files to be inspected at once.

Figure 19 shows what happens after we select the file *output_mydesign.json*. GeoGen has found 58 non-trivial tasks related to the designed configurations. The tasks and configurations can be visualised, for each task you can also see the relevance, the symmetry and the complexity estimates, and the number of objects. In the visualisation you can also see auxiliary objects that are bleached, so that they are more or less visible, as well as the emphasised objects that illustrate the property found.

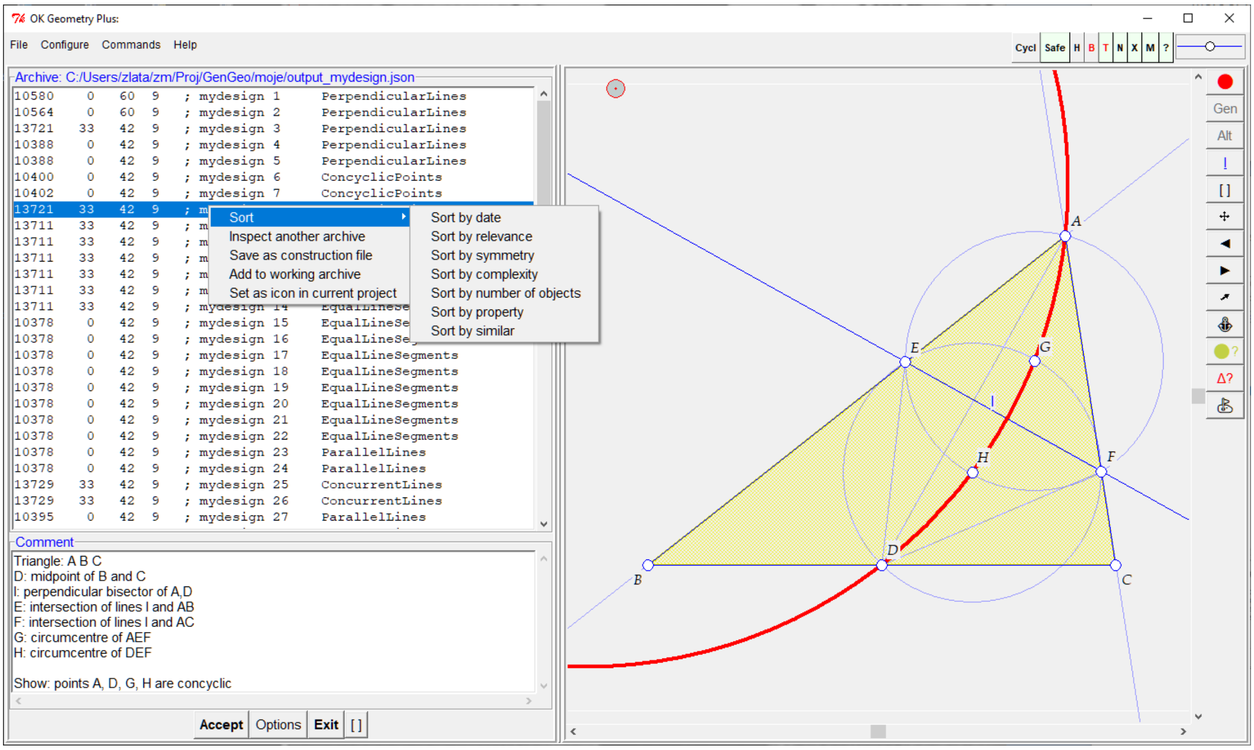


Figure 19

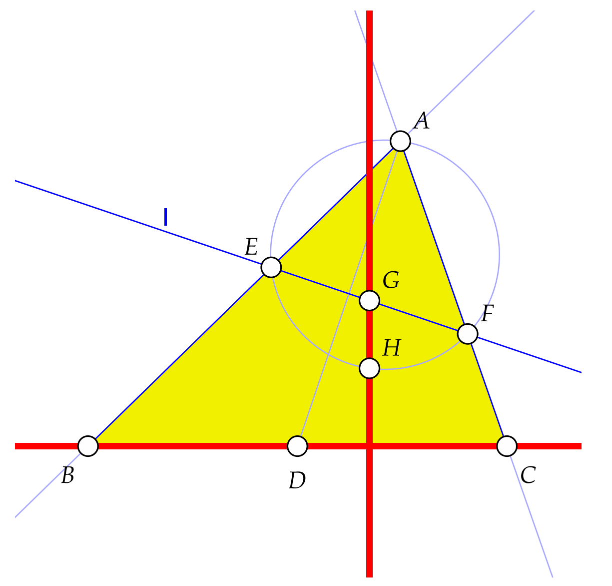
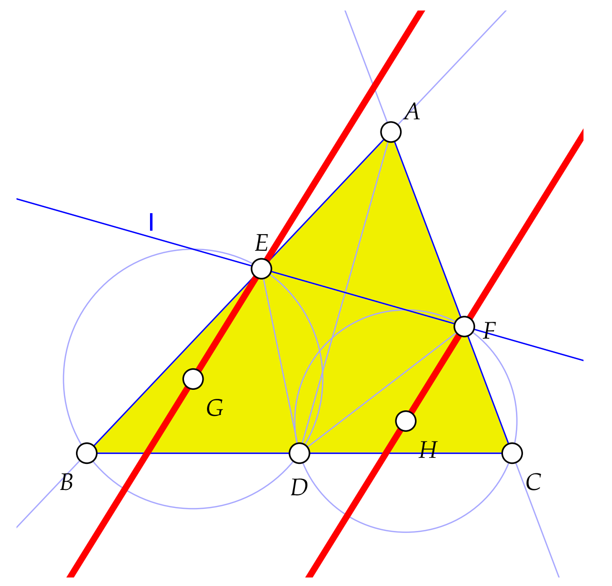
As you can see, the tasks can be sorted according to various criteria.

The configurations and properties can

- be saved as ordinary OK Geometry constructions,
- be set as icons in the current OK Geometry project,
- be added to the current work archive.

The **Accept** button makes the visualised configuration the current construction in OK Geometry – so that you can analyse it immediately with the tools provided by OK Geometry. To exit without affecting the current OK Geometry construction, press **Exit**.

We present two examples that we have selected from the GeoGen archive (Figure 20). Both tasks are indeed beautiful and certainly not trivial.

<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Triangle: A B C
D: midpoint of B and C
l: perpendicular bisector of A,D
E: intersection of lines l ans AB
F: intersection of lines l ans AC
G: midpoint of E and F
H: antipode of A in circumcircle AEF

Show: lines [B, C] and [G, H] are perpendicular</td>
<td>Given is a triangle ABC.
Let D be the midpoint of the segment BC. Let l be the perpendicular bisector of the segment AD. Let E be the intersection of lines l and AB. Let F be the intersection of lines l and AC. Let G be the circumcentre of the triangle BDE. Let H be the circumcentre of the triangle CDF.

Show that the lines [E, G] and [F, H] are parallel.</td>
</tr>
</table>

Figure 20

A second example

In the second example we present, we can see how OK Geometry can shed additional light and make more sense of a configuration of non-trivial tasks created by GeoGen. Figure 21 shows one of the tasks created from a rather simple GeoGen design input file[8].

[8] OkExamples\GeoGen_examples\input_incenter.txt

Triangle: A B C
D: incentre of ABC
E: projection of D onto line BC
F: projection of D onto line AC
G: projection of D onto line AB
H: intersection of lines AB and EF

