PERGAMON

Computers & Education 38 (2002) 21–35

COMPUTERS & EDUCATION

www.elsevier.com/locate/compedu

A dynamic–symbolic interface for geometric theorem discovery

Francisco Botana[a,*], José L. Valcarce[b]

[a]Department of Applied Mathematics, EUET Forestal, University of Vigo at Pontevedra, Campus A Xunqueira, 36005 Pontevedra, Spain

[b]Department of Mathematics, IES Pontepedriña, 15701 Santiago de Compostela, Spain

Abstract

This paper describes Discover, a program for learning and teaching geometry with the help of a computer. The program is a dynamic geometry environment that can communicate with Mathematica, using its symbolic capabilities to perform geometric discovery or rediscovery. Discover is specially suited to be used as a learning tool for geometry from the ages of 12 up to University. It permits the replacement of the traditional ruler and compass by electronic substitutes, as in standard dynamic geometry environments. Through its link with the computer algebra software, it enhances the process of conjecturing and proving. The results can be expressed in natural language or through the use of equations. The mathematical methods that Discover uses are sound, although not complete. Despite this last fact, almost all parts of the school curricula in plane geometry can be adequately treated with the program.

Keywords: Human–computer interface; Interactive learning environments; Media in education; Dynamic geometry; Symbolic methods

1. Introduction

Since the 1990s dynamic geometry systems (DGS) have been increasingly used for teaching, mainly in secondary schools. As the logarithmic tables and the slide rules had been universally replaced by pocket calculators and computers, the traditional Euclidean tools, the ruler and the compass, are still being substituted by virtual tools in computers. Although there is a wide variety of programs for computer drawing, most of them are not suitable in the study of geometry or geometric-based disciplines. Simple interactive illustration systems such as Paint or MacDraw or sophisticated CAD software work fine when producing single static pictures, but they are inadequate

* Corresponding author. Fax: +34-986-801-907.
E-mail addresses: fbotana@uvigo.es (F. Botana), jvalcarce@edu.xunta.es (J.L. Valcarce).

0360-1315/02/$ - see front matter
PII: S0360-1315(01)00089-6

tools for studying the geometric invariants. You can test, for example, the ratio 1:2 defined by the centroid in a median of a triangle. While you drag a vertex the triangle is continuously deforming, but DGS lets the centroid and the median remain well defined. This fact is certainly essential for feeling that this ratio is constant in any triangle.

A DGS provides certain primitive objects (e.g. points, lines, circles) and basic tools (e.g. parallel to a line through a point, circle with its centre in a point passing through another point) for assembling these objects into the constructions. The main contribution of DGS is the dragging feature. Once a construction is completed, the user drags certain elements of it and the whole construction behaves in such a way that specified constraints are maintained. Besides the dragging feature, automatic loci generation and some mechanisms to record the history of the constructions are standard options in most DGS.

Current implementations of DGS include The Geometer's Sketchpad (Jackiw, 1995), Cabri Geometry (Baulac, Bellemain, & Laborde, 1994), Cinderella (Richter-Gebert & Kortenkamp, 1999) and Geometry Expert (Gao, Zhu, & Chou, 1998). The program, Discover is a standard DGS that can communicate with a computer algebra system (CAS), Mathematica. In this way, common numerical approaches to dynamic geometry can be complemented with the symbolic capabilities of CAS, allowing a step forward for discovery and proof in geometry.

2. Proof and discovery in dynamic geometry systems

Although proof is a central concept in mathematics, some attempts to relegate it to a lesser role in the secondary curricula have been reported (Hanna, 1997; Tall, 1995). It can also be argued that visual evidence will replace proof of theorems, but empirical findings support the opposite situation (Jiang & McClintock, 1997). The accuracy of constructions done with any DGS easily allows to reject statements based in erroneous figures. A paradigmatic illustration is the century-old fallacious theorem stating that all triangles are isosceles (Maxwell, 1959), that exhibits a correct reasoning over an impossible figure: D, the intersection of the angle bisector m of A and the perpendicular bisector n of BC, is never inside ΔABC (Fig. 1).

Another proof-related issue in DGS has been highlighted in Goldenberg and Cuoco (1998). It deals with the difference between Euclidean theorems and DGS design properties and it can be

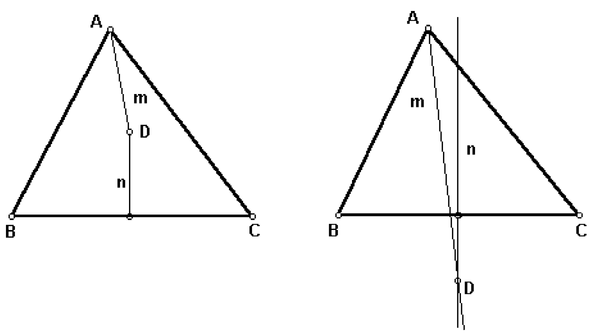

Fig. 1. (a) A malicious figure for proving that all triangles are isosceles. (b) The construction in a dynamic geometry system (DGS).

illustrated through a ratio invariance. Construct (Fig. 2, left) a segment *AB*, a point *C* on it and measure the ratio *AC:BC*. Moving an endpoint, say *A*, this ratio remains constant in the majority of DGS. Now construct (Fig. 2, right) a triangle *ABC*, a point *D* on segment *AB*, the segment *DE* parallel to *AC* and measure the ratios *BD:BE* and *BA:BC*. They are equal and they so remain when moving *D* along line *AB*. Although this last result is an Euclidean theorem about similar triangles, the first one is due to a developer decision. And they cannot be distinguished by a student using the standard calculator tool within current DGS.

