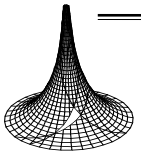

10/1 (2012), 57–76
tmcs@math.klte.hu
http://tmcs.math.klte.hu

Teaching
Mathematics and
Computer Science

Teaching Gröbner bases

György Maróti

Abstract. In this article we offer a demonstration of how the StudentGroebner package, a didactic oriented Maple package for Gröbner basis theory, could assist the teaching/learning process. Our approach is practical. Instead of expounding on deep didactic theory we simply give examples on how we imagine experimental learning in classroom. The educational goal is to prepare the introduction of two sophisticated algorithms, the division algorithm and Buchberger's algorithm, by gathering preliminary knowledge about them.

Key words and phrases: Gröbner bases, Maple.

ZDM Subject Classification: U65.

> "Why is Gröbner Bases Theory Attractive?
> The main problem solved by the theory can be explained is five minutes. The algorithm that solves the problem can be learned in fifteen minutes the theorem on which the algorithm is based is nontrivial to (invent and to) prove, many problems in seemingly quite different areas of mathematics can be reduced to the problem of computing Gröbner bases."
>
> Bruno Buchberger

Introduction

Most computer algebra systems like Maple, Mathematica, CoCoA and others offer tools to work with multivariate polynomials and polynomial ideals (see [10], [11] and [12]). Comprehensive list of different Gröbner Basis Implementations can be found in [13]. The main effort behind these tools is to be as general and

effective as possible. There is only one point lacking and this is the point of view of didactics. For novice students, who meet first time in their life with multivariate polynomials, to see how division algorithm works or simply to calculate an S-polynomial or a Gröbner basis of an ideal could become easily undoable. Procedures do not help them in understanding. Students specify the input and get an output without any clarification.

The StudentGroebner package (see [14]) addresses this kind of problems. It offers a teaching/learning tool, which helps students to understand the work of different algorithms by revealing the main steps of calculation, by showing the loop invariant of cycles, by displaying the calling sequences of procedures up to a user defined level and last but not least by clarifying the input and output data.

This article is devoted to show the didactic relevance of the StudentGroebner package. The first section presents the main ideas behind the package without the demand for completeness. Next two sections demonstrate the process of experimental learning with the StudentGroebner package in the classroom. The method is to carry out different experiments with procedures which implement the division algorithm and Buchberger's algorithm (see [1], [6] and [7]). These procedures are considered as black boxes during the process of experiments. Students call them with different parameters and options, observe the output and try to formulate facts, which are at this stage not more than conjectures. The laboratory work should end with the exact description of the algorithms and the proof of facts to be conjectured. The last section illustrates the connection between Euclidean Algorithm and Buchberger's algorithm.

It has to be emphasized that each section below consists of one independent sample whose aim is to present one possible usage of the StudentGroebner package. The main point is to show its didactic usefulness. It is always assumed that the reader/students own all the prerequisites required for the understanding the specific section. As for the notion and theorems of polynomial ideal theory the reader is referred to [2], [4] and [5]. Detailed description of Maple can be found in [8] and [9].

How to use the StudentGroebner package

The StudentGroebner package is an educational aid which on the one hand promotes teachers and educators to make abstract mathematical objects and complicated relations perceptible and on the other hand provides students to uncover the inner behavior of sophisticated algorithmsin the field of Gröbner bases theory.

The package was developed by the author in 2010 summer and autumn and freely available to any Maple users, students and lecturers. It consists of 38 procedures, all together 1800 lines of concise Maple code. Besides some service functions new types like 'monomial', 'term' and others are introduced by extending the functionality of Maple's procedure 'type'. Four procedures are devoted to univariate polynomials, while the rest of the package handle the monomial orders, the conversions between the different inner representations and offer functionalities from determining LT and LM of a polynomial through Dickson' lemma to S-polynomials and Buchbergher's algorithm. In this section we give some introductory examples which demonstrate how these concepts have been realized.

Monomial orders play a central role in the theory of Gröbner bases. Most theorems and arguments in textbooks begin with the phrase "fix a monomial order" (whatever it is). To implement this phenomenon the introduction of default monomial order seemed to be appropriate. Procedure **SetP** is devoted to manage the default monomial order. Without parameter **SetP** returns its current value.
$> SetP()$

$$[\,plex, [x, y, z, t, u, v]\,] \tag{2.1}$$

This output specifies the type of monomial order ($plex$), and the descending list of variables: $x > y > z > t > u > v$. This means the symbols x, y, z, t, u, v are considered to be variables, while any other symbols appearing in the polynomials are parameters. Of course the user is allowed to define a different default monomial order. The next two commands are self-explaining.
$> SetP(tdeg)$

