Using GRÖBNER as a "Black Box"

Wolfgang Windsteiger[*]
RISC-Linz

—

Research Institute for Symbolic Computation

Johannes Kepler Universität Linz

A-4040 Linz, Austria

Tel: +43 (7236) 3231-27, Fax: +43 (7236) 3231-30

email: wwindste@risc.uni-linz.ac.at

—

March 16, 1995

Abstract

This document should serve as a User's Guide to the library GRÖBNER and its accompanying I/O library GRÖBNER-IO for those users who *do not* intend to *change any part* of the library but rather want to use it as a "Gröbner–basis–calculator". Therefore, *no deep insight* into the structure of the programs, software-technological issues, and data structures is required but some *simple recipes*, how to solve certain classes of problems by Gröbner bases are desired.

1 Introduction

GRÖBNER is a library for computing Gröbner bases, based on the Computer Algebra library SACLIB and written in C language. Some of its features that distinguish it from other Gröbner bases implementations are

[*]This work was done during a two-months visit at the Department of Electronics and Electric Engineering at the University of Pretoria, South Africa, Autumn/Spring 1993.

- ◇ *public availability* of source code and
- ◇ a simple but efficient mechanism of *polymorphism* that makes the library applicable to a wide variety of combinations of coefficient domains, admissible term orderings, selection strategies for S-polynomials, etc. without the necessity of code duplication.

The availability of the source code is intended for those users who want to study the details of the Gröbner bases algorithm and need direct access to the code for experimenting. A basic knowledge about using SACLIB is a prerequisite in order to be able to modify the library appropriately, see [Buchberger *et al.*, 1993]. Furthermore, a *complete understanding* of the methods used to establish polymorphism as well as a certain amount of knowledge in Gröbner basis theory is inevitable.

The concept of *virtual domains* as the key to polymorphic design and the its motivations are described in [Windsteiger and Buchberger, 1993]. For an introduction to the Gröbner basis method we refer the reader to [Buchberger, 1985] and [Buchberger, 1987].

However, the library should also be a useful tool for those users who want to use GRÖBNER as a "Black Box Gröbner Basis Solver", without being concerned about the background in software design and algebraic matters. Still, they should be able to exploit the whole power of the library, namely its adjustability to several coefficient domains, term orderings, selection strategies, and data representations.

The present document is intended for the latter group of users, although we recommend [Windsteiger and Buchberger, 1993] also for those users, since it might give some insight into the inner workings and the limitations of applicability of GRÖBNER.

2 Possibilities of Variation

In this section, we shall give a survey on some possibilities of variations of Gröbner bases computations. The purpose of this section is to give the user an idea of which aspects have to be observed in order to get the desired result. The aspects treated in this section will have their counterparts in Section 5, where possibilities of adjusting the library according to those aspects will be discussed.

2.1 Coefficient Domain

In this section, the impact of the chosen domain of coefficients will be discussed. It should be obvious that the resulting Gröbner basis depends on the chosen domain of coefficients. We shall illustrate this fact by some examples.

Consider the set of polynomials

$$F = \{16x^2 + 4xy^2 - 4z + 1, 4x + 2y^2z + 1, 2x^2z - x - 2y^2\}$$

We fix *total degree inverse lexical* ordering w.r.t. $x > y > z$ ($x > y$ in Example 3) for monomials.

Example 1 (Rational Numbers) Interpreting the set F as a set of polynomials in the variables x, y, and z over rational numbers, i.e. $F \subset \mathbb{Q}[x, y, z]$, the Gröbner basis of F is the set

$$\begin{aligned} G_{\mathbb{Q}} \;=\; & \{x^2 - 2y^2 + 1/2z^2 - 3/4x - 1/8z, \\ & z^3 + 2y^2 - 3/2xz - 1/4z^2 + 9x + 2, \\ & y^2z + 2x + 1/2, \\ & xy^2 + 8y^2 - 2z^2 + 3x - 1/2z + 1/4, \\ & y^4 - xz^2 - 32y^2 + 1/4xz + 8z^2 - 12x + 5/2z - 9/8\} \end{aligned}$$