Show: lines [C, G] and [D, H] are perpendicular

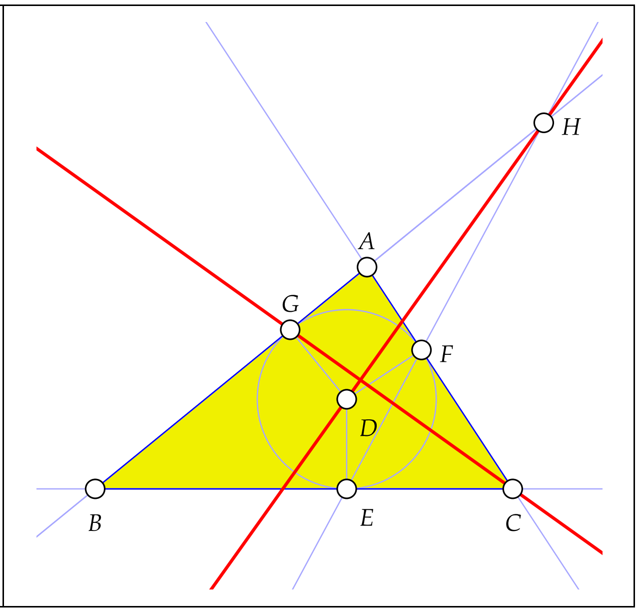
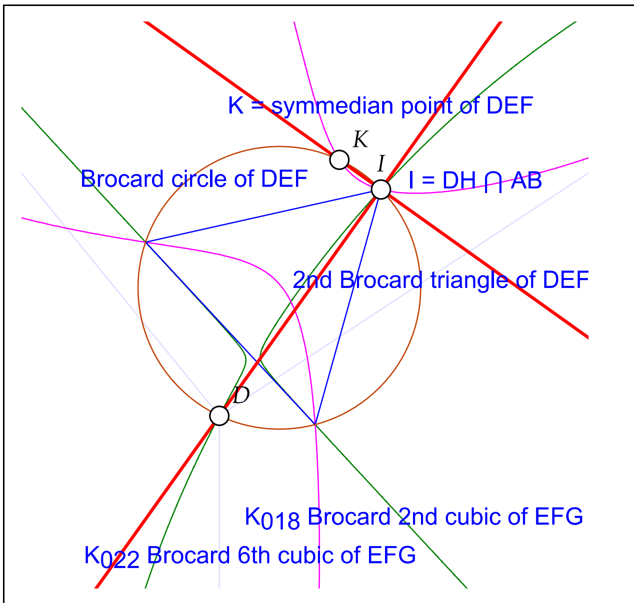


Figure 21

In order to better understand the situation (not necessarily for proof purposes), we make the configuration of the task the current OK Geometry construction. Obviously, a point I is created at the intersection of the lines CG and DH. We analyse the point I with respect to the triangle ABC – observation shows that I lays on the Brocard circle of the intouch triangle of ABC, i.e. on the Brocard circle of triangle EFG. We therefore analyse the point I with respect to the triangle EFG as a reference triangle (Figure 22).



According to the observation, the point I lays on the Brocard circle of DEF, which has as diameter KD, where K is the symmedian point of DEF. The fact that K (by observation) lays on the line CD and I lays on the Brocard circle is thus consistent with the fact that the lines HD and CG are perpendicular to each other.

In addition, the observation shows that I is a vertex of the 2nd Brocard triangle of EFG and that the cubics K022 and K018 pass through I.

Figure 22

OkExamples\GeoGen_examples\output_incenter.json

Additional comments

When inspecting GeoGen archives, you can set several options (most of which apply when a new archive is read):

Description of constructions: Can be set to *Symbolic* (see Figure 20 left) or *Narrative* (see Figure 20 right).

Bleach auxiliary objects: When this option is enabled, you can control the visibility of auxiliary objects with Alt+MouseScroll. Some figures can have several overlapping auxiliary objects, this can cause nuisance for some operations if such a construction is imported into OK Geometry. The X button in the upper right corner of the OK Geometry window can be used to ignore auxiliary objects.

Emphasise theorem objects: When this option is enabled, you can control the emphasis of theorem objects with Shift+MouseScroll.
Include source data of constructions: If this option is set, the descriptions of the configurations will contain the information from which GeoGen archive and which example in the archive is the construction from.

Generate also simple problems: When this option is enabled, GeoGen generates a large archive with prefix 'output_' and extension '.txt'. The archive contains besides non-trivial problems (with relevance > 10 000) also easier tasks (with relevance ≥ 9200). The relevance of simple problems is related to the number of inferences GeoGen used in the proof.

Show potential ambiguities in theorems: When this option is enabled, you will be warned that the considered property may not be valid generally. It may even occur that the displayed property is false in the visualised example. OK Geometry will also display possible alternatives for the property.

Relevance limit: Only tasks-configurations for which the relevance exceeds this limit will be displayed.

There is a certain similarity in the creation of GeoGen archives of non-trivial proof tasks and generic constructions used in OK Geometry. But the similarity is rather superficial. Generic constructions are families of constructionally isomorphic configurations based on a large number of admissible operations. In such families one can observe and search for different properties. Unless the searched properties are bound to specific instances of objects, the observations are slow. On the other hand, GeoGen can deal with larger families of configurations that are freely generated with a limited number of specific operations. GeoGen observes fewer properties, but takes into account all objects. GeoGen is not only is very fast in observing, but determines which observed facts of configurations are non-trivial by proving or attempting to prove the observed facts.

6. JGEX connections