Introduction to Gröbner Bases

Bruno Buchberger

Talk at the Summer School
Emerging Topics in Cryptographic Design and Cryptanalysis

30 April - 4 May, 2007 Samos, Greece

Gröbner Bases: What and How?

Applications of Gröbner Bases

Discussion

Gröbner Bases: What and How?

Applications of Gröbner Bases

Discussion

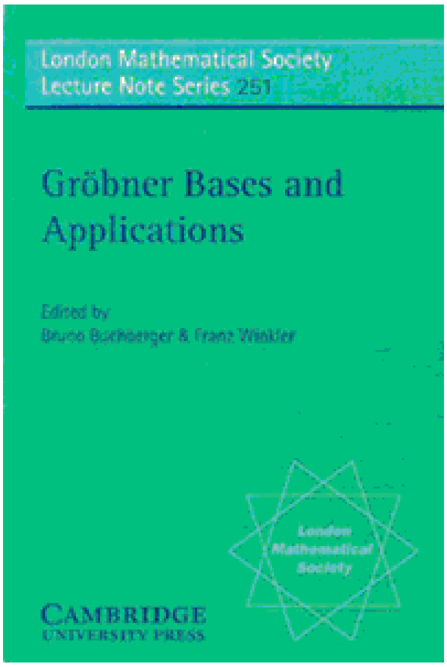
Motivation

- Dozens of (difficult) problems turned out to be reducible to the construction of Gröbner bases. (~ 1000 papers, 10 textbooks, ~ 3000 citations in Research Index, extra entry 13P10 in AMS index).
- This is based on the fact that Gröbner bases have many nice properties (e.g. canonicality property, elimination property, syzygy property).
- For the construction of Gröbner bases we have (an) algorithm(s), [BB 1965, ...]
- A "beautiful" theory: The notion of Gröbner bases and the algorithm is easy to explain, but correctness is based on a non-trivial theory.

○ A very active research area: more efficient algorithms based on more theory, and more applications (e.g. cryptography).

A First Entry to Literature

For an overview on theory and applications see:



This talk is based on the paper B. Buchberger, "Introduction to Gröbner Bases", pp. 3-31, in this book (1998). (The presentation in the paper is more formal than the presentation in this talk).

Also see web site of the Special Semester on Gröbner Bases organized by BB in 2006 at RICAM / RISC, Johannes Kepler University, Austria:

www.ricam.oeaw.ac.at/specsem/srs/groeb/

In particular, see the interactive bibliography on Gröbner bases at

www.ricam.oeaw.ac.at/specsem/srs/groeb/bibliography.htm

The Linear Combination of Polynomials

```
f1 = -2 y + x y
f2 = -x^2 + y^2
```

$-2\,y + x\,y$

$-x^2 + y^2$

Leading power products: w.r.t. an ordering of the power products (e.g. lexicographically, by total degreee or ...)

(There are infinitely many "admissible" orderings for Gröbner bases theory that can be characterized by two easy axioms.)

Consider now the following linear combination of f_1 and f_2:

```
g = (y) f1 + (-x + 2) f2
```

$y\,(-2\,y + x\,y) + (2 - x)\,(-x^2 + y^2)$

```
g = (y) f1 + (-x + 2) f2 // Expand
```

$-2\,x^2 + x^3$

Observation: The leading power product x^3 of g is

neither a multiple of the leading power product $x\,y$ of f_1

nor a multiple of the leading power product y^2 of f_2.

Definition of Gröbner Bases (B. Buchberger 1965, P. Gordon 1899)

A set F of polynomials is called a Gröbner basis (w.r.t. the chosen ordering of power products) iff the above phenomenon cannot happen, i.e. iff

for all f_1, ..., $f_m \in$ F and all polynomials h_1, ..., h_m,

the leading power product of $h_1\,f_1 + \ldots + h_m\,f_m$

is a multiple of the leading power product of

at least one of the polynomials in F.

Counterexample: The Set $F = \{f_1, f_2\}$ of the Above Example is not a Gröbner basis.

Example of a Gröbner Basis

The following set G (results from F by adding $-2x^2 + x^3$) is a Gröbner basis:

```
G = {-2 x^2 + x^3, -2 y + x y, -x^2 + y^2}
```

$\{-2x^2 + x^3, -2y + xy, -x^2 + y^2\}$

For example,

```
(1 + 3 y) (-2 x^2 + x^3) + (8 x + 3 x y) (-2 y + x y) + (2 - x - y^2) (-x^2 + y^2) // Expand
```

$-4x^2 + 2x^3 - 16xy + 2x^2y + 3x^3y + 2y^2 - 7xy^2 + 4x^2y^2 - y^4$

```
(1) (-2 x^2 + x^3) + (8 x) (-2 y + x y) + (y) (-x^2 + y^2) // Expand
```

$-2x^2 + x^3 - 16xy + 7x^2y + y^3$

Why is it difficult to check whether a given F is a Gröbner basis?

How can we check whether a given F is a Gröbner basis?

How can we get an "equivalent" Gröbner basis G for a given F (which may not be a Gröbner basis)?

The "Main Theorem" of Algorithmic Gröbner Bases Theory (B. Buchberger 1965):

F is a Gröbner basis $\Longleftrightarrow \underset{f_1,f_2\in F}{\forall}$ remainder[F, S–polynomial[f_1, f_2]] = 0.

Proof: Nontrivial. Combinatorial. Some details in the Appendix.

The theorem reduces an infinite check to a finite check: Recall definition of "F is a Gröbner basis":

for all f_1, ..., $f_m \in$ F and polynomials h_1, ..., h_m,

the leading power product of $h_1 f_1 + ... + h_m f_m$

is a multiple of the leading power product of at least one of the polynomials in F.

The power of the Gröbner bases method is contained in this theorem and its proof.

Alternative approach: Establish upper bound on degree of polynomials that may occur in the linear combinations: [Hermann 1926], see also lectures by J.C. Faugere.

S-Polynomials

```
f1 = -2 y + x y
f2 = -x^2 + y^2
```

$-2\,y + x\,y$

$-x^2 + y^2$

```
S-polynomial[f1, f2] = y f1 - x f2
```

$y\,(-2\,y + x\,y) - x\,(-x^2 + y^2)$

```
S-polynomial[f1, f2] = y f1 - x f2    // Expand
```

$x^3 - 2\,y^2$

The Algorithm 'remainder'

Roughly, remainder[F, g] results from replacing power products in *g* by a lower products using the polynomials in *F* until no more replacements are possible.

Example:

Consider, again,

```
f1 = -2 x^2 + x^3;
f2 = -2 y + x y;
f3 = -x^2 + y^2;
```

```
F = {f1, f2, f3};
```

and

```
g = x y - 3 x y^2;
```

A "reduction" ("division") step on g w.r.t. F:

```
g1 = g + (3 x) f3
```

$x\,y - 3\,x\,y^2 + 3\,x\,(-x^2 + y^2)$

```
g1 = g + (3 x) f3 // Expand
```

$-3\,x^3 + x\,y$

A next division step w.r.t. F:

```
F = {f1, f2, f3}
```

$\{-2\,x^2 + x^3,\ -2\,y + x\,y,\ -x^2 + y^2\}$

```
g2 = g1 + (-1) f2 // Expand
```

$-3\,x^3 + 2\,y$

A next division step w.r.t. F:

```
F = {f1, f2, f3}
```

$\{-2\,x^2 + x^3,\ -2\,y + x\,y,\ -x^2 + y^2\}$

```
g3 = g2 + (3) f1 // Expand
```

$-6\,x^2 + 2\,y$

This is the remainder of the division of g w.r.t. F because ...