Example 2 (Finite Fields $\mathbb{Z}_p$) Alternatively, we can consider the set F as a set of polynomials in the variables x, y, and z over $\mathbb{Z}_{17}$, i.e. $F \subset \mathbb{Z}_{17}[x, y, z]$. The Gröbner basis of F is in this case the set

$$\begin{aligned} G_{\mathbb{Z}_{17}} \;=\; & \{x^2 + 15y^2 + 9z^2 + 12x + 2z, \\ & z^3 + 2y^2 + 7xz + 4z^2 + 9x + 2, \\ & y^2z + 2x + 9, \\ & xy^2 + 8y^2 + 15z^2 + 3x + 8z + 13, \\ & y^4 + 16xz^2 + 2y^2 + 13xz + 8z^2 + 5x + 11z + 1\} \end{aligned}$$

Example 3 (Rational Functions over $\mathbb{Q}$) We could as well interpret F as a set of polynomials in the variables x and y over rational numbers and regard z as an element of the coefficient domain (a *parameter*), i.e. $F \subset \mathbb{Q}(z)[x, y]$. In this case the Gröbner basis of F is the set

$$G_{\mathbb{Q}(z)} \;=\; \{1\}$$

Example 4 (Galois Field $GF(p^n)$) Consider now the set of polynomials

$$F = \{x + y + z + \alpha^6, x^3 + y^3 + z^3 + \alpha^8, x^5 + y^5 + z^5 + \alpha^{10}\}$$

Again we fix total degree inverse lexical ordering w.r.t. $x > y > z$ for monomials.

We interpret F as a set of polynomials in the variables x, y, and z over $GF(2^5)$, where α is a primitive element in $GF(2^5)$, i.e. $F \subset GF(2^5)[x, y, z]$. The Gröbner basis of F is the set

$$\begin{aligned} G_{GF(2^5)} &= \{x + y + z + (\alpha^3 + \alpha), \\ &\quad y^2 + yz + z^2 + (\alpha^3 + \alpha)y + (\alpha^3 + \alpha)z + (\alpha^2), \\ &\quad z^3 + (\alpha^3 + \alpha)z^2 + (\alpha^2)z + (\alpha + 1)\} \end{aligned}$$

Example 5 (Rational Functions over $\mathbb{Z}_p$) A slight modification of Example 4 yields the set

$$F = \{x + y + z + S_1, x^3 + y^3 + z^3 + S_2, x^5 + y^5 + z^5 + S_3\}$$

Again we fix total degree inverse lexical ordering w.r.t. $x > y > z$ for monomials.

We interpret F as a set of polynomials in the variables x,y, and z over Z_2 and regard S_1, S_2, and S_3 as parameters, i.e. $F \subset Z_2(S_1, S_2, S_3)[x, y, z]$. The Gröbner basis of F is then the set

$$\begin{aligned} G_{Z_2(S_1,S_2,S_3)} &= \{x + y + z + S_1, \\ &\quad y^2 + yz + z^2 + S_1 y + S_1 z + \frac{S_3 + S_1^2 S_2}{S_2 + S_1^3}, \\ &\quad z^3 + S_1 z^2 + \frac{S_3 + S_1^2 S_2}{S_2 + S_1^3} z + \frac{S_1 S_3 + S_2^2 + S_1^3 S_2 + S_1^6}{S_2 + S_1^3}\} \end{aligned}$$

Remark 1 (to Examples 4 and 5) From an algebraic point of view, α, S_1, S_2, and S_3 are *field extensions* of the field $\mathbb{Z}_2$. α is an *algebraic extension of degree 5*, whereas S_1, S_2, and S_3 are *transcendental extensions.*

An instance of Example 5 is to view F as a set of polynomials in the variables x,y, and z over $GF(2^n)$, for unspecified n, and think of S_1 as α^i, S_2 as α^j, and S_3 as α^k, where α is a primitive element in $GF(2^n)$. In contrary to Example 4 the degree of α as an algebraic extension of $\mathbb{Z}_2$ cannot be determined, hence, α must be treated as a transcendental extension of $\mathbb{Z}_2$.