$$[\,tdeg, [x, y, z, t, u, v]\,] \tag{2.2}$$

$> SetP([a, b])$

$$[\,plex, [b, c]\,] \tag{2.3}$$

SetP allows to use the same syntax for short monomial orders like Maple. Notice that indexed variables are allowed and the number of variables is not limited.
$> SetP(grlex(t_1, t_2, t_3, u_1, u_2, v, z, w))$

$$[\,grlex, [t_1, t_2, t_3, u_1, u_2, v, z, w]\,] \tag{2.4}$$

Let us return to the default monomial order, which can be done by the first command below. It is a pleasant feature that **SetP** accepts a positive integer in the range 1..6. By giving this integer we can simply specify how many variables are intended to be used in the foregoing computation.

> $SetP(default)$

$$[\,plex, [x, y, z, t, u, v]\,] \tag{2.5}$$

> $SetP(3)$

$$[\,plex, [x, y, z]\,] \tag{2.6}$$

The procedure **OrderP** returns the object of type short monomial order as defined in Maple. **OrderP** can be used for example in the **sort** command and/or in any call of procedures of the Groebner package where the monomial order is a compulsory parameter.

> $OrderP()$

$$plex(x, y, z) \tag{2.7}$$

> $sort(z^4 - 2 + 3 \cdot x^2 \cdot y^5,\ order{=}OrderP())$

$$3x^2y^5 + z^4 - 2 \tag{2.8}$$

> $Groebner{:}\text{-}LeadingMonomial(f, OrderP())$

$$x^2y^5 \tag{2.9}$$

There are several options which can be given as arguments in the call of any procedure of the StudentGroebner package. In the following examples we illustrate their use by applying them to the procedure LT which computes the leading term of a polynomial. Let us define a polynomial f and then call the procedure **LT**.

> $f{:=}z^4 - 2 + 3 \cdot x^3 \cdot y^5$

$$f := z^4 - 2 + 3x^3y^5 \tag{2.10}$$

> $LT(f)$

$$3x^3y^5 \tag{2.11}$$

When we are confronted with this output for the first time it may happen that we are not able to interpret it correctly. The option **input** helps in understanding.

> $LT(f,\ input)$

`.Input(LT)`:

$$poly = z^4 - 2 + 3x^3y^5$$

`.Output(LT)`:

$$leading\ term = 3x^3y^5$$

`.Result(LT)`:

$$3x^3y^5 \tag{2.12}$$

Let's see in more detail what happens behind the scenes. In the next command we use the option **steps**.

$> LT(f,\ steps)$

.Input(LT):

$$poly = z^4 - 2 + 3x^3y^5$$

.Applied monomial order(LT):

$$[\,plex, [x, y, z]\,]$$

.List of terms(LT):

$$[\,z^4, -2, 3x^3y^5\,]$$

.List of termvectors(LT):

$$[\,[\,1, [\,0, 0, 4\,]\,], [\,-2, [\,0, 0, 0\,]\,], [3, [\,3, 5, 0\,]\,]\,]$$

.Maximal termvector(LT):

$$[\,3, [\,3, 5, 0\,]\,]$$

.Output(LT):

$$leading\ term = 3x^3y^5$$

.Result(LT):

$$3x^3y^5 \tag{2.13}$$

In return for the option **steps**, we learn that the applied monomial order is $plex(x, y, z)$. Next the procedure **LT** forms the list of terms of the polynomial f, converts each term to a termvector. There is one-to-one correspondent between terms and termvectors, which are two element lists consisting of the coefficient and the list of exponents of variables. Next **LT** computes the maximal termvector according to the applied monomial order. In the end the procedure converts the maximal termvector back to a term.

Notice that we have not specified the monomial orders to be applied in the previous command. Therefore the current value of default monomial order has been chosen. We are allowed, however, to prescribe arbitrary monomial order to be applied.

$> LT(f,\ input,\ tdeg(z, x))$

.Input(LT):

$$poly = z^4 - 2 + 3x^3y^5$$

.Output(LT):

$$leading\ term = z^4$$

.Result(LT):

$$z^4 \tag{2.14}$$

The option **calculation** offers another way to make the steps of calculation process clear. In return for this option a matrix - which will be called calculation table henceforth - is displayed, which contains results of milestones of the computational process. Of course the user has to learn how to read calculation tables, but this is not a very strong requirement as the content of calculation tables is easy to interpret. For example in our case the terms of the input polynomial are listed in the first row. Next a term-to-termvector conversion has been performed. In the end the algorithm determined the maximal element of termvectors and converted the maximal termvector back to a term.