Remainder Algorithms are Available in all Math Systems

```
PolynomialReduce[g, F, {y, x}]
```

$\{\{0,\ 1 - 3\,y,\ -6\},\ -6\,x^2 + 2\,y\}$

Note: the remaindering algorithm can be extended to a "remaindering with co-factors":

```
g - 0  (-2 x^2 + x^3) - (1 - 3 y) ( -2 y + x y) - (-6 ) (-x^2 + y^2) // Expand
```

$-6x^2 + 2y$

Now We Can *Check* Gröbnerianity

Let's again look to the above example:

```
F = {f1, f2, f3}
```

$\{-2x^2 + x^3, -2y + xy, -x^2 + y^2\}$

```
PolynomialReduce[f1 y - f2 x^2, F, {y, x}]
```

{{0, 0, 0}, 0}

```
PolynomialReduce[f1 y^2 - f3 x^3, F, {y, x}]
```

$\{\{4 + 2x + x^2, -4y - 2xy, -8\}, 0\}$

```
PolynomialReduce[f2 y - f3 x, F, {y, x}]
```

{{1, 0, -2}, 0}

The Problem of *Constructing* Gröbner Bases

Given F, find G s.t. Ideal(F) = Ideal(G) and G is a Gröbner basis.

(Ideal(F) := the set of all linear combinations $h_1 f_1 + \ldots + h_m f_m$

with $f_1, \ldots, f_m \in$ F and $h_1, \ldots, h_m$ arbitrary polynomials.)

An Algorithm for *Constructing* Gröbner Bases (B. Buchberger, 1965)

Recall the main theorem:

$$F \text{ is a Gröbner basis} \iff \underset{f_1, f_2 \in F}{\forall} \text{remainder}[F, \text{S–polynomial}[f_1, f_2]] = 0.$$

Based on the main theorem, the problem can be solved by the following algorithm:

Start with G:= F.

For any pair of polynomials f_1, $f_2 \in G$:

h := remainder[G, S–polynomial[f_1, f_2]]

If $h = 0$, consider the next pair.

If $h \neq 0$, add h to G and iterate.

The algorithm allows many refinements and variants which, however, are all based on the notion of S-polynomial and variants of the main theorem.

Many improvements to this crude form of the algorithm have been proposed and investigated over the years, see literature hints below. First significant improvement: Use of "criteria" (for detecting unnecessary reductions), see [Buchberger 1979].

Correctness and Termination of the Algorithm

Correctness: Easy as soon as we know the main theorem.

Termination: by Dickson's Lemma (Dickson 1913, BB 1970).

A sequence p_1, p_2, ... of power products with the property that, for all i < j, p_i does not divide p_j, must be finite.

Specializations

The Gröbner bases algorithm,

for linear polynomials, specializes to Gauss' algorithm, and

for univariate polynomials, specializes to Euclid's algorithm.

Example

Let's again look at

```
f1 = -2 y + x y
f2 = -x^2 + y^2
```

$-2\,y + x\,y$

$-x^2 + y^2$

```
F = {f1, f2}
```

$\{-2\,y + x\,y,\ -x^2 + y^2\}$

F is not a Gröbner basis.

The S-polynomial of f_1, f_2:

```
S-polynomial[f1, f2] = y f1 - x f2    // Expand
```

$x^3 - 2\,y^2$

Its remainder w.r.t. F is:

$$-2\,x^2 + x^3.$$

All the other S-polynomials have remainder 0. Hence, we arrived at a Gröbner basis.

The Gröbner basis algorithm is available now available in all math software systems, e.g. in *Mathematica*:

```
G = GröbnerBasis[F, {y, x}]
```

$\{-2\,x^2 + x^3,\ -2\,y + x\,y,\ -x^2 + y^2\}$

Reduced Gröbner Bases

A set F of polynomials is called a reduced Gröbner basis (w.r.t. the chosen ordering of power products) iff

F is a Gröbner bases and,

for all $f \in F$,

remainder[F-{f}, f] = f and

f is monic.

Algorithm for obtaining a reduced Gröbner basis: Compute a Gröbner basis and then "auto-reduce" the basis.

Extended Gröbner Basis Algorithm

Keeps track of how the polynomials in the Gröbner basis G can be linearly combined from the polynomials in F.

Gröbner Bases: What and How?

Applications of Gröbner Bases

Discussion

Applications are Based on Three Main Properties of Gröbner Bases

Canonicality Property

Elimination Property

Syzygy Property

Canonicality

Remaindering modulo a Gröbner basis F is a "canonical simplifier" for congruence modulo F:

$f \equiv_F g \iff \text{remainder}[F, f] = \text{remainder}[F, g]$

$f \equiv_F \text{remainder}[F, f]$

"Second order" canonicality: "Reduced Gröbner basis" is a "canonical simplifier" for "have same congruence":

$\text{Ideal}[F] = \text{Ideal}[G] \iff \text{reduced–Gröbner–basis}[F] = \text{reduced–Gröbner–basis}[G]$

$\text{Ideal}[F] = \text{Ideal}[\text{ reduced–Gröbner–basis}[F]\ ].$

Elimination Ideals

Let $\prec$ be the lexicographic ordering defined by $x_1 \prec x_2 \prec \ldots \prec x_n$. If F is a Gröbner basis w.r.t. $\prec$, then

If F is a Gröbner basis w.r.t. $\prec$ then, for all $i \leq n$,

$$\text{Ideal}[F] \cap K[x_1, \ldots, x_i] = \text{Ideal}[\ F \cap K[x_1, \ldots, x_i]\ ]$$

The "elimination ideals" of an ideal can be easily computed if we have a Gröbner basis for the ideal.

Syzygy Property (Linear Syzygies)

Given a tuple $\langle f_1, \ldots, f_m \rangle$ of polynomials. How can we obtain a finite basis for the set of all possible polynomial solutions ("syzygies") $\langle h_1, \ldots, h_m \rangle$ of the linear diophantine equation

$$h_1 . f_1 + \ldots + h_m . f_m = 0 \quad ?$$

In the case that F:= $\{f_1, \ldots, f_m\}$ is a Gröbner basis the following set of tuples is a finite basis for the infinite set of all syzgies:

consider all pairs f_i, f_j :

m := LCM[LPP[f_i], LPP[f_j]], u_i := m / LPP[f_i], u_j := m / LPP[f_j]

$\langle H_1, \ldots, H_m \rangle$:= the cofactors obtained by remaindering

S-polynomial[f_i, f_j] modulo F

$\langle h_1, \ldots, h_i, \ldots, h_j, \ldots, h_m \rangle$:=

$$\langle -H_1, \ldots, u_i - H_i, \ldots, -u_j - H_j, \ldots, -H_m \rangle$$

Summarizing: the S-polynomials give a handle for obtaining a finite basis for the set of all syzygies!

(The inhomogeneous equation

$$h_1 . f_1 + \ldots + h_m . f_m = g$$

can be solved by finding one solution of the inhomogeneous equation and adding the solutions of the homogeneous equations.)

(In case $\{f_1, \ldots, f_m\}$ is not a Gröbner basis, transform to Gröbner basis by the extended Gröbner basis algorithm, solve, and transform solutions back.)

(The case of several linear diophantine equations with polynomial coefficients can be reduced to the case of one equation. Alternatively, the entire Gröbner bases approach can be formulated for polynomial "modules" instead of polynomial rings.)

Application: Solving Polynomial Systems

Is based on the elimination property of Gröbner bases (w.r.t. lexicographic orderings).