In this sense Example 5 is a generalization of Example 4.

2.2 Term Ordering

In all previous examples examples, total degree inverse lexical ordering for terms in a polynomial was used. In the following examples, we shall discuss the influence of the term ordering on both the resulting Gröbner basis as well as on the computing time.

Example 6 Consider the set of polynomials over the rational numbers

$$\begin{aligned} F &= \{x_1^2x_2x_3 - 15x_1x_2^2x_3 - 17x_1x_3 - 13x_1^2 - 7, \\ &\quad x_1x_2x_3 - 19x_2x_3 - 5x_3, \\ &\quad x_1^3 - 2x_1x_3 - 17x_3\} \end{aligned}$$

Using lexical ordering w.r.t. $x_1 > x_2 > x_3$ for terms the Gröbner basis of F is the set

$$\begin{aligned} G_{\mathrm{lex}} &= \{x_3^6 - \frac{3882524366}{3856705}x_3^5 + \frac{29336079632}{65563985}x_3^4 - \frac{146544200644}{65563985}x_3^3 + \\ &\quad \frac{1076732348381}{65563985}x_3^2 - \frac{36579851959}{65563985}x_3 + \frac{312900721}{65563985}, \\ &\quad x_2 - \frac{41069006293050575626147465502171977091687 5}{7819894144655475719560813127815784366331865 76}x_3^5 + \\ &\quad \frac{5906177835719736205919503222449721118104207 5}{1117127734950782245651544732545112052333123 68}x_3^4 - \\ &\quad \frac{1766646973076432434207495627321125170076776 05}{7819894144655475719560813127815784366331865 76}x_3^3 + \\ &\quad \frac{9154569614313175547020705254475700022893245 85}{7819894144655475719560813127815784366331865 76}x_3^2 - \\ &\quad \frac{221707896522396461929966229679768506295585 00}{257233360021561701301342537099203433103021 9}x_3 + \\ &\quad \frac{653198367426461631074993067771729904700324 5}{159589676421540320807363533220730293190446 24}, \\ &\quad x_1 + \frac{404007826869973491486300617838690584 0}{1063519080586933895486594191504541419 37}x_3^5 - \\ &\quad \frac{581007978071157445826112380980499812369}{151931297226704842212370598786363059 91}x_3^4 + \end{aligned}$$

$$
\begin{aligned}
&\frac{1737825008878312700745790017463831412281}{1063519080586933895486594191504541419 37}x_3^3 - \\
&\frac{8999883565406957360794520256882814016945}{10635190805869338954865941915045414193 7}x_3^2 + \\
&\frac{66301874193909905023853925626281168642046}{1063519080586933895486594191504541419 37}x_3 - \\
&\frac{2300454246671652522780266088087635674 8}{21704471032386406030338656969480437 13}\}
\end{aligned}
$$

Using total degree inverse lexical ordering w.r.t. $x_1 > x_2 > x_3$ for terms the Gröbner basis of F is the set

