An observation tool as an aid for building proofs

Zlatan Magajna
e-mail: Zlatan.magajna@pef.uni-lj.si
University of Ljubljana
Faculty of Education
Ljubljana, Slovenia

Abstract

School geometry is endowed with an empirical and a deductive aspect. Though in history their role and importance varied, it is probably out of question that both aspects are essential for learning and knowing in geometry. Dynamic geometry systems by their nature favour the empirical aspect of geometry and promote conceptual understanding. We propose a different type of software that is aimed at promoting the deductive aspect of geometry. In simple terms, this software analyses geometric configurations (constructed in a dynamic geometry environment) and brings to light its geometric properties. The student can focus on selecting the relevant properties, organise the selected properties into a proof and supply the deductive arguments. We present the basic functionalities that we found useful or necessary for such type of software. Further research will show whether such approach improves the students' ability to make deductive argumentations.

1. Proofs in school geometry

A glance to aims of learning mathematics in secondary schools in curricular documents in any country shows that students should learn the principles of deductive reasoning. Most mathematics teachers will agree with Rowlands ([8]) that "geometry can provide an ideal venue for inducting students into proof, the formalism of mathematics and to encourage them to think as mathematicians". Herbst ([4]) shows that in the American schools the role of proof in school mathematics and the way in which proofs are incorporated in school mathematics have gone through a series of significant transformations. In the mid 19th century school geometry consisted of replicating proofs. Only later proofs were given a didactic value (in the sense that authors tried to make them elegant and easy to understand). Still later (in the beginning of 20th century student became engaged in doing proofs by themselves. Only at this stage exercises that required building up proofs and two column proofs became part of school mathematics. A side effect of making proofs as a separate object of study was the separation of practices of proving from the practices of knowing. Herbs claims that proof "is essential in mathematics education not only as a valuable process for students to engage in (such as developing the capacity for mathematical reasoning) but, more important, as a necessary aspect of knowledge construction".

In what way are proofs related to knowledge construction? Proofs in school mathematics are not just sequences of logical deductions that serve as an irrefutable verification of the truth of a statement. Hanna ([3]) lists several functions of proofs and proving: verification (concerned with the truth of a statement); explanation (providing insight into why it is true); systematisation (the organisation of various results); discovery (the discovery or invention of new results); communication (the transmission of mathematical knowledge); construction of an empirical theory; exploration of the meaning of a definition or the consequences of an assumption; incorporation of a fact into a new framework and viewing it from a fresh perspective. This holds for mathematics in general and, even more, for school mathematics.

A proof of a geometric fact may serve the above functions to a different extent. Let us consider, for example a simple fact from geometry: the three medians of a triangle meet in a common point. A synthetic proof of this fact relies on previously learnt facts about triangles and is quite explanatory. A proof, which is based on algebraic manipulation of coordinates in Cartesian plane, is less explanatory (but connects geometry to algebra). Finally, think of an automated proof, i.e. a proof that is automatically generated with a computer software program. Of course one needs to formulate the involved geometric objects and properties as algebraic entities in Cartesian plane. The software for automated proving, if the process is successful, using advanced algebraic machinery (e.g. Groebner basis of polynomials) de facto proves the theorem ([10]). Certainly, in secondary schools, besides the undeniable verification function, such a proof has little didactic value.

Students consider proofs as a difficult part of mathematics and, as stated above, they often do not see proving as an aspect of knowledge construction. In other words, they are not aware of the explanatory, systematisation and other functions of proofs and proving. A simple interview with students in secondary schools reveals that most students find difficult to grasp the meaning of proofs, they find intriguing where proofs ’come from’ and how can one build up a proof. Several approaches have been suggested to overcome these difficulties. Brandell ([1]), for example suggested the introduction of a sort of flowchart form as an intermediate step towards the paragraph form of proofs.