A Simple System of Equations

```
f1 = -2 y + x y
f2 = -x^2 + y^2
```

$-2\,y + x\,y$

$-x^2 + y^2$

Find x, y such that

```
-2 y + x y = 0
-x^2 + y^2 = 0
```

We compute

```
G = GröbnerBasis[F, {y, x}]
```

$\{-2\,x^2 + x^3,\ -2\,y + x\,y,\ -x^2 + y^2\}$

```
Solve[-2 x^2 + x^3 == 0, x]
```

$\{\{x \to 0\},\ \{x \to 0\},\ \{x \to 2\}\}$

```
{G[[2]], G[[3]]} /. {x → 2}
```

$\{0,\ -4 + y^2\}$

```
Solve[-4 + y^2 == 0, y]
```

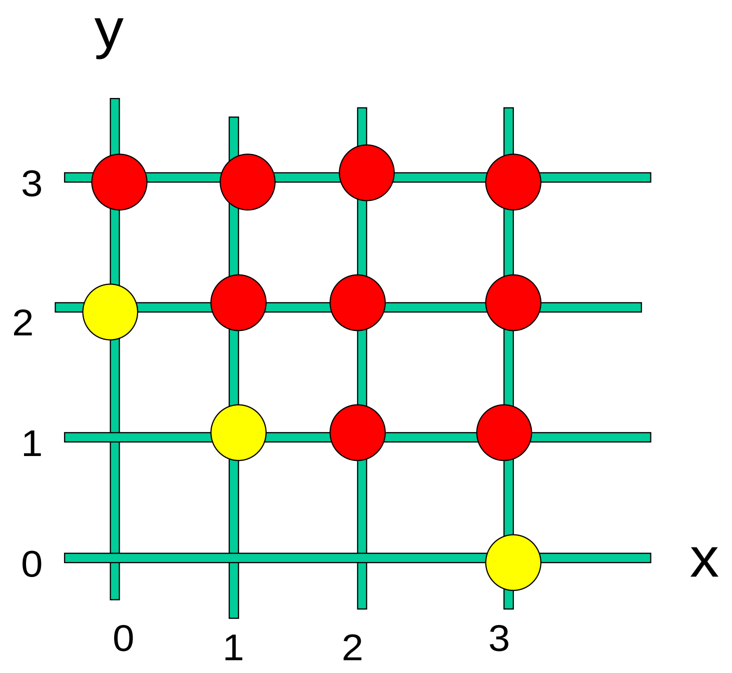
$\{\{y \to -2\},\ \{y \to 2\}\}$

All this is already implemented in the *Mathematica* general Solve function:

```
Solve[{f1 == 0, f2 == 0}, {x, y}]
```

$\{\{x \to 0, y \to 0\}, \{x \to 0, y \to 0\}, \{x \to 2, y \to -2\}, \{x \to 2, y \to 2\}\}$

Theorem (Buchberger 1970): Solvability and the number of solutions can be predicted from the form of the Gröbner basis.



A More Complicated System of Equations

```
f1 = x y - 2 y z - z;
f2 = y^2 - x^2 z + x z;
f3 = z^2 - y^2 x + x;

F = {f1, f2, f3};
```

```
{time, G} = GröbnerBasis[F] // Timing
```

$\{0.\ \text{Second},$
$\{-z - 4 z^3 + 17 z^4 - 3 z^5 + 45 z^6 - 60 z^7 + 29 z^8 - 124 z^9 + 48 z^{10} - 64 z^{11} + 64 z^{12},$
$-22001 z + 14361 y z + 16681 z^2 + 26380 z^3 + 226657 z^4 + 11085 z^5 - 90346 z^6 - 472018 z^7 - 520424 z^8 - 139296 z^9 - 150784 z^{10} + 490368 z^{11},$
$43083 y^2 - 11821 z + 267025 z^2 - 583085 z^3 + 663460 z^4 - 2288350 z^5 + 2466820 z^6 - 3008257 z^7 + 4611948 z^8 - 2592304 z^9 + 2672704 z^{10} - 1686848 z^{11},$
$43083 x - 118717 z + 69484 z^2 + 402334 z^3 + 409939 z^4 + 1202033 z^5 - 2475608 z^6 + 354746 z^7 - 6049080 z^8 + 2269472 z^9 - 3106688 z^{10} + 3442816 z^{11}\}\}$

```
zsolexact1 = Solve[G[[1]] == 0, z]
```

```
{{z → 0},
 {z → Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 - 124 #1^8 + 48 #1^9 -
      64 #1^10 + 64 #1^11 &, 1]},
 {z → Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 2]},
 {z → Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 3]},
 {z → Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 4]},
 {z → Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 5]},
 {z → Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 6]},
 {z → Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 7]},
 {z → Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 8]},
 {z → Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 9]},
 {z → Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 10]},
 {z → Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 11]}}
```

One can compute with these "abstract roots" like with "square roots". For example

```
G0 = G /. zsolexact1[[2]] // Simplify
```

```
{Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 -
     60 #1^6 + 29 #1^7 - 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1] +
  4 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^3 -
  17 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^4 +
  3 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^5 -
  45 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^6 +
  60 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^7 -
  29 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^8 +
  124 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^9 -
  48 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^10 +
  64 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
```

```
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^11 -
  64 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^12,
-22001 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1] +
  14361 y Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1] +
  16681 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^2 +
  26380 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^3 +
  226657 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^4 +
  11085 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^5 -
  90346 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^6 -
  472018 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^7 -
  520424 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^8 -
  139296 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^9 -
  150784 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^10 +
  490368 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^11,
43083 y^2 - 11821 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 +
      29 #1^7 - 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1] +
  267025 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^2 -
  583085 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^3 +
  663460 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^4 -
  2288350 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^5 +
  2466820 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^6 -
  3008257 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^7 +
  4611948 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^8 -
  2592304 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^9 +
  2672704 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
        124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^10 -
  1686848 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
```

```
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^11,
 43083 x - 118717 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 +
     29 #1^7 - 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1] +
  69484 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^2 +
  402334 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^3 +
  409939 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^4 +
  1202033 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^5 -
  2475608 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^6 +
  354746 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^7 -
  6049080 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^8 +
  2269472 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^9 -
  3106688 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^10 +
  3442816 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^11}
```

```
G0 /. {Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 -
      60 #1^6 + 29 #1^7 - 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1] → α}
```

```
{α + 4 α^3 - 17 α^4 + 3 α^5 - 45 α^6 + 60 α^7 - 29 α^8 + 124 α^9 - 48 α^10 + 64 α^11 - 64 α^12,
 -22001 α + 14361 y α + 16681 α^2 + 26380 α^3 + 226657 α^4 + 11085 α^5 -
  90346 α^6 - 472018 α^7 - 520424 α^8 - 139296 α^9 - 150784 α^10 + 490368 α^11,
 43083 y^2 - 11821 α + 267025 α^2 - 583085 α^3 + 663460 α^4 - 2288350 α^5 +
  2466820 α^6 - 3008257 α^7 + 4611948 α^8 - 2592304 α^9 + 2672704 α^10 - 1686848 α^11,
 43083 x - 118717 α + 69484 α^2 + 402334 α^3 + 409939 α^4 + 1202033 α^5 -
  2475608 α^6 + 354746 α^7 - 6049080 α^8 + 2269472 α^9 - 3106688 α^10 + 3442816 α^11}
```

No more details in this talk!