A common approach to proof in DGS consists of giving the students a linguistic description of a construction and asking them to investigate some property. An archetypal example is the Simson theorem. The construction can be described by the following sentences: "Inscribe a triangle in a circle. Select a point on the circle and draw perpendiculars from the point to each side of the triangle. Draw a line through two of the three feet of the perpendiculars", and the proposed investigation is making a conjecture about the foot of the third perpendicular with respect to the line segment (Fig. 3, left). Any DGS (in fact, all of those that we have tested) will convince the student that the three feet are collinear, for any possible position of the vertices of the triangle. This kind of conviction, generically referred to as a visual proof, can be even gone farther with some of the cited DGS. Cinderella, using randomized theorem checking, shows the message "The

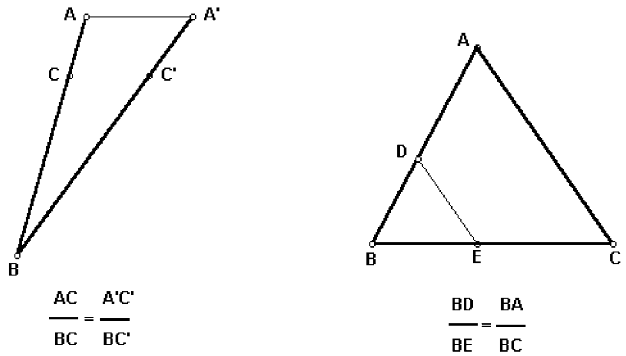

Fig. 2. (a) The ratio *AC:BC* is constant due to the dynamic geometry system (DGS) design. (b) The equality of ratios *BD:BE* and *BA:BC* is an Euclidean theorem.

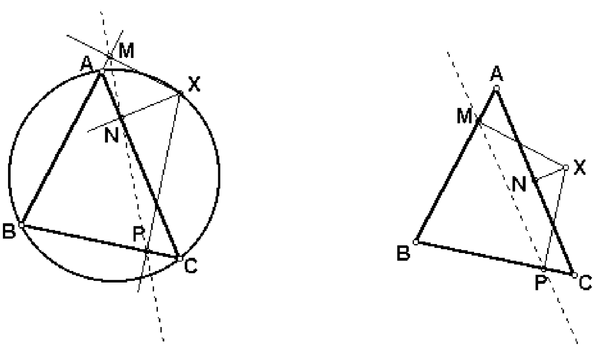

Fig. 3. (a) A visual proof of Simson theorem. (b) Finding the locus of *X* for which *M*,*N* and *P* are collinear is a hard task in dynamic geometry system (DGS).

third_foot lies on line *first_foot-second_foot*" when drawing the line through two of the three feet of the perpendiculars. Also Cabri detects this alignment if using its property checker on the three feet. Finally, Geometry Expert can give a formal proof of the statement. But let us consider this situation with a light difference. Without knowing in advance the Simson theorem, investigate the position of a point for which the feet of the perpendiculars to each side of a triangle are collinear (Fig. 3, right).

None of the three programs, or the others, will allow an ordinary student to discover that the locus is the circumcircle of the triangle. Worst of all, the property checker of Cabri will refute the visual evidence that the three feet are collinear if the point lies on the circumcircle (at least until the user redefines the point to be on the circle; Fig. 4).

We think that a capital enhancement for a DGS would be a method allowing not only theorem proving but mainly theorem discovery. With respect to the Simson theorem, a DGS capable of finding a fact like "A necessary condition for the collinearity of the feet of the perpendiculars is that the vertices of the triangle and the point are on a circle" would be, without doubt, an interesting and profitable program. It could be used according recent tendencies that emphasize cognitive processes and heuristic reasoning over axiomatic reasoning in geometry (Clemens & Battista, 1992; Tall, 1995).

3. A symbolic approach to geometric discovery

Considerable attention and efforts had been addressed to automatic proving theorems in elementary geometry. Most of the earlier systems were based on purely logical or axiomatic approaches (Gelernter, 1963; Nevins, 1975) until the 1980s when the use of constructive methods in computer algebra was proposed (Buchberger, 1985; Chou, 1987; Wu, 1994). Comparatively much less work has been done in the area of automatic discovery in geometry. A landmark in mathematical

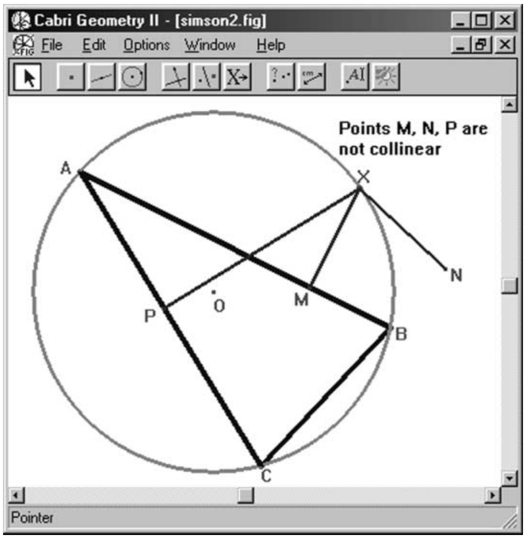

Fig. 4. Cabri refutes visual evidence in Simson theorem.

automatic discovery was the AM program (Lenat, 1984) but the use of algebraic geometric methods for discovery in geometry can be traced back to Kapur and Mundy (1989). Recently, Recio and Vélez (1999) have systematically addressed this issue using Groebner basis tools.