Dynamic geometry systems gave rise to new ways of presenting geometry and, consequently, to new approaches in learning and doing geometry ([9]). These software tools turned out to be very effective for executing certain general geometric constructions, for generating conceptual understanding and for exploring geometric configurations. In short, they are magnificent exploratory tools and an excellent aid for gaining conceptual knowledge. But do dynamic geometry systems also promote deductive reasoning? Are they of any help in proving theorems? The answer is not simple. First, by their nature, dynamic geometry systems favour the empirical (and not the formal, deductive) aspect of geometry. Second, the documentation ability and the dynamic nature of the objects make these systems very appropriate tools for presentation of proofs. Third, by promoting conceptual understanding these systems implicitly promote also the ability to build up proofs. On the other hand the empirical nature of these systems does not suit well the formal deductive nature of proofs and proving. Several authors find dynamic geometry system a useful means for proving theorems. A possible theoretical model that describes how this can be done is the Toulmin argumentation model ([7]). The process of building a proof is a continuous interplay of conjecture production (from the empirical observation) and proof production. Dynamic geometry system may serve as a means for empirical observation as well as a source for the warrant of the claimed conjecture. A similar approach was used by Marrades and Gutierez ([6]) in their research of proof production.

2. A different approach to proofs

We present an approach aimed to help student in building up proofs of geometry theorems. Usual geometry tasks that require proving can be considered as problems with given initial conditions (hypotheses, what is given) and a goal (what is to be proved or what is to be constructed). To solve a task, e.g. to prove a theorem means to find a sequence of evidences connecting the initial conditions to the goal. Similarly, the solution of a construction task is, basically, a sequence of construction steps connecting the given data to the required construction. In terms of information theory ([5]) one needs to find in problem space a path from the initial state to the goal. Thus, in principle, solving a geometry task requires that one delineates the problem space and then finds appropriate deductive steps in problem space from the initial state to the goal. In other words, a solver should be aware of 'potentially true and relevant' facts related to a geometric configuration (we may think of this as an analysis step). Then using some strategy the solver selects some facts and puts them in an order so that each steps follows deductively from previously established facts (Figure 1).

A good problem solver is able to use a variety of strategies for selecting and connecting facts. Furthermore, a good solver has also a rich knowledge base. This means that s/he is able to recall many facts related to a geometric configuration and that in his knowledge base these facts are structured in several relevant and specific ways. On the other hand, a poor solver lacks all the mentioned abilities. In practice it means that given a geometric configuration a poor solver is able recall a only a few related theorems, is aware of a few facts s/he considers true, and can think of a few facts that s/he considers to be possibly true. Thus, a poor problem space prevents a student to move on. A poor problem space is thus also an obstacle for developing strategies of connecting facts in problem space, to structure the facts in knowledge base, and to develop deductive reasoning (Figure 1).

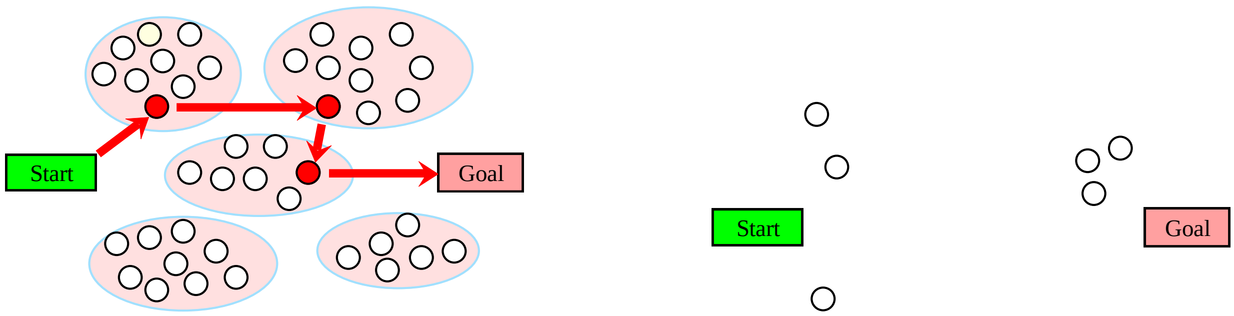

Figure 1. *Schematic representation of problem spaces. The problem space of a good problem solver (left) is reach and well structured. A poor and unstructured problem space (right) is an obstacle for solving problems.*

