The Algorithmic Invention of a Groebner Basis Algorithm

Computer Algebra Seminar, JINR, Dubna, May 25−26,2004

Bruno Buchberger

RISC (Research Institute for Symbolic Computation)
Johannes Kepler University
RICAM (Radon Institute for Computational and Applied Mathematics)
Academy of Science
Linz, Austria

Dedicated to the Memory of Mikhail G. Mescsheryakov

■ First Floor: An Algorithm for Nonlinear Problems

■ Some Nonlinear Problems

■ Nonlinear Polynomial Equations

$$x y - 2 y z - z = 0$$
$$y^2 - x^2 z + x z = 0$$
$$z^2 - y^2 x + x = 0$$

(x, y, z) ?

■ Linear Representability of Nonlinear Polynomials

Do there exist polynomials (h1, h2, h3) such that

$$(x\,y - 2\,y\,z - z)\ \ h1 + \\ (y^2 - x^2\,z + x\,z)\ \ h2 + \\ (z^2 - y^2\,x + x)\ \ h3 \ = \ x^2\,y - 3\,z^2 + z$$

?

(If yes, find them "all".)

■ Nonlinear Representability of Nonlinear Polynomials

Can

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

be expressed as a polynomial in

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

?

Note: The above polynomials forms a system of fundamental invariants for $\mathbb{Z}_4$, i.e. a set of generators for the ring

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

■ Other Problems

- determine dimension of algebraic manifolds,
- ideal and radical membership decision,
- effective operations on ideals
- effective computation in residue class rings modulo polynomial ideals,
- Hilbert functions,
- implicitization,
- inverse polynomial mappings
- ....
- dozens of problems in invariant theory, automated geometric theorem proving, coding theory, integer programming, symbolic summation, statistics, systems theory, ...

■ The Problem of Constructing Gröbner Bases

■ All these Problems Can be Reduced to the Construction of Gröbner Bases

Find algorithm Gb such that

$$\forall_{F} \begin{pmatrix} \text{is-finite}[\ \text{Gb}[F]\ ] \\ \text{is-Gröbner-basis}[\ \text{Gb}[F]] \\ \text{ideal}[F] = \text{ideal}[\ \text{Gb}[F]]. \end{pmatrix}$$

Definitions [BB 1965, 1970]:

$$\text{is-Gröbner-basis}[G] \Leftrightarrow \text{is-confluent}[\ \rightarrow_G\ ].$$

■ h1 $\rightarrow_G$ h2 : h2 results from h1 by one "division step" using divisors from G

$$(h1 \rightarrow_G h2) \Leftrightarrow \exists_{g \in G} \left(\begin{cases} \text{lp}[g] \mid \text{lp}[h1] \\ h2 = h1 - (\text{lm}[h1] / \text{lm}[g])\, g \end{cases} \right),$$

■ Confluence (Uniqueness) of Division

$$\text{is-confluent}[\ \rightarrow\ ] :\ \Leftrightarrow \forall_{f1,f2} (f1 \leftrightarrow^* f2 \Rightarrow f1 \downarrow^* f2)$$

■ An Algorithm for the Construction of Gröbner Bases [BB 1965, 1970]

$$\mathrm{Gb}[F] = \mathrm{Gb}[F, \mathrm{pairs}[F]]$$
$$\mathrm{Gb}[F, \langle\rangle] = F$$

$$\mathrm{Gb}[F, \langle\langle g1, g2\rangle, \overline{p}\rangle] =$$
$$\text{with}\Big[f = \mathbf{lcm}[\,\mathrm{lp}[g1], \mathrm{lp}[g2]],$$
$$h1 = \mathrm{trd}[\,\mathrm{rd}[f, g1], F], \qquad h2 = \mathrm{trd}[\,\mathrm{rd}[f, g2], F],$$

$$\begin{cases} \mathrm{Gb}[F, \langle\overline{p}\rangle] & \Leftarrow \quad h1 = h2 \\ \mathrm{Gb}\Big[F \frown (h1 - h2), \langle\overline{p}\rangle \asymp \Big\langle \langle F_k, h1 - h2\rangle \underset{k=1,\ldots,|F|}{|} \Big\rangle\Big] & \Leftarrow \quad \text{otherwise} \end{cases} \Big]$$

The algorithm terminates by Dickson's lemma.