$> LT(f,\ calc,\ tdeg(z,x),\ inp)$

.Input(LT):

$$poly = z^4 - 2 + 3x^3y^5$$

.Applied monomial order(LT):

$$[\,[tdeg, [\,[z, x\,]\,]\,]\,]$$

.Calculation table(LT):

$$\begin{bmatrix} \textit{Terms:} & & z^4 & -2 & 3x^3y^5 \\ \textit{Termvectors:} & & [1,[4,0]] & [-2,[0,0]] & [3y^5,[0,3]] \\ \textit{Maxtermvector}(0..\mathrm{i}) & [-\infty,[0,0]] & [1,[4,0]] & [1,[4,0]] & [1,[4,0]] \\ \textit{Maxterm:} & & & & z^4 \end{bmatrix}$$

.Output(LT):

$$leading\ term = z^4$$

.Result(LT):

$$z^4 \tag{2.15}$$

Notice that we are allowed to use as many options as we want to and the order of options is irrelevant. On the other hand the system checks only the first three characters of the options. Therefore we could write **calc** instead of **calculation** and **inp** instead of **input**.

The scope of every options is the procedure whose argument it is listed in. This means that the option is not used for those procedures which are called within the body of the main procedure. This can be overwritten by using the option **level=n**, where n is a positive integer. In the next command the option

steps is applied to the procedure LM and to any procedures called from within LM. Be aware of using this option as higher levels could result in tremendous output. All the same this is an impressive feature which clearly shows the calling tree and the computations at different levels.

$> LM(f, ste, level{=}2)$

.`Input(LM)`:

$$poly = z^4 - 2 + 3x^3y^5$$

.`Applied monomial order(LM)`:

$$[\,plex, [x, y\,]\,]$$

..`Input(LT)`:

$$poly = z^4 - 2 + 3x^3y^5$$

..`Applied monomial order(LT)`:

$$[\,plex, [x, y\,]\,]$$

..`List of terms(LT)`:

$$[\,z^4, -2, 3x^3y^5\,]$$

..`List of termvectors(LT)`:

$$[\,[\,z^4, [0, 0\,]\,], [\,-2, [0, 0\,]\,], [3, [3, 5\,]\,]\,]$$

..`Maximal termvector(LT)`:

$$\textit{[leading coefficient, leading monomial]} = [\,3, [\,3, 5\,]\,]$$

.`Output(LM)`:

$$\textit{leading monomial} = x^3y^5$$

.`Result(LM)`:

$$x^3y^5 \tag{2.16}$$

The procedures of StudentGroebner package are less effective than that of Groebner package of Maple. This is because StudentGroebner's procedures spend a lot of time and use a lot of extra memory for administrative works. Therefore, if someone is interested in the results only the fastest way is the use of Maple's Groebner package. But this can be achieved by the option **groebner** as well. In this way the StudentGroebner package offers another interface to access the procedures of Maple's Groebner package.

$> LT(z^2 - y^2,\ groebner,\ ste,\ [y, z])$
.Input(LT):

$$poly = z^2 - y^2$$

.Applied monomial order(LT):

$$[\,plex, [y, z\,]\,]$$

.Output by Groebner:-LeadingTerm(LT):

$$-y^2$$

.Result(LT):

$$-y^2 \tag{2.17}$$

Experimental learning 1: understanding the division algorithm

The division algorithm formultivariate polynomials is one of the key ingredients of Gröbner bases theory. Its detailed exposition can be found in all surveys and text books (see [2], [4] and [5]). In this section we present one possible preparation of the introduction of the division algorithm for multivariate polynomials. What we do is to show a sequence of experiments. Every experiment is an interaction between the teacher and the students. The teacher performs a few commands and the students are asked to interpret what they see on the screen and try to formulate statements. Student's remarks can of course be either partly or totally wrong or correct. It is the teachers job to prepare sufficiently many different procedure calls for guiding the students to recognize the right facts.

It has to be emphasized what we do is just the preparation process which precedes and by no means replaces the exact discussion. We use procedure ***DivP*** as black box and call it by a dividend and divisors. In each experiment we set forth the motivation and the phenomena to be observed. By the end of such common work students should have a strong impression how the division algorithm works, which in turn should make the abstract discussion easier and much more efficient.

Experiment 1

Motivation

The procedure **DivP** works on polynomials in an arbitrary number of variables. We expect it to work in the same or at least in a similar way as the division

of univariate polynomials.