In order to help the student in the solving process we propose the following approach[1]. Given a (planar) geometry problem a student constructs the configuration using dynamic geometry software (Cabri, Geogebra or an internal geometry editor). To overcome the obstacle of unsuitable problem space the solver asks the software for a help in observation. The software produces a rich list of facts related to the studied configuration. To move on the solver has to ignore the facts s/he finds trivial or irrelevant to his tasks. In other words, the solver constructs the problem space not by bringing to mind facts (that, indeed, s/he cannot think of), but rather by selecting facts from the list offered by the software. Then s/he has to connect the facts by deductive arguments. The process is illustrated in Figure 2 and Figure 3. Note that in order to facilitate the process a teacher himself may select the relevant properties, so that a student is focused just on organising the properties and give deductive arguments.

[1] The described software is called OK Geometry (Observing and Knowing in Geometry) and was developed by the author. The Beta version is available on www.z-maga.si. The software is freeware.

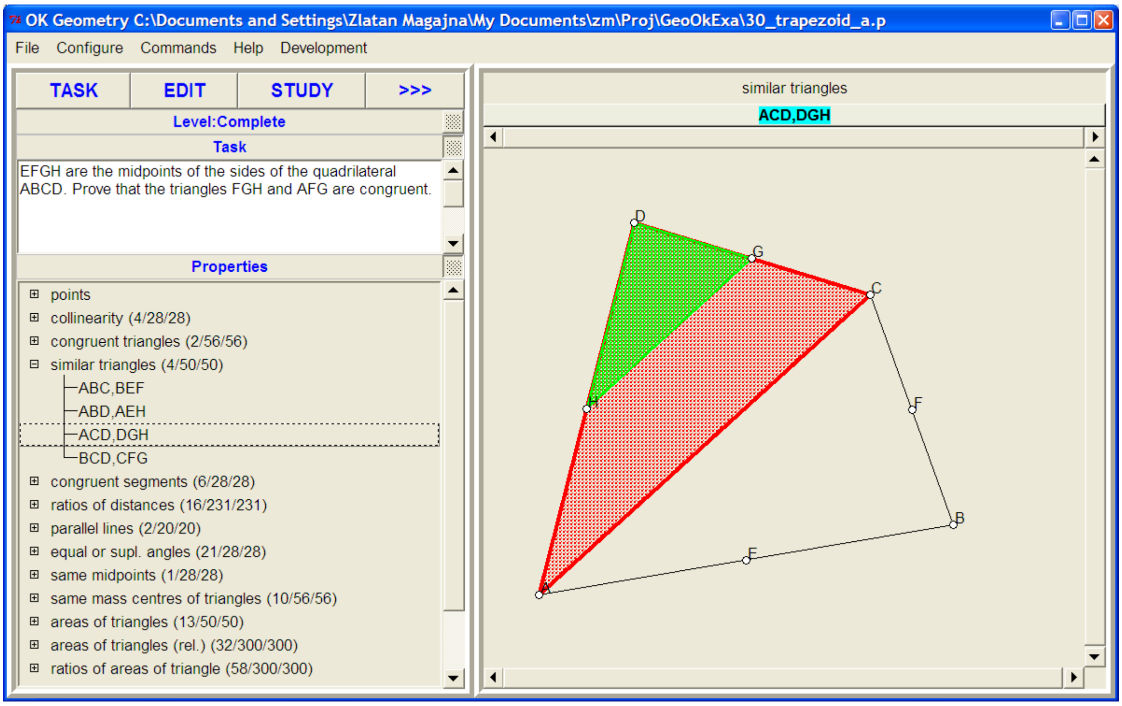

Figure 2. *The first phase of the described approach in proving geometry theorems. A geometric construction is imported from a dynamic geometry system. The software produces a list of properties of the considered construction.*

