

Gröbner bases in GAP

To start using it, one has to define indeterminates and polynomials. Indeterminates are displayed either by their internal numbers or you can prescribe names. (Note however that the names hide the internal numbers and these numbers are basis for the monomial orderings. the best is to define a set of variables at the start and then not to redefine them afterwards.

```
gap> x:=X(Rationals,1); # number
x_1
gap> x:=X(Rationals,"x");
x
gap> y:=X(Rationals,"y");
y
gap> IndeterminateNumberOfUnivariateLaurentPolynomial(y);
2
```

The *lex* and *deglex* ordering are defined as `MonomialLexOrdering()`, `MonomialGrlexOrdering()`. (There also is `MonomialGrevlexOrdering()` for degree reverse-lexicographic.) These functions may take a number of variables and then assume that these variables are given in descending order.

`LeadingTerm` gives the leading term of a polynomial.

```
gap> LeadingTerm(x*y^2+2*x^2*y,MonomialGrlexOrdering());
2*x^2*y
```

`GrobnerBasis` computes a Gröbner basis. (You can set the info level as done here to get some information about the calculations.)

```
gap> B:=[x*y-y^2,y^2-x];
[ x*y-y^2, -x+y^2 ]
gap> GrobnerBasis(B,MonomialGrlexOrdering());
[ x*y-y^2, y^2-x, -x^2+x ]
gap> G:=ReducedGroebnerBasis(B,MonomialLexOrdering());
[ y^2-x, x*y-x, x^2-x ]
gap> G:=ReducedGroebnerBasis(B,MonomialGrlexOrdering(y,x));
[ x^2-x, x*y-x, y^2-x ]
```

`PolynomialReduction` can be used to determine remainders. The first entry is the remainder, the second the coefficients with respect to the list of basis elements. `PolynomialReducedRemainder` only returns the remainder.

```
gap> PolynomialReduction(x^5*y,G,MonomialGrlexOrdering());
[ x, [ x^4+x^3*y+x^2*y^2+x*y^3+y^4+y^3+y^2+y+1, y^4+y^3+y^2+y+1, 0 ] ]
```

Constants

Sometimes we want to solve a Gröbner basis for situations where there are constants involved. We cannot simply consider these constants as further variables (e.g. one cannot solve for these.) Instead, suppose that we have constants c_1, \dots, c_k and variables x_1, \dots, x_n .

We first form the polynomial ring in the *constants*, $C := k[c_1, \dots, c_k]$. This is an integral domain, we therefore (Theorem 2.25 in the book) can form the field of fractions (analogous to how the rationals are formed as fractions of integers) $F = \text{Frac}(C) = k(c_1, \dots, c_k)$ ¹.

¹The round parentheses denote that we permit also division

We then set $R = F[x_1, \dots, x_n] = k(c_1, \dots, c_k)[x_1, \dots, x_n]$ and work in this polynomial ring. (The only difficulty arises once one specializes the constants: It is possible that denominators evaluate back to zero.) For example, suppose we want to solve the equations

$$x^2y + y + a = y^2x + x = 0$$

In GAP we first create the polynomial ring for the constants:

```
gap> C:=PolynomialRing(Rationals,["a"]);
Rationals[a]
gap> a:=IndeterminatesOfPolynomialRing(S)[1];
a
```

(There is no need to create a separate fraction field of C .) Now we define the new variables x, y to have coefficients from C :

```
gap> x:=X(C, "x");;
#I You are creating a polynomial *over* a polynomial ring (i.e. in an
#I iterated polynomial ring). Are you sure you want to do this?
#I If not, the first argument should be the base ring, not a polynomial ring
#I Set ITER_POLY_WARN:=false; to remove this warning.
gap> y:=X(C, "y");;
```

As one sees, GAP warns that these variables are defined over a ring which has already variables, but we can ignore this warning², or set `ITER_POLY_WARN:=false;` to turn it off.

Now we can form polynomials and calculate a Gröbner basis:

```
gap> f:=x^2*y+y+a;;
gap> g:=y^2*x+x;;
gap> ReducedGroebnerBasis([f,g],MonomialLexOrdering());
[ y^3+a*y^2+y+a, x*y^2+x, x^2-y^2-a*y ]
```

We thus need that $y^3 + ay^2 + y + a = 0$, and then can solve this for y and substitute back.