$> SetP(3)$

$$[\,plex, [x, y, z]\,] \tag{3.1}$$

$> f := (x + y) \cdot (x^2 \cdot y + y^3)$

$$f := (x + y)(x^2y + y^3) \tag{3.2}$$

$> g := x + y$

$$g := x + y \tag{3.3}$$

$> DivP(f,\ g,\ inp,\ cal)$

.`Input(DivP)`:

$$dividend: f = (x + y)(x^2y + y^3)$$

$$divisor: G = [\,x + y\,]$$

.`Applied monomial order(DivP)`:

$$[\,[\,plex, [\,[x, y, z]\,]\,]\,]$$

.`Calculation table(DivP)`:

$$\begin{bmatrix} & & & & G_1 = x + y & \\ p & r & q_1 & LT(p) & LT(G_1) & Aux \\ x^3y + x^2y^2 + xy^3 + y^4 & 0 & 0 & x^3y & x & x^2y(x + y) = x^3y + x^2y^2 \\ xy^3 + y^4 & 0 & x^2y & xy^3 & x & y^3(x + y) = xy^3 + y^4 \\ 0 & 0 & x^2y + y^3 & & & \end{bmatrix}$$

.`Output(DivP)`:

$$[\,remainder = 0, quotient = [\,x^2y + y^3\,]\,]$$

.`Result(DivP)`:

$$0, [\,x^2y + y^3\,] \tag{3.4}$$

Facts to be explored by students

1. The two steps if the calculation table of ***DivP*** are the same as that in the univariate case.
2. In the last step the current dividend equals 0.
3. The remainder equals zero in every row (what does it mean?).

Experiment 2

Motivation

What makes it possible to perform all the steps in the computation above? The leading term of the divisor, $LT(G_1)$ divides the leading term of the current dividend $LT(p)$ in every row of the calculation table. As it is unlikely that this always happens, we have to check other dividend and divisors.

$> f := (x^2 + x) \cdot (x^2 \cdot y)$

$$f := (x^2 + x)x^2y \tag{3.5}$$

$> g := x \cdot y - 1$

$$g := xy - 1 \tag{3.6}$$

$> DivP(f, g, inp, cal)$

.`Input(DivP)`:

$$dividend: f = (x^2 + x)x^2y$$

$$divisor: G = [\, xy - 1 \,]$$

.`Applied monomial order(DivP)`:

$$[\,[\, plex, [\,[\, x, y, z \,]\,]\,]\,]$$

.`Calculation table(DivP)`:

$$\left[\begin{array}{cccccc} & & & & G_1 = xy - 1 & \\ p & r & q_1 & LT(p) & LT(G_1) & Aux \\ x^4y + x^3y & 0 & 0 & x^4y & xy & x^3(xy-1) = x^4y - x^3 \\ x^3y + x^3 & 0 & x^3 & x^3y & xy & x^2(xy-1) = x^3y - x^2 \\ x^3 + x^2 & 0 & x^3 + x^2 & x^3 & & \\ x^2 & x^3 & x^3 + x^2 & x^2 & & \\ 0 & x^3 + x^2 & x^3 + x^2 & & & \end{array}\right]$$

.`Output(DivP)`:

$$[\, remainder = x^3 + x^2, quotient = [\, x^3 + x^2 \,]\,]$$

.`Result(DivP)`:

$$x^3 + x^2, [\, x^3 + x^2 \,] \tag{3.7}$$

Facts to be explored by students

1. When the leading term of the current dividend is not divisible by the leading term of the divisor, we add this leading term the remainder, while subtract it from the dividend.

2. Next we continue the calculation in the same way.

Experiment 3

Motivation

It is well known that in case of univariate polynomials every ideal is principal, i.e. can be generated by one polynomial. On the other hand the division algorithm is capable of solving the ideal membership problem for univariate ideals as

$$f \in \langle g \rangle \text{ if and only if } \mathit{Rem}(f, g) = 0.$$

On the other hand in case of multivariate polynomials there are ideals which are not principal. Hence, if we expect the division algorithm to solve the ideal membership problem we have to allow several divisors. This is extremely strange at first glance. Of course, if we have several divisors, we must have the same number of quotients.

$> f := x \cdot y^2 + x^2 \cdot y + y^2$

$$f := x^2y + xy^2 + y^2 \tag{3.8}$$

$> g_1 := y^2 - 1$

$$g_1 := y^2 - 1 \tag{3.9}$$

$> g := x \cdot y - 1$

$$g_2 := xy - 1 \tag{3.10}$$

$> \mathit{DivP}(f, [g_1, g_2], \mathit{inp}, \mathit{cal}, \mathit{invariant})$

`.Input(DivP)`:

$$\mathit{dividend:}\, f = x^2y + xy^2 + y^2$$

$$\mathit{divisor:}\, G = [\, y^2 - 1, xy - 1 \,]$$

`.Applied monomial order(DivP)`:

$$[\,[\mathit{plex}, [\,[x, y, z\,]\,]\,]\,]$$

`.Calculation table(DivP)`:

					$G_1 = y^2 - 1$	$G_2 = xy - 1$		
p	r	q_1	q_2	$LT(p)$	$LT(G_1)$	$LT(G_2)$	Aux	$p + G_1q_1 + G_2q_2 + r$
$x^2y + xy^2 + y^2$	0	0	0	x^2y		xy	$x(xy-1) = x^2y - x$	$x^2y + xy^2 + y^2$
$xy^2 + x + y^2$	0	0	x	xy^2	y^2		$x(y^2-1) = xy^2 - x$	$x^2y + xy^2 + y^2$
$2x + y^2$	0	x	x	$2x$				$x^2y + xy^2 + y^2$
y^2	$2x$	x	x	y^2	y^2		$y^2 - 1 = y^2 - 1$	$x^2y + xy^2 + y^2$
1	$2x$	$x+1$	x	1				$x^2y + xy^2 + y^2$
0	$2x+1$	$x+1$	x					$x^2y + xy^2 + y^2$

`.Output(DivP)`:

$$[\, remainder = 2x + 1, quotient = [\, x + 1, x \,] \,]$$

`.Result(DivP)`:

$$2x + 1, [\, x + 1, x \,] \tag{3.11}$$

Facts to be explored by students

1. **DivP** chooses the first of divisor from left to right, whose leading term divides the leading term of the current dividend. In other words the algorithm chooses one possible divisor and perform the same steps as before.
2. The algorithm results in the following equality (see loop invariant in the last column of calculation table

$$f = g_1 \cdot q_1 + \ldots + g_s \cdot q_s + r$$

3. For the remainder we either have $r = 0$ or no term of r is divisible by the leading term of any divisor.

At this point the our lecture should be continued by an exact mathematical discussion of the division algorithm, which transforms the procedure **DivP** to a white box. This is, however, beyond the scope of this lecture.

Experimental learning 2: understanding Buchberger's algorithm

The original source for the algorithm computing a Gröbner basis of polynomial ideal published by Bruno Buchberger can be found in [3]. As for the detailed discussion of Buchberger's algorithm including the proof of correctness, the immediate consequences and its typical usage the reader is referred to [2], [4] and [5].In this section we present two experiments for discovering Buchberger's algorithm. Our method is the same as in the previous section.

Let us set the default monomial order and create a list of polynomials which is considered to be a basis of an ideal.

$> SetP(tdeg(x, y, z))$

$$[\, tdeg, [x, y, z] \,] \tag{4.1}$$

$> F := [\, x - z^4, y - x \cdot z^5 \,]$

$$F := [\, x - z^4, y - xz^5 \,] \tag{4.2}$$

Experiment 1

Motivation

Buchberger's algorithm generates Gröbner basis from a given basis of an ideal. Let us see how it works and collect our first experiences.

> *BuchbergerP*(F, *bas*)

.`Input(BuchbergerP)`:

$$G = [\, -z^4 + x, -xz^5 + y \,]$$

.`1.New basis(BuchbergerP)`:

$$G = [\, -z^4 + x, -xz^5 + y, -x^2z + y \,]$$

.`2.New basis(BuchbergerP)`:

$$G = [\, -z^4 + x, -xz^5 + y, -x^2z + y, yz^3 - x^3 \,]$$

.`3.New basis(BuchbergerP)`:

$$G = [\, -z^4 + x, -xz^5 + y, -x^2z + y, yz^3 - x^3, x^5 - y^2z^2 \,]$$

.`Result(BuchbergerP)`:

$$[\, -z^4 + x, -xz^5 + y, -x^2z + y, yz^3 - x^3, x^5 - y^2z^2 \,] \tag{4.3}$$

Facts to be explored by students

1. The algorithm extends the basis step by step. More specifically one new polynomial is added to the current basis.
2. Explain that the ideal generated by the polynomials listed in step 2 contains the ideal generated by the polynomials in step 1. Is the similar statement true for step 3 and step 2?

Experiment 2

Motivation

On the evidence of the first experiment we do not know what kind of polynomials are added to the current base. It would be desirable to have more detailed information.

> *BuchbergerP*(F, *cal*)

.`Calculation table:(BuchbergerP)`:

G	*Indexes*	*Index*	*Spoly*	*Rem*(*Spoly*, G)	*Action*
$[-z^4+x, -xz^5+y]$	$[[1,2]]$	$[1,2]$	$-x^2z+y$	$-x^2z+y$	*added*
$[-z^4+x, -xz^5+y, -x^2z+y]$	$[[1,3],[2,3]]$	$[1,3]$	yz^3-x^3	yz^3-x^3	*added*
$[-z^4+x, -xz^5+y, -x^2z+y, yz^3-x^3]$	$[[2,3],[1,4],[2,4],[3,4]]$	$[2,3]$	yz^4-xy	0	*discarded*
	$[[1,4],[2,4],[3,4]]$	$[1,4]$	x^3z-xy	0	*discarded*
	$[[2,4],[3,4]]$	$[2,4]$	$x^4z^2-y^2$	0	*discarded*
	$[[3,4]]$	$[3,4]$	$x^5-y^2z^2$	$x^5-y^2z^2$	*added*
$[-z^4+x, -xz^5+y, -x^2z+y, yz^3-x^3, x^5-y^2z^2]$	$[[1,5],[2,5],[3,5],[4,5]]$	$[1,5]$	$y^2z^6-x^6$	0	*discarded*
	$[[2,5],[3,5],[4,5]]$	$[2,5]$	$y^2z^7-x^4y$	0	*discarded*
	$[[3,5],[4,5]]$	$[3,5]$	$y^2z^3-x^3y$	0	*discarded*
	$[[4,5]]$	$[4,5]$	$-x^8+y^3z^5$	0	*discarded*

.`Result(BuchbergerP)`:

$$[-z^4+x, -xz^5+y, -x^2z+y, yz^3-x^3, x^5-y^2z^2] \tag{4.4}$$

Facts to be explored by students

1. The current bases is extended by the S-polynomial of two of its elements. Is this sufficient to state that the ideals generated by the polynomials in different steps of the algorithm are the same?
2. *Spoly*(f,g) is added to the basis if and only if *Rem*(*Spoly*(f,g), G)=0.
3. What is the condition which holds for the last basis and does not hold for the first three ones? The answer for this question is very important as this is the necessary and sufficient condition for a basis to be of Gröbner type.
4. What is the role of the row named *Indexes*? Explain that in each row the list of indexes contains pair $[i,j]$ if and only if the ith and jth polynomial of the current bases has not been checked yet. Checking means testing whether *Rem*(*Spoly*(f,g), G)=0.

Explore Buchberger's algorithm

Every ideal of the ring of univariate polynomials is principal, i.e. it can be generated by one polynomial. If the ideal in concern is generated by the two polynomials f and g then the one element basis can be found by forming the greatest common divisor of f and g. This computation can be performed by the Euclidean Algorithm (EA). On the other hand one element bases are necessarily of Gröbner type and reduced, provided the generating polynomial is monic. This yields that Buchberger's algorithm with option **reduced** must give the same result as EA if we apply to two univariate polynomials. Let us see an example.

$> f :=expand((x-3)\cdot(x^4+5\cdot x-1))$

$$f := x^5 - 3x^4 + 5x^2 - 16x + 3 \tag{5.1}$$

$> g :=expand((x-3)\cdot(x^2+4))$

$$g := x^3 - 3x^2 + 4x - 12 \tag{5.2}$$

$> EA(f,g,\ cal,\ monic)$
`.Calculation table(EA)`:

$$\begin{bmatrix} Dividend & Divisor & Remainder \\ x^5-3x^4+5x^2-16x+3 & x^3-3x^2+4x-12 & 5x^2-45 \\ x^3-3x^2+4x-12 & 5x^2-45 & 13x-39 \\ 5x^2-45 & 13x-39 & 0 \end{bmatrix}$$

`.Result(EA)`:

$$x - 3 \tag{5.3}$$

$> BuchbergerP([f,g],\ reduced)$

$$[\,x-3\,] \tag{5.4}$$

The didactic challenge here is how can we illustrate this phenomenon.? In other words how can we explore and clarify this close connection between EA and Buchberger's algorithm?

At first let us investigate S-polynomials of univariate polynomials. For the sake of simplicity suppose that

$$deg(f) \geq deg(g) \text{ and } LC(f) = 1.$$

In this case

$$LCM(LM(f), LM(g)) = x^{max(deg(f),deg(g))} = LM(f) = LT(f),$$

which yields

$$Spoly(f,g) =$$

$$\frac{LCM(LM(f),LM(g))}{LT(f)}\cdot f - \frac{LCM(LM(f),LM(g))}{LT(g)}\cdot g = \frac{LT(f)}{LT(f)}\cdot f - \frac{LT(f)}{LT(g)}\cdot g = f - \frac{LT(f)}{LT(g)}\cdot g.$$

This equality tells us that the S-polynomial of univariate polynomials is nothing else than the result of the iteration step in the division algorithm.