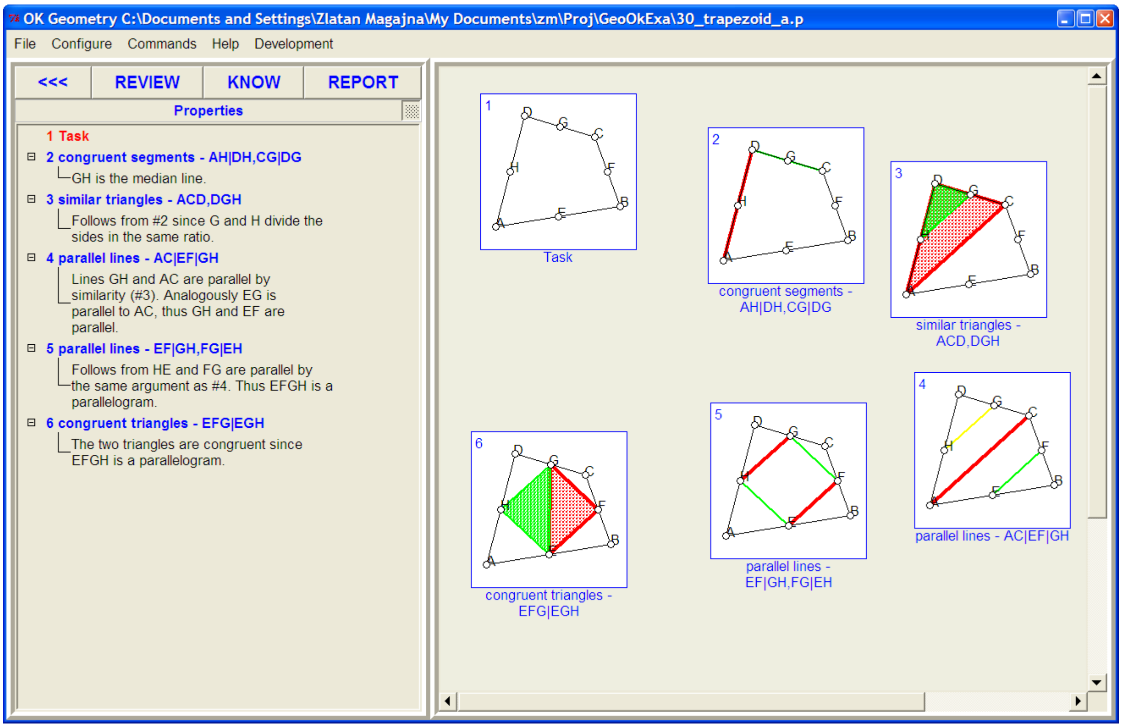

Figure 3. *The second phase of the described approach in proving geometry theorems. The student selects the properties s/he finds relevant, organises them and supplies deductive arguments.*

The produced list of properties of a geometric configuration may be used also for other purposes. One may simply look for interesting (novel) properties, for generating geometry tasks (though this is not the aim of the software), or just to recall facts in geometry.

3. Using a computer software for delineating the problem space

We focus now on problems related to the detection and presentation of properties of geometric constructions. We shall present some aspects of the analysis that we found helpful or even necessary in such type of software.