The Nullstellensatz and the Radical

In applications (see below) we sometimes want to check not whether $g \in I \triangleleft R$, but whether g has “the same roots” as I . (I.e. we want that $g(x_1, \dots, x_n) = 0$ if and only if $f(x_1, \dots, x_n) = 0$ for all $f \in I$.) Clearly this can hold for polynomials not in I , for example consider $I = \langle x^2 \rangle \triangleleft \mathbb{Q}[x]$, then $g(x) = x \notin I$, but has the same roots.

An important theorem (which we will prove later) shows that over an algebraically closed field such a power condition is all that can happen.

Theorem (Hilbert’s Nullstellensatz): Let F be algebraically closed and $R = F[x_1, \dots, x_n]$ and $I \triangleleft R$. If $g \in R$ such that $g(x_1, \dots, x_n) = 0$ if and only if $f(x_1, \dots, x_n) = 0$ for all $f \in I$, then $g \in r(I)$.

We investigate in the homework how to test membership in the radical.

Proving Theorems from Geometry

Suppose we describe points in the plane by their (x, y) -coordinates. We then can describe many geometric properties by polynomial equations:

Theorem: Suppose that $A = (x_a, y_a)$, $B = (x_b, y_b)$, $C = (x_c, y_c)$ and $D = (x_d, y_d)$ are points in the plane. Then the following geometric properties are described by polynomial equations:

²The reason for the warning is that this was a frequent error of users when defining variables and polynomial rings. Variables were taken accidentally not from the polynomial ring, but created over the polynomial ring.

\overline{AB} is parallel to \overline{CD} : $\frac{y_b - y_a}{x_b - x_a} = \frac{y_d - y_c}{x_d - x_c}$, which implies $(y_b - y_a)(x_d - x_c) = (y_d - y_c)(x_b - x_a)$.

$\overline{AB} \perp \overline{CD}$: $(x_b - x_a)(x_d - x_c) + (y_b - y_a)(y_d - y_c) = 0$.

$|AB| = |CD|$: $(x_b - x_a)^2 + (y_b - y_a)^2 = (x_d - x_c)^2 + (y_d - y_c)^2$.

C lies on the circle with center A though the point B : $|AC| = |AB|$.

C is the midpoint of \overline{AB} : $x_c = \frac{1}{2}(x_b + x_a)$, $y_c = \frac{1}{2}(y_b + y_a)$.

C is collinear with \overline{AB} : self

\overline{BD} bisects $\angle ABC$: self

A theorem from geometry now sets up as prerequisites some points (and their relation with circles and lines) and claims (the conclusion) that this setup implies other conditions. If we suppose that (x_i, y_i) are the coordinates of all the points involved, the prerequisites thus are described by a set of polynomials $f_j \in \mathbb{Q}[x_1, \dots, x_n, y_1, \dots, y_n] =: R$. A conclusion similarly corresponds to a polynomials $g \in R$.

The statement of the geometric theorem now says that whenever the $\{(x_i, y_i)\}$ are points fulfilling the prerequisites (i.e. $f_j(x_1, \dots, x_n, y_1, \dots, y_n) = 0$), then also the conclusion holds for these points (i.e. $g(x_1, \dots, x_n, y_1, \dots, y_n) = 0$).

This is exactly the condition we studied in the previous section and translated to $g^m \in \langle f_1, \dots, f_m \rangle$, which we can test. (Caveat: This is formally over \mathbb{C} , but we typically want real coordinates. There is some extra subtlety in practice.)

For example, consider the theorem of THALES: Any triangle suspended under a half-circle has a right angle.

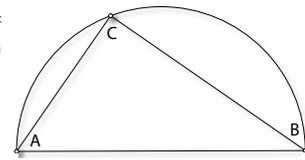
We assume (after rotation and scaling) that $A = (-1, 0)$ and $B = (1, 0)$ and set $C = (x, y)$ with variable coordinates. The fact that C is on the circle (whose origin is $(0, 0)$ and radius is 1) is specified by the polynomial

$$f = x^2 + y^2 - 1 = 0$$

The right angle at C means that $\overline{AC} \perp \overline{BC}$. We encode this as

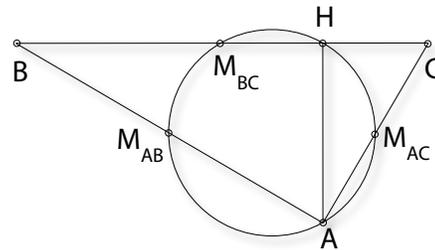
$$g = (x - (-1)) + (1 - x) + y(-y) = -x^2 - y^2 + 1 = 0.$$

Clearly g is implied by f .



Apollonius' theorem We want to use this approach to prove a classical geometrical theorem (it is a special case of the "Nine point" or "Feuerbach circle" theorem):

Circle Theorem of APOLLONIUS: Suppose that ABC is a right angled triangle with right angle at A . The midpoints of the three sides, and the foot of the altitude drawn from A onto \overline{BC} all lie on a circle.



To translate this theorem into equations, we choose coordinates for the points A, B, C . For simplicity we set (translation) $A = (0, 0)$ and $B = (b, 0)$ and $C = (0, c)$, where b and c are constants. (That is, the calculation takes place in a polynomial ring not over the rationals, but over the quotient field of $\mathbb{Q}[b, c]$.) Suppose that $M_{AB} = (x_1, 0)$, $M_{AC} = (0, x_2)$ and $M_{BC} = (x_3, x_4)$. We get the equations

$$\begin{aligned} f_1 &= 2x_1 - b = 0 \\ f_2 &= 2x_2 - c = 0 \\ f_3 &= 2x_3 - b = 0 \\ f_4 &= 2x_4 - c = 0 \end{aligned}$$

Next assume that $H = (x_5, x_6)$. Then $AH \perp BC$ yields

$$f_5 = x_5b - x_6c = 0$$

Because H lies on BC , we get

$$f_6 = x_5c + x_6b - bc = 0$$

To describe the circle, assume that the middle point is $O = (x_7, x_8)$. The circle property then means that $|M_{AB}O| = |M_{BC}O| = |M_{AC}O|$ which gives the conditions

$$\begin{aligned} f_7 &= (x_1 - x_7)^2 + (0 - x_8)^2 - (x_3 - x_7)^2 - (x_4 - x_8)^2 = 0 \\ f_8 &= (x_1 - x_7)^2 + x_8^2 - x_7^2 - (x_8 - x_2)^2 = 0 \end{aligned}$$

The conclusion is that $|HO| = |M_{AB}O|$, which translates to

$$g = (x_5 - x_7)^2 + (x_6 - x_8)^2 - (x_1 - x_7)^2 - x_8^2 = 0.$$

We now want to show that there is an m , such that $g^m \in \langle f_1, \dots, f_8 \rangle$. In GAP, we first define a ring for the two constants, and assign the constants. We also define variables *over* this ring.

```
R:=PolynomialRing(Rationals, ["b", "c"]);
ind:=IndeterminatesOfPolynomialRing(R);b:=ind[1];c:=ind[2];
x1:=X(R, "x1"); ... x8:=X(R, "x8");
```

We define the ideal generators f_i as well as g :

```
f1:=2*x1-b;
f2:=2*x2-c;
f3:=2*x3-b;
f4:=2*x4-c;
f5:=x5*b-x6*c;
f6:=x5*c+x6*b-b*c;
f7:=(x1-x7)^2+(0-x8)^2-(x3-x7)^2-(x4-x8)^2;
f8:=(x1-x7)^2+x8^2-x7^2-(x8-x2)^2;
g:=(x5-x7)^2+(x6-x8)^2-(x1-x7)^2-x8^2;
```

For the test whether $g^m \in I$, we need an auxiliary variable y . Then we can generate a Gröbner basis for \tilde{I} :

```
y:=X(R, "y");
order:=MonomialLexOrdering();
bas:=ReducedGroebnerBasis([f1,f2,f3,f4,f5,f6,f7,f8,1-y*g], order);
```

The basis returned is (1), which means that indeed $g^m \in I$.

If we wanted to know for which m , we can test membership in I itself, and find that already $g \in I$:

```
gap> bas:=ReducedGroebnerBasis([f1,f2,f3,f4,f5,f6,f7,f8], order);
[ x8-1/4*c, x7-1/4*b, x6+(-b^2*c/(b^2+c^2)), x5+(-b*c^2/(b^2+c^2)),
  x4-1/2*c, x3-1/2*b, x2-1/2*c, x1-1/2*b ]
gap> PolynomialReducedRemainder(g,bas, order);
0
```