$> SetP(1)$

$$[\,plex, [\,x\,]\,] \tag{5.5}$$

> $SpolyP(f, g)$

$$-4x^3 + 17x^2 - 16x + 3 \tag{5.6}$$

> $DivuniP(f, g,\ cal)$:
.Calculation table(DivuniP):

$$\begin{bmatrix} q = quotient & r = remainder & deg(r) >= deg(g) & [LT(r), LT(g)] & LT(r)/LT(g) \\ 0 & x^5 - 3x^4 + 5x^2 - 16x + 3 & true & [x^5, x^3] & x^2 \\ x^2 & -4x^3 + 17x^2 - 16x + 3 & true & [-4x^3, x^3] & -4 \\ x^2 - 4 & 5x^2 - 45 & false & & \end{bmatrix}$$

.Result(DivuniP):

In particular when the division procedure consists of one step, we have

$$Spoly(f, g) = Rem(f, g).$$

> $f :=expand((x - 3) \cdot (x^3 + 5 \cdot x - 1))$

$$f := x^4 - 3x^3 + 5x^2 - 16x + 3 \tag{5.7}$$

> $SpolyP(f, g)$=$DivuniP(f, g$,$rem)$

$$x^2 - 4x + 3 = x^2 - 4x + 3 \tag{5.8}$$

Now let us perform in parallel the Euclidean algorithm and Buchberger's algorithm for the same univariate dividend and divisor.

Euclidean algorithm

This column shows the calculation table of EA. The first division of this EA consists of one step, while calculation table of the second division is shown in 5.10 below.

> $EA(f,\ g,\ inp,\ cal)$
.Input(EA):

$$f = x^4 - 3x^3 + 5x^2 - 16x + 3$$

$$g = x^3 - 3x^2 + 4x - 12$$

.Calculation table(EA):

$$\begin{bmatrix} Dividend & Divisor & Remainder \\ x^4 - 3x^3 + 5x^2 - 16x + 3 & x^3 - 3x^2 + 4x - 12 & x^2 - 4x + 3 \\ x^3 - 3x^2 + 4x - 12 & x^2 - 4x + 3 & 5x - 15 \\ x^2 - 4x + 3 & 5x - 15 & 0 \end{bmatrix}$$

.Output(EA):

$$gcd(f, g) = 5x - 15$$

.Result(EA):

$$5x - 15 \tag{5.9}$$

Next command shows the details of the second division of the Euclidean Algorithm above.

> $DivuniP(x^3 - 3x^2 + 4x - 12,\ x^2 - 4x + 3,\ rem,\ cal,\ inp)$

.Input(DivuniP):

$$dividend:\ f = x^3 - 3x^2 + 4x - 12$$

$$divisor:\ g = x^2 - 4x + 3$$

.Calculation table(DivuniP):

$$\begin{bmatrix} q = quotient & r = remainder & deg(r) >= deg(g) & [LT(r), LT(g)] & LT(r)/LT(g) \\ 0 & x^3 - 3x^2 + 4x - 12 & true & [x^3, x^2] & x \\ x & x^2 + x - 12 & true & [x^2, x^2] & 1 \\ x + 1 & 5x - 15 & false & & \end{bmatrix}$$

.Output(DivuniP):

$$quotient = x + 1$$

$$remainder = 5x - 15$$

.Result(DivuniP):

$$5x - 15 \tag{5.10}$$

Buchberger's algorithm

Step 0. The initial basis is $G = [f, g]$.

> $G := [f, g]$

$$G := [x^4 - 3x^3 + 5x^2 - 16x + 3, x^3 - 3x^2 + 4x - 12] \tag{5.11}$$

G contains the divisor of the FIRST iteration of EA.

Step 1. Condition $Rem(Spoly(f, g), G)$=0 is to be checked now.

> $SpolyP(f, g)$

$$x^2 - 4x + 3 \tag{5.12}$$

As the condition is false S-polynomial which equals $Rem(f, g)$ is added to the basis.

$> G := [\, f, g, x^2 - 4x + 3 \,]$

$$G := [\, x^4 - 3x^3 + 5x^2 - 16x + 3, x^3 - 3x^2 + 4x - 12, x^2 - 4x + 3 \,] \tag{5.13}$$

G contains the divisor of FIRST TWO iteration of EA.

Step 2. Continue the extension of the basis. Let us check the S-polynomial of g and the newly added basis element.
$> \mathit{SpolyP}(x^3 - 3x^2 + 4x - 12, x^2 - 4x + 3)$

$$x^2 + x - 12 \tag{5.14}$$

Because the remainder differs from zero this S-polynomial is also added to G.
$> G := [\, f, g, x^2 - 4x + 3, x^2 + x - 12 \,]$

$$G := [\, x^5 - 3x^4 + 5x^2 - 16x + 3, x^3 - 3x^2 + 4x - 12, x^2 - 4x + 3, x^2 + x - 12 \,] \tag{5.15}$$