3.1. Reducing the complexity

According to Gestalt psychologist humans have the ability (based on certain principles) to simplify and organise perceptions and make them meaningful. As it turns out, even simple geometric configuration (e.g. consisting of a few lines, circles, implicitly or explicitly defined points), are quite complex and may easily involve hundreds of properties. Most of them are obvious, are completely irrelevant, and an observer is not even aware of them. If a software program would state them this would not help a solver. We found that simple principles are necessary in order to reduce the unnecessary complexity. In this respect we found that an easy and useful principle is to focus just on selected points in the configuration. Given a geometric configuration the user states which points should be taken into consideration in the analysis (by appropriately labelling such points). The software then considers all possible lines, circles, etc. defined by the selected points, and looks for properties among these object (e.g. that some points are collinear, that certain triangles have the same area, etc. It is very important that during the analysis the user can repeatedly change the points to be used in the analysis. A reasonable number of points to be considered at time is 5 to 12, depending on the configuration. Another advantage of reducing the analysis to declared (selected) points is the possibility of symbolic notation of objects. Given, say 8 points in general position, there exists tens of lines and circles through these points (whether drawn or not) and hundreds of angles defined by the points. A notation not based on points on objects would be very hard to follow.

3.2. Level of expertise

If a software is used for analysing geometric configurations at various levels of schooling then a mechanism for setting the level of expertise (analogously to dynamic geometry software) is required. We have decided to set four levels of expertise (Figure 4). The Simple level basically looks for basic properties (e.g. collinearity of points) and for isometric invariants (e.g. congruency of segments, angles, circles, triangles). The Medium level takes into account also ratio and similarity (e.g. finds that the ratio of two lengths of segments is 1 : 2-sqrt(3) or that two triangles are similar). The Advanced level looks also for some projective invariants (e.g. points lying on a same conic, quadruples of points with the same cross-ratio). The Complete level looks also for rather specific properties that are normally not considered. Furthermore, there is a User level, where users can specify exactly which properties they are is interested in.

