Hagenberg Research

N.N.*

October 22, 2010

1 Computer-Assisted Discovery and Proving

First we comment on *computer-assisted guessing* in the context of mathematical discovery. Then we turn to the activity of proving, more precisely, to *proving methods* where *computer algebra algorithms* are used. Here we restrict to this special type of computed-assisted proving; for *general mathematical proving machines* like the THEOREMA system developed at RISC.

1.1 I.Q. Tests, Rabbits, and the Golden Section

Let us consider the following problem taken from an I.Q. test [10, Aufgabe 13, Denksport I fuer Superintelligente] from the sixties of the last century:

Continue the sequence $1, 1, 2, 3, 5, 8, 13, 21$.

In the 21st century we let the computer do the problem. To this end we load the RISC package `GeneratingFunctions` written by C. Mallinger [19] in the computer algebra system Mathematica:

```
In[1]:= ≪GeneratingFunctions.m
```

In the next step we input a little program that can be used to solve such I.Q. tests automatically:

```
In[2]:= GuessNext2Values[Li_] := Module[{rec},
          rec = GuessRE[Li,c[k],{1,2},{0,3}];
          RE2L[rec[[1]],c[k],Length[Li]+1]]
```

*This is an exercise for the RISC course "Computer based working environments".

Finally the problem is solved automatically with

In[3]:= `GuessNext2Values[{1, 1, 2, 3, 5, 8, 13, 21}]`

Out[3]= {1,1,2,3,5,8,13,21,34,55}

To produce additional values is no problem:

In[4]:= `GuessNext2Values[{1, 1, 2, 3, 5, 8, 13, 21, 34, 55}]`

Out[4]= {1,1,2,3,5,8,13,21,34,55,89,144}

Note. The same automatic guessing can be done in the Maple system; there B. Salvy and P. Zimmermann [24] developed the poineering package `gfun` which has served as a model for the development of Mallinger's `GeneratingFunctions`.

What is the mathematical basis for such automatic guessing? The answer originates in a simple observation: Many of the sequences $(x_n)_{n\geq 0}$ arising in practical applications (and in I.Q. tests!) are produced from a very simple pattern; namely, linear recurrences of the form

$$p_d(n)x_{n+d} + p_{d-1}(n)x_{n+d-1} + \cdots + p_0(n)x_n = 0, \qquad n \geq 0,$$

with coefficients $p_i(n)$ being polynomials in n. So packages like Mallinger's `GeneratingFunctions` try to compute-via an ansatz using undetermined coefficients-a recurrence of exactly this type. For the I.Q. example above a recurrence is obtained by

In[5]:= `GuessRE[{1, 1, 2, 3, 5, 8, 13, 21},f[n]]`

Out[5]= {{-f[n]-f[1+n]+f[2+n]==0,f[0]==1,f[1]==1}, ogf}

Since only finitely many values are given as input, the output recurrence $f_{n+2} = f_{n+1} + f_n$ $(n \geq 0)$ can be only a *guess* about a possible building principle of an *infinite* sequence. However, such kind of automated guessing is becoming more and more relevant to concrete applications. For instance, an application from mathematical chemistry can be found in [8] where a prediction for the total number of benzenoid hydrocarbons was made. Three years later this predication was confirmed [27]. Recently, quite sophisticated applications arose in connection with the enumeration of lattice paths and also with quantum field theory.

In 1202 Leonard Fibonacci introduced the numbers f_n. The fact that $f_0 = f_1 = 1$, and

$$f_{n+2} = f_{n+1} + f_n, \qquad n \geq 0,$$

in Fibonacci's book was given the following interpretation: If baby rabbits become adults after one month, and if each pair of adult rabbits produces one pair of baby rabbits every month, how many pairs of rabbits, starting with one pair, are present after n months?

A non-recursive representation is the celebrated Euler-Binet formula

$$f_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right), \qquad n \geq 0.$$

The number $(1+\sqrt{5})/2 \approx 1.611803$, the *golden ratio*, is important in many parts of mathematics as well as in the art world. For instance, Phidias is said to have used it consciously in his sculpture.