$$
\begin{aligned}
G_{\mathrm{tdi}} \quad = \quad &\{x_1x_3 - \frac{33630672}{951979}x_2x_3 - \frac{45373}{951979}x_3^2 - \frac{27554056}{951979}x_1 - \\
&\frac{9956414656}{4759895}x_2 - \frac{8816660}{951979}x_3 - \frac{523241163}{951979}, \\
&x_2^2 - \frac{1172324412319}{55998055089536}x_2x_3 - \frac{57850575}{2947266057344}x_3^2 - \frac{122874580595}{55998055089536}x_1 + \\
&\frac{20556157141745}{55998055089536}x_2 - \frac{308205541205}{55998055089536}x_3 + \frac{81094451805}{2947266057344}, \\
&x_1x_2 - 19x_2 - 5, \\
&x_1^2 + \frac{843035439}{7615832}x_2x_3 + \frac{771341}{7615832}x_3^2 + \frac{377980947}{7615832}x_1 + \\
&\frac{136610929347}{38079160}x_2 + \frac{221281645}{7615832}x_3 + \frac{7183441529}{7615832}, \\
&x_3^3 - \frac{18262443665210101 13}{71332042144360}x_2x_3 - \frac{864848325435447}{839200495816}x_3^2 - \\
&\frac{3681916563603807 93}{14266408428872}x_1 - \frac{13312906452010443373 7}{71332042144360}x_2 - \\
&\frac{9411190419244980 7}{14266408428872}x_3 - \frac{69963654591902014 59}{14266408428872}, \\
&x_2x_3^2 + \frac{34187314201}{647345720}x_2x_3 + \frac{9227175}{7615832}x_3^2 + \frac{4067004761}{129469144}x_1 +
\end{aligned}
$$

$$\frac{294057504725}{129469144}x_2 + \frac{1781561079}{129469144}x_3 + \frac{77268344459}{129469144}\}$$

Example 7 Consider the set of polynomials over the rational numbers

$$\begin{aligned} F \;=\; & \{x_1^2x_2x_3 - 15x_1x_2^2x_3 - 17x_1x_3 - 13x_1^2 - 7, \\ & x_1x_2x_3 - 19x_2x_3 - 5x_3, \\ & x_1^3 - 2x_1x_3 - 17x_3\} \end{aligned}$$

The Gröbner bases look similar to the results in the previous example. We use this example to show the unpredictable behaviour of the algorithm depending on the term ordering and the selection strategy for S-polynomials used. See Section 2.3 and Table 1.

Remark 2 In most cases a total degree ordering is faster than the lexical ordering, see also Table 1. Nevertheless, lexical ordering is often used since the Gröbner basis w.r.t. to the lexical ordering has some nice properties that can be utilized when Gröbner bases are used to solve systems of non-linear algebraic equations. Namely, the lexical Gröbner basis w.r.t. $x_1 > \ldots > x_n$ yields a kind of *triangularization* of the input set in the following sense (in case there are only finitely many solutions):

- ◇ the Gröbner basis contains exactly one univariate polynomial in x_n,
- ◇ for all $1 \leq i < n$ there exists a finite number r_i, s.t. the Gröbner basis contains r_i polynomials that depend only on the variables $x_{n-i}, \ldots, x_n$.

This structure can also be seen in Example 6. See [Buchberger, 1985] for a detailed description of the properties of a lexical Gröbner basis, an algorithm for solving systems of non-linear algebraic equations based on Gröbner bases, and further applications of both lexical and total degree Gröbner bases.

2.3 Selection Strategy for S-Polynomials, Pair Ordering

It is known that the *sequence, in which pairs are considered for S-polynomial computation*, has an impact on the computation time needed for computing the Gröbner basis. This is particularly true when lexical ordering for terms is used.

Usually, pairs are ordered w.r.t. the least common multiples (*lcm*) of the leading terms of the polynomials in the pair. Pairs, whose *lcm* is *smaller* in a particular term ordering are considered first. A canonic choice for the term

Term Ordering	Pair Ordering	Example 6	Example 7
lexical	lexical	41.97	1051.40
lexical	total degree	146.42	85.82
total degree	total degree	2.5	1.23

Table 1: Timings in seconds on an HP9000/725 workstation.

ordering used for pairs is to use the same ordering as for terms in the polynomials, but this need not in all examples be the best choice. In fact, it turned out that – for lexical ordering of terms – total degree ordering for pairs is in some examples significantly *faster* than lexical ordering whereas in other examples the opposite is true. See also [Windsteiger, 1992] and [Windsteiger and Buchberger, 1993].

Remark 3 Up to now no reliable heuristic for choosing a "fast" pair ordering is known. Don't hesitate to experiment!

