Geometry : A Small Package for Mechanized (Plane) Geometry Manipulations

Version 1.1

Hans-Gert Gräbe, Univ. Leipzig, Germany

September 7, 1998

1 Introduction

Geometry is not only a part of mathematics with ancient roots but also a vivid area of modern research. Especially the field of geometry, called by some negligence "elementary", continues to attract the attention also of the great community of leisure mathematicians. This is probably due to the small set of prerequisites necessary to formulate the problems posed in this area and the erudition and non formal approaches ubiquitously needed to solve them. Examples from this area are also an indispensable component of high school mathematical competitions of different levels upto the International Mathematics Olympiad (IMO) [6].

The great range of ideas involved in elementary geometry theorem proving inspired mathematicians to search for a common toolbox that allows to discover such geometric statements or, at least, to prove them in a more unified way. These attempts again may be traced back until ancient times, e.g., to Euclid and his axiomatic approach to geometry.

Axiomatic approaches are mainly directed towards the introduction of coordinates that allow to quantify geometric statements and to use the full power of algebraic and even analytic arguments to prove geometry theorems. Different ways of axiomatization lead to different, even non-commutative, *rings of scalars*, the basic domain of coordinate values, see [10].

Taking rational, real or even complex coordinates for granted (as we will do in the following) it turns out that geometry theorems may be classified due to their symmetry group as statements in, e.g., projective, affine or Euclidean (Cartesian) geometry. Below such a distinction will be important for the freedom to choose appropriate coordinate systems.

It may be surprising that tedious but mostly straightforward manipulations of the algebraic counterparts of geometric statements allow to prove many theorems in geometry with even ingenious "true geometric" proofs. With the help of a Computer Algebra System supporting algebraic manipulations this approach obtains new power. The method is not automatic, since one often needs a good feeling how to encode a problem efficiently, but mechanized in the sense that one can develop a tool box to support this encoding and some very standard tools to derive a (mathematically strong !) proof from these encoded data.

The attempts to algorithmize this part of mathematics found their culmination in the 80's in the work of W.-T. Wu [10] on "the Chinese Prover" and the fundamental book [2] of S.-C. Chou who proved 512 geometry theorems with this mechanized method, see also [1], [3], [8], [9].

Since the geometric interpretation of algebraic expressions depends heavily on the properties of the field of scalars, we get another classification of geometry theorems: Those with coordinate version valid over the algebraically closed field **C** and those with coordinate version valid (or may be even formulated) only over **R**. The latter statements include *ordered geometry*, that uses the distinction between "inside" and "outside", since **C** doesn't admit monotone orderings.

This package GEOMETRY, written in the algebraic mode of Reduce, should provide the casual user with a couple of procedures that allow him/her to mechanize his/her own geometry proofs. Together with the Reduce built-in simplifier for rational functions, the `solve` function, and the Gröbner utilities[1] of the author's package CALI [5] (part of the Reduce library) it allows for proving a wide range of theorems of unordered geometry, see the examples below and in the test file `geometry.tst`.

This package grew up from a course of lectures for students of computer science on this topic held by the author at the Univ. of Leipzig in fall 1996 and was updated after a similar lecture in spring 1998.

2 Mechanizing Geometry Proving

Most geometric statements are of the following form:

> Given certain (more or less) arbitrarily chosen points and/or lines we construct certain derived points and lines from them. Then the (relative) position of these geometric objects is of a certain specific kind regardless of the (absolute) position of the chosen data.

To obtain evidence for such a statement (recommended before attempting to prove it !) one makes usually one or several drawings, choosing the independent data appropriately and constructing the dependent ones out of them (best with ruler and compass, if possible). A computer may be helpful in such a task, since the constructions are purely algorithmic and computers are best suited for algorithmic tasks. Given appropriate data structures such construction steps may be encoded into *functions* that afterwards need only to be called with appropriate parameters.

Even more general statements may be transformed into such a form and must be transformed to create drawings. This may sometimes involve constructions that can't be executed with ruler and compass as, e.g., angle trisection in Morley's theorem or construction of a conic in Pascal's theorem.

2.1 Algorithmization of (plane) geometry

The representation of geometric objects through coordinates is best suited for both compact (finite) data encoding and regeometrization of derived objects, e.g., through graphic output. Note that the target language for realization of these ideas on a computer can be almost every computer language and is not restricted to those supporting symbolic computations. Different geometric objects may be collected into a *scene*. Rapid graphic output of such a scene with different parameters may be collected into animations or even interactive drag-and-move pictures if supported by the programming system. (All this is not (yet) supported by GEOMETRY.)

We will demonstrate this approach on geometric objects, containing points and lines, represented as pairs `P:Point=`(p_1, p_2) or tripels `g:Line=`(g_1, g_2, g_3) of a certain basic type `Scalar`, e.g., floating point reals. Here g represents the homogeneous coordinates of the line $\{(x, y) : g_1x + g_2y + g_3 = 0\}$. In this setting geometric

[1] Unfortunately, the built in Gröbner package of Reduce doesn't admit enough flexibility for our purposes.

constructions may be understood as functions constructing new geometric objects from given ones. Implementing such functions variables occur in a natural way as formal parameters that are assigned with special values of the correct type during execution.

1) For example, the equation

$$(x - p_1)(q_2 - p_2) - (y - p_2)(q_1 - p_1) = 0$$

of the line through two given points $P = (p_1, p_2)$, $Q = (q_1, q_2)$ yields the function

`pp_line(P,Q:Point):Line ==` $(q_2 - p_2, p_1 - q_1, p_2 q_1 - p_1 q_2)$

that returns the (representation of the) line through these two points. In this function P and Q are neither special nor general points but formal parameters of type `Point`.

2) The (coordinates of the) intersection point of two lines may be computed solving the corresponding system of linear equations. We get a partially defined function, since there is no or a not uniquely defined intersection point, if the two lines are parallel. In this case our function terminates with an error message.

```
intersection_point(a,b:Line):Point ==
      d:=a1b2 − a2b1;
      if d = 0 then error ‘‘Lines are parallel’’
      else return ((a2b3 − a3b2)/d, (a3b1 − a1b3)/d)
```

Again a and b are formal parameters, here of the type `Line`.

3) In the same way we may define a line l through a given point P perpendicular to a second line a as

`lot(P:Point,a:Line):Line ==` $(a_2, -a_1, a_1 p_2 - a_2 p_1)$

and a line through P parallel to a as

`par(P:Point,a:Line):Line ==` $(a_1, a_2, -a_1 p_1 - a_2 p_2)$

4) All functions so far returned objects with coordinates being rational expressions in the input parameters, thus especially well suited for algebraic manipulations. To keep this nice property we introduce only the *squared Euclidean distance*

`sqrdist(P,Q:Point):Scalar ==` $(p_1 - q_1)^2 + (p_2 - q_2)^2$

5) Due to the relative nature of geometric statements some of the points and lines may be chosen arbitrarily or with certain restrictions. Hence we need appropriate constructors for points and lines given by their coordinates

`Point(a,b:Scalar):Point ==` (a, b)
`Line(a,b,c::Scalar):Line ==` (a, b, c)

and also for a point on a given line. For this purpose we provide two different functions

```
choose_Point(a:Line,u:Scalar):Point ==
      if a2 = 0 then
            if a1 = 0 then error ‘‘a is not a line’’
            else return (−a3/a1, u)
      else return (u, −(a3 + a1 u)/a2)
```

that chooses a point on a line a and

`varPoint(P,Q:Point,u:Scalar):Point ==` $(u\,a_1 + (1-u)\,b_1, u\,a_2 + (1-u)\,b_2)$

that chooses a point on the line through two given points. The main reason to have also the second definition is that u has a well defined geometric meaning in this case. For example, the midpoint of PQ corresponds to $u = \frac{1}{2}$:

```
midPoint(P,Q:Point):Point == varPoint(P,Q,1/2)
```

6) One can compose these functions to get more complicated geometric objects as, e.g., the pedal point of a perpendicular

```
pedalPoint(P:Point,a:Line):Point == intersection_point(lot(P,a),a),
```

the midpoint perpendicular of BC

```
mp(B,C:Point):Line == lot(midPoint(B,C),line(B,C)),
```

the altitude to BC in the triangle $\Delta\, ABC$

```
altitude(A,B,C:Point):Line == lot(A,line(B,C))
```

and the median line

```
median(A,B,C:Point):Line == line(A,midPoint(B,C))
```

7) We can also test geometric conditions to be fulfilled, e.g., whether two lines a and b are parallel or orthogonal

`parallel(a,b:Line):Boolean ==` $(a_1 b_2 - a_2 b_1 = 0)$

resp.