Mathematicians gradually began to discover more and more interesting things about Fibonacci numbers f_n; see e.g. [13]. For example, a typical sunflower has a large head that contains spirals of tightly packed florets, usually with $f_8 = 34$ winding in one direction and $f_9 = 55$ in another.

Another observation is this: Define g_n as a sum over binomial coefficients of the form

$$g_n := \sum_{k=0}^{n} \binom{n-k}{k}.$$

From the values $g_0 = 1$, $g_1 = 1$, $g_2 = 2$, $g_3 = 3$, $g_4 = 5$, and $g_5 = 8$ it is straight-forward to conjecture that the sequence $(g_n)_{n\geq 0}$ is nothing but the Fibonacci sequence $(f_n)_{n\geq 0}$. In the next subsection we shall see that nowadays such statements can be proved automatically with the computer.

1.2 Pi, Inequalities, and Finite Elements

We have seen that linear recurrences can be used as a basis for automated *guessing*. Concerning symbolic computation, this is only the beginning. Namely, following D. Zeilberger's holonomic paradigm [29], the description of mathematical sequences in terms of linear recurrences, and of mathematical functions in terms of linear differential equations, is also of great importance to the design of computer algebra algorithms for automated *proving*.

For example, consider the sequence $(g_n)_{n\geq 0}$ defined above. To prove the statement

$$f_n = g_n, \qquad n \geq 0,$$

in completely automatic fashion, we use the RISC package `Zb` [21], an implementation of D. Zeilberger's algorithm [28]:

In[6]:= ≪`Zb.m`

In[7]:= `Zb[Binomial[n-k,k],{k,0,Infinity},n,2]`
Out[7]= $\{\mathrm{SUM}[n] + \mathrm{SUM}[1+n] - \mathrm{SUM}[2+n] == 0\}$

The output tells us that $g_n = \mathrm{SUM}[n]$ indeed satisfies the same recurrence as the Fibonacci numbers. A proof for the correctness of the output recurrence can be obtained automatically, too; just type the command:

In[8]:= `Prove[]`

For further details concerning the mathematical background of this kind of proofs, see e.g. Zeilberger's articles [29] and [28] which were the booster charge for the development of a new subfield of symbolic computation; namely, the design of computer algebra algorithms for special functions and sequences. For respective RISC developments the interested reader is referred to the web page

`http://www.risc.uni-linz.ac.at/research/combinat`

For various applications researchers are using such algorithms in their daily research work-sometimes still in combination with tables. However, there are particular problem classes where symbolic (and numeric) algorithms are going to replace tables almost completely.

Concerning *special sequences* the most relevant table is N. Sloane's handbook [25], [26]. Sloane's home page provides an extended electronic version of it; also symbolic computation algorithms are used to retrieve information about sequences .

Concerning *special functions* one of the most prominent tables is the 'Handbook' [1] from 1964. Soon it will be replaced by its strongly revised successor, the NIST Digital Library of Mathematical Functions (DLMF); see `http://dlfm.nist.gov`. The author of this section is serving as an associate editor of this new handbook (and author, together with F. Chyzak, of a new chapter on computer algebra) that will be freely available via the web.

We expect the development of special provers will intensify quite a bit. By special provers we mean methods based on computer algebra algorithms specially tailored for certain families of mathematical objects. *Special function inequalities* provide a classical domain that so far has been considered as

being hardly accessible by such methods. To conclude this section we briefly describe that currently this situation is about to change.

Consider the famous Wallis product formula for π:

$$\pi = 2 \cdot \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \frac{8}{7} \cdot \frac{8}{9} \cdot \cdots .$$