Alternatively, compute numerically:

```
zsol = NSolve[G[[1]] == 0, z]
```

```
{{z → -0.331304 - 0.586934 i}, {z → -0.331304 + 0.586934 i},
 {z → -0.296413 - 0.705329 i}, {z → -0.296413 + 0.705329 i},
 {z → -0.163124 - 0.37694 i}, {z → -0.163124 + 0.37694 i},
 {z → 0.}, {z → 0.0248919 - 0.89178 i}, {z → 0.0248919 + 0.89178 i},
 {z → 0.468852}, {z → 0.670231}, {z → 1.39282}}
```

```
Gsubnum = G /. zsol[[1]]
```

$\{1.33227\times10^{-15} + 9.71445\times10^{-17}\, i,$
$(-523.519 - 4967.65\, i) - (4757.86 + 8428.97\, i)\, y,$
$(-7846.9 - 8372.06\, i) + 43083\, y^2,\ (-16311.7 + 16611.\, i) + 43083\, x\}$

```
PolynomialGCD[Gsubnum[[2]], Gsubnum[[3]]]
```

1

```
ysol = NSolve[ Gsubnum[[2]] == 0, y]
```

$\{\{y \to -0.473535 - 0.205184\, i\}\}$

```
ysol = NSolve[ Gsubnum[[3]] == 0, y]
```

$\{\{y \to -0.473535 - 0.205184\, i\},\ \{y \to 0.473535 + 0.205184\, i\}\}$

Theorem (Roider, Kalkbrener et al. 1990): It suffices to consider the poly in y with lowest degree.

```
xsol = NSolve[ Gsubnum[[4]] == 0, x]
```

$\{\{x \to 0.378611 - 0.385558\, i\}\}$

```
F /. zsol[[1]] /. ysol[[1]] /. xsol[[1]]
```

$\{-1.88738\times10^{-15} + 2.35922\times10^{-16}\, i,$
$1.02696\times10^{-15} + 1.52656\times10^{-16}\, i,\ -1.94289\times10^{-15} - 1.11022\times10^{-16}\, i\}$

Application: Invariant Theory

A Question: Can

```
h = x1^7 x2 - x1 x2^7
```

$x_1^7\, x_2 - x_1\, x_2^7$

be expressed as a polynomial in

```
F = {x1^2 + x2^2, x1^2 x2^2, x1^3 x2 - x1 x2^3}
```

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

?

Note: These polynomials are fundamental invariants for $\mathbb{Z}_4$, i.e. a set of generators for the ring

$$\{f \in \mathbb{C}[x_1, x_2] \mid f(x_1, x_2) = f(-x_2, x_1)\},$$

i.e.

```
{x1^2 + x2^2, x1^2 x2^2, x1^3 x2 - x1 x2^3} /. {x1 -> -x2, x2 -> x1}
```

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

and all invariants can be expressed as polynomials in these invariants.

Reduction to Gröbner Bases Computation

```
{time, GB} = GröbnerBasis[
   {-i1 + x1^2 + x2^2, -i2 + x1^2 x2^2, -i3 + x1^3 x2 - x1 x2^3}, {x2, x1, i3, i2, i1}] // Timing
```

$$\{0.\ \text{Second},\ \{i_1^2 i_2 - 4 i_2^2 - i_3^2,\ -i_2 + i_1 x_1^2 - x_1^4,\ i_1^2 i_3 x_1 - 2 i_2 i_3 x_1 - i_1 i_3 x_1^3 + i_1^2 i_2 x_2 - 4 i_2^2 x_2,\ i_1^2 x_1 - 2 i_2 x_1 - i_1 x_1^3 + i_3 x_2,\ -i_1 i_3 + 2 i_3 x_1^2 - i_1^2 x_1 x_2 + 4 i_2 x_1 x_2,\ -i_3 x_1 - 2 i_2 x_2 + i_1 x_1^2 x_2,\ -i_3 - i_1 x_1 x_2 + 2 x_1^3 x_2,\ -i_1 + x_1^2 + x_2^2\}\}$$

```
PolynomialReduce[x1^7 x2 - x1 x2^7, GB,
 {x2, x1, i3, i2, i1}, MonomialOrder -> Lexicographic]
```

$$\left\{\left\{0,\ -i_3 - \frac{1}{2} i_1 x_1 x_2 - x_1^3 x_2,\ 0,\ \frac{3 i_1 x_2}{4} - \frac{1}{2} x_1^2 x_2 + \frac{x_2^3}{2},\ i_1 - \frac{x_1^2}{2} + \frac{3 x_2^2}{4},\ \frac{3 i_1 x_1}{2} + x_1 x_2^2,\ \frac{x_2^4}{2},\ -\frac{1}{4} i_1^2 x_1 x_2 - \frac{1}{2} i_1 x_1 x_2^3 - x_1 x_2^5\right\},\ i_1^2 i_3 - i_2 i_3\right\}$$

Theorem (Sweedler, Sturmfels et al. 1988): *h* can be represented in terms of *I* iff remainder of *h* w.r.t. "Gröbner basis of *I* with slack variables" is a polynomial in the slack variables (which gives the representation).

```
i1^2 i3 - i2 i3 /. {i1 -> x1^2 + x2^2, i2 -> x1^2 x2^2, i3 -> x1^3 x2 - x1 x2^3} // Expand
```

$$x_1^7 x_2 - x_1 x_2^7$$

```
R = PolynomialReduce[x1^6 x2 - x1 x2^6, GB,
  {x2, x1, i3, i2, i1}, MonomialOrder -> Lexicographic]
```

$$\left\{\left\{0, \frac{i_1 x_1}{2} - i_1 x_2 - x_1^2 x_2, 0, \frac{3 i_1}{4} - \frac{x_1^2}{2} + \frac{x_2^2}{2}, -\frac{x_1}{4} + \frac{3 x_2}{4}, \frac{3 i_1}{4} + x_1 x_2, \frac{x_2^3}{2}, -\frac{1}{4} i_1^2 x_1 - \frac{1}{2} i_1 x_1 x_2^2 - x_1 x_2^4\right\}, -i_1^3 x_1 + 2 i_1 i_2 x_1 + \frac{1}{2} i_1 i_3 x_1 + i_1^2 x_1^3 - i_2 x_1^3 + \frac{1}{2} i_3 x_1^3 + \frac{1}{2} i_1 i_2 x_2\right\}$$

$x_1^6 x_2 - x_1 x_2^6$ can not be expressed by the fundamental invariants in I.

```
x1^6 x2 - x1 x2^6 /. {x1 -> -x2, x2 -> x1}
```

$$x_1^6 x_2 + x_1 x_2^6$$

Application: Automated (Dis-) Proving in Geometry

Reduction of the Problem to Gröbner bases computation:

Geo Theorem $\longrightarrow$ (by coordinatization)

$$\underset{x,y,\ldots}{\forall} \left(\text{poly1}(x,y,\ldots)=0 \wedge \ldots \Rightarrow \text{poly}(x,y,\ldots)=0 \right) \longrightarrow$$

$$\neg \underset{x,y,\ldots}{\exists} \left(\text{poly1}(x,y,\ldots)=0 \wedge \ldots \wedge \text{poly}(x,y,\ldots)\neq 0 \right) \longrightarrow$$

$$\neg \underset{x,y,\ldots,a}{\exists} \left(\text{poly1}(x,y,\ldots)=0 \wedge \ldots \wedge a \,.\, \text{poly}(x,y,\ldots) - 1 = 0 \right)$$