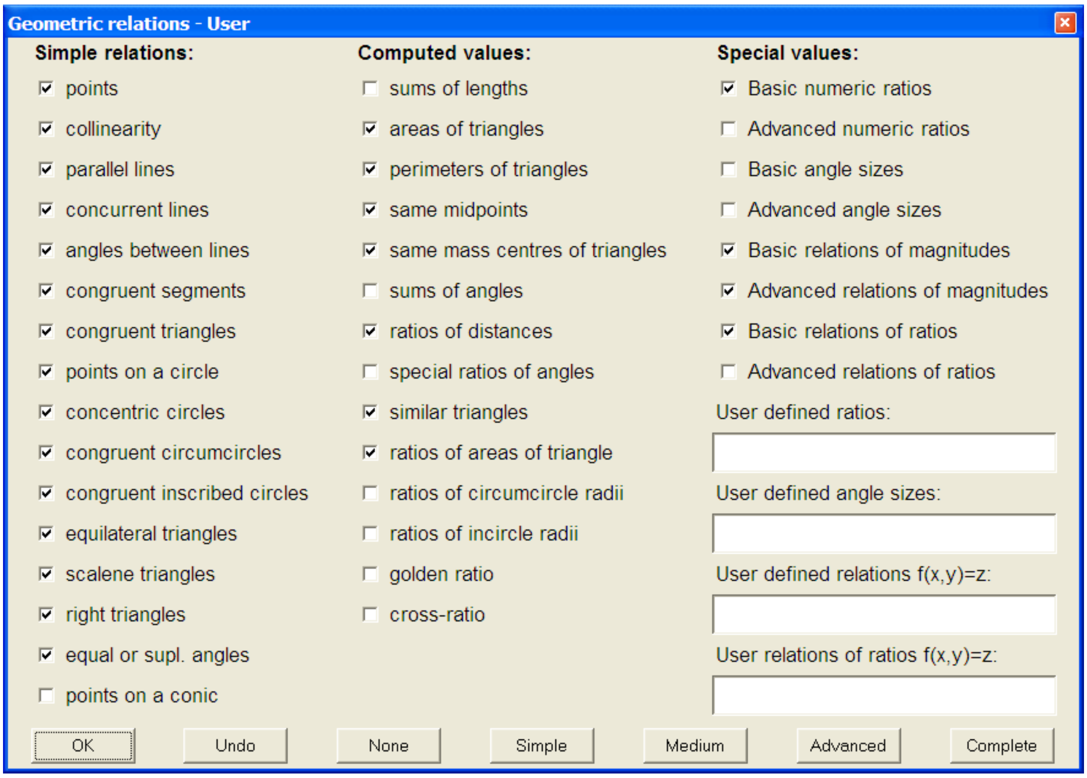

Figure 4. *Setting the level of expertise*

3.3. Structuring the facts

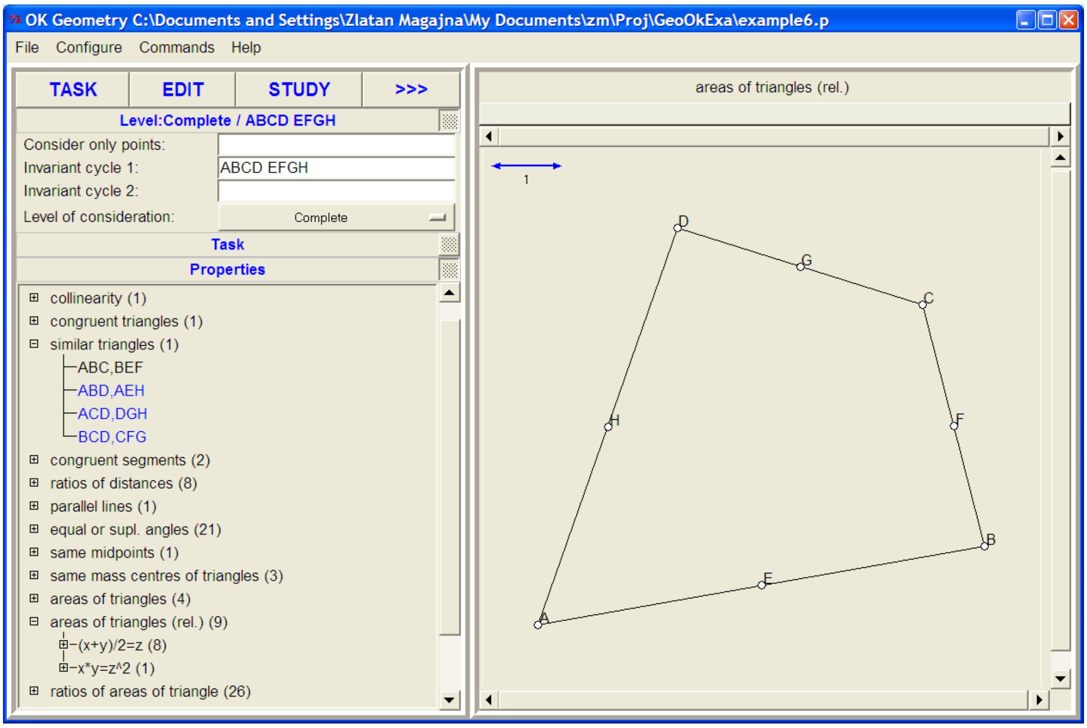

Figure 5. *Structuring the properties for easier access*

An ordinary geometric configuration (based on 5 to 12 points) gives rise to tens or even hundreds of properties, thus it is necessary to structure them. This can be done in several ways. We have opted for a rather formal hierarchical grouping of properties. For example, all instances of collinear points and all instances of similar triangles are grouped together. Thus we first decide which properties we are interested in, and then find an appropriate instance of the property (Figure 5). Note that other methods of structuring properties are possible (perhaps based on subsets of points or based on logical content of properties), but we have opted for an organisation that allows an easy inspection of properties.

3.4. Truthfulness of presented facts

This aspect is central from both, didactic and technical aspect of the software. We state clearly that in order to detect the properties of geometric configurations the considered software relies on algorithms that are essentially based on numeric computations (though also algorithms for text manipulation are extensively used). In other words, the software just very accurately ‘observes’ the configuration. For example (in principle) it ‘finds’ that two line segments are congruent by detecting that the lengths of the two segments differ by less than a very tiny amount. This characteristic of the software requires a didactic and a technical comment.