This product is an immediate consequence ($n \to \infty$) of the following inequality (John Wallis, Arithmetica Infinitorum, 1656):

$$\frac{2n}{2n+1} \leq \frac{c_n}{\pi} \leq 1, \qquad n \geq 0,$$

where

$$c_n := \frac{2^{4n+1}}{2n+1} \binom{2n}{n}^{-2}.$$

In analysis one meets such inequalities quite frequently. Another example, similar to that of Wallis, is

$$\frac{1}{4n} \leq a_n \leq \frac{1}{3n+1}, \qquad n \geq 0,$$

where

$$a_n := \frac{1}{2^{4n}} \binom{2n}{n}^{2}.$$

We shall prove the right hand side, i.e. $a_n \leq 1/(3n+1)$, (the left hand side goes analogously) to exemplify the new Gerhold-Kauers method [12] for proving special function/sequence inequalities. As proof strategy they use mathematical induction combined with G. Collins' cylindrical algebraic decomposition (CAD). First observe that

$$a_{n+1} = a_n \frac{(2n+1)^2}{(2n+2)^2} \leq \frac{1}{3n+1} \frac{(2n+1)^2}{(2n+2)^2},$$

where for the inequality the induction hypothesis is used. In order to show that this implies $a_{n+1} \leq 1/(3n+4)$, it is sufficient to establish that

$$\frac{1}{3n+1} \frac{(2n+1)^2}{(2n+2)^2} \leq \frac{1}{3n+4}.$$

But this step can be carried out automatically with any implementation of Collins' CAD; for instance, in Mathematica:

In[9]:= `Reduce[`$\frac{1}{3n+1}\frac{(2n+1)^2}{(2n+2)^2} \leq \frac{1}{3n+4}, n$`]`
Out[9]= $-\frac{4}{3} < n < -1 \;||\; -1 < n < -\frac{1}{3} \;||\; n \geq 0$

The Gerhold-Kauers method already found quite a number of non-trivial applications. They range from new refinements of Wallis' inequality [20] like

$$\frac{32n^2+32n+7}{4(2n+1)(4n+3)} \leq \frac{c_n}{\pi} \leq \frac{16(n+1)(2n+1)}{32n^2+56n+25}, \qquad n \geq 0,$$

to a proof of the long-standing log-concavity conjecture of V. Moll [17]. Further applications and details about the method are given in [16].

We want to conclude by referring to results that emerged from numerical-symbolic SFB collaboration in the context of finite element methods (FEM). In order to set up a new FEM setting, J. Schöberl (RWTH Aachen, formerly JKU) needed to prove the following special function inequality:

$$\sum_{j=0}^{n}(4j+1)(2n-2j+1)P_{2j}(0)P_{2j}(x) \geq 0$$

for $-1 \leq x \leq 1$, $n \geq 0$, and with $P_{2j}(x)$ being the Legendre polynomials. Using the Gerhold-Kauers method together with RISC symbolic summation software, V. Pillwein [22] was able to settle this conjecture. Remarkably, there is still no human proof available!

Last but not least, we mention a recent collaboration of J. Schöberl with C. Koutschan (RISC), which led to a new tool for engineering applications in the context of electromagnetic wave simulation. Formulas derived by Koutschan's symbolic package `HolonomicFunctions` resulted in a significant speed-up of numerical FEM algorithms e.g. for the construction of antennas or mobile phones. The method is planned to be registered as a patent.

2 Gröbner Bases Theory for Nonlinear Polynomial Systems

2.1 The Relevance of Gröbner Bases Theory

To a great extent, Gröbner bases theory was the starting point of the Research Institute for Symbolic Computation and, hence, the Softwarepark

Hagenberg. Gröbner bases theory was initiated in the PhD thesis [3, 4] and turned out to be one of the first coherent results in the emerging area of what was later called "computer algebra". Gröbner bases theory allows to handle a big variety of fundamental problems connected to systems of multivariate polynomials, for example the problem of solving such systems (finding all common roots of such systems) or the problem of deciding whether two given multivariate polynomials are "equivalent" with respect to a given system of multivariate polynomials.

Since nonlinear polynomial systems are a mathematical model for a large class of problems in science and engineering, it is no surprise that a general algorithmic method like the Gröbner bases method for handling such systems has an unlimited range of applications. In fact, in many fields of science and engineering, prior to the advent of Gröbner bases theory only linear approximations of the actual problems could be studied. In some cases, if we are satisfied with approximate solutions, linear approximations to the original models may be good enough. However, there are many areas in which only the exact treatment of the exact non-linear problems gives meaningful answers. For example, graph coloring problems can be translated into the problem of solving certain non-linear polynomial systems, see below, where each solution corresponds to a possible coloring. Linear approximations to the systems or approximations to the solutions of the original systems would not make it possible to distinguish between or identify the various colorings.