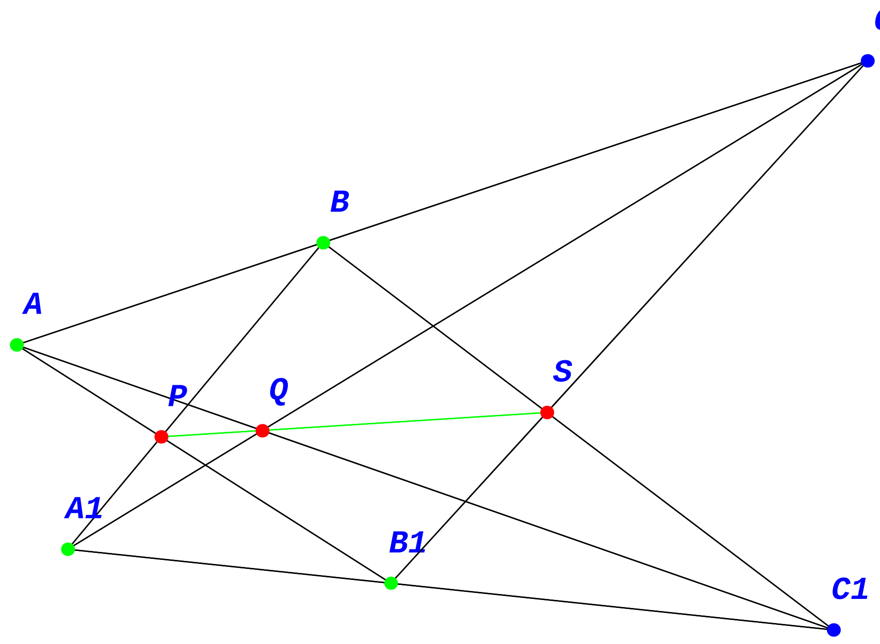
The latter question can be decided by the Gröbner basis method!

The method is implemented in the Theorema System:

B. B., C. Dupre, T. Jebelean, F. Kriftner, K. Nakagawa, D. Vasaru, W. Windsteiger. The Theorema Project: A Progress Report. In: Symbolic Computation and Automated Reasoning (Proceedings of CALCULEMUS 2000, Symposium on the Integration of Symbolic Computation and Mechanized Reasoning, August 6-7, 2000, St. Andrews, Scotland), M. Kerber and M. Kohlhase (eds.), A.K. Peters, Natick, Massachusetts, ISBN 1-56881-145-4, pp. 98-113.

Example: Pappus Theorem

- What does the theorem say geometrically?



- Textbook formulation:

Let A,B, C and A1,B1, C1 be on two lines and P = AB1 ⋂ A1B, Q = AC1 ⋂ A1C, S = BC1 ⋂ B1C. Then P, Q, and S are collinear.

- Input to the system:

```
Proposition["Pappus", any[A, B, A1, B1, C, C1, P, Q, S],
 point[A, B, A1, B1] ∧ pon[C, line[A, B]] ∧ pon[C1, line[A1, B1]] ∧
   inter[P, line[A, B1], line[A1, B]] ∧ inter[Q, line[A, C1], line[A1, C]] ∧
   inter[S, line[B, C1], line[B1, C]] ⇒ collinear[P, Q, S]]
```

- Input to the system:

```
Prove[Proposition["Pappus"], by → GeometryProver,
 ProverOptions → {Method -> "GröbnerProver", Refutation → True}]
```

- Notebook generated automatically by the proving algorithm based on Gröbner basis algorithm:

Prove:

(Proposition (Pappus))

$$\underset{A,B,A1,B1,C,C1,P,Q,S}{\forall} (\mathrm{point}[A, B, A1, B1] \wedge \mathrm{pon}[C, \mathrm{line}[A, B]] \wedge \mathrm{pon}[C1, \mathrm{line}[A1, B1]] \wedge \mathrm{inter}[P, \mathrm{line}[A, B1], \mathrm{line}[A1, B]] \wedge \mathrm{inter}[Q, \mathrm{line}[A, C1], \mathrm{line}[A1, C]] \wedge \mathrm{inter}[S, \mathrm{line}[B, C1], \mathrm{line}[B1, C]] \Rightarrow \mathrm{collinear}[P, Q, S])$$

with no assumptions.

To prove the above statement we shall use the Gröbner basis method. First we have to transform the problem into algebraic form.

Algebraic Form:

To transform the geometric problem into algebraic form we have to chose first an orthogonal coordinate system.

Let's have the origin in point A, and points $\{B, C\}$ on the two axes.

Using this coordinate system we have the following points:

$$\{\{A, 0, 0\}, \{B, 0, u_1\}, \{A1, u_2, u_3\}, \{B1, u_4, u_5\}, \{C, 0, u_6\}, \{C1, u_7, x_1\}, \{P, x_2, x_3\}, \{Q, x_4, x_5\}, \{S, x_6, x_7\}\}$$

The algebraic form of the assertion is:

(1)

$$\underset{x_1,x_2,x_3,x_4,x_5,x_6,x_7}{\forall} (u_3 u_4 + -u_2 u_5 + -u_3 u_7 + u_5 u_7 + u_2 x_1 + -u_4 x_1 == 0 \land u_5 x_2 + -u_4 x_3 == 0 \land -u_1 u_2 + u_1 x_2 + -u_3 x_2 + u_2 x_3 == 0 \land x_1 x_4 + -u_7 x_5 == 0 \land -u_2 u_6 + -u_3 x_4 + u_6 x_4 + u_2 x_5 == 0 \land u_1 u_7 + -u_1 x_6 + x_1 x_6 + -u_7 x_7 == 0 \land -u_4 u_6 + -u_5 x_6 + u_6 x_6 + u_4 x_7 == 0 \Rightarrow x_3 x_4 + -x_2 x_5 + -x_3 x_6 + x_5 x_6 + x_2 x_7 + -x_4 x_7 == 0)$$

This problem is equivalent to:

(2)

$$\neg \left(\underset{x_1,x_2,x_3,x_4,x_5,x_6,x_7}{\exists} (u_3 u_4 + -u_2 u_5 + -u_3 u_7 + u_5 u_7 + u_2 x_1 + -u_4 x_1 == 0 \land u_5 x_2 + -u_4 x_3 == 0 \land -u_1 u_2 + u_1 x_2 + -u_3 x_2 + u_2 x_3 == 0 \land x_1 x_4 + -u_7 x_5 == 0 \land -u_2 u_6 + -u_3 x_4 + u_6 x_4 + u_2 x_5 == 0 \land u_1 u_7 + -u_1 x_6 + x_1 x_6 + -u_7 x_7 == 0 \land -u_4 u_6 + -u_5 x_6 + u_6 x_6 + u_4 x_7 == 0 \land x_3 x_4 + -x_2 x_5 + -x_3 x_6 + x_5 x_6 + x_2 x_7 + -x_4 x_7 \neq 0)\right)$$

To remove the last inequality, we use the Rabinowitsch trick: Let v_0 be a new variable. Then the problem becomes:

(3)

$$\neg \left(\underset{x_1,x_2,x_3,x_4,x_5,x_6,x_7,v_0}{\exists} (u_3 u_4 + -u_2 u_5 + -u_3 u_7 + u_5 u_7 + u_2 x_1 + -u_4 x_1 == 0 \land u_5 x_2 + -u_4 x_3 == 0 \land -u_1 u_2 + u_1 x_2 + -u_3 x_2 + u_2 x_3 == 0 \land x_1 x_4 + -u_7 x_5 == 0 \land -u_2 u_6 + -u_3 x_4 + u_6 x_4 + u_2 x_5 == 0 \land u_1 u_7 + -u_1 x_6 + x_1 x_6 + -u_7 x_7 == 0 \land -u_4 u_6 + -u_5 x_6 + u_6 x_6 + u_4 x_7 == 0 \land 1 + -v_0 (x_3 x_4 + -x_2 x_5 + -x_3 x_6 + x_5 x_6 + x_2 x_7 + -x_4 x_7) == 0)\right)$$

This statement is true iff the corresponding Gröbner basis is {1}.

The Gröbner bases is $\{1\}$.

Hence, the statement and the original assertion is true.