After termination: The finitely many **lcm** conflue.

Correctness Theorem and Proof: From the finitely many confluences infer all the infinitely many confluences:

lcm[lp[g1], lp[g2]]

■ Application of the Algorithm: Example

For example, solve nonlinear equations:

$$x\,y - 2\,y\,z - z = 0$$
$$y^2 - x^2\,z + x\,z = 0$$
$$z^2 - y^2\,x + x = 0$$

$$G = \mathrm{GroebnerBasis}[\{x\,y - 2\,y\,z - z,\ y^2 - x^2\,z + x\,z,\ z^2 - y^2\,x + x\},\ \{x, y, z\}]$$

$$\{z + 4z^3 - 17z^4 + 3z^5 - 45z^6 + 60z^7 - 29z^8 + 124z^9 - 48z^{10} + 64z^{11} - 64z^{12},$$

$$-22001z + 14361yz + 16681z^2 + 26380z^3 + 226657z^4 + 11085z^5 - 90346z^6 - 472018z^7 - 520424z^8 - 139296z^9 - 150784z^{10} + 490368z^{11},$$

$$43083y^2 - 11821z + 267025z^2 - 583085z^3 + 663460z^4 - 2288350z^5 + 2466820z^6 - 3008257z^7 + 4611948z^8 - 2592304z^9 + 2672704z^{10} - 1686848z^{11},$$

$$43083x - 118717z + 69484z^2 + 402334z^3 + 409939z^4 + 1202033z^5 - 2475608z^6 + 354746z^7 - 6049080z^8 + 2269472z^9 - 3106688z^{10} + 3442816z^{11}\}$$

The Groebner basis has the "elimination" property: one can solve "one equation after the other".

■ Summary of Gröbner Bases Theory

My Gröbner bases algorithm is now routinely available in all math software systems like Mathematica, Maple, etc.

Approx. 500 papers on Gröbner bases and 10 textbooks.

Dozens of non−trivialproblems reducible to the construction of Gröbner bases.

■ Second Floor: An Algorithm for Inventing Algorithms

■ The Algorithm Invention ("Synthesis") Problem: "One Floor Higher Up"

Given a problem specification P, find an algorithm A such that

$$\forall_x P[x, A[x]].$$

Higher "order": Find an algorithm ("method") S such that

$$\forall_P \forall_x P[x, S[P][x]].$$

Examples of specifications P:

$$P[x, y] \Leftrightarrow \text{is-greater}[x, y]$$

$$P[x, y] \Leftrightarrow \text{is-sorted-version}[x, y]$$

$$P[x, y] \Leftrightarrow \text{is-finite-Gröbner-basis}[x, y]$$

■ The "Lazy Thinking" Method [BB 2001]

Given a problem specification P

- consider various "algorithm schemes" for A

- and try to prove (automatically) $\forall_x P[x, A[x]]$.
- This proof will normally fail because nothing is known on the auxiliary functions in the algorithm scheme.
- From the temporary assumptions and goals in the failing proof situation (automatically) generate specifications for the auxiliary functions that would make the proof possible.

Now, apply the method recursively to the auxiliary functions.

■ 2003: Synthesis of Easy Algorithms

Example: Synthesize A such that

$$\forall_x \text{ is-sorted-version}[x, A[x]].$$

Example of algorithm scheme: "divide and conquer"

$$\forall_x \left(A[x] = \begin{cases} s[x] & \Leftarrow \text{ is-trivial-tuple}[x] \\ m[A[l[x]], A[r[x]]] & \Leftarrow \text{ otherwise} \end{cases} \right)$$

Lazy Thinking automatically (in approx. 2 minutes), using the *Theorema* system, finds the following specifications for the auxiliary functions

$$\forall_x \ (s[x] = x)$$

$$\forall_{y,z} \left(\begin{cases} \text{is-sorted}[y] \\ \text{is-sorted}[z] \end{cases} \Rightarrow \begin{cases} \text{is-sorted}[m[y, z]] \\ m[y, z] \approx (y \asymp z) \end{cases} \right)$$

$$\forall_x \ (l[x] \asymp r[x] \approx x)$$

■ 2004: Synthesis of My Gröbner Bases Algorithm

■ Algorithm Scheme "Critical Pair / Completion"