Over the years, a many applications of Gröbner bases theory, some of them quite surprising, have been found. An overview on these applications, up to 1998, can be found in the proceedings [6]. An online-bibliography has been compiled at the occasion of the Special Semester on Gröbner Bases at the Radon Institute for Computational and Applied Mathematics (RICAM) in Linz 2006, which contains over 1000 papers on Gröbner bases, see `www.ricam.oeaw.ac.at/spec_sem/srs/groeb/` (follow link "Bibliography"). A quick way of getting access to the growing literature on Gröbner bases is to use the online citation index "citeseer" (at researchindex.org/). If one enters "Gröbner" or "Buchberger", one will obtain several thousand citations of papers containing contributions to the development, extension and improvement of the Gröbner bases method and its many applications. Also, there are a couple of textbooks available on Gröbner bases, see for example [2] and [18]. The latter contains a list of most other textbooks on Gröbner bases in its introduction.

Applications of Gröbner bases reach from algebraic geometry or poly-

nomial ideal theory (the original field for which Gröbner bases theory was invented in [3, 4]) to invariant theory, coding theory, cryptography and cryptoanalysis, systems theory, control theory, automated geometrical theorem proving, graph theory, invention and proof of combinatorial identities, software engineering, integration of differential equations and many others. Here are some surprising recent applications of Gröbner bases in quite distinct areas:

Origami Construction The Japanese art of Origami aims at constructing two-dimensional and three-dimensional objects by certain folding operations starting from a square paper sheet. Six classes of folding operations are permitted. The mathematical problem consists in deciding whether a given sequence of operations provenly leads to an object having prescribed properties. For example, ways were proposed to fold a regular heptagon from the initial square using only Origami folding operations. In this case, the question is to *prove* rigorously that a proposed sequence of operations results, indeed, in a heptagon. Gröbner bases can be used for proving or disproving the correctness of arbitrary such sequences of operations for arbitrary properties (that can be described by multivariate polynomials) completely automatically. The method consists, roughly, in translating the sequence of operations into a set of polynomial relations (which is easily possible) and to check whether or not the polynomial that describes the desired property is in the "ideal" generated by the polynomial relations, which is always possible by the Gröbner bases methodology. For details, see for example [15].

Solution of Linear Boundary Value Problems Initial value problems for a wide class of differential equations can be solved by symbolic methods. For boundary value problems, there were hardly any symbolic methods available. A generalization of Gröbner bases theory for non-commutative polynomials allows now to obtain symbolic solutions also for boundary value problems. In this new application, the strength of the Gröbner bases method is demonstrated by the fact that the invention of the Green's functions, which was deemed to be an ad hoc creative process for each boundary value problem, is replaced by a completely algorithmic procedure, which is nothing else than just the reduction ("remaindering") operation w.r.t. a (non-commutative)

Gröbner basis, which represents the relations between the fundamental operations of functional analysis for boundary problems, see [23].

Optimization of Oil Platforms (the "Algebraic Oil Project") In this surprising application, the fundamental problem of improved control of the valves on an oil platform, with unknown geometry of the oil caverns under the sea, is attacked. In a "learning phase" the quantity of oil produced in dependence on the position of the valves on the platform is measured. The assumption is made that this dependence can be described by a system of multivariate polynomials (whose coefficients are unknown in the learning phase). With the data collected from sufficiently many measurements, the Gröbner bases method allows then to determine these coefficients (in fact, the system of polynomials generated for modelling the flow will be a Gröbner basis). Now, this multivariate polynomial model of the flow can be used, in the "application phase", to optimize the flow w.r.t. various criteria. This new application of (a numerical variant of) Gröbner bases was proposed in a cooperation between Shell company and the CoCoA Group, see [14]. The results are practically promising.

Automated Synthesis of Loop Invariants for Programs The proof that programs meet their specification is one of the fundamental problems in computer science. The method of "loop invariants" for solving this problem requests that, for certain points in the given program, an assertion (formula), called a "loop invariant" is invented for which one can prove that, for every moment the program gets to that point the respective assertion is true for the values of the program variables. The invention of these loop invariants often needs quite some creativity and this is a major obstacle for the practical use of the method of loop invariants. In the *Theorema* Group at RISC, a method was developed by which, for a wide class of programs, these loop invariants can be generated by a combination of symbolic execution of the program, solution of the resulting recursive equations, see Section 1 above, and the use of the Gröbner bases method.