Statistics:

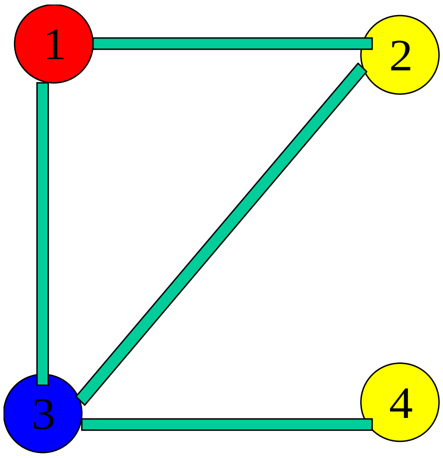
Time needed to compute the Gröbner bases: 0.42 Seconds.

Application: Graph Coloring

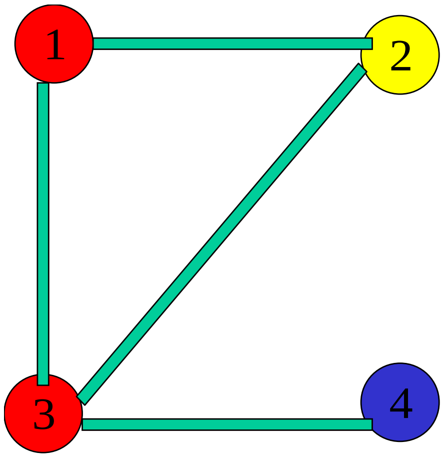
The Problem:

Find all admissible colorings in k colors of a graph with n vertices and edges E:

An admissible coloring in 3 colors of a graph with 4 vertices and edges {1,2}, {1,3}, {2,3}, {3,4}:



Not an admissible coloring in 3 colors of the same graph:



The Translation into a Gröbner Bases Problem

Theorem: The possible colorings of the above graph correspond 1-1 to the common solutions of the following set of polynomials:

```
(x1^2 + x1 x2 + x2^2) (x1 - x2) // Expand
```

$x_1^3 - x_2^3$

```
{ -1 + x1^3,    ... at vertex 1 color is a 3 - ary root of 1
  - 1 + x2^3,   ... at vertex 2 color is a 3 - ary root of 1
  - 1 + x3^3,
 -1 + x4^3,
 x1^2 + x1 x2 + x2^2,   ... the colors at 1 and 2 must be different,
 x1^2 + x1 x3 + x3^2,
 x2^2 + x2 x3 + x3^2,
 x3^2 + x3 x4 + x4^2}
```

Solution by Gröbner Bases

Compute a Gröbner basis of this polynomial set and compute all solutions.

```
GB = GröbnerBasis[{-1 + x1^3, -1 + x2^3, -1 + x3^3, -1 + x4^3,
   x1^2 + x1 x2 + x2^2, x1^2 + x1 x3 + x3^2, x2^2 + x2 x3 + x3^2, x3^2 + x3 x4 + x4^2},
  {x4, x3, x2, x1}]
```

$\{-1 + x_1^3,\ x_1^2 + x_1 x_2 + x_2^2,\ -x_1 - x_2 - x_3,\ -x_1 x_2 + x_1 x_4 + x_2 x_4 - x_4^2\}$

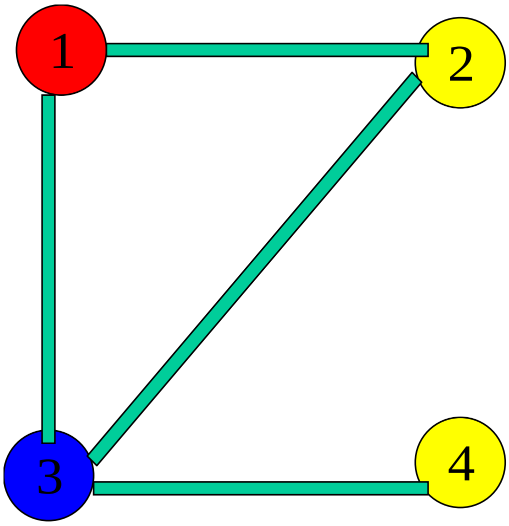
```
Solve[{-1 + x1^3 == 0, -1 + x2^3 == 0, -1 + x3^3 == 0, -1 + x4^3 == 0,
  x1^2 + x1 x2 + x2^2 == 0, x1^2 + x1 x3 + x3^2 == 0, x2^2 + x2 x3 + x3^2 == 0, x3^2 + x3 x4 + x4^2 == 0},
 {x4, x3, x2, x1}]
```

$\{\{x_4 \to 1,\ x_2 \to 1,\ x_1 \to -1 + (-1)^{1/3},\ x_3 \to -(-1)^{1/3}\},$
$\{x_4 \to 1,\ x_2 \to -1 + (-1)^{1/3},\ x_1 \to 1,\ x_3 \to -(-1)^{1/3}\},$
$\{x_4 \to 1,\ x_2 \to -1 - (-1)^{2/3},\ x_1 \to 1,\ x_3 \to (-1)^{2/3}\},$
$\{x_4 \to 1,\ x_2 \to (-1)^{1/3} - (-1)^{2/3},\ x_1 \to -(-1)^{1/3},\ x_3 \to (-1)^{2/3}\},$
$\{x_4 \to -(-1)^{1/3},\ x_2 \to -1 + (-1)^{1/3},\ x_1 \to -(-1)^{1/3},\ x_3 \to 1\},$
$\{x_4 \to -(-1)^{1/3},\ x_2 \to -1 - (-1)^{2/3},\ x_1 \to 1,\ x_3 \to (-1)^{2/3}\},$
$\{x_4 \to -(-1)^{1/3},\ x_2 \to -1 - (-1)^{2/3},\ x_1 \to (-1)^{2/3},\ x_3 \to 1\},$
$\{x_4 \to -(-1)^{1/3},\ x_2 \to (-1)^{1/3} - (-1)^{2/3},\ x_1 \to -(-1)^{1/3},\ x_3 \to (-1)^{2/3}\},$
$\{x_4 \to (-1)^{2/3},\ x_2 \to -1 + (-1)^{1/3},\ x_1 \to -(-1)^{1/3},\ x_3 \to 1\},$
$\{x_4 \to (-1)^{2/3},\ x_2 \to -1 - (-1)^{2/3},\ x_1 \to (-1)^{2/3},\ x_3 \to 1\},$
$\{x_4 \to -1 + (-1)^{1/3},\ x_2 \to 1,\ x_1 \to -1 + (-1)^{1/3},\ x_3 \to -(-1)^{1/3}\},$
$\{x_4 \to -1 + (-1)^{1/3},\ x_2 \to -1 + (-1)^{1/3},\ x_1 \to 1,\ x_3 \to -(-1)^{1/3}\}\}$

Slightly re-organized output:

$\{\{x_1 \to 1,\ x_2 \to -(-1)^{1/3},\ x_3 \to -1 + (-1)^{1/3},\ x_4 \to 1\},$
$\{x_1 \to 1,\ x_2 \to -(-1)^{1/3},\ x_3 \to -1 + (-1)^{1/3},\ x_4 \to -(-1)^{1/3}\},$
$\{x_1 \to 1,\ x_2 \to (-1)^{2/3},\ x_3 \to -1 - (-1)^{2/3},\ x_4 \to 1\},$
$\{x_1 \to 1,\ x_2 \to (-1)^{2/3},\ x_3 \to -1 - (-1)^{2/3},\ x_4 \to (-1)^{2/3}\},$
$\{x_1 \to -(-1)^{1/3},\ x_2 \to 1,\ x_3 \to -1 + (-1)^{1/3},\ x_4 \to 1\},$
$\{x_1 \to -(-1)^{1/3},\ x_2 \to 1,\ x_3 \to -1 + (-1)^{1/3},\ x_4 \to -(-1)^{1/3}\},$
$\{x_1 \to -(-1)^{1/3},\ x_2 \to -1 + (-1)^{1/3},\ x_3 \to 1,\ x_4 \to -(-1)^{1/3}\},$
$\{x_1 \to -(-1)^{1/3},\ x_2 \to -1 + (-1)^{1/3},\ x_3 \to 1,\ x_4 \to -1 + (-1)^{1/3}\},$
$\{x_1 \to (-1)^{2/3},\ x_2 \to 1,\ x_3 \to -1 - (-1)^{2/3},\ x_4 \to 1\},$
$\{x_1 \to (-1)^{2/3},\ x_2 \to 1,\ x_3 \to -1 - (-1)^{2/3},\ x_4 \to (-1)^{2/3}\},$
$\{x_1 \to (-1)^{2/3},\ x_2 \to -1 - (-1)^{2/3},\ x_3 \to 1,\ x_4 \to (-1)^{2/3}\},$
$\{x_1 \to (-1)^{2/3},\ x_2 \to -1 - (-1)^{2/3},\ x_3 \to 1,\ x_4 \to -1 - (-1)^{2/3}\}\}$

For example, $\{x_1 \to 1,\ x_2 \to -(-1)^{1/3},\ x_3 \to -1 + (-1)^{1/3},\ x_4 \to -(-1)^{1/3}\}$ corresponds to



Application: Integer Optimization

Example (B. Sturmfels):

What is the minimum number of coins (e.g. p Pennies, n Nickels, d Dimes, q Quarters) for composing a given value, e.g. 117?

Reduction to Gröbner Bases Problem (C. Traverso et al. 1986):

Code the integer values p, n, d, q as exponents of power products!

Code the goal function as the (generalized) degree of the power products!

Code the exchange rules of the coins (the relations between the quantities) as polynomials consisting of power products:

```
F = {P^5 - N, P^10 - D, P^25 - Q}
```

$\{-N + P^5,\ -D + P^{10},\ P^{25} - Q\}$

Now compute the Gröbner basis of F (w.r.t. degree ordering):

```
G = GröbnerBasis[F, MonomialOrder → DegreeLexicographic]
```

$\{-D + N^2,\ -D^3 + N\,Q,\ D^2\,N - Q,\ -N + P^5\}$

Now you can be sure that, starting with any admissible solution (e.g. (p=17, n=10, d=5, q=0), by reduction modulo *G*, you will end up with a minimal solution:

```
PolynomialReduce[P^17 N^10 D^5, G, , MonomialOrder → DegreeLexicographic]
```

$\{\{D^9\,P^{17} + D^8\,N^2\,P^{17} + D^7\,N^4\,P^{17} + D^6\,N^6\,P^{17} + D^5\,N^8\,P^{17} + D^4\,P^{17}\,Q^2 + P^7\,Q^4,$
$-D^7\,P^{17} - D^4\,N\,P^{17}\,Q - D^2\,P^{17}\,Q^2,\ P^{17}\,Q^3,\ D\,P^2\,Q^4 + N\,P^7\,Q^4 + P^{12}\,Q^4\},\ D\,N\,P^2\,Q^4\}$

Answer: take 4 quarters, 1 dime, 1 nickel, 2 pennies.

Other Applications

Algebraic Geometry

Coding Theory

Cryptography

Invariant Theory

Integer Optimization

Statistics

Symbolic Integration

Symbolic Summation

Differential Equations: Boundary Value Problems

Systems Theory

Gröbner Bases: What and How?

Applications of Gröbner Bases

Discussion

How Difficult is the Construction of Gröbner Bases?

Very Easy

The structure of the algorithm is easy. The operations needed in the algorithm are elementary. "Every high-school student can execute the algorithm." (See palm-top TI-98.)

Very Difficult

The inherent complexity of the problems that can be solved by the GB method (e.g. graph colorings) is "exponential". Hence, the worst-case complexity of the GB algorithm *must* be high.

Sometimes Easy

Mathematically interesting examples often have a lot of "structure" and, in concrete examples, GB computations can be reasonably, even surprisingly, fast.

Enormous Potential for Improvement

More *mathematical* theorems can lead to drastic speed-up:

- The use of "criteria" for eliminating the consideration of certain S-polynomials.
- *p*-adic approaches and floating point approaches.
- The "Gröbner Walk" approach.
- The "linear algebra" approach (see lectures by J.C. Faugere).
- The "numerics" approach.

Tuning of the algorithm:

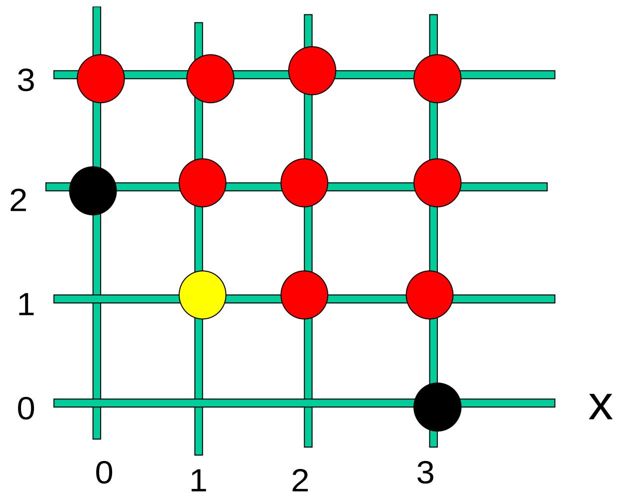
- Heuristics, strategies for choosing orderings, selecting S-polynomials etc.
- Good implementation techniques.

  A huge literature.

Why "Gröbner" Bases?

Professor Wolfgang Gröbner (1899-1980) was my PhD thesis supervisor.

He gave me the problem of finding "the uncovered points if the black points are given".



My PhD Thesis:

B. Buchberger.

Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.

English Translation (by M. Abramson): An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal. Journal of Symbolic Computation, Special Issue on Logic, Mathematics, and Computer Science: Interactions. Volume 41, Number 3-4, Pages 475-511, 2006.

In this thesis and the subsequent (1970) journal publication I introduced:

* the concept of Gröbner bases and reduced Gröbner bases
* the S-polynomials
* the main theorem with proof
* the algorithm with termination and correctness proof
* the uniqueness of Gröbner bases
* first applications (computing in residue rings, Hilbert function, algebraic systems)
* the technique of base-change w.r.t. to different orderings

* a complete computer implementation

* first complexity considerations.

Also various details of the algorithm are discussed, which later have been forgotten:

* store intermediate polynomials

* presentation of power products by numbers

* polynomial reduction as linear algebra row reductions

(In the early days of computing, as a tendency, one tried to save memory and compensate by time. Thus, the S-poly theorem allowed to forget about intermediate reductions!)

However, in the thesis, I did not use the name "Gröbner bases". I introduced this name only in 1976, for honoring Gröbner, when people started to become interested in my work.

My later contributions:

* the technique of criteria for eliminating unnecessary reductions [Buchberger 1979]

* an abstract characterization of "Gröbner bases rings", [Buchberger 1983 ff.]

More Info on Gröbner Bases?

Gröbner Bases 98 Conference:

B. Buchberger, F. Winkler (eds.). Gröbner *Bases: Theory and Applications.* Cambridge University Press, 1998. 560 pages.

This book contains tutorials and original papers.