Table 1 shows timings for the input as given in Examples 6 and 7.

3 Basic Structure

In this section, we assume that the contents of the file `GB.tar` has already been extracted, hence, the directory `Groebner` exists. We will often call this directory "Gröbner-directory". In the Gröbner-directory you can find a file called `README`, which contains a brief description of the directory structure.

Basically, the Gröbner-directory contains

- ◇ GRÖBNER, the library responsible for computing Gröbner bases,
- ◇ a library responsible for in- and output of data, and
- ◇ a sample application that shows how to use those libraries.

Remark 4 Note that GRÖBNER is *no executable programm* but only a *library that can be used by other programs.* The actually executable programm ("the application") is the program `gb` in the directory `APP`. This program repeats the following loop as long as the user desires to continue:

1. *read* data (i.e. variables, set of polynomials, etc.) by means of the I/O library,
2. *compute* the Gröbner basis of the input polynomial set by means of GRÖBNER. 4 different variants of the Gröbner bases algorithm can be used:
   - ◇ *Crude Version*: See description of this variant at the beginning of the source file `GROEBNER/src/gb_crude.c.trace` or the description of the function `GB_crude` in [Windsteiger and Buchberger, 1993].
   - ◇ *Criteria*: See description of this variant at the beginning of the source file `GROEBNER/src/gb_criteria.c.trace` or the description of the function `GB_criteria` in [Windsteiger and Buchberger, 1993].
   - ◇ *Reduce All*: See description of this variant at the beginning of the source file `GROEBNER/src/gb_red_all.c.trace` or the description of the function `GB_reduce_all` in [Windsteiger and Buchberger, 1993].
   - ◇ *Small Pair Set*: See description of this variant at the beginning of the source file `GROEBNER/src/gb_small_ps.c.trace` or the description of the function `GB_small_pair_set` in [Windsteiger and Buchberger, 1993].

   and finally
3. *print* the Gröbner basis by means of the I/O library.

4 Possibilities of Adjusting the Library

In this section, we shall discuss, *how the library can be adjusted* according to the possibilities of variation explained in Section 2.

The following programs should be of help when using GRÖBNER and the sample application program `gb`. They are located in the directory `Groebner/bin`, which should therefore be in the shell's search path. See Section 5 for examples how those tools can be used.

4.1 groebner_setup

The program `groebner_setup` can be used for setting up the library according to the possibilities of variation described in Section 2.

```
unix% groebner_setup
============================================
          GENERAL SETUP MENU
============================================
Coefficient Domain .................. C
Term Ordering ....................... TO
Pair Ordering ....................... PO
Normal Form Algorithm ............... N
List Operations ..................... L
Representation of Power Products ..... PoPr
Representation of Pure Polynomials ... PuPo
Polynomial Structure ................ PoSt
Representation of Polynomial Sets .... PoSe
Tracing ............................. T
EXIT ................................ X
============================================
Enter your choice:
```

Figure 1: General Setup Menu

Starting `groebner_setup` leads to the *general setup menu*, see Figure 1. Depending on the choice entered the setup procedure continues.

4.1.1 Coefficient Domain

GRÖBNER is capable of computing Gröbner bases for sets of polynomials that are subsets of $\mathcal{K}[x_1, \ldots, x_n]$, for various fields $\mathcal{K}$. $\mathcal{K}$ is called the *coefficient domain*, $x_1, \ldots, x_n$ are called the *ring variables* and have to be entered at run-time.

If *Coefficient Domain* is chosen from the general setup menu the actual setting for $\mathcal{K}$ can be chosen from the following menu:

```
=================================================
Current setting is: rational_numbers
=================================================
Rational Numbers ........ RN
Rational Functions ...... RF
Finite Field Zp ......... FF
```

```
Galois Field ............ GF
Floting Point Numbers ... FPN
Leave Unmodified ........ <return>
```

First, the *current setting* of the coefficient domain is displayed, the following choices are possible then:

RN chooses *rational numbers* as coefficient domain, i.e. $\mathcal{K} = \mathbb{Q}$.