Breaking Cryptographic Codes Gröbner bases are being used both for constructing cryptosystems as well as for trying to break such systems (cryptoanalysis). Breaking an (algebraic) crypto-code basically amounts to solving a system of nonlinear algebraic equations with

Boolean coefficients for the values that constitute the bits of the unknown code, i.e. the number of unknowns in the system is the number of unknown bits in the code. Typically, this number is 80 or more. Recently, proposals for algebraic codes that have been deemed to be sufficiently safe have been broken using the Gröbner bases method, see [11]. This was one of the most exciting recent applications of Gröbner bases.

The Determination of Species Relationship in Evolution In this research
area, the probabilities of one species being closer in the evolution with some species than with some other species are determined from an analysis of the genetic codes of species. The result of such an analysis is called the phylogenetic tree of the species. In [7] it has recently been shown how this problem of finding the mutual neighborhood probabilities can be cast into the language of multivariate polynomial ideals in Gröbner bases form.

Wavelets Wavelets are spectra of functions. Each function in a spectrum is determined by a couple of parameters. By combining the functions in a spectrum, i.e. by specifying the values of the individual parameters in a spectrum, (graphical) information can be presented in highly condensed form ("data compression"). The search for suitable spectra of wavelets is an important research topic in wavelet theory. This search leads to systems of algebraic equations that recently have been solved by the Gröbner bases method, see [9].

Gröbner bases theory is still a very active research area with focus on generalizations of the method (e.g. the non-commutative case), specializations for certain classes of polynomial sets (e.g. toric sets) with higher efficiency, new approaches to compute Gröbner bases for improving the efficiency, numeric variants of the method, and new applications in a big spectrum of different areas.

2.2 Gröbner Bases: Basic Notions and Results

Gröbner bases are sets of multivariate polynomials that enjoy certain uniqueness properties, which make it possible to solve many fundamental problems on such sets algorithmically. The main result of algorithmic Gröbner bases

theory is that any finite set of multivariate polynomials can be transformed, by an algorithm, into an equivalent Gröbner basis and that, hence, many fundamental problems on arbitrary sets of multivariate polynomials can be solved algorithmically by, first, transforming the sets into Gröbner bases form and then using the respective algorithms for Gröbner bases. Three examples of such fundamental problems that can be solved algorithmically by transformation into Gröbner bases form are:

- the exact solution of systems of multivariate polynomial equations,
- the problem of deciding whether or not two given multivariate polynomials are equivalent w.r.t. to a given set of multivariate polynomials that define the equivalence,
- the problem of solving "diophantine" equations, i.e. the problem of finding (all) multivariate polynomials that satisfy linear relations whose coefficients are also multivariate polynomials.

We explain here one of the many different, equivalent, ways of defining the notion of Gröbner bases. For this, consider for example the two quadratic bivariate polynomials f_1 and f_2 in the indeterminates x and y:

$$f_1 := -2y + xy \qquad\qquad f_2 := x^2 + y^2.$$

If we fix an ordering on the power products (for example, the lexicographic ordering that ranks y higher than x), each polynomial has a "leading power product", in our case xy and y^2, respectively. Consider now the following linear combination g of f_1 and f_2:

$$g := (y)f_1 + (-x + 2)f_2 = 2x^2 - x^3.$$

Observation: The leading power product x^3 of g is neither a multiple of the leading power product xy of f_1 nor a multiple of the leading power product y^2 of f_2. Now, a set F of multivariate polynomials is called a *Gröbner basis* (w.r.t. the chosen ordering of power products) iff the above phenomenon cannot happen, i.e.

for all $f_1, \ldots, f_m \in F$ and all (*infinitely many* possible) polynomials $h_1, \ldots, h_m$, 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.

Example 2.1 *The Set $F := \{f_1, f_2\}$ is not a Gröbner basis. The equivalent Gröbner basis is $\{f_1, f_2, f_3\}$, where $f_3 := 2x^2 - x^3$, which can only be checked by the theorem below.*

The following theorem is the crucial result on which the algorithmic usefulness of Gröbner bases hinges.

Theorem 2.2 (Buchberger) *F is a Gröbner basis iff, for all f_1, f_2, the remainder of the S-polynomial of f_1 and f_2 w.r.t. F is 0.*

The remainder of a multivariate polynomial w.r.t. a set of such polynomials is the rest in a generalized polynomial division, which is an algorithmic process. The S-polynomial of two multivariate polynomials is obtained by multiplying the two polynomials with the lowest possible power products that make the leading power products equal and by subtracting the resulting two polynomials. In the above example, the S-polynomial of f_1 and f_2 is

$$y(-2y + xy) - x(x^2 + y^2) = -x^3 - 2y^2.$$

The proof of this theorem is difficult, see [5] for a concise version. The algorithmic power of the Gröbner bases method is based on this theorem and its proof because the theorem shows, essentially, that the infinite test appearing in the definition of Gröbner bases for checking whether or not a given set F is a Gröbner basis can be replaced by the finite, algorithmic, test on the right-hand side of the theorem! This theorem can now be transformed into an algorithm for *constructing* Gröbner bases, i.e. for the problem to find, for any given multivariate polynomial set F, a set G such that G is a Gröbner basis and F and G generate the same set of linear combinations, see Algorithm 1.

The notion of Gröbner bases, the theorem on the characterization of Gröbner bases by S-polynomials, and the algorithm for the construction of Gröbner bases, together with termination proof, first applications and complexity considerations, were introduced in the PhD thesis [3] and the corresponding journal publication [4]. Buchberger gave the name "Gröbner" to his theory for honoring his PhD thesis advisor Wolfgang Gröbner (1899–1980).

Example 2.3 *Solving the problem of graph coloring by Gröbner bases. This problem consists in finding all admissible colorings in k colors of a graph with n vertices and edges E. A coloring of the vertices of a graph is admissible if*

Algorithm 1 Buchberger's Algorithm

Start with $G \leftarrow F$
for all pairs of polynomials $f_1, f_2 \in G$ **do**
  $h \leftarrow$ remainder of the S-polynomial of f_1 and f_2 w.r.t G
  if $h = 0$ **then**
    consider the next pair
  else
    add h to G and iterate
  end if
end for

no two adjacent vertices obtain the same color. For example, the left picture in Figure 1 is an admissible coloring in 3 colors of a graph with 4 vertices and edges $\{1,2\}$, $\{1,3\}$, $\{2,3\}$, $\{3,4\}$, *whereas the right picture in Figure 1 is not an admissible coloring in 3 colors of the same graph.*

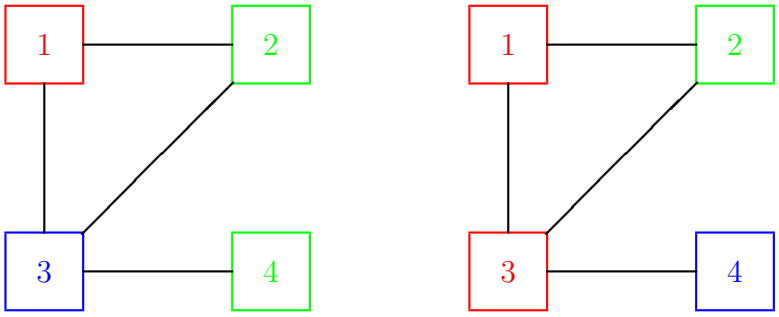

Figure 1: An admissible and a non-admissible coloring of a graph.

It is easy to see that the possible colorings of a graph can be obtained by considering all solutions of a certain system of polynomial equations (where the n *indeterminates appearing in the polynomials correspond to the colors a the* n *vertices). We illustrate the construction of the polynomial system in*

the example:

$$
\begin{array}{ll}
\{-1+x_1^3, & \ldots \text{ the color at vertex 1 is a ternary root of 1, i.e. the three ternary roots of 1 encode the three colors} \\
-1+x_2^3, & \ldots \text{ the color at vertex 2 is a ternary root of 1,} \\
-1+x_3^3, & \\
-1+x_4^3, & \\
x_1^2+x_1x_2+x_2^2, & \ldots \text{ the colors at 1 and 2 must be different} \\
x_1^2+x_1x_3+x_3^2, & \\
x_2^2+x_2x_3+x_3^2, & \\
x_3^2+x_3x_4+x_4^2\} &
\end{array}
$$

Now, compute a Gröbner basis of this polynomial set (this can be done by using for example Mathematica because, nowadays, Buchberger's algorithm is routinely available in all mathematical software systems) and compute all solutions. The corresponding Gröbner basis is:

$$\left\{-1+x_1^3, x_1^2+x_1x_2+x_2^2, -x_1-x_2-x_3, -x_1x_2+x_1x_4+x_2x_4-x_4^2\right\}.$$

One sees that the corresponding Gröbner basis is "decoupled" (this is one of the fundamental properties of Gröbner bases w.r.t. to lexicographic orderings), i.e. it can be completely solved by determining the values of one indeterminate after the other, starting with the first polynomial, which is always a polynomial in the first indeterminate only.

For example, the solution

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

corresponds to the coloring illustrated in Figure 2.

Oversimplified, the strategy for solving problems with Gröbner bases consists of the following steps:

1. Describe the problem (e.g. "coloring"), if possible, by sets of multivariate polynomials (e.g. polynomials on "roots of unity" instead of "colors").

2. Transform the occurring sets of polynomials into Gröbner basis form.

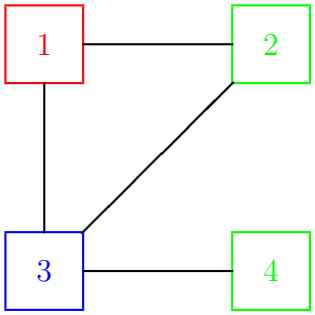

Figure 2: The graph coloring corresponding to the solution (1) of a system of polynomial equations.

3. Solve the problem for the corresponding Gröbner bases (which, typically, is simpler than for the original sets). (For instance, find all solutions of the Gröbner basis.)

4. Translate the solutions back to the original sets. (In the case of finding solutions, the solutions of the Gröbner basis are the same as the solutions of the original system.)

5. Interpret the results in the language of the original problem (e.g. translate "roots of unity" into "colors").

References

[1] M. Abramowitz and I. Stegun, editors. *Handbook of Mathematical Functions*. United States Government Printing Office, 1964. Reprinted by Dover, 1965.

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

[3] B. Buchberger. *An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal*. PhD thesis, University Innsbruck, Mathematical Institute, 1965. German, English translation in: J. of Symbolic Computation, Special Issue on Logic, Mathematics, and Computer Science: Interactions. Volume 41, Number 3–4, Pages 475–511, 2006.

[4] B. Buchberger. An Algorithmical Criterion for the Solvability of Algebraic Systems of Equations. *Aequationes mathematicae*, 4(3):374–383, 1970. German. English translation in: B. Buchberger, F. Winkler (eds.), Groebner Bases and Applications, London Mathematical Society Lecture Note Series, Vol. 251, Cambridge University Press, 1998, pp. 535–545.

[5] B. Buchberger. Introduction to Groebner Bases. In B. Buchberger and F. Winkler, editors, *Groebner Bases and Applications*, number 251 in London Mathematical Society Lecture Notes Series, pages 3–31. Cambridge University Press, 1998.

[6] Bruno Buchberger and Franz Winkler, editors. *Gröbner Bases and Applications. Proc. of the International Conference "33 Years of Groebner Bases"*, volume 251 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1998. 560 pages.

[7] J. Chifman and S. Petrovic. Toric Ideals of Phylogenetic Invariants for the General Group-based Model on Claw Trees K1,n. In H. Anai, K. Horimoto, and T. Kutsia, editors, *Algebraic Biology, Proc. of the Second International Conference on Algebraic Biology*, volume 4545 of *Lecture Notes in Computer Science*, pages 307–321, RISC, Hagenberg, Austria, July 2007. Springer.