This book contains also:

B. B. *Introduction to Gröbner Bases*, pp. 3-31.

B. B. *An Algorithmic Criterion for the Solvability of Systems of Algebraic Equations, pp. 540-560.* (English translation of the original paper from 1970, the first journal publication of my work.)

Also

Gröbner Bases on Your Desk and in Your Palm

GB implementations are contained in all the current math software systems like *Mathematica* (see demo), Maple, Magma, Macsyma, Axiom, Derive, Reduce, Mupad, ...

Software systems specialized on Gröbner bases: RISA-ASIR (M. Noro, K. Yokoyama), CoCoA, Macaulay, Singular, ...

Gröbner bases are now availabe even on the TI-98 (implemented in Derive).

Textbooks on Gröbner Bases

T. Kreuzer, L. Robbiano: *Algorithmic Commutative Algebra I.* Springer, Heidelber, 2000: Contains a list of all other, approx. 10, textbooks on GB.

W.W.Adams, P. Loustenau. *Introduction to* Gröbner *Bases*. Graduate Studies in Mathematics: Amer. Math. Soc., Providence, R.I., 1994.

T.Becker, V.Weispfenning. Gröbner *Bases: A Computational Approach to Commutative Algebra.* Springer, New York, 1993.

D.Cox, J.Little, D.O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Springer, New York, 1992.

....

M. Maruyama. Gröbner Bases and Applications. 2002.

M. Noro, K. Yokoyama. Computational Fundamentals of Gröbner Bases. University of Tokyo Press, 2003.

Original Publications on Gröbner Bases

Approximately 1000 papers appeared meanwhile on Gröbner bases.

J of Symbolic Computation, in particular, special issues.

ISSAC Conferences.

Mega Conferences.

ACA Conferences.

...

The essential additional original ideas in the literature:

- Gröbner bases can be constructed w.r.t. arbitrary "admissible" orderings (W. Trinks 1978)
- Gröbner bases w.r.t. to "lexical" orderings have the elimination property (W. Trinks 1978)
- Gröbner bases can be used for computing syzygies and the S-polys generate the module of syzygies (G. Zacharias 1978)
- A given *F*, w.r.t. the *infinitely* many admissible orderings, has only *finitely* many Gröbner bases and, hence, we can construct a "universal" Gröbner bases for *F* (L. Robbiano, V. Weispfenning, T. Schwarz 1988)

- Starting from a Gröbner bases for *F* for ordering O_1 one can "walk", by changing the basis only slightly, to a basis for a "nearby" ordering O_2 and so on ... until one arrives at a Gröbner bases for a desired ordering O_k (Kalkbrener, Mall 1995, Nam 2000).
- Use arbitrary linear algebra algorithms for the reduction (remaindering) process: (Faugère 1997 ff.).
- The numerics of Gröbner bases computation.
- ... numerours applications,

Early forerunners:

Paul Gordon.

Ein neuer Beweis des Hilbertschen Satzes über homogene Funktionen (A New Proof of Hilbert's Theorem on Homogeneous Functions), Nachr. der Königl. Ges. der Wiss. zu Göttingen 3 (1899), pp. 240-242.

Grete Hermann.

Die Frage der endlich vielen Schritte in der Theorie der Polynomideale (The Question of Finitely Many Steps in Polynomial Ideal Theory), Math. Ann. 95 (1926), pp. 736-788.

Research Topics

- the inner structure of Gröbner bases: generalized Sylvester matrices
- the numerics of GB computations
- axiomatic characterization of Gröbner rings
- generalizations (e.g. non-commutative poly-rings)
- speeding up the computation
- Gröbner bases for particular classes of ideals (avoid computation)
- the study of admissible orderings
- new applications

Appendix: Sketch of the Proof of the Main Theorem

Details see B. Buchberger, Introduction to Gröbner Bases, in [Buchberger, Winkler 1998], pp. 1 - 31.

Equivalent definition of Gröbner bases:

F is a Gröbner basis $\Longleftrightarrow$ $\rightarrow_F$ has the Church-Rosser property.

f $\to_F$ g ... f reduces to g in one remaindering step using divisors from F.

$$\to \text{ is Church-Rosser} \iff \underset{g_1, g_2}{\forall} \left(g_1 \leftrightarrow^* g_2 \Longrightarrow g_1 \downarrow^* g_2 \right)$$

Main Theorem:

$$\boxed{F \text{ is a Gröbner basis} \iff \underset{f_1, f_2 \in F}{\forall} \text{remainder}[F, \text{S–polynomial}[f_1, f_2]] = 0.}$$

Proof: "$\Longrightarrow$": Easy.

For the direction "$\Longleftarrow$" one can use the Newman Lemma (Newman 1942). (For the version of the algorithm with criteria one needs the generalized Newman lemma by BB.) For Noetherian $\to$:

$$\to \text{ is Church-Rosser} \iff \underset{g_1, g_2, h}{\forall} \left(g_1 \leftarrow \text{h} \to g_2 \Longrightarrow g_1 \downarrow^* g_2 \right)$$

The proof of this lemma uses Noetherian induction. By using Newman's lemma in the proof of the main theorem, one takes induction out of the proof and is left with the specific technicalities of polynomial reduction.

Hence, we have to consider, for arbitrary polynomials g_1, g_2, h, the situation that

$$g_1 \leftarrow_F h \to_F g_2$$

and we have to show that we can always find a polynomial p such that

$$g_1 \to_F{}^* \text{ p } \leftarrow_F{}^* g_2.$$

By the assumption, there exist polynomials f_1 and f_2 in F such that h reduces w.r.t. f_1 and f_2. Let t_1 and t_2 be the power products in h on which these reductions work.

$$\text{h} = \ldots + \square\, t_1 + \ldots. + \square\, t_1 + \ldots..$$

$$\begin{array}{cc} -\, u_1 f_1 & -\, u_2 f_2 \\ \text{yields } g_1 & \text{yields } g_2 \end{array}$$

Cases $t_1 \prec t_2$ and $t_2 \prec t_1$ easy (but not trivial!): by "semi-compatibility" of polynomial reduction.

Cases t:= $t_1 = t_2$:

$$\text{h} = \ldots + \square\, t + \ldots..$$

$- u_1 f_1 \qquad - u_2 f_2$

In this case t is a multiple of the LCM m of LPP[f_1] and LPP[f_2]: t = v . m.

Since, by assumption of the theorem, the S-polynomial of f_1 and f_2 can be reduced to 0, the reduction of m in the two essentially different ways (starting once by using f_1 and once by using f_2) has a common successor.

Hence, by "stability" of polynomial reduction, by multiplication of all the steps by v, g_1 and g_2 have a common successor.

My Recent Research Interest: Automated Theory Exploration

For example: How can one invent (and verify) notions like "S-polynomial", theorems like the main theorem, and algorithms like the Gröbner bases algorithm automatically, i.e. by algorithms that work on formulae.

For example, algorithm synthesis:

Given the specification P of a problem.

Find an algorithm A such that $\underset{F}{\forall}$ P[F, A[F]].

I succeeded to come up with a method which, for many P, yields A automatically. In particular, with this method, starting from the specification of the Gröbner bases construction problem:

Given: F.

Find: G such that

G is finite

G is a Gröbner basis

Ideal[F] = Ideal[G],

one arrives automatically at the notion of S-polynomials and the above Gröbner bases algorithm based on the notion of S-polynomials.

For details see the recent publication

B. Buchberger

Towards the Automated Synthesis of a Gröbner Bases Algorithm

RACSAM, 98/1 (Rev. Acad. Cienc., Spanish Royal Academy of Science), 98/1, pp. 65-75, 2005.

and the Workshop C "Formal Gröbner Bases Theory", March 6-10, 2006, in the course of the Special Semester on Gröbner bases.