RF chooses *rational functions* as coefficient domain, i.e. $\mathcal{K} = \mathcal{B}(f_1, \ldots, f_m)$. $\mathcal{B}$ is called the *rational function base domain* and can be chosen from an immediately following menu:

> RN chooses $\mathcal{B} = \mathbb{Q}$,
>
> FF chooses $\mathcal{B} = \mathbb{Z}_p$. p is called the *characteristic* of the finite field and must be entered at run-time.

$f_1, \ldots, f_m$ are called the *field variables* and have to be entered at run-time.

FF chooses *finite fields* of the form $\mathbb{Z}_p$ as coefficient domain, i.e. $\mathcal{K} = \mathbb{Z}_p$. p is called the *characteristic* of the finite field and must be entered at run-time.

GF chooses *finite fields* of the form $GF(p^n)$ as coefficient domain, i.e. $\mathcal{K} = GF(p^n)$. p is called the *characteristic* of the finite field and n is called the *degree* of the field extension. Both p and n have to be entered at run-time.

$GF(p^n)$ is isomorphic to $\mathbb{Z}_p[x]/q$, where q is an irreducible polynomial over $\mathbb{Z}_p$ of degree n. This isomorphism is used to represent elements of $GF(p^n)$ as univariate polynomials over $\mathbb{Z}_p$ of degree less than n. The actual name of the variable x has to be entered at run-time.

In order to perform arithmetic in $GF(p^n)$, p and q are required. At run-time p has to be entered first. In case $p = 2$ the degree of the field extension, n, has to be entered. For $n < 10$ irreducible polynomials over $\mathbb{Z}_2$ of degree n are predefined. For $p \neq 2$ or $n \geq 10$ an irreducible polynomial over $\mathbb{Z}_p$ of degree n must be supplied at run-time.

Although $\mathbb{Z}_p$ is a special case of $GF(p^n)$, namely $n = 1$, we implemented finite fields of the form $\mathbb{Z}_p$ seperately, since arithmetic can be done more efficiently.

FPN Chooses *floating point numbers* as coefficient domain, i.e. $\mathcal{K} = \mathbb{R}$. This is an experimental implementaion of floating point coefficients.

Note: The menu for setting up the coefficient domain can also be accessed directly by calling the program `coef_setup`.

See Section 2.1 for the influence of the chosen coefficient domain.

4.1.2 Term Ordering

GRÖBNER can compute Gröbner bases w.r.t. several orderings $<_T$ of terms within a polynomial.

If *Term Ordering* is chosen from the general setup menu the actual setting for $<_T$ can be chosen from the following menu:

```
=================================================
Current setting is: lexical
=================================================
Total Degree Inverse Lexical ... TDI
Total Degree Lexical ........... TD
Lexical ........................ Lex
Matrix ......................... Mat
Leave Unmodified ............... <return>
```

First, the *current setting* of the term ordering is displayed, the following choices are possible then:

TDI chooses *total degree inverse lexical* term ordering, i.e. $<_T = <_{\mathrm{tdilex}}$.

TD chooses *total degree lexical* term ordering, i.e. $<_T = <_{\mathrm{tdeg}}$.

Lex chooses *lexical* term ordering, i.e. $<_T = <_{\mathrm{lex}}$.

Mat chooses *matrix* term ordering, i.e. $<_T = <_M$, for a matrix M.

Note: The menu for setting up the term ordering can also be accessed directly by calling the program `to_setup`.

See [Windsteiger and Buchberger, 1993] for the definitions of those orderings and Section 2.2 for the influence of the chosen term ordering.

4.1.3 Pair Ordering

GRÖBNER can compute Gröbner bases w.r.t. several orderings of pairs of polynomials.

If *Pair Ordering* is chosen from the general setup menu the same choices are possible like for the term ordering.
Note: The menu for setting up the pair ordering can also be accessed directly by calling the program `po_setup`.