`orthogonal(a,b:Line):Boolean ==` $(a_1 b_1 + a_2 b_2 = 0)$

or whether a given point is on a given line

`point_on_line(P:Point,a:Line):Boolean ==` $(a_1 p_1 + a_2 p_2 + a_3 = 0)$

The corresponding procedures implemented in the package return the value of the expression to be equated to zero instead of a boolean.

Even more complicated conditions may be checked as, e.g., whether three lines have a point in common or whether three points are on a common line. For a complete collection of the available procedures we refer to the section 6.

Note that due to the linearity of points and lines all procedures considered so far return data with coordinates that are rational in the input parameters. One can easily enlarge the ideas presented in this section to handle also non linear objects as circles and angles, compute intersection points of circles, tangent lines etc., if the basic domain `Scalar` admits to solve non-linear (mainly quadratic) equations. Since non-linear equations usually have more than one solution, branching ideas should be incorporated, too. For example, intersecting a circle and a line the program should consider both intersection points.

2.2 Mechanized evidence of geometric statements

With a computer and these prerequisites at hand one may obtain evidence of geometric statements not only from plots but also computationally, converting the statement to be checked into a function depending on the variable coordinates as parameters and plugging in different values for them.

For example, the following function tests whether the three midpoint perpendiculars in a triangle given by the coordinates of its vertices A, B, C pass through a common point

```
test(A,B,C:Point):Boolean ==
                concurrent(mp(A,B,C),mp(B,C,A),mp(C,A,B))
```

Plugging in different values for A, B, C we can verify the theorem for many different special geometric configurations. Of course this is not yet a **proof**.

Lets add another remark: `Point` and `Line` are not only the basic data types of our geometry, but data type functions parametrized by the data type `Scalar`. To have the full functionality of our procedures `Scalar` must be a field with effective zero test.

3 Geometry Theorems of Constructive Type

Implementing the functions described above in a system, that admits also symbolic computations, we can execute the same computations also with symbolic values, i.e. taking a pure transcendental extension of $\mathbf{Q}$ as scalars. The procedures then return (simplified) symbolic expressions that specialize under (almost all) substitutions of "real" values for these symbolic ones to the same values as if they were computed by the original procedures with the specialized input. This leads to the notion of *generic geometric configurations.* A geometric statement holds in this generic configuration, i.e., the corresponding symbolic expression simplifies to zero, if and only if it is "generically true", i.e., holds for all special coordinate values except degenerate ones.

3.1 Geometric configurations of constructive type

This approach is especially powerful, if all geometric objects involved into a configuration may be constructed step by step and have *rational* expressions in the algebraically independent variables as symbolic coordinates.

DEFINITION: We say that a geometric configuration is of *constructive type*[2], if its generic configuration may be constructed step by step in such a way, that the coordinates of each successive geometric object may be expressed as rational functions of the coordinates of objects already available or algebraically independent variables, and the conclusion may be expressed as vanishing of a rational function in the coordinates of the available geometric objects.

Substituting the corresponding rational expressions of the coordinates of the involved geometric objects into the coordinate slots of newly constructed objects and finally into the conclusion expression, we obtain successively rational expressions in the given algebraically independent variables.

> *A geometry theorem of constructive type is generically true if and only if (its configuration is not contradictory and) the conclusion expression simplifies to zero.*

Indeed, if this expression simplifies to zero, the algebraic version of the theorem will be satisfied for all "admissible" values of the parameters. If the expression doesn't simplify to zero, the theorem fails for almost all such parameters.

Note that due to cancelation of denominators the domain of definition of the simplified expression may be greater than the (common) domain of definition of the different parts of the unsimplified expression. The correct non degeneracy conditions describing "admissibility" may be collected during the computation. Collecting up the zero expression indicates, that the geometric configuration is contradictory.

[2] This notion is different from [2].

Hence the statement, that a certain geometric configuration of constructive type is contradictory, is of constructive type, too.

The package GEOMETRY provides procedures `clear_ndg(), print_ndg()` to manage and print these non degeneracy conditions and also a procedure `add_ndg(d)` as a hook for their user driven management.

3.2 Some one line proofs

Take independent variables $a_1, a_2, b_1, b_2, c_1, c_2$ and

```
A:=Point(a1,a2); B:=Point(b1,b2); C:=Point(c1,c2);
```

as the vertices of a generic triangle. We can prove the following geometric statements about triangles computing the corresponding (compound) symbolic expressions and proving that they simplify to zero. Note that Reduce does simplification automatically.

1) The midpoint perpendiculars of $\Delta\, ABC$ pass through a common point since

```
concurrent(mp(A,B),mp(B,C),mp(C,A));
```

simplifies to zero.

2) The intersection point of the midpoint perpendiculars

```
M:=intersection_point(mp(A,B),mp(B,C));
```

is the center of the circumscribed circle since

```
sqrdist(M,A) - sqrdist(M,B);
```

simplifies to zero.

3) *Euler's line*:

> The center M of the circumscribed circle, the orthocenter H and the barycenter S are collinear and S divides MH with ratio 1:2.

Compute the coordinates of the corresponding points

```
M:=intersection_point(mp(a,b,c),mp(b,c,a));
H:=intersection_point(altitude(a,b,c),altitude(b,c,a));
S:=intersection_point(median(a,b,c),median(b,c,a));
```

and then prove that

```
collinear(M,H,S);
sqrdist(S,varpoint(M,H,2/3));
```

both simplify to zero.

4) *Feuerbach's circle*:

> The midpoint N of MH is the center of a circle that passes through nine special points, the three pedal points of the altitudes, the midpoints of the sides of the triangle and the midpoints of the upper parts of the three altitudes.

```
N:=midpoint(M,H);

sqrdist(N,midpoint(A,B))-sqrdist(N,midpoint(B,C));
sqrdist(N,midpoint(A,B))-sqrdist(N,midpoint(H,C));

D:=intersection_point(pp_line(A,B),pp_line(H,C));
sqrdist(N,midpoint(A,B))-sqrdist(N,D);
```

Again the last expression simplifies to zero thus proving the theorem.

4 Non-linear Geometric Objects

GEOMETRY provides several functions to handle angles and circles as non-linear geometric objects.