In order to describe the use of the Groebner method by Discover we discuss a simple example. A more mathematical development of the subject can be found in Botana and Valcarce (2001). The task consists of discovering the necessary conditions, if any, for which two vertices *A*,*B* of a triangle *ABC* and its circumcenter *O* are collinear (Fig. 5). Although the problem is almost trivial and can be easily done with any DGS, we just propose it for illustration.

The first step consists of making the construction. The user constructs three points *A*, *B* and *C*, and the perpendicular bisectors of sides *AB* and *BC* (some elements are hidden in the figure). Their meet is *O*, the center of the circumcircle, which is also drawn. The construction is internally described as follows:

A = FreePoint(U_1,U_2)
B = FreePoint(U_3,U_4)
C = FreePoint(U_5,U_6)
S_4 = Segment(A,B)
S_5 = Segment(B,C)
S_6 = Segment(A,C)
A_7 = MidPoint(S_4)
S_8 = Perpendicular(A_7, S_4)
A_9 = MidPoint(S_5)
S_{10} = Perpendicular(A_9, S_5)
O = Meet(S_8, S_{10})
C_{12} = Circle(O,A)

Once we get this geometric information, it is an easy task to translate it into an algebraic form. On one side we have six points (some of them are hidden). Three of them, *A*,*B*,*C*, are arbitrarily placed by the user on the screen, so their coordinates can have any value. No restriction will be made if the user or the system selects two of them as origin (*A*) and unit (*B*) of a reference, so simplifying the algebraic expressions. The coordinates of the other points are then completely known. Lets call them, for example, *A*7(X_1,X_2), *A*9(X_3,X_4) and *O*(X_5,X_6).

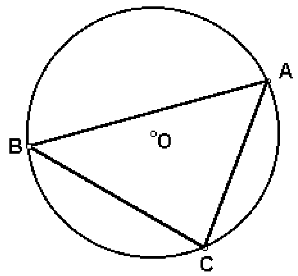

Fig. 5. When are *A*,*B* and *O* collinear?

On the other side, the constraints introduced by the user are translated into the following polynomial equations:

$X_1 = 1$
$X_2 = 0$; A_7 is the midpoint of AB
$X_3 = (1 + U_5)/2$
$X_4 = U_6/2$; A_9 is the midpoint of BC
$X_5 = X_1$; A_7O is perpendicular to AB
$U_6(X_4 - X_6) = (U_5 - 1)(X_5 - X_3)$; A_9O is perpendicular to BC

Finally, the user graphically imposes the condition of collinearity of points *A*,*B* and *O*, and its equation

$$X_6(1 - X_5) = -X_6X_5$$

is added to the polynomial system. The new system can be seen as a set of constraints satisfied by any triangle where two of its vertices and its circumcenter are collinear. Because the Xs coordinates values are known from any assignment for the Us coordinates, we look for an equivalent system without Xs, i.e. we try to eliminate them. This task, easy in this case but usually very hard, is accomplished with the CAS implementation of the Groebner basis method. The elimination returns

$$U_6^2 = U_5\,(1 - U_5)$$

meaning that the segments *AC* and *BC* are perpendicular.

4. The Discover system

Discover is designed to run under Windows 9x in any 486 machine or greater. It uses Mathematica 3.0 or greater, although due to the high cost of this program for many institutions we are planning to employ a free CAS, such as CoCoA (Capani, Niesi, & Robbiano, 2001). Discover has evolved from a standard DGS developed by the authors, mathematics teachers at University and secondary level, for use in the classroom and at home by our students and ourselves to become a DGS specialized in geometric discovery, used in teaching but also in our investigation. Currently we are planning to develop a study of its effectiveness and a comparison with Cabri and The Geometer's Sketchpad for classroom uses with 16 year students. The results of this work will be reported in a later work.

There are two main modules in Discover (Fig. 6): a DGS, written in Visual Prolog, and a Mathematica package. They intercommunicate via file reading and writing.

As already mentioned the DGS is a standard one, offering dragging capability, macro operations, loci and the general characteristics of this software. In addition, it can export the user-made constructions as a parametric code (a possibility that will probably also be offered by the release 4.0 of The Geometer's Sketchpad). The points and the constraints in a construction can be recorded in Mathematica or Maple algebraic format. Note that the system has enough knowledge to detect which type the coordinates of a point are (where U stands for a free coordinate and X for a bounded one).

When trying to discover geometric properties about a construction the user imposes conditions on it (Fig. 7a). Through the Discover dialog window, (Fig. 7b) he/she adds the relevant properties for the discovery (it is possible that a construction property would not be relevant: just imagine a new segment, without relation to the main figure, and its perpendicular bisector). Some of the points can also be added if a natural language statement of the discovered facts is wanted. With the chosen points a database of equations and linguistic sentences is made. All these data are passed to a package of Mathematica (Fig. 7c), which, acting as a black box for the case of elementary use, accomplishes symbolic tasks such as eliminating bounded variables, factoring the obtained equivalent system, matching each factor with the database, if it exists, and taking into account degenerated conditions, if any has been detected. Finally, the linguistic sentences that match some factor or the equations are returned to the DGS, where all the discovery findings are shown in a window (Fig. 7d).

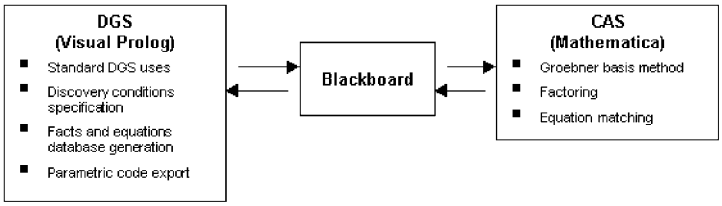

Fig. 6. The architecture and main tasks of Discover modules.