$$A[F] = A[F, \text{pairs}[F]]$$
$$A[F, \langle\rangle] = F$$

$$A[F, \langle\langle g1, g2\rangle, \overline{p}\rangle] =$$
$$\text{where}\Big[f = \mathbf{lc}[g1, g2],\ h1 = \text{trd}[\text{rd}[f, g1], F],\ h2 = \text{trd}[\text{rd}[f, g2], F],$$
$$\begin{cases} A[F, \langle\overline{p}\rangle] & \Leftarrow \ h1 = h2 \\ A\Big[F \frown \mathbf{df}[h1, h2], \langle\overline{p}\rangle \asymp \Big\langle \langle F_k, \mathbf{df}[h1, h2]\rangle \ \Big|_{k=1,\ldots,|F|} \Big\rangle\Big] & \Leftarrow \ \text{otherwise} \end{cases} \Big]$$

This scheme can be tried in any domain, in which we have a reduction operation rd that depends on sets F of objects and a Noetherian relation $\succ$ which interacts with rd in the following natural way:

$$\forall_{f,g} \ (f \succeq \text{rd}[f, g]).$$

■ The Essential Problem

The problem of synthesizing a Gröbner bases algorithm can now be also stated by asking whether starting with the proof of

$$\forall_{F} \text{ is-finite-Gröbner-basis}[F, A[F]]$$

we can *automatically* arrive at the idea that

$$\mathbf{lc}[g1, g2] = \mathbf{lcm}[lp[g1], lp[g2]]$$

and

$$\mathbf{df}[h1, h2] = h1 - h2$$

are suitable functions that specialize the algorithm scheme to an algorithm that constructs a Gröbner basis for the input F.

(Detecting that lcm enables us to "master the infinite by the finite" was the main invention in algorithmic Gröbner bases theory!)

■ Now Start the (Automated) Correctness Proof

Details cannot be presented in one talk.

With current theorem proving technology, in the *Theorema* system, the proof can be done automatically.

■ Roughly,

It should be clear that, if the algorithm terminates, the final result is a finite set (of polynomials) G that has the property

$$\forall_{g1,g2 \in G} \left(\text{with}\left[f = lc[g1, g2], \right.\right.$$
$$h1 = trd[rd[f, g1], F], \quad h2 = trd[rd[f, g2], F],$$
$$\left.\left. \bigvee \left\{ \begin{array}{l} h1 = h2 \\ df[h1, h2] \in G \end{array} \right. \right] \right).$$

■ Roughly,

We now try to prove that, if G has this property, then

$$\text{is-finite}[G],$$

$$\text{ideal}[F] = \text{ideal}[G],$$

and

$$\text{is-Gröbner-basis}[G], \text{ i.e. is-confluent}[\rightarrow_G].$$

We only deal with the third, most important, property. For this, we assume

$$\begin{cases} p \to_G f1 \\ p \to_G f2. \end{cases}$$

and have to find a polynomial g such that

$$f1 \to_G{}^* g,$$
$$f2 \to_G{}^* g.$$

■ The Proof Fails but ...

by an (automated) analysis of the failing proof situation we detect that the proof could be completed if the unknown **lc** satisfied the following property:

$$\mathrm{lp}[g1] \mid \mathbf{lc}[g1, g2],$$
$$\mathrm{lp}[g2] \mid \mathbf{lc}[g1, g2],$$

$$\forall_{p,g1,g2} \left(\left(\begin{cases} \mathrm{lp}[g1] \mid p \\ \mathrm{lp}[g2] \mid p \end{cases} \right) \Rightarrow (\mathbf{lc}[g1, g2] \mid p) \right).$$

Heureka! It is clear that this specification is (only) met by

$$\mathbf{lc}[g1, g2] = \mathbf{lcm}[\mathrm{lp}[g1], \mathrm{lp}[g2]].$$

Similarly, it can be (automatically) detected that

$$\mathbf{df}[h1, h2] = h1 - h2.$$

■ Conclusion

▫ The Status

BB 1970 invents an algorithm for Gröbner bases construction (and, hence, many other problems).

BB 2001−2004invents algorithm for the invention of algorithms (including the BB 1970 algorithm).

Hence, "BB automated BB".

- **What Does this Mean?**

The algorithmization of mathematics goes higher and higher becoming more and more "symbolic" ....

more symbolic = more automated proving + more mathematics.