4.1 Angles and bisectors

(Oriented) angles between two given lines are presented as tangens of the difference of the corresponding slopes. Since

$$\tan(\alpha - \beta) = \frac{\tan(\alpha) - \tan(\beta)}{1 + \tan(\alpha)\,\tan(\beta)}$$

we get for the angle between two lines g, h

l2_angle(g,h:Line):Scalar == $\frac{g_2 h_1 - g_1 h_2}{g_1 h_1 + g_2 h_2}$

Note that in unordered geometry we can't distinguish between inner and outer angles. Hence we cannot describe (rationally) the parameters of the angle bisector of a triangle. For a point P the equation

```
l2_angle(pp_line(A,B),pp_line(P,B)) =
           l2_angle(pp_line(P,B),pp_line(C,B))
```

i.e., $\angle\, ABP = \angle\, PBC$, describes the condition to be located on either the inner or outer bisector of $\angle\, ABC$. Clearing denominators yields a procedure

```
point_on_bisector(P,A,B,C)
```

that returns on generic input a polynomial of (total) degree 4 and quadratic in the coordinates of P that describes the condition for P to be on (either the inner or the outer) bisector of $\angle\, ABC$.

With some more effort one can also employ such indirect geometric descriptions. For example, we can prove the following unordered version of the bisector intersection theorem.

5) There are four common points on the three bisector pairs of a given triangle $\Delta\, ABC$. Indeed, due to Cartesian symmetry we may choose a special coordinate system with origin A and (after scaling) x-axes unit point B. The remaining point C is arbitrary. Then the corresponding generic geometric configuration is described with two independent parameters u_1, u_2 – the coordinates of C:

```
A:=Point(0,0); B:=Point(1,0); C:=Point(u1,u2);
```

A point `P:=Point(x1,x2)` is an intersection point of three bisectors iff it is a common zero of the polynomial system

```
polys:={   point_on_bisector(P,A,B,C),
           point_on_bisector(P,B,C,A),
           point_on_bisector(P,C,A,B)},
```

i.e., of the polynomial system

$$\{\, x_1^2\, u_2 - 2\, x_1\, x_2\, u_1 + 2\, x_1\, x_2 - 2\, x_1\, u_2 - x_2^2\, u_2 + 2\, x_2\, u_1 - 2\, x_2 + u_2,$$
$$2\, x_1^2\, u_1\, u_2 - x_1^2\, u_2 - 2\, x_1\, x_2\, u_1^2 + 2\, x_1\, x_2\, u_1 + 2\, x_1\, x_2\, u_2^2 - 2\, x_1\, u_1^2\, u_2 - 2\, x_1\, u_2^3 - 2\, x_2^2\, u_1\, u_2 + x_2^2\, u_2 + 2\, x_2\, u_1^3 - 2\, x_2\, u_1^2 + 2\, x_2\, u_1\, u_2^2 - 2\, x_2\, u_2^2 + u_1^2\, u_2 + u_2^3,$$
$$x_1^2\, u_2 - 2\, x_1\, x_2\, u_1 - x_2^2\, u_2\}$$

with indeterminates x_1, x_2 over the coefficient field $\mathbf{Q}(u_1, u_2)$. A Gröbner basis computation with CALI

```
load cali;
setring({x1,x2},{},lex);
setideal(polys,polys);
gbasis polys;
```

yields the following equivalent system:

$$\{\ 4\,x_2{}^4\,u_2 - 8\,x_2{}^3\,u_1{}^2 + 8\,x_2{}^3\,u_1 - 8\,x_2{}^3\,u_2{}^2 + 4\,x_2{}^2\,u_1{}^2\,u_2 - 4\,x_2{}^2\,u_1\,u_2 + 4\,x_2{}^2\,u_2{}^3 - 4\,x_2{}^2\,u_2 + 4\,x_2\,u_2{}^2 - u_2{}^3,$$
$$2\,x_1\,u_1\,u_2{}^2 - x_1\,u_2{}^2 + 2\,x_2{}^3\,u_2 - 4\,x_2{}^2\,u_1{}^2 + 4\,x_2{}^2\,u_1 - 2\,x_2{}^2\,u_2{}^2 - 2\,x_2\,u_1{}^2\,u_2 + 2\,x_2\,u_1\,u_2 - 2\,x_2\,u_2 - u_1\,u_2{}^2 + u_2{}^2\}$$

The first equation has 4 solutions in x_2 and each of them may be completed with a single value for x_1 determined from the second equation. Hence the system *polys* has four generic solutions corresponding to the four expected intersection points. The solutions have algebraic coordinates of degree 4 over the generic field of scalars $\mathbf{Q}(u_1, u_2)$ and specialize to the correct "special" intersection points for almost all values for the parameters u_1 and u_2.

Although it is hard to give an explicit description through radicals of these symbolic values, one can compute with them knowing their minimal polynomials. Since in this situation x_2 is the distance from P to the line AB, we can prove that each of the four points has equal distance to each of the 3 lines through two vertices of $\Delta\, ABC$, i.e., that these points are the centers of its incircle and the three excircles. First we compute the differences of the corresponding squared distances

```
con1:=sqrdist(P,pedalpoint(p,pp_line(A,C)))-x2^2;
con2:=sqrdist(p,pedalpoint(p,pp_line(B,C)))-x2^2;
```

The numerator of each of these two expressions should simplify to zero under the special algebraic values of x_1, x_2. This may be verified computing their normal forms with respect to the above Gröbner basis:

```
con1 mod gbasis polys;
con2 mod gbasis polys;
```

Note that [10] proposes also a constructive proof for the bisector intersection theorem:

Start with A, B and the intersection point P of the bisectors through A and B. Then $g(AC)$ and $g(BC)$ are symmetric to $g(AB)$ wrt. $g(AP)$ and $g(BP)$ and P must be on their bisector:

```
A:=Point(0,0); B:=Point(1,0); P:=Point(u1,u2);
l1:=pp_line(A,B);
l2:=symline(l1,pp_line(A,P));
l3:=symline(l1,pp_line(B,P));

point_on_bisector(P,A,B,intersection_point(l2,l3));
```

As desired the last expression simplifies to zero.

4.2 Circles

The package GEOMETRY supplies two different types for encoding circles. The first type is `Circle1` that stores the pair (M, s), the center and the squared radius of the circle. The implementation of `point_on_circle1(P,c)` and p3_circle1(A,B,C) is almost straightforward. The latter function finds the circle through 3 given points, computing its center as the intersection point of two midpoint perpendiculars.

For purposes of analytic geometry it is often better to work with the representation `Circle` derived from the description of the circle as the set of points (x, y) for which the expression

$$(x - m_1)^2 + (y - m_2)^2 - r^2 = (x^2 + y^2) - 2\, m_1\, x - 2\, m_2\, y + m_1^2 + m_2^2 - r^2$$

vanishes. We use homogeneous coordinates `k:Circle` $= (k_1, k_2, k_3, k_4)$ for the circle

$$k := \{\, (x, y) \ : \ k_1 * (x^2 + y^2) + k_2 * x + k_3 * y + k_4 = 0\}$$

since they admit denominator free computations and include also lines as special circles with infinite radius: The line $g = (g_1, g_2, g_3)$ is the circle $(0, g_1, g_2, g_3)$.

Its easy to derive formulas `circle_center(k)` for the center of the circle k and `circle_sqradius(k)` for its squared radius. It is also straightforward to test `point_on_circle(P,k)`. The parameters of the circle `p3_circle(A,B,C)` through 3 given points

```
A:=Point(a1,a2); B:=Point(b1,b2); C=Point(c1,c2);
```

may be obtained from a nontrivial solution of the corresponding homogeneous linear system with coefficient matrix

$$\begin{pmatrix} a_1^2 + a_2^2 & a_1 & a_2 & 1 \\ b_1^2 + b_2^2 & b_1 & b_2 & 1 \\ c_1^2 + c_2^2 & c_1 & c_2 & 1 \end{pmatrix}$$

The condition that 4 points are on a common circle then may be expressed as

```
p4_circle(A,B,C,D) == point_on_circle(D,p3_circle(A,B,C));
```

For generic points A, B, C, D this yields a polynomial p_4 of degree 4 in their coordinates.

Note that this condition is equivalent to the circular angle theorem: For generic points A, B, C, D

```
u:=angle(pp_line(A,D),pp_line(B,D));
v:=angle(pp_line(A,C),pp_line(B,C));
(num(u)*den(v)-den(u)*num(v));
```

yields the same condition p_4. The common denominator `den(u)*den(v)` corresponds to the degeneracy condition that either A, B, C or A, B, D are collinear.

This condition is also equivalent to *Ptolemy's theorem*:

> For points A, B, C, D are (in that order) on a circle iff
>
> $$l(AB) * l(CD) + l(AD) * l(BC) = l(AC) * l(BD),$$
>
> i.e., the sum of the products of the lengths of opposite sides of the cyclic quadrilateral $ABCD$ equals the product of the lengths of its diagonals.

For an elementary proof see [4, 2.61]. To get a mechanized proof with the tools developed so far we are faced with several problems. First the theorem invokes distances and not their squares. Second the theorem uses the order of the given points. Unordered geometry can't even distinguish between sides and diagonals of a quadrilateral.

The fist problem may be solved by repeated squaring. Denoting the lengths appropriately we get step by step

$$p \cdot r + q \cdot s = t \cdot u$$
$$(p\,r)^2 + (q\,s)^2 - (t\,u)^2 = -(2\,p\,q\,r\,s)$$
$$((p\,r)^2 + (q\,s)^2 - (t\,u)^2)^2 - (2\,p\,q\,r\,s)^2 = 0$$

arriving at an expression that contains only squared distances. This expression

$$\texttt{poly} := p^4\,r^4 - 2\,p^2\,q^2\,r^2\,s^2 - 2\,p^2\,r^2\,t^2\,u^2 + q^4\,s^4 - 2\,q^2\,s^2\,t^2\,u^2 + t^4\,u^4$$

is symmetric in pairs of opposite sides thus solving also the second problem. Substituting the corresponding squared distances of generic points A, B, C, D we obtain exactly the square of the condition p_4.

As for bisector coordinates the coordinates of intersection points of a circle and a line generally can't be expressed rationally in terms of the coordinates of the circles. For a generic circle `c:= Circle(`c_1, c_2, c_3, c_4`)` and a generic line `d:=Line(`d_1, d_2, d_3`)` we may solve the line equation for y and substitute the result into the circle equation to get a single polynomial $q(x)$ of degree 2 with zeroes being the x-coordinate of the two intersection points of `c` and `d`:

```
vars:={x,y};
polys:={c1*(x^2+y^2)+c2*x+c3*y+c4, d1*x+d2*y+d3};
s:=solve(second polys,y);
q:=num sub(s,first polys);
```

$$q := x^2\,c_1\,({d_1}^2 + {d_2}^2) + x\,(2\,c_1\,d_1\,d_3 + c_2\,{d_2}^2 - c_3\,d_1\,d_2) + (c_1\,{d_3}^2 - c_3\,d_2\,d_3 + c_4\,{d_2}^2)$$

In many cases `d` is the line through a specified point `P:= Point(`p_1, p_2`)` on the circle. Fixing these coordinates as generic ones we get the algebraic relations

```
polys:={point_on_line(P,d), point_on_circle(P,c)};
```

$$\{d_1\,p_1 + d_2\,p_2 + d_3, c_1\,{p_1}^2 + c_1\,{p_2}^2 + c_2\,p_1 + c_3\,p_2 + c_4\}$$

between the coordinates of c, d and P. This dependency may be removed solving these equations for d_3 and c_4. In the new coordinates the polynomial $q(x)$ factors

```
s:=solve(polys,{d3,c4});
factorize sub(s,q);
```

into $x - p_1$ and a second factor that is linear in x. This yields the coordinates for the intersection point of `c` and `d` different from `P` that are saved into a function `other_cl_point(P,c,d)`. Similarly we computed the coordinates of the second intersection point of two circles c_1 and c_2 passing through a common point `P` and saved into a function `other_cc_point(P,c1,c2)`.

Also conditions on the coordinates of a circle and a line resp. two circles to be tangent may be derived in a similar way.

6) These functions admit a constructive proof of *Miquels theorem*:

> Let $\Delta\,ABC$ be a triangle. Fix arbitrary points P, Q, R on the sides AB, BC, AC. Then the three circles through each vertex and the chosen points on adjacent sides pass through a common point.

Take as above

```
A:=Point(0,0); B:=Point(1,0); C:=Point(c1,c2);
```

Generic points on the sides may be introduced with three auxiliary indeterminates:

```
P:=choose_pl(pp_line(A,B),u1);
Q:=choose_pl(pp_line(B,C),u2);
R:=choose_pl(pp_line(A,C),u3);
```

Then

```
X:=other_cc_point(P,p3_circle(A,P,R),p3_circle(B,P,Q));
```

is the intersection point of two of the circles different from `P` (its generic coordinates contain 182 terms) and since

```
point_on_circle(X,p3_circle(C,Q,R));
```

simplifies to zero the third circle also passes through `X`.

5 Geometry Theorems of Equational Type

As already seen in the last section non-linear geometric conditions are best given through implicit polynomial dependency conditions on the coordinates of the geometric objects. In this more general setting a geometric statement may be translated into a *generic geometric configuration*, involving different geometric objects with coordinates depending on (algebraically independent) variables $\mathbf{v} = (v_1, \ldots, v_n)$, a system of *polynomial conditions* $F = \{f_1, \ldots, f_r\}$ expressing the implicit geometric conditions and a polynomial g encoding the geometric conclusion, such that, for a certain polynomial non degeneracy condition h, the following holds:

> *The geometric statement is true iff for all non degenerate correct special geometric configurations, i.e., with coordinates, obtained from the generic ones by specialization $v_i \mapsto c_i$ in such a way, that $f(\mathbf{c}) = 0$ for all $f \in F$ but $h(\mathbf{c}) \neq 0$, the conclusion holds, i.e., $g(\mathbf{c})$ vanishes.*

Denoting by $Z(F)$ the set of zeroes of the polynomial system F and writing $Z(h) = Z(\{h\})$ for short, we arrive at *geometry theorems of equational type*, that may be shortly stated in the form

$$Z(F) \setminus Z(h) \subseteq Z(g).$$

Over an algebraically closed field, e.g. $\mathbf{C}$, this is equivalent to the ideal membership problem

$$g \cdot h \in rad\ I(F),$$

where $rad\ I(F)$ is the radical of the ideal generated by F. Even if h is unknown a detailed analysis of the different components of the ideal $I(F)$ allows to obtain more insight into the geometric problem.

Note the symmetry between g and h in the latter formulation of geometry theorems. This allows to derive *non degeneracy conditions* for a given geometry theorem of equational type from the stable ideal quotient

$$h \in rad\ I(F) : g^{\infty}.$$

Since every element of this ideal may serve as non degeneracy condition there is no weakest condition among them, if the ideal is not principal.

5.1 Dependent and independent variables

Let $S = R[v_1, \dots, v_n]$ be the polynomial ring in the given variables over the field of scalars R. The polynomial system F describes algebraic dependency relations between these variables in such a way that the values of some of the variables may be chosen (almost) arbitrarily whereas the remaining variables are determined upto a finite number of values by these choices.