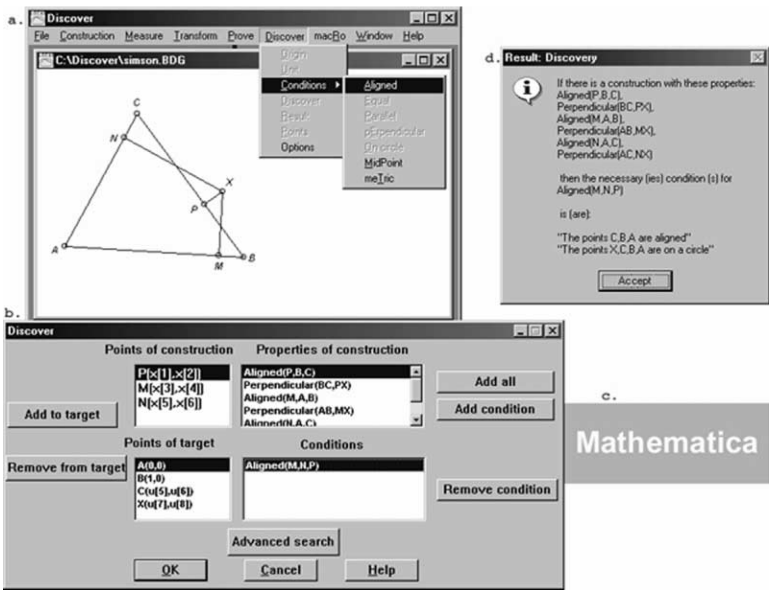

Fig. 7. An example illustrating discovery of Simson theorem.

5. Examples

Discover can (re)discover an impressive set of well-known Euclidean theorems and a lot of new geometric facts. We briefly mention some old and new theorems previously published, and a couple of, as far as we know, new results. We indicate in each of the following results the time used by Mathematica 3.0 in a Pentium II, that is, the elapsed time since we close the Discover window until the result is shown.

Theorem 1. *The diagonals of a quadrilateral meet at their midpoint if it is a parallelogram.*

We draw a quadrilateral $ABCD$, its diagonals AC and BD, and their intersection point M (Fig. 8 a).

Via menu Discover (Fig. 7a), we impose M to be the midpoint of diagonals AC and BD. In the Discover dialog window, we add both construction properties (Fig. 8b; note that the quadrilateral sides are merely graphic traces, not properties of the construction), and the discovered facts are two equations, meaning the parallelism of sides $AB-CD$ and $AD-BC$ (Fig. 8c). The time used by Mathematica is less than a second.

In order to get the parallelism statements, it would just be necessary to add the quadrilateral vertices also (Fig. 9). In this case the Mathematica time is about 4 s.

Theorem 2. *Given a triangle ABC and a point X on its plane, the necessary conditions for the collinearity of the symmetrical images of X with respect to the three sides of the triangle are: i) X lies on the circumcircle of ABC, or ii) the triangle ABC collapses to a line.*

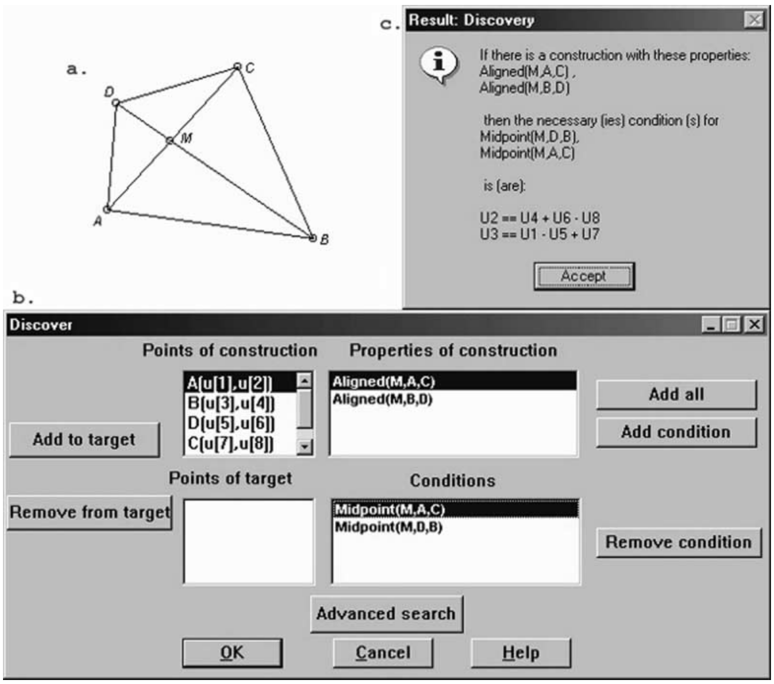

Fig. 8. The diagonals of a parallelogram meet at their midpoint: equations.

Once the construction has been made (Fig. 10a), the user imposes the alignment of the symmetrical points X_1, X_2 andX_3. Apart from adding the construction properties, the user also adds points A, B, C and X, and declares A as origin and B as unit (Fig. 10b). The program returns two linguistic conditions as necessary for the imposed alignment (Fig. 10c), with a Mathematica time of 5 s. Note that the first one is a degenerated condition, whereas the last one just states the cocircularity of A, B, C and X. In its current version, Discover has no semantic knowledge about facts like that the circle passing through the vertices of a triangle is called circumcircle.

Theorem 3. Pythagoras. *Given a triangle ABC, the area of the square drawn on BC is equal to the sum of those drawn on AC and AB if the angle BAC is a right angle or, equivalently, AC is perpendicular to AB.*

Just draw a triangle and impose the equality of the areas as metric condition (see the option in Fig. 7a). Since there are no properties and just three points, the settings in the Discover dialog window lead us to rediscover this well-known theorem, with less than 3 s of Mathematica time (Fig. 11).