Note that $x^2 + x - 12$ is not the remainder, just the result of the first iteration of the division algorithm (see 'remainder' column in 4.20). On the same time the base G still contains the divisor, so we can choose it again in the next extension step.
$> \mathit{SpolyP}(x^2 + x - 12, x^2 - 4x + 3)$

$$5x - 15 \tag{5.16}$$

We obtained the result of the division algorithm's second iteration (see 4.20 below), which equals the last divisor of EA. As this S-polynomial must be added to G as well, we have
$> G := [\, f, g, x^2 - 4x + 3, x^2 + x - 12, 5x - 15 \,]$

$$G := [\, x^4 - 3x^3 + 5x^2 - 16x + 3, x^3 - 3x^2 + 4x - 12, x^2 - 4x + 3, x^2 + x - 12, 5x - 15 \,] \tag{5.17}$$

G contains the divisor of the FIRST THREE iterations of EA.

In this way G contains ALL divisors of the Euclidean Algorithm, among them the last one: $GCD(f, g)$. By this step we have reached a Gröbner bases of the ideal in question.
$> \mathit{IsGroebnerP}(G)$

$$\mathit{true} \tag{5.18}$$

This example clearly demonstrates that in the course of basis extensions all polynomials which appear during the series of divisions in Euclidean Algorithm

become elements of the bases generated by Buchberger's algorithm. In this way all divisors of EA, in particular the latest one, which is nothing else than the greatest common divisor, are also added to the basis.

This basis contains superfluous elements. As we know that $[\, GCD(f,g)\,]$ is the reduced Gröbner basis of the ideal $\langle f, g\rangle$ we must obtain a one element basis if we perform the reduction of G.

$>$ *reduceP*(G)

$$[\, x-3\,] \tag{5.19}$$

Conclusion

This lecture is the first attempt to illustrate the usage of the StudentGroebner package. Unlike geometrical objects polynomials are symbolic and abstract. Working with multivariate polynomials is not easy for several reasons. Polynomials of this type consist of a lot of letters, in most cases they are too long to read and perceive them at a glance. More important, making calculations with long strings by hand is exhausting and as a consequence of too many handwriting, it may be a source of different mistakes. All these yield that students need tools, which help them in performing long computations, in displaying the inner structure of sophisticated algorithms and last but not least in showing the preliminary result of complex symbolic computations. StudendGroebner package could be one possible answer to this challenge.

References

[1] B. Buchberger, Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory, in: *Multidimensional systems theory*, Reidel, 1985, 184–232.

[2] B. Buchberger, An Introduction to Gröbner Bases, in: *Groebner Bases and Applications*, (B. Buchberger, F. Winkler, eds.), London Mathematical Society Lecture Notes Series 251, Cambridge University Press, 1998, 3–31, ISBN 0-521-63298-6.

[3] B. Buchberger, Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems (An Algorithmic Criterion for the Solvability of Algebraic Systems of Equations), *Aequationes mathematicae* **4** (1970), 374–383, (English transl.: B. Buchberger, F. Winkler: Groebner Bases and Applications, Proc. of the International Conference "33 Years of Groebner Bases", 1998, RISC, Austria, London Mathematical Society Lecture Notes Series 251, Cambridge University Press, 1998, 535–545).

[4] D. A. Cox, J. B. Little and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Third Edition, Springer, 2007.

[5] R. Fröberg, *An Introduction to Gröbner Bases*, John Wiley & Sons, 1997.

[6] R. Gebauer and H. Moller, On an installation of Buchberger's algorithm., *J. Symbolic Comput.* **6** (1988), 275–286.

[7] A. Heck, Bird's-eye view of Gröbner Bases, *Nuclear Inst. and Methods in Physics Research A* **389** (1997), 16–21,
http://staff.science.uva.nl/∼heck/AIHENP96/groebnerbasis.pdf.

[8] A. Heck, *Introduction to Maple*, Third Edition, Springer, 2003.

[9] Gy. Maróti and M. Klincsik, *Maple (in Hungarian)*, Livermore, 2006.

[10] Overview of the Groebner Package,
http://www.maplesoft.com/support/help/Maple/view.aspx?path=Groebner.

[11] Gröbner Basis, http://mathworld.wolfram.com/GroebnerBasis.html.

[12] CoCoA, Computations in Commutative Algebra,
http://cocoa.dima.unige.it/flyer4.html.

[13] Gröbner Basis Implementation,
shttp://www.risc.jku.at/Groebner-Bases-Implementations/view/SystemList.php.

[14] StudentGroebner, a didactic oriented Maple package, http://maroti.epigramma.hu.

GYÖRGY MARÓTI
DEPARTMENT OF MATHEMATICS
POLLACK MIHÁLY FACULTY OF ENGINEERING
UNIVERSITY OF PÉCS
HUNGARY

E-mail: maroti@epigramma.hu

(Received March, 2011)