A set of variables $\mathbf{u} \subset \mathbf{v}$ is called *independent* wrt. the ideal $I = I(F)$ iff $I \cap R[\mathbf{u}] = (0)$, i.e., the variables are algebraically independent modulo I. If $\mathbf{u}$ is a maximal subset with this property the remaining variables $\mathbf{x} = \mathbf{v} \setminus \mathbf{u}$ are called *dependent*.

Although a maximal set of independent variables may be read off from a Gröbner basis of I there is often a natural choice of dependent and independent variables induced from the geometric problem. $\mathbf{u}$ is a maximal independent set of variables iff F has a finite number of solutions as polynomial system in $\mathbf{x}$ over the generic scalar field $R(\mathbf{u})$. In many cases this may be proved with less effort than computing a Gröbner basis of I over S.

If F has an infinite number of solutions then $\mathbf{u}$ was independent but not maximal. If F has no solution then $\mathbf{u}$ was not independent.

5.2 Geometry theorems of linear type

We arrive at a particularly nice situation in the case when F is a non degenerate quadratic linear system of equations in $\mathbf{x}$ over $R(\mathbf{u})$. Such geometry theorems are called *of linear type*.

In this case there is a unique (rational) solution $\mathbf{x} = \mathbf{x}(\mathbf{u})$ that may be substituted for the dependent variables into the geometric conclusion $g = g(\mathbf{x}, \mathbf{u})$. We obtain as for geometry theorems of constructive type a rational expression in $\mathbf{u}$ and

> *the geometry theorem holds (under the non degeneracy condition $h = det(F) \in R[\mathbf{u}]$, where $det(F)$ is the determinant of the linear system F) iff this expression simplifies to zero.*

7) As an example consider the *theorem of Pappus*:

> Let A, B, C and P, Q, R be two triples of collinear points. Then the intersection points $g(AQ) \wedge g(BP), g(AR) \wedge g(CP)$ and $g(BR) \wedge g(CQ)$ are collinear.

The geometric conditions put no restrictions on A, B, P, Q and one restriction on each C and R. Hence we may take as generic coordinates

```
A:=Point(u1,u2); B:=Point(u3,u4); C:=Point(x1,u5);
P:=Point(u6,u7); Q:=Point(u8,u9); R:=Point(u0,x2);
```

with $u_0, \dots, u_9$ independent and x_1, x_2 dependent, as polynomial conditions

```
F:={collinear(A,B,C), collinear(P,Q,R)};
```

and as conclusion

```
con:=collinear(
          intersection_point(pp_line(A,Q),pp_line(P,B)),
          intersection_point(pp_line(A,R),pp_line(P,C)),
          intersection_point(pp_line(B,R),pp_line(Q,C)));
```

a rational expression with 462 terms. The polynomial conditions are linear in x_1, x_2 and already separated. Hence

```
sol:=solve(polys,{x1,x2});
sub(sol,con);
```

proves the theorem since the expression obtained from *con* substituting the dependent variables by their rational expressions in $\mathbf{u}$ simplifies to zero.

As for most theorems of linear type the linear system may be solved "geometrically" and the whole theorem may be translated into a constructive geometric statement:

```
A:=Point(u1,u2); B:=Point(u3,u4);
P:=Point(u6,u7); Q:=Point(u8,u9);

C:=choose_pl(pp_line(A,B),u5);
R:=choose_pl(pp_line(P,Q),u0);

con:=collinear(
      intersection_point(pp_line(A,Q),pp_line(P,B)),
      intersection_point(pp_line(A,R),pp_line(P,C)),
      intersection_point(pp_line(B,R),pp_line(Q,C)));
```

5.3 Geometry theorems of non-linear type

Lets return to the general situation of a polynomial system $F \subset S$ that describes algebraic dependency relations, a subdivision $\mathbf{v} = \mathbf{x} \cup \mathbf{u}$ of the variables into dependent and independent ones, and the conclusion polynomial $g(\mathbf{x}, \mathbf{u}) \in S$. The set of zeros $Z(F)$ may be decomposed into irreducible components that correspond to prime components P_α of the ideal $I = I(F)$ generated by F over the ring $S = R[\mathbf{x}, \mathbf{u}]$.

Since $P_\alpha \supset I$ the variables $\mathbf{u}$ may become dependent wrt. P_α. Prime components where $\mathbf{u}$ remains independent are called *generic*, the other components are called *special*. Note that each special component contains a non zero polynomial in $R[\mathbf{u}]$. Multiplying them all together yields a non degeneracy condition $h = h(\mathbf{u}) \in R[\mathbf{u}]$ on the independent variables such that a zero $P \in Z(F)$ with $h(P) \neq 0$ necessarily belongs to one of the generic components. Hence they are the "essential" components and we say that the geometry theorem is *generically true*, when the conclusion polynomial g vanishes on all these generic components.

If we compute in the ring $S' = R(\mathbf{u})[\mathbf{x}]$, i.e., consider the independent variables as parameters, exactly the generic components remain visible. Indeed, this corresponds to a localization of S by the multiplicative set $R[\mathbf{u}] \setminus \{0\}$. Hence the geometry theorem is generically true iff $g \in rad(I) \cdot S'$, i.e. g belongs to the radical of the ideal I in this special extension of S. A sufficient condition can be derived from a Gröbner basis G of F with the $\mathbf{u}$ variables as parameters: Test whether $g \ mod \ G = 0$, i.e., the normal form vanishes. More subtle examples may be analyzed with the Gröbner factorizer or more advanced techniques from the authors package CALI, [5].

8) As an application we consider the following nice theorem from [4, ch. 4, § 2] about Napoleon triangles:

> Let $\Delta\, ABC$ be an arbitrary triangle and P, Q and R the third vertex of equilateral triangles erected externally on the sides BC, AC and AB of the triangle. Then the lines $g(AP), g(BQ)$ and $g(CR)$ pass through a common point, the *Fermat point* of the triangle $\Delta\, ABC$.

A mechanized proof again will be faced with the difficulty that unordered geometry can't distinguish between different sides wrt. a line. A straightforward formulation of the geometric conditions starts with independent coordinates for A, B, C

and dependent coordinates for P, Q, R. W.l.o.g. we may fix the coordinates in the following way:

```
A:=Point(0,0); B:=Point(0,2); C:=Point(u1,u2);
P:=Point(x1,x2); Q:=Point(x3,x4); R:=Point(x5,x6);
```

There are 6 geometric conditions for the 6 dependent variables.

```
polys:={   sqrdist(P,B)-sqrdist(B,C), sqrdist(P,C)-sqrdist(B,C),
           sqrdist(Q,A)-sqrdist(A,C), sqrdist(Q,C)-sqrdist(A,C),
           sqrdist(R,B)-sqrdist(A,B), sqrdist(R,A)-sqrdist(A,B)};
```

$$
\begin{array}{l}
{x_1}^2 + {x_2}^2 - 4\,x_2 - {u_1}^2 - {u_2}^2 + 4\,u_2 \\
{x_1}^2 - 2\,x_1\,u_1 + {x_2}^2 - 2\,x_2\,u_2 + 4\,u_2 - 4 \\
{x_3}^2 + {x_4}^2 - {u_1}^2 - {u_2}^2 \\
{x_3}^2 - 2\,x_3\,u_1 + {x_4}^2 - 2\,x_4\,u_2 \\
{x_5}^2 + {x_6}^2 - 4\,x_6 \\
{x_5}^2 + {x_6}^2 - 4
\end{array}
$$

These equations may be divided into three groups of two quadratic relations for the coordinates of each of the points P, Q, R. Each of this pairs has (only) two solutions, the inner and the outer triangle vertex, since it may easily be reduced to a quadratic and a linear equation, the line equation of the corresponding midpoint perpendicular. Hence the whole system has 8 solutions and by geometric reasons the conclusion

```
con:=concurrent(pp_line(A,P), pp_line(B,Q), pp_line(C,R));
```

will hold on at most two of them. Due to the special structure the interreduced polynomial system is already a Gröbner basis and hence can't be split by the Gröbner factorizer. A full decomposition into isolated primes yields four components over $R(\mathbf{u})$, each corresponding to a pair of solutions over the algebraic closure. On one of them the conclusion polynomial reduces to zero thus proving the geometry theorem.

```
vars:={x1,x2,x3,x4,x5,x6};
setring(vars,{},lex);
iso:=isolatedprimes polys;

for each u in iso collect con mod u;
```

With a formulation as in [2, p. 123], that uses oriented angles, we may force all Napoleon triangles to be erected on the *same* side (internally resp. externally) and prove a more general theorem as above. Taking isosceles triangles with equal base angles and (due to one more degree of freedom) x_5 as independent the conclusion remains valid:

```
polys2:={  sqrdist(P,B)-sqrdist(P,C),
           sqrdist(Q,A)-sqrdist(Q,C),
           sqrdist(R,A)-sqrdist(R,B),
           num(p3_angle(R,A,B)-p3_angle(P,B,C)),
           num(p3_angle(Q,C,A)-p3_angle(P,B,C))};

sol:=solve(polys2,{x1,x2,x3,x4,x6});
sub(first sol,con);
```

again simplifies to zero. Note that the new theorem is of linear type.

6 The Procedures Supplied by Geometry

This section contains a short description of all procedures available in GEOMETRY. We refer to the data types `Scalar, Point, Line, Circle1` and `Circle` described above. Booleans are represented as extended booleans, i.e. the procedure returns a `Scalar` that is zero iff the condition is fulfilled. In some cases also a non zero result has a geometric meaning. For example, `collinear(A,B,C)` returns the signed area of the corresponding parallelogram.

`angle_sum(a,b:Scalar):Scalar`
: Returns $\tan(\alpha+\beta)$, if $a=\tan(\alpha), b=\tan(\beta)$.

`altitude(A,B,C:Point):Line`
: The altitude from A onto $g(BC)$.

`c1_circle(M:Point,sqr:Scalar):Circle`
: The circle with given center and sqradius.

`cc_tangent(c1,c2:Circle):Scalar`
: Zero iff c_1 and c_2 are tangent.

`choose_pc(M:Point,r,u):Point`
: Chooses a point on the circle around M with radius r using its rational parametrization with parameter u.

`choose_pl(a:Line,u):Point`
: Chooses a point on a using parameter u.

`Circle(c1,c2,c3,c4:Scalar):Circle`
: The `Circle` constructor.

`Circle1(M:Point,sqr:Scalar):Circle1`
: The `Circle1` constructor.

`circle_center(c:Circle):Point`
: The center of c.

`circle_sqradius(c:Circle):Point`
: The sqradius of c.

`cl_tangent(c:Circle,l:Line):Scalar`
: Zero iff l is tangent to c.

`collinear(A,B,C:Point):Scalar`
: Zero iff A, B, C are on a common line. In general the signed area of the parallelogram spanned by $\vec{AB}$ and $\vec{AC}$.

`concurrent(a,b,c:Line):Scalar`
: Zero iff a, b, c have a common point.

`intersection_point(a,b:Line):Point`
: The intersection point of the lines a, b.

`l2_angle(a,b:Line):Scalar`
: Tangens of the angle between a and b.

`Line(a,b,c:Scalar):Line`
: The `Line` constructor.

`lot(P:Point,a:Line):Line`
: The perpendicular from P onto a.

`median(A,B,C:Point):Line`

The median line from A to BC.

`midpoint(A,B:Point):Point`
The midpoint of AB.

`mp(B,C:Point):Line`
The midpoint perpendicular of BC.

`orthogonal(a,b:Line):Scalar`
zero iff the lines a, b are orthogonal.

`other_cc_point(P:Point,c1,c2:Circle):Point`
c_1 and c_2 intersect at P. The procedure returns the second intersection point.

`other_cl_point(P:Point,c:Circle,l:Line):Point`
c and l intersect at P. The procedure returns the second intersection point.

`p3_angle(A,B,C:Point):Scalar`
Tangens of the angle between $\vec{BA}$ and $\vec{BC}$.

`p3_circle(A,B,C:Point):Circle` or
`p3_circle1(A,B,C:Point):Circle1`
The circle through 3 given points.

`p4_circle(A,B,C,D:Point):Scalar`
Zero iff four given points are on a common circle.

`par(P:Point,a:Line):Line`
The line through P parallel to a.

`parallel(a,b:Line):Scalar`
Zero iff the lines a, b are parallel.

`pedalpoint(P:Point,a:Line):Point`
The pedal point of the perpendicular from P onto a.

`Point(a,b:Scalar):Point`
The `Point` constructor.

`point_on_bisector(P,A,B,C:Point):Scalar`
Zero iff P is a point on the (inner or outer) bisector of the angle $\angle ABC$.

`point_on_circle(P:Point,c:Circle):Scalar` or
`point_on_circle1(P:Point,c:Circle1):Scalar`
Zero iff P is on the circle c.

`point_on_line(P:Point,a:Line):Scalar`
Zero iff P is on the line a.

`pp_line(A,B:Point):Line`
The line through A and B.

`sqrdist(A,B:Point):Scalar`
Square of the distance between A and B.

`sympoint(P:Point,l:Line):Point`
The point symmetric to P wrt. the line l.

`symline(a:Line,l:Line):Line`
The line symmetric to a wrt. the line l.

`varpoint(A,B:Point,u):Point`
: The point $D = u \cdot A + (1-u) \cdot B$.

GEOMETRY supplies as additional tools the functions

`extractmat(polys,vars)`
: Returns the coefficient matrix of the list of equations *polys* that are linear in the variables *vars*.

`red_hom_coords(u:{Line,Circle})`
: Returns the reduced homogeneous coordinates of u, i.e., divides out the content.

7 More Examples

Here we give a more detailed explanation of some of the examples collected in the test file `geometry.tst` and give a list of exercises. Their solutions can be found in the test file, too.

7.1 Theorems that can be translated into theorems of constructive or linear type

There are many geometry theorems that may be reformulated as theorems of constructive type.

9) The affine version of *Desargue's theorem*:

> If two triangles $\Delta\, ABC$ and $\Delta\, RST$ are in similarity position, i.e., $g(AB) \| g(RS)$, $g(BC) \| g(ST)$ and $g(AC) \| g(RT)$, then $g(AR)$, $g(BS)$ and $g(CT)$ pass through a common point (or are parallel).