Theorem 4 (Guzmán, 1999). *Given a triangle ABC, an arbitrary point X on its plane, and three projection directions, not all three equal, nor parallel to the sides of the triangle, let M, N, P be the projections along these directions on the sides. The locus of all points X such that the oriented triangle MNP has constant area is a conic.*

The construction in Fig. 12a shows the triangle ABC, the point X, three directions given by four unlabeled points near the left top, and the projections M, N and P. The construction properties

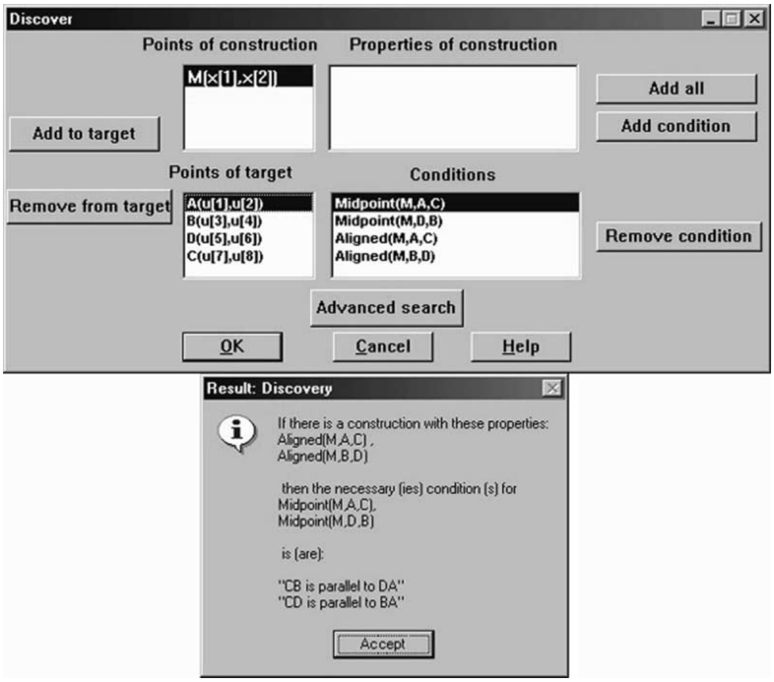

Fig. 9. The diagonals of a parallelogram meet at their midpoint: linguistic statements.

are listed in Table 1. Selecting *A* as origin and *B* as unit, and imposing as condition that the area of triangle *MPN* is 1, the program returns, after 9 s, as necessary condition an equation which would have 154 terms if expanded, and is reproduced in Table 2 as a curiosity.

In order to conclude that the locus of points X is a conic, the user must select X as the locus point in the Advanced search window (Fig. 12b). The discovered fact is then returned as shown in Fig. 12c.

New 1. *Given a quadrilateral ABCD, and the intersection point of its diagonals, M, a necessary condition for the equality of areas of the triangles AMD and BCM is the parallelism of sides AB and CD.*

Fig. 13 shows the construction (a) and the discovered condition (b). Here the time is about 3 s.

Table 1
Geometric properties in Guzmán theorem

Aligned(M,B,C)
Parallel(A8A9,MX)
Aligned(N,A,C)
Parallel(A10A8,NX)
Aligned(P,A,B)
Parallel(A11A8,PX)

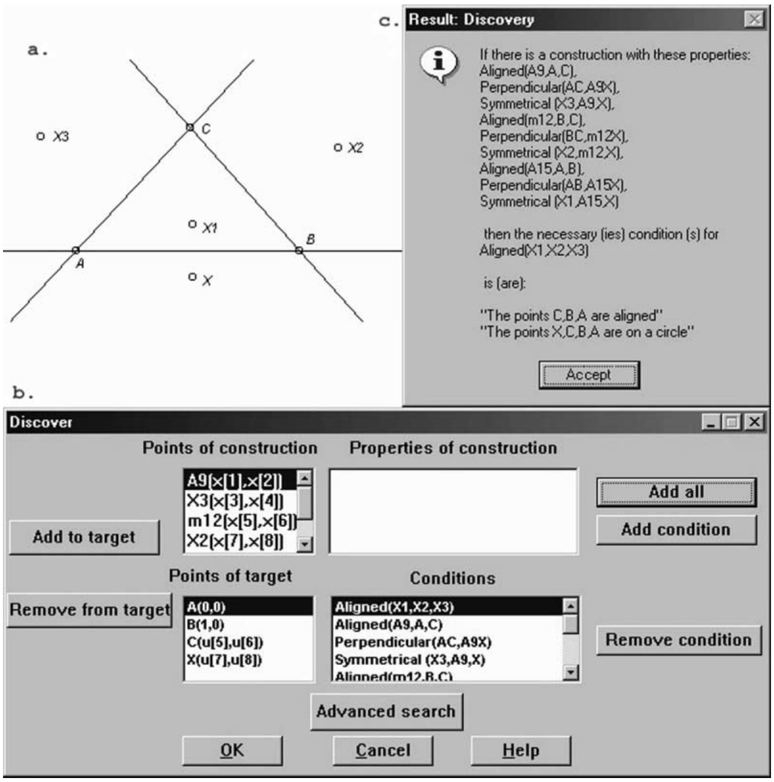

Fig. 10. Discovering conditions for the alignment of a point and its symmetrical images.