[8] F. Chyzak, I. Gutman, and P. Paule. Predicting the Number of Hexagonal Systems with 24 and 25 Hexagons. *MATCH*, 40:139–151, 1999.

[9] F. Chyzak, P. Paule, O. Scherzer, A. Schoisswohl, and B. Zimmermann. The Construction of Orthonormal Wavelets using Symbolic Methods and a Matrix Analytical Approach for Wavelets on the Interval. *Experiment. Math.*, 10:67–86, 2001.

[10] Hans J. Eysenck. *Check Your Own I.Q.* Rowohlt, 1966.

[11] J.C. Faugere and A. Joux. Algebraic Cryptoanalysis of Hidden Field Equation (HFE) Cryptosystems Using Groebner Bases. In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 44–60, 2003.

[12] S. Gerhold and M. Kauers. A Procedure for Proving Special Function Inequalities Involving a Discrete Parameter. In *Proceedings of ISSAC'05*, pages 156–162. ACM Press, 2005.

[13] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, 2nd edition edition, 1994.

[14] D. Heldt, M. Kreuzer, S. Pokutta, and H. Poulisse. Algebraische Modellierung mit Methoden der approximativen Computeralgebra und Anwendungen in der Ölindustrie. *OR-News*, 28, 2006. issn 1437-2045.

[15] T. Ida, D. Tepeneu, B. Buchberger, and J. Robu. Proving and Constraint Solving in Computational Origami. In B. Buchberger and John Campbell, editors, *Proceedings of AISC 2004 (7 th International Conference on Artificial Intelligence and Symbolic Computation)*, volume 3249 of *Springer Lecture Notes in Artificial Intelligence*, pages 132–142. Copyright: Springer-Berlin, 22-24 September 2004.

[16] Manuel Kauers. Computer Algebra for Special Function Inequalities. In Tewodros Amdeberhan and Victor Moll, editors, *Tapas in Experimental Mathematics*, volume 457 of *Contemporary Mathematics*, pages 215–235. AMS, 2008.

[17] Manuel Kauers and Peter Paule. A Computer Proof of Moll's Log-Concavity Conjecture. *Proceedings of the AMS*, 135(12):3847–3856, December 2007.

[18] M. Kreuzer and L. Robbiano. *Computational Commutative Algebra I*. Springer New York–Heidelberg, 2000.

[19] Christian Mallinger. Algorithmic Manipulations and Transformations of Univariate Holonomic Functions and Sequences. Master's thesis, RISC-Linz, August 1996.

[20] P. Paule and V. Pillwein. Automatic Improvements of Wallis' Inequality. Technical Report 08–18, RISC Report Series, University of Linz, Austria, 2008.

[21] P. Paule and M. Schorn. A Mathematica version of Zeilberger's Algorithm for Proving Binomial Coefficient Identities. *J. Symbolic Comput.*, 20(5-6):673–698, 1995.

[22] V. Pillwein. Positivity of Certain Sums over Jacobi Kernel Polynomials. *Advances Appl. Math.*, 41:365–377, 2007.

[23] M. Rosenkranz, B. Buchberger, and H. W. Engl. Solving Linear Boundary Value Problems Via Non-commutative Groebner Bases. *Applicable Analysis*, 82(7):655–675, July 2003.

[24] B. Salvy and P. Zimmermann. Gfun: A Package for the Manipulation of Generating and Holonomic Functions in One Variable. *ACM Trans. Math. Software*, 20:163–177, 1994.

[25] N.J.A. Sloane. *A Handbook of Integer Sequences.* Academic Press, 1973.

[26] N.J.A. Sloane. *The New Book of Integer Sequences.* Springer, 1994.

[27] Markus Voege, Anthony J. Guttmann, and Iwan Jensen. On the Number of Benzenoid Hydrocarbons. *Journal of Chemical Information and Computer Sciences*, 42(3):456–466, 2002.

[28] D. Zeilberger. A Fast Algorithm for Proving Terminating Hypergeometric Identities. *Discrete Math.*, 80:207–211, 1990.

[29] D. Zeilberger. A Holonomic Systems Approach to Special Function Identitites. *J. Comput. Appl. Math.*, 32:321–368, 1990.