The given configuration may be constructed step by step in the following way: Take A, B, C, R arbitrarily, choose S arbitrarily on the line through R parallel to $g(AB)$ and T as the intersection point of the lines through R parallel to $g(AC)$ and through S parallel to $g(BC)$.

```
A:=Point(a1,a2); B:=Point(b1,b2);
C:=Point(c1,c2); R:=Point(d1,d2);
S:=choose_pl(par(R,pp_line(A,B)),u);
T:=intersection_point(
          par(R,pp_line(A,C)),par(S,pp_line(B,C)));

con:=concurrent(pp_line(A,R),pp_line(B,S),pp_line(C,T));
```

Another proof may be obtained translating the statement into a theorem of linear type. Since the geometric conditions put no restrictions on A, B, C, R, one restriction on S ($g(AB) \| g(RS)$) and two restrictions on T ($g(BC) \| g(ST)$, $g(AC) \| g(RT)$), we may take as generic coordinates

```
A:=Point(u1,u2); B:=Point(u3,u4); C:=Point(u5,u6);
R:=Point(u7,u8); S:=Point(u9,x1); T:=Point(x2,x3);
```

with $u_1, \ldots, u_9$ independent and x_1, x_2, x_3 dependent, as polynomial conditions

```
polys:={  parallel(pp_line(R,S),pp_line(A,B)),
          parallel(pp_line(S,T),pp_line(B,C)),
          parallel(pp_line(R,T),pp_line(A,C))};
```

and as conclusion

```
con:=concurrent(pp_line(A,R),pp_line(B,S),pp_line(C,T));
```

The polynomial conditions are linear in x_1, x_2, x_3 and thus

```
sol:=solve(polys,{x1,x2,x3});
sub(sol,con);
```

proves the theorem since the expression obtained from *con* substituting the dependent variables by their rational expressions in **u** simplifies to zero.

The general version of *Desargue's theorem*:

> The lines $g(AR)$, $g(BS)$ and $g(CT)$ pass through a common point iff the intersection points $g(AB) \wedge g(RS)$, $g(BC) \wedge g(ST)$ and $g(AC) \wedge g(RT)$ are collinear.

may be reduced to the above theorem by a projective transformation mapping the line through the three intersection points to infinity. Its algebraic formulation

```
A:=Point(0,0); B:=Point(0,1); C:=Point(u5,u6);
R:=Point(u7,u8); S:=Point(u9,u1); T:=Point(u2,x1);

con1:=collinear(
      intersection_point(pp_line(R,S),pp_line(A,B)),
      intersection_point(pp_line(S,T),pp_line(B,C)),
      intersection_point(pp_line(R,T),pp_line(A,C)));
con2:=concurrent(pp_line(A,R),pp_line(B,S),pp_line(C,T));
```

contains a polynomial con_2 linear in x_1 and a rational function con_1 with numerator quadratic in x_1 that factors as

$$\text{num}(con_1) = con_2 \cdot \text{collinear}(R, S, T)$$

thus also proving the general theorem.

10) Consider the following theorem about the *Brocard points* ([2, p. 336])

> Let $\Delta\, ABC$ be a triangle. The circles c_1 through A, B and tangent to $g(AC)$, c_2 through B, C and tangent to $g(AB)$, and c_3 through A, C and tangent to $g(BC)$ pass through a common point.

It leads to a theorem of linear type that can't be translated into constructive type in an obvious way. The circles may be described each by 3 dependent variables and 3 conditions

```
A:=Point(0,0); B:=Point(1,0); C:=Point(u1,u2);

c1:=Circle(1,x1,x2,x3);
c2:=Circle(1,x4,x5,x6);
c3:=Circle(1,x7,x8,x9);

polys:={  cl_tangent(c1,pp_line(A,C)),
          point_on_circle(A,c1),
          point_on_circle(B,c1),
          cl_tangent(c2,pp_line(A,B)),
          point_on_circle(B,c2),
          point_on_circle(C,c2),
          cl_tangent(c3,pp_line(B,C)),
          point_on_circle(A,c3),
          point_on_circle(C,c3)};
```

that are linear in the dependent variables. Hence the coordinates of the circles and the intersection point of two of them may be computed and checked for incidence with the third circle:

```
vars:={x1,x2,x3,x4,x5,x6,x7,x8,x9};
sol:=solve(polys,vars);

P:=other_cc_point(C,sub(sol,c1),sub(sol,c2));
con:=point_on_circle(P,sub(sol,c3));
```

Again *con* simplifies to zero thus proving the theorem.

Even some theorems involving nonlinear objects as circles may be translated into theorems of constructive type using a rational parametrization of the non linear object. For a circle with radius r and center $M = (m_1, m_2)$ we may use the rational parametrization

$$\{(\frac{1-u^2}{1+u^2}r + m_1, \frac{2u}{1+u^2}r + m_2) \mid u \in \mathbf{C}\}.$$

This way we can prove

11) *Simson's theorem* ([1, p. 261], [4, thm. 2.51]):

> Let P be a point on the circle circumscribed to the triangle $\Delta\,ABC$ and X, Y, Z the pedal points of the perpendiculars from P onto the lines passing through pairs of vertices of the triangle. These points are collinear.

Take the center M of the circumscribed circle as the origin and r as its radius. The proof of the problem may be mechanized in the following way:

```
M:=Point(0,0);
A:=choose_pc(M,r,u1);
B:=choose_pc(M,r,u2);
C:=choose_pc(M,r,u3);
P:=choose_pc(M,r,u4);
X:=pedalpoint(P,pp_line(A,B));
Y:=pedalpoint(P,pp_line(B,C));
Z:=pedalpoint(P,pp_line(A,C));

con:=collinear(X,Y,Z);
```

Since *con* simplifies to zero this proves the theorem.

7.2 Theorems of equational type

An "almost" constructive proof of Simson's theorem may be obtained in the following way:

```
A:=Point(0,0); B:=Point(u1,u2);
C:=Point(u3,u4); P:=Point(u5,x1);
X:=pedalpoint(P,pp_line(A,B));
Y:=pedalpoint(P,pp_line(B,C));
Z:=pedalpoint(P,pp_line(A,C));

poly:=p4_circle(A,B,C,P);

con:=collinear(X,Y,Z);
```

There is a single dependent variable bound by the quadratic condition *poly* that the given points are on a common circle. *con* is a rational expression with numerator equal to

$$poly \cdot \mathrm{collinear}(A, B, C)^2.$$

Since the second factor may be considered as degeneracy condition this also proves Simson's theorem. The factors of the denominator

$$\mathrm{den}(con) = \mathrm{sqrdist}(A, B) \cdot \mathrm{sqrdist}(A, C) \cdot \mathrm{sqrdist}(B, C)$$

are exactly the non degeneracy conditions collected during the computation. They may be printed with `print_ndg()`.

One may also substitute the rational coordinate construction of X, Y, Z through `pedalpoint` with additional dependent variables and polynomial conditions:

```
M:=Point(0,0); A:=Point(0,1);
B:=Point(u1,x1); C:=Point(u2,x2); P:=Point(u3,x3);
X:=varpoint(A,B,x4);
Y:=varpoint(B,C,x5);
Z:=varpoint(A,C,x6);
```

The polynomial conditions

```
polys:={   sqrdist(M,B)-1, sqrdist(M,C)-1, sqrdist(M,P)-1,
           orthogonal(pp_line(A,B),pp_line(P,X)),
           orthogonal(pp_line(A,C),pp_line(P,Z)),
           orthogonal(pp_line(B,C),pp_line(P,Y))};
```

contain three quadratic polynomials in x_1, x_2, x_3 and three polynomials linear in x_4, x_5, x_6. The quadratic polynomials correspond to different points on the circle with given x-coordinate. The best variable order eliminates linear variables first. Thus the following computations prove the theorem

```
con:=collinear(X,Y,Z);

vars:={x4,x5,x6,x1,x2,x3};
setring(vars,{},lex);
setideal(polys,polys);
con mod gbasis polys;
```

since the conclusion polynomial reduces to zero.

12) The Butterfly Theorem ([1, p. 269], [4, thm. 2.81]) :

> Let A, B, C, D be four points on a circle with center O, P the intersection point of AC and BD and F resp. G the intersection point of the line through P perpendicular to OP with AB resp. CD. Then P is the midpoint of FG.