Table 2
The conic in Guzmán theorem

−2*U10∧3*U5 + 2*U10∧2*U12*U5 + 2*U10∧2*U14*U5−2*U10*U12*U14*U5 +
2*U10∧2*U16*U5−2*U10*U12*U16*U5−2*U10*U14*U16*U5 + 2*U12*U14*U16*U5 +
2*U10∧3*U5∧2−2*U10∧2*U12*U5∧2−2*U10∧2*U14*U5∧2 + 2*U10*U12*U14*U5∧2-
2*U10∧2*U16*U5∧2 + 2*U10*U12*U16*U5∧2 + 2*U10*U14*U16*U5∧2-
2*U12*U14*U16*U5∧2−2*U10∧2*U13*U6 + 2*U10*U12*U13*U6 + 2*U10*U13*U16*U6-
2*U12*U13*U16*U6 + 2*U10∧2*U13*U5*U6−2*U10*U12*U13*U5*U6-
2*U10*U13*U16*U5*U6 + 2*U12*U13*U16*U5*U6−U10∧2*U13*U6∧2*U7 +
U10*U12*U13*U6∧2*U7 + U10*U13*U16*U6∧2*U7−U12*U13*U16*U6∧2*U7 +
U10∧2*U13*U6∧2*U7∧2−U10*U12*U13*U6∧2*U7∧2−U10*U13*U16*U6∧2*U7∧2 +
U12*U13*U16*U6∧2*U7∧2 + U10∧2*U13*U5*U6*U8−U10*U12*U13*U5*U6*U8-
U10∧2*U15*U5*U6*U8 + U10*U12*U15*U5*U6*U8 + U10*U14*U15*U5*U6*U8-
U12*U14*U15*U5*U6*U8−U10*U13*U16*U5*U6*U8 + U12*U13*U16*U5*U6*U8-
U10*U13*U15*U6∧2*U8 + U12*U13*U15*U6∧2*U8 + U10∧2*U15*U6*U7*U8-
U10*U12*U15*U6*U7*U8−U10*U14*U15*U6*U7*U8 + U12*U14*U15*U6*U7*U8-
U10∧2*U13*U5*U6*U7*U8 + U10*U12*U13*U5*U6*U7*U8 + U10*U13*U16*U5*U6*U7*U8-
U12*U13*U16*U5*U6*U7*U8 + U10*U13*U15*U6∧2*U7*U8−U12*U13*U15*U6∧2*U7*U8 +
U10*U13*U15*U6*U8∧2−U12*U13*U15*U6*U8∧2−U10*U13*U15*U5*U6*U8∧2 +
U12*U13*U15*U5*U6*U8∧2 + 2*U10∧2*U6*U9−2*U10*U12*U6*U9-
2*U10*U16*U6*U9 + 2*U12*U16*U6*U9−4*U10∧2*U5*U6*U9 + 2*U10*U12*U5*U6*U9 +
2*U10*U14*U5*U6*U9 + 4*U10*U16*U5*U6*U9−2*U12*U16*U5*U6*U9-
2*U14*U16*U5*U6*U9−2*U10*U13*U6∧2*U9 + 2*U13*U16*U6∧2*U9-
U10*U12*U6∧2*U7*U9 + U10*U14*U6∧2*U7*U9 + U12*U16*U6∧2*U7*U9-
U14*U16*U6∧2*U7*U9 + U10*U12*U6∧2*U7∧2*U9−U10*U14*U6∧2*U7∧2*U9-
U12*U16*U6∧2*U7∧2*U9 + U14*U16*U6∧2*U7∧2*U9−U10*U14*U5*U6*U8*U9 +
U12*U14*U5*U6*U8*U9 + U10*U16*U5*U6*U8*U9−U12*U16*U5*U6*U8*U9-
U12*U13*U6∧2*U8*U9 + U10*U15*U6∧2*U8*U9−U12*U15*U6∧2*U8*U9 +
U13*U16*U6∧2*U8*U9 + U10*U12*U6*U7*U8*U9−U12*U14*U6*U7*U8*U9-
U10*U16*U6*U7*U8*U9 + U14*U16*U6*U7*U8*U9−U10*U12*U5*U6*U7*U8*U9 +
U10*U14*U5*U6*U7*U8*U9 + U12*U16*U5*U6*U7*U8*U9−U14*U16*U5*U6*U7*U8*U9-
U10*U13*U6∧2*U7*U8*U9 + U12*U13*U6∧2*U7*U8*U9 + U12*U15*U6∧2*U7*U8*U9-
U14*U15*U6∧2*U7*U8*U9 + U12*U13*U6*U8∧2*U9−U10*U15*U6*U8∧2*U9 +
U12*U15*U6*U8∧2*U9−U13*U16*U6*U8∧2*U9 + U10*U13*U5*U6*U8∧2*U9-
U12*U13*U5*U6*U8∧2*U9−U12*U15*U5*U6*U8∧2*U9 + U14*U15*U5*U6*U8∧2*U9 +
2*U10*U6∧2*U9∧2−2*U16*U6∧2*U9∧2 + U12*U6∧2*U8*U9∧2−U16*U6∧2*U8*U9∧2-
U12*U6∧2*U7*U8*U9∧2 + U14*U6∧2*U7*U8*U9∧2−U12*U6*U8∧2*U9∧2 +
U16*U6*U8∧2*U9∧2 + U12*U5*U6*U8∧2*U9∧2−U14*U5*U6*U8∧2*U9∧2 = =
U11*U6*(-2*U10∧2*U5 + 2*U10*U14*U5 + 2*U10*U16*U5−2*U14*U16*U5-
2*U10*U13*U6 + 2*U13*U16*U6−U10∧2*U6*U7 + U10*U14*U6*U7 +
U10*U16*U6*U7−U14*U16*U6*U7 + U10∧2*U6*U7∧2−U10*U14*U6*U7∧2-
U10*U16*U6*U7∧2 + U14*U16*U6*U7∧2−U10*U13*U6*U8 + U13*U16*U6*U8 +
U10∧2*U7*U8−U10*U14*U7*U8−U10*U16*U7*U8 + U14*U16*U7*U8-
U10∧2*U5*U7*U8 + U10*U14*U5*U7*U8 + U10*U16*U5*U7*U8−U14*U16*U5*U7*U8 +
U10*U15*U6*U7*U8−U14*U15*U6*U7*U8 + U10*U13*U8∧2−U13*U16*U8∧2-
U10*U15*U5*U8∧2 + U14*U15*U5*U8∧2 + 2*U10*U6*U9−2*U16*U6*U9 +
U10*U6*U8*U9−U16*U6*U8*U9−U10*U6*U7*U8*U9 + U14*U6*U7*U8*U9-
U10*U8∧2*U9 + U16*U8∧2*U9 + U10*U5*U8∧2*U9−U14*U5*U8∧2*U9)