- **Mathematical Knowledge Management: The Future of Symbolic Computation**

All this is part of a major new worldwide endeavor: "Mathematical Knowledge Management".

MKM 2001 at RISC
MKM 2003 in Bologna
MKM 2004 in Byalostok
...

EU MKM Network, North American MKM Network.

The *Theorema* system is one of the systems that aim at being a frame for MKM.

The higher we go in MKM, the more "Symbolic" Computation is needed.

bruno.buchberger@jku.at

Come to AISC 2004 !

■ References

■ On Gröbner Bases

[Buchberger 1970]
B. Buchberger. Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems (An Algorithmical Criterion for the Solvability of Algebraic Systems of Equations). Aequationes mathematicae 4/3, 1970, pp. 374−383. (English translation in: [Buchberger, Winkler 1998], pp. 535 −545.) Published version of the PhD Thesis of B. Buchberger, University of Innsbruck, Austria, 1965.

[Buchberger 1998]
B. Buchberger. Introduction to Gröbner Bases. In: [Buchberger, Winkler 1998], pp.3−31.

[Buchberger, Winkler, 1998]
B. Buchberger, F. Winkler (eds.). Gröbner Bases and Applications, Proceedings of the International Conference "33 Years of Gröbner Bases", 1998, RISC, Austria, London Mathematical Society Lecture Note Series, Vol. 251, Cambridge University Press, 1998.

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

■ On Mathematical Knowledge Management

B. Buchberger, G. Gonnet, M. Hazewinkel (eds.)
Mathematical Knowledge Management.
Special Issue of Annals of Mathematics and Artificial Intelligence, Vol. 38, No. 1−3, May 2003, Kluwer Academic Publisher, 232 pages.

A.Asperti, B. Buchberger, J.H.Davenport (eds.)
Mathematical Knowledge Management.
Proceedings of the Second International Conference on Mathematical Knowledge Management (MKM 2003), Bertinoro, Italy, Feb.16−18, 2003, Lecture Notes in Computer Science, Vol.2594, Springer, Berlin−Heidelberg-NewYork, 2003, 223 pages.

■ On Theorema

[Buchberger et al. 2000]
B. Buchberger, C. Dupre, T. Jebelean, F. Kriftner, K. Nakagawa, D. Vasaru, W. Windsteiger. The Theorema Project: A Progress Report. In: M. Kerber and M. Kohlhase (eds.), 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), A.K. Peters, Natick, Massachusetts, ISBN 1−56881−145−4, pp. 98−113.

■ On Theory Exploration and Algorithm Synthesis

[Buchberger 2000]
B. Buchberger. Theory Exploration with *Theorema*.
Analele Universitatii Din Timisoara, Ser. Matematica−Informatica, Vol. XXXVIII, Fasc.2, 2000, (Proceedings of SYNASC 2000, 2nd International Workshop on Symbolic and Numeric Algorithms in Scientific Computing, Oct. 4−6,2000, Timisoara, Rumania, T. Jebelean, V. Negru, A. Popovici eds.), ISSN 1124−970X,pp. 9−32.

[Buchberger 2003]
B. Buchberger. Algorithm Invention and Verification by Lazy Thinking.
In: D. Petcu, V. Negru, D. Zaharie, T. Jebelean (eds), Proceedings of SYNASC 2003 (Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, October 1-4, 2003), Mirton Publishing, ISBN 973-661-104-3, pp. 2-26.

[Buchberger 2004]
B. Buchberger. The Four Parallel Threads of Formal Theory Exploration. Technical Report of the SFB (Special Research Area) Scientific Computing, Johannes Kepler University, Linz, in preparation.

[Buchberger, Craciun 2003]
B. Buchberger, A. Craciun. Algorithm Synthesis by Lazy Thinking: Examples and Implementation in Theorema. in: Fairouz Kamareddine (ed.), Proc. of the Mathematical Knowledge Management Workshop, Edinburgh, Nov. 25, 2003, Electronic Notes on Theoretical Computer Science, volume dedicated to the MKM 03 Symposium, Elsevier, ISBN 044451290X, to appear.

[Buchberger 2004]
B. Buchberger.
Towards the Automated Synthesis of a Gröbner Bases Algorithm.
RACSAM (Review of the Royal Spanish Academy of Science), to appear, 10 pages.