Taking P as the origin and the lines $g(FG)$ and $g(OP)$ as axes we get the following coordinatization:

```
P:=Point(0,0); O:=Point(u1,0);
A:=Point(u2,u3); B:=Point(u4,x1);
C:=Point(x2,x3); D:=Point(x4,x5);
F:=Point(0,x6); G:=Point(0,x7);
```

```
polys:={   sqrdist(O,B)-sqrdist(O,A),
           sqrdist(O,C)-sqrdist(O,A),
```

```
          sqrdist(O,D)-sqrdist(O,A),
          point_on_line(P,pp_line(A,C)),
          point_on_line(P,pp_line(B,D)),
          point_on_line(F,pp_line(A,D)),
          point_on_line(G,pp_line(B,C))};

con:=num sqrdist(P,midpoint(F,G));
```

Note that the formulation of the theorem includes $A \neq C$ and $B \neq D$. Hence the conclusion may (and will) fail on some of the components of $Z(polys)$. This can be avoided supplying appropriate constraints to the Gröbner factorizer:

```
vars:={x6,x7,x3,x5,x1,x2,x4};
setring(vars,{},lex);

sol:=groebfactor(polys,{sqrdist(A,C),sqrdist(B,D)});

for each u in sol collect con mod u;
```

sol contains a single solution that reduces the conclusion *con* to zero. Hence the Gröbner factorizer could split the components and remove the auxiliary ones.

Note that there is also a constructive proof of the Butterfly theorem, see `geometry.tst`.

13) Let's prove another property of Feuerbach's circle ([4, thm. 5.61]):

> For an arbitrary triangle $\Delta\, ABC$ Feuerbach's circle is tangent to its in- and excircles (tangent circles for short).

Take the same coordinates as in example 5 and construct the coordinates of the center N of Feuerbach's circle c_1 as in example 4:

```
A:=Point(0,0); B:=Point(2,0); C:=Point(u1,u2);
M:=intersection_point(mp(A,B),mp(B,C));
H:=intersection_point(altitude(A,B,C),altitude(B,C,A));
N:=midpoint(M,H);

c1:=c1_circle(N,sqrdist(N,midpoint(A,B)));
```

The coordinates of the center `P:=Point(x1,x2)` of one of the tangent circles are bound by the conditions

```
polys:={point_on_bisector(P,A,B,C), point_on_bisector(P,B,C,A)};
```

Due to the choice of the coordinates x_2 is the radius of this circle. Hence the conclusion may be expressed as

```
con:=cc_tangent(c1_circle(P,x2^2),c1);
```

The polynomial conditions *polys* have four generic solutions, the centers of the four tangent circles, as derived in example 5. Since

```
vars:={x1,x2};
setring(vars,{},lex);
setideal(polys,polys);
num con mod gbasis polys;
```

yields zero this proves that all four circles are tangent to Feuerbach's circle. [4, ch.5,§6] points out that Feuerbach's circle of $\Delta\, ABC$ coincides with Feuerbach's circle of each of the triangles $\Delta\, ABH$, $\Delta\, ACH$ and $\Delta\, BCH$. Hence there are another 12 circles tangent to c_1. This may be proved

Note that the proof in [4] uses inversion geometry. The author doesn't know about a really "elementary" proof of this theorem.

8 Exercises

1. ([1, p. 267]) Let $ABCD$ be a square and P a point on the line parallel to BD through C such that $l(BD) = l(BP)$, where $l(BD)$ denotes the distance between B and D. Let Q be the intersection point of BF and CD. Show that $l(DP) = l(DQ)$.

2. The altitudes' pedal points theorem: Let P, Q, R be the altitudes' pedal points in the triangle $\Delta\, ABC$. Show that the altitude through Q bisects $\angle\, PQR$.

3. Let $\Delta\, ABC$ be an arbitrary triangle. Consider the three altitude pedal points and the pedal points of the perpendiculars from these points onto the the opposite sides of the triangle. Show that these 6 points are on a common circle, the *Taylor circle*.

4. Prove the formula
$$F(\Delta\, ABC) = \frac{a\,b\,c}{4\,R},$$
for the area of the triangle $\Delta\, ABC$, if a, b, c are the lengths of its sides and R the radius of its circumscribed circle.

5. ([1, p. 283]) Let k be a circle, A the contact point of the tangent line from a point B to k, M the midpoint of AB and D a point on k. Let C be the second intersection point of DM with k, E the second intersection point of BD with k and F the second intersection point of BC with k. Show that EF is parallel to AB.

6. (35th IMO 1995, Toronto, problem 1, [6]) Let A, B, C, D be four distinct points on a line, in that order. The circles with diameters AC and BD intersect at the points X and Y. The line XY meets BC at the point Z. Let P be a point on the line XY different from Z. The line CP intersects the circle with diameter AC at the points C and M, and the line BP intersects the circle with diameter BD at the points B and N. Prove that the lines AM, DN and XY are concurrent.

7. (34th IMO 1994, Hong Kong, problem 2, [6]) ABC is an isosceles triangle with $AB = AC$. Suppose that

   (i) M is the midpoint of BC and O is the point on the line AM such that OB is perpendicular to AB;

   (ii) Q is an arbitrary point on the segment BC different from B and C;

   (iii) E lies on the line AB and F lies on the line AC such that E, Q and F are distinct and collinear.

   Prove that OQ is perpendicular to EF if and only if $QE = QF$.

8. (4th IMO 1959, Czechia, problem 6, [7]) Show that the distance d between the centers of the inscribed and the circumscribed circles of a triangle $\Delta\, ABC$ satisfies $d^2 = r^2 - 2r\rho$, where r is the radius of the circumscribed circle and ρ the radius of the inscribed circle.

9. (1th IMO 1959, Roumania, problem 5, [7]) Let M be a point on AB, $AMCD$ and $MBEF$ squares to the same side of $g(AB)$ and N the intersection point of their circumscribed circles, different from M.

   (i) Show that $g(AF)$ and $g(BC)$ intersect at N.

   (ii) Show that all lines $g(MN)$ for various M meet at a common point.

References

[1] S.-C. Chou. Proving elementary geometry theorems using Wu's algorithm. In *Contemp. Math.*, volume 19, pages 243 – 286. AMS, Providence, Rhode Island, 1984.

[2] S.-C. Chou. *Mechanical geometry theorem proving.* Reidel, Dortrecht, 1988.

[3] S.-C. Chou. Automated reasoning in geometries using the characteristic set method and Gröbner basis method. In *Proc. ISSAC-90*, pages 255–260. ACM Press, 1990.

[4] H.S.M. Coxeter and S.L. Greitzer. *Geometry revisted.* Random House, The L.W. Singer Comp., New York, 1967.

[5] H.-G. Gräbe. *CALI – A Reduce package for commutative algebra. Version 2.2.1.* Uni Leipzig, Available from `http://www.informatik.uni-leipzig.de/~compalg`, June 1995.

[6] International Mathematics Olympiad (IMO). Available from `http://olympiads.win.tue.nl/imo`.

[7] E.A. Morozova and I.S. Petrakov. *International Mathematics Olympiads.* Prosveščenie, Moscow, 1968. (in russian).

[8] W.-T. Wu. On the decision problem and the mechanization of theorem-proving in elementary geometry. In *Contemp. Math.*, volume 19, pages 213 – 234. AMS, Providence, Rhode Island, 1984.

[9] W.-T. Wu. Some recent advances in mechanical theorem proving of geometry. In *Contemp. Math.*, volume 19, pages 235 – 241. AMS, Providence, Rhode Island, 1984.

[10] W.-T. Wu. *Mechanical Theorem Proving in Geometries.* Number 1 in Texts and Monographs in Symbolic Computation. Springer, Wien, 1994.