The aim of the software is not to prove facts but rather to help the student to make sense of the configuration, to make him aware of geometric properties, and in this way to help him to build a proof and to develop deductive reasoning. Students should be aware that observing something is not the same as knowing something (the name of the software, OK Geometry, is an acronym for Observing and Knowing). It is up to the student to establish the truthfulness by logical arguments. The students should be well aware that they should not take as granted the properties listed by the software. This may sound as weakness of the software, but we prefer to treat this as a didactic strength.

From the technical point of view it should be stressed that geometric properties are not detected just by relying on geometric drawings, but rather on several instances of dynamic constructions. Using this approach (and some other techniques) we significantly reduce the possibility that a property in a dynamic construction is observed by chance.

3.5. Requiring the invariance of properties

Specifying the required invariance of geometric properties may significantly reduce the complexity of geometric configurations, additionally structure the found properties, and also additionally increase the reliability of the results. Let us illustrate this with a simple and well known example. Consider a quadrilateral ABCD and the midpoints E,F,G,H of its sides (see Figure 5). From the construction it follows that we are interested in properties that are invariant with respect to the cyclic permutation (ABCD)(EFGH) of the labels. The software lists only such properties. For example, there are 4 instances of pairs of similar triangles in the configuration, but they are considered as a single one since each can be derived from the any other with a cyclic permutation of the point labels (see Figure 5).

3.6. Implicit constructions

In order to study a geometric construction with the considered software it is necessary to execute the construction in a dynamic geometry system. Sometimes we simply do not know how to do this. Here are two simple examples:

> *In a given acute triangle we would like to inscribe a triangle with minimal perimeter (on each side of the triangle ABC there should be one vertex of the inscribed triangle, see Figure 6 left).*

Or:

> *For a given triangle ABC the pedal triangle of a point P is the triangle whose vertices are the feet of the perpendiculars from P to the side lines of the triangle ABC. Is it possible (for some point P) that P is the centre of the inscribed circle of its pedal triangle? (See* Figure *6 right.)*

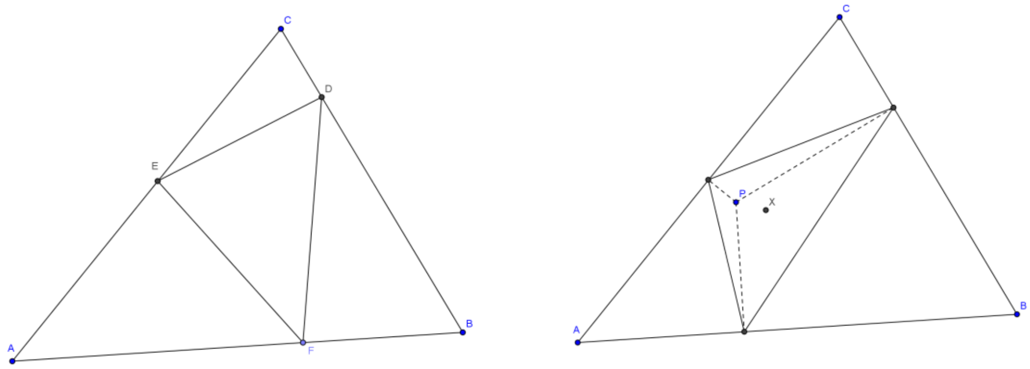

Figure 6. *Examples of implicit constructions*

Both constructions are not trivial. In the first case one does not know how where to put the vertices along the sides, and in the second case one does not know where to place the point P. In order to study such problems the software allows implicit constructions. We cannot enter into details here, we give just a rough description. Basically, one makes a dynamic construction that is not correct in the sense that not all the constraints are met. The computer software then tries to modify elements in the construction so that all the constraints are fulfilled. For example, in the second task one may draw a point P, a pedal triangle of the point, and the centre X of the inscribed circle of the pedal triangle (Figure 6, right). The point X, in general, will not coincide with the point P. Using an appropriate command the solver asks (the computer software) to position the point P so that P and X coincide.

There is nothing mystical in implicit constructions. The computer program tries to make the implicit construction using some sort of numeric optimisation. The method is, indeed, irrelevant, what counts is that the optimisation task is stated in terms of a (dynamic) geometric language (no need to translate the problem in algebraic terms). Also, note that the implicit construction (more precisely: the found configuration) itself is not relevant, what counts are the listed properties of the found configuration. These properties may help to find an ordinary construction or even to prove the correctness of the solution. When solving the second task, for example, the software is able to position the point P s that coincides with the point X. The listed properties of the found configuration clearly indicate that the required position for P is the orthocentre of the triangle ABC. Also, using the listed properties, it is rather easy to prove this.

3.7. Properties search

This is another useful function when studying a configuration. Often we are interested in properties of some specific objects in a configuration. One may focus on a triangle in a configuration and question whether it has some important properties. Or one may look for properties that involve two given line segments. Note that the software may give a list of tens or even hundreds of properties and it is not easy to spot those that involve specific objects. Thus, search functionality is of great help.

4. Conclusion

There are two systems of reasoning in (school) geometry: one is based on empirical observation, and the other is based on deductive argumentation. Their role and relevance changes with time, but probably it is out of question that both aspects of geometry are essential and mutually support each other. Dynamic geometry systems by their nature favour the empirical aspect of geometry and also promote conceptual understanding. We propose a different type of software for promoting the deductive aspect of geometry. In simple terms, this software brings to light the properties of a given geometric configuration. In this way the student can focus on selecting the relevant properties, organise the selected properties into a proof and supply the deductive arguments. In such approach we see the potential for developing students' deductive reasoning and for promoting the perception that the geometric properties of a configuration form a logically organised structure. Further research will show to what extent this is true.

5. References

[1] Brandell, J. L. (1994). Helping Students Write Paragraph Proofs in Geometry, *The Mathematics Teacher,* Volume 87(7), Number 7, October, pp. 498- 502.

[2] Edwards, L. D. (1997). Exploring the Territory Before Proof: Student's Generalizations in a Computer Microworld for Transformation Geometry. *International Journal of Computers for Mathematical Learning*, Volume 2, Number 3, pp. 187-215.

[3] Hanna, G. (2000). Proof, Explanation and Exploration: An Overview. *Educational Studies in Mathematics*, Volume 44, Numbers 1-2, pp. 5-23.

[4] Herbst, P. G. (2002). Establishing a Custom of Proving in American School Geometry: Evolution of the Two-column Proof in the Early Twentieth Century. *Educational Studies in Mathematics,* Volume 49, Number 3, pp. 283-312.

[5] Kahney, H. (1993). *Problem Solving: Current Issues*. Buckingham: Open University Press.

[6] Marades, J., Guttierrez, A. (2000). Proofs produced by secondary school students learning geometry in a dynamic computer environment. *Educational Studies in Mathematics,* Volume 44, Numbers 1-2, pp. 87-125.

[7] Pedemonte, B. (2007). How can the Relationship Between Argumentation and Proof be Analysed?. *Educational Studies in Mathematics*, Volume 66, Number 1, pp. 23-41.

[8] Rowlands, S. (2010). A Pilot Study of a Cultural-Historical Approach to Teaching Geometry, *Science & Education*, Volume 19, Number 1, pp. 55-73.

[9] Straesser, R. (2001). Cabri-géomètre: Does Dynamic Geometry Software (DGS) Change Geometry and its Teaching and Learning?. *International Journal of Computers for Mathematical Learning*, Volume 6, Number 3, pp. 319-333.

[10] Wu, W.T. (1994). *Mechanical Theorem Proving in Geometries*. Springer: Vienna.

[11] Yiu, P. (2005). Elegant Geometric Constructions, *Forum Geometricorum*, Volume 5, pp. 75-96.