New 2. *Given a quadrilateral ABCD, let R be the intersection point of the lines AB and CD, let S be the intersection point of the lines BC and DA, and M, N and P the midpoints of AC, BD and RS, respectively (The Gauss line theorem states that M, N and P are collinear). A necessary condition for P to be the midpoint of MN is that C lies on a conic passing through B and D.*

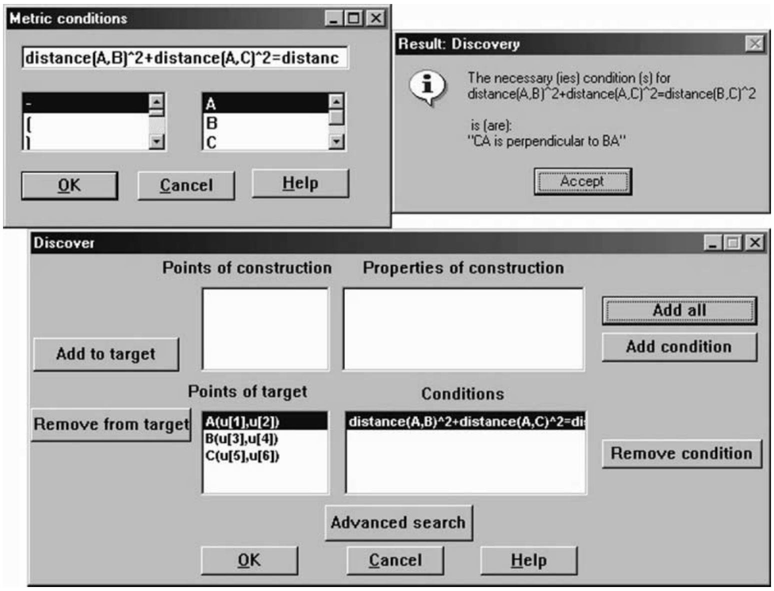

Fig. 11. Pythagoras theorem rediscovered.

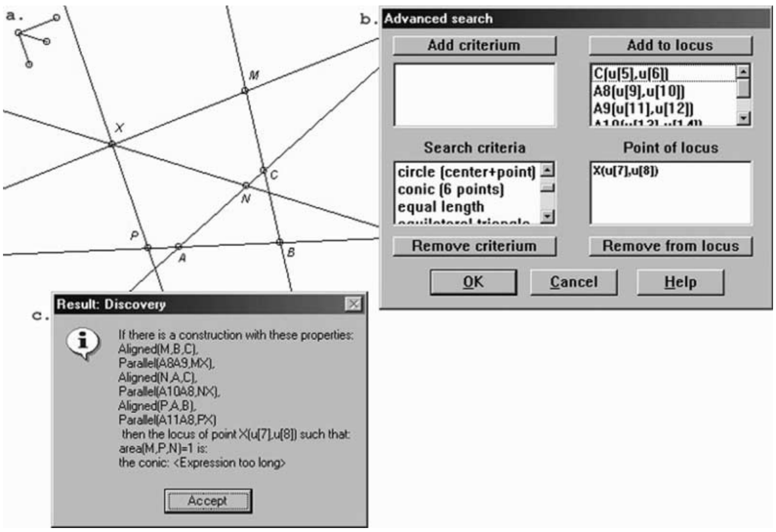

Fig. 12. A recent result generalizing Simson theorem.

Setting A as origin and B as unit in the construction in Fig. 14a, and imposing P to be the midpoint of MN, the program returns, within a second, as necessary condition the equation in Fig. 14b. No linguistic statement is produced in the actual version of Discovery.

So, this time some extra work must be done by the (advanced) student. The equation has four variables. They can be identified through a program option that lists the coordinates of the construction points (Fig. 14c).

The equation has degree two in the coordinates of C, that is, is a conic. Furthermore, the equation is satisfied when substituting the C coordinates by those of B or D, proving that the conic passes through these points. In fact, a similar analysis can be carried out changing C by D.

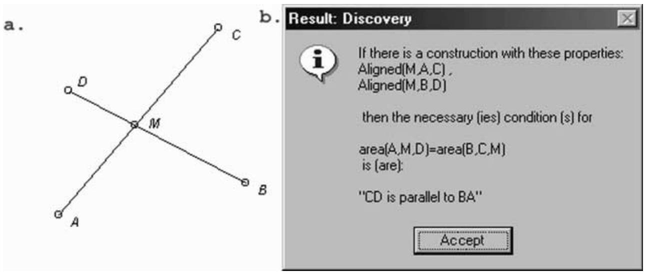

Fig. 13. A new theorem discovered by the program.

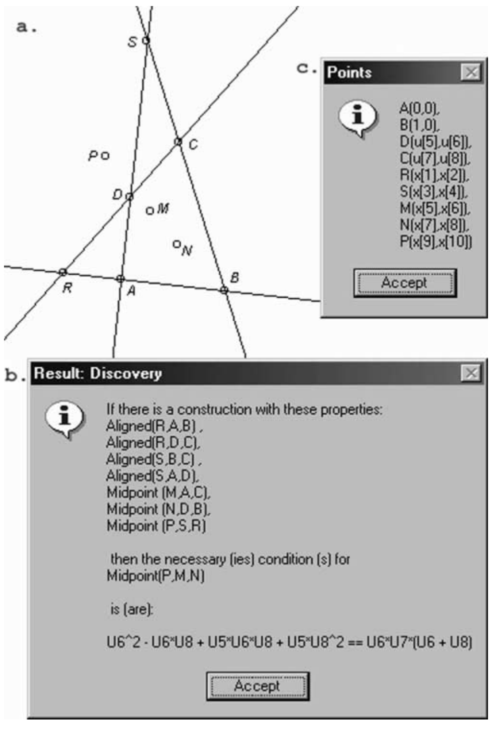

Fig. 14. A new property related to the Gauss line theorem.

In this case the necessary condition for P to be the midpoint of MN is that D lies on a conic passing through A and C.

Although the results obtained by Discover are sound, sometimes the program will not discover a quasi evident fact. Properties involving relations such as *between* cannot be described, in general, by equations, so prohibiting their study.

6. Conclusions

We have developed an interface between a dynamic geometry environment and a computer algebra system. Besides the standard uses of dynamic geometry, the program offers an interactive, direct, and graphical way for investigating and conjecturing in geometry. It is based in solid algebraic methods here used for geometric discovery. Dynamic geometry systems had been commonly based in numerical methods. The integration of this approach with a symbolic one (a trend also announced by other dynamic geometry software developers) had provided a learning environment where novice users can explore open geometric situations.

References

Baulac, Y., Bellemain, F., & Laborde, J. M. (1994). *Cabri geometry II*. Dallas: Texas Instruments.

Botana F., & Valcarce J. L. (2001). Cooperation between a dynamic geometry environment and a computer algebra system for geometric discovery. In V. G. Ganzha, E. W. Mayr, & E. V. Vorozhtsov (Eds.), Computer algebra and scientific computing (pp. 63–74). Springer: Berlin.

Buchberger, B. (1985). Groebner bases: an algorithmic method in polynomial ideal theory. In N. K. Bose (Ed.), *Multidimensional systems theory* (pp. 184–232). Reidel: Dordrecht.

Capani, A, Niesi, G. & Robbiano, L. (2001). *CoCoA, a system for doing computations in conmutative algebra*. Available via anonymous ftp: cocoa.dima.unige.it.

Chou, S. C. (1987). *Mechanical geometry theorem proving*. Reidel: Dordrecht.

Clemens, C. H., & Battista, M. T. (1992). Geometry and spatial reasoning. In D. A. Grouws (Ed.), *Handbook of research on mathematical teaching and learning* (pp. 420–464). New York: MacMillan.

Gao, X. S., Zhu, C. C., & Chou, S. C. (1998). *Geometry expert*. Taiwan: Nine Chapters Publishers.

Gelernter, H. (1963). Realization of a geometry theorem proving machine. In E. Feigenbaum, & J. Feldman (Eds.), *Computers and thought* (pp. 134–152). New York: McGraw-Hill.

Goldenberg, E. P., & Cuoco, A. (1998). What Is Dynamic Geometry?. In R. Lehrer, & D. Chazan (Eds.), *Designing learning environments for developing understanding of geometry and space* (pp. 351–367). Mahwah, NJ: Lawrence Erlbaum Associates.

Hanna, G. (1997). The ongoing value of proof. *Journal mathematik didaktik*, *18*(2), 171–185.

Jackiw, N. R. (1995). *The Geometer's Sketchpad, v3.0*. Berkeley, CA: Key Curriculum Press.

Guzmán, M. (1999). An extension of the Wallace-Simson theorem: projecting in arbitrary directions. *American Math. Monthly*, *106*(6), 574–580.

Jiang, Z., & McClintock, E. (1997). Using the geometer's sketchpad with preservice teachers. In J. King, & D. Schattschneider (Eds.), *Geometry turned on* (pp. 129–136). Washington: MAA.

Kapur, D., & Mundy, J. L. (1988). Wu's method and its application to perspective viewing. *Artificial Intelligence*, *37*, 15–26.

Lenat, D. B. (1984). Automated theory formation in mathematics. In W. W. Bledsoe, & D. W. Loveland (Eds.), *Automated theorem proving: After 25 years* (pp. 287–301). Providence: AMS.

Maxwell, E. A. (1959). *Fallacies in mathematics*. Cambridge: Cambridge University Press.
Nevins, A. J. (1975). Plane geometry theorem proving using forward chaining. *Artificial Intelligence*, *6*, 1–23.
Recio, T., & Vélez, M. P. (1999). Automatic discovery of theorems in elementary geometry. *Journal of automated reasoning*, *23*, 63–82.
Richter-Gebert, J., & Kortenkamp, U. (1999). *The interactive geometry software Cinderella*. Berlin: Springer.
Tall, D. (1995). Cognitive development, representations and proof. In *Proceedings of the conference Justifying and Proving in School Mathematics*, Institute of Education, London, 27–38.
Wu, W. T. (1994). *Mechanical theorem proving in geometries*. Viena: Springer.