See Section 2.3 for the influence of the chosen pair ordering.

4.1.4 Tracing

Various levels of information during the computation of a Gröbner basis are possible.

If *Tracing* is chosen from the general setup menu the file responsible for *activating groups of trace statements* can be edited.

See [Windsteiger and Buchberger, 1993] for an explanation of the organization of tracing information during the algorithm.

4.1.5 Further choices

Further choices available in the general setup menu *should not concern the "Black Box User"*. Most of them deal with underlying data structures used for representation of domains. We refer to [Windsteiger and Buchberger, 1993] for a detailed description of data structures available in GRÖBNER. After having read the sections about "Virtual Domains" and "Implementations of Virtual Domains" the respective items in the general setup menu will be self-explaining.

4.2 mkgroebner

After having finished the setup the GRÖBNER library, the I/O library, and the application program eventually have to be recompiled. This step can be done by the program `mkgroebner`.

`mkgroebner` does nothing more than issueing a `make` command in the appropriate directories. This implies that only those files that are affected by changes of the setup are actually recompiled.

4.3 groebner

`groebner` is the program that actually starts the application program `gb`. The GRÖBNER library as well as the I/O library expect certain environment variables to be set. Those variables are set in `groebner`, then `mkgroebner` is called to check, whether recompilation is necessary, then the application program is called.

Issue the `-f` option if you want to skip recompilation (of course, only if *you are sure that recompilation is superflous!*.

Issue the `-h` option to get information about *all command-line parameters* that are understood by `groebner`.

4.4 xgroebner

Same as `groebner` except that the application program and on-line help information (if requested) are started in seperate `xterm`s.

5 Using Available Tools

In this section, we shall illustrate how the tools described in Section 4 can be used.

Before starting `groebner` the user has to decide the following issues:

- ◇ Which *coefficient domain* do I want to use?
- ◇ Which *term ordering* do I want to use?
- ◇ Which *pair ordering* do I want to use?
- ◇ Which *level of trace information* do I want to use?

After having decided about the above questions, `groebner_setup` is employed to accomplish the requested changes in the setup.

After finishing the setup, call `groebner` or `xgroebner`. Unless the `-f` option is given, the libraries are recompiled (if necessary) before the application program is started. If you do not want to start the application immediately, call `mkgroebner` instead.

Therefore, the sequence of steps is as follows:

1. Decide about desired setup,
2. Setup libraries by using `groebner_setup`,
3. Call `groebner`, `xgroebner`, or `mkgroebner`.

5.1 Examples

Suppose we want to compute the Gröbner basis of the set

$$F = \{16x^2 + 4xy^2 - 4z + 1, 4x + 2y^2z + 1, 2x^2z - x - 2y^2\}$$

(see Section 2.1) for the following setup:

- ◇ Coefficient Domain: Rational Numbers
- ◇ Term Ordering: Total Degree Inverse Lexical
- ◇ Pair Ordering: Total Degree Inverse Lexical
- ◇ Tracing: Inform about setup, leading power product of non-zero normal forms, application of criteria, and zero reductions.

The dialog would look as follows:

```
unix% groebner_setup
============================================
          GENERAL SETUP MENU
============================================
Coefficient Domain .................. C
Term Ordering ....................... TO
Pair Ordering ....................... PO
Normal Form Algorithm ............... N
List Operations ..................... L
Representation of Power Products ..... PoPr
Representation of Pure Polynomials ... PuPo
Polynomial Structure ................ PoSt
Representation of Polynomial Sets .... PoSe
Tracing ............................. T
EXIT ................................ X
============================================
Enter your choice: C
=================================================
Current setting is: rational_numbers
=================================================
Rational Numbers ........ RN
Rational Functions ...... RF
Finite Field Zp ......... FF
Galois Field ............ GF
```

```
Floting Point Numbers ... FPN
Leave Unmodified ........ <return>

Enter your choice:
============================================
          GENERAL SETUP MENU
============================================
Coefficient Domain .................. C
Term Ordering ....................... TO
Pair Ordering ....................... PO
Normal Form Algorithm ............... N
List Operations ..................... L
Representation of Power Products ..... PoPr
Representation of Pure Polynomials ... PuPo
Polynomial Structure ................ PoSt
Representation of Polynomial Sets .... PoSe
Tracing ............................. T
EXIT ................................ X
============================================
Enter your choice: TO
=================================================
Current setting is: lexical
=================================================
Total Degree Inverse Lexical ... TDI
Total Degree Lexical ........... TD
Lexical ........................ Lex
Matrix ......................... Mat
Leave Unmodified ............... <return>

Enter your choice: TDI
============================================
          GENERAL SETUP MENU
============================================
Coefficient Domain .................. C
Term Ordering ....................... TO
Pair Ordering ....................... PO
Normal Form Algorithm ............... N
List Operations ..................... L
Representation of Power Products ..... PoPr
```

```
Representation of Pure Polynomials ... PuPo
Polynomial Structure ................ PoSt
Representation of Polynomial Sets .... PoSe
Tracing ............................ T
EXIT ............................... X
============================================
Enter your choice: PO
=================================================
Current setting is: total_degree_inverse_lexical
=================================================
Total Degree Inverse Lexical ... TDI
Total Degree Lexical ........... TD
Lexical ........................ Lex
Matrix ......................... Mat
Leave Unmodified ............... <return>

Enter your choice:
============================================
          GENERAL SETUP MENU
============================================
Coefficient Domain .................. C
Term Ordering ....................... TO
Pair Ordering ....................... PO
Normal Form Algorithm ............... N
List Operations ..................... L
Representation of Power Products ..... PoPr
Representation of Pure Polynomials ... PuPo
Polynomial Structure ................ PoSt
Representation of Polynomial Sets .... PoSe
Tracing ............................ T
EXIT ............................... X
============================================
Enter your choice: T

Group     1: Include IO-library header file:                                 Y
Group     2: Print algorithm variant :                                       N
Group     4: Print the current setup (general):                              Y
Group     8: Print the current setup (details):                              Y
Group    16: Statistics:                                                     N
```

```
Group    32: Inform about reduction:                                  N
Group    64: Inform about new polynomial retrieved from reduction:    Y
Group   128:    Label (i.e. number) of this polynomial:               N
Group   256:    Leading power product:                                Y
Group   512:    Size (i.e. number of terms):                          N
Group  1024:    Whole polynomial:                                     N
Group  2048: Inform about application of criteria:                    Y
Group  4096: Announce zero-reduction:                                 Y
Group  8192: Print size of basis and pair set:                        N
Group 16384: Details during reduction:                                N
============================================
          GENERAL SETUP MENU
============================================
Coefficient Domain .................. C
Term Ordering ....................... TO
Pair Ordering ....................... PO
Normal Form Algorithm ............... N
List Operations ..................... L
Representation of Power Products ..... PoPr
Representation of Pure Polynomials ... PuPo
Polynomial Structure ................ PoSt
Representation of Polynomial Sets .... PoSe
Tracing ............................. T
EXIT ................................ X
============================================
Enter your choice: X
Setup finished
unix% groebner

  <Compiling Messages>

Done.
Path for Input-file: /disk2/users/groebner/Groebner/input
Type input-file name ( <RETURN> = stdin, ? = help):
Path for Output-file: /disk2/users/groebner/Groebner/input
Output-file name ( <RETURN> = stdout, ? = help):
***********************************************************
Input File:  stdin
***********************************************************
```

```
***********************************************************
Output File: stdout
***********************************************************
============================
WELCOME TO GROEBNER

  Version 2.1
  I/O library Version 2.1
  by the RISC-Linz Institute

Good Luck ...
============================
SACLIB initialized with 500000 cells in SPACE.

Compute GROEBNER BASIS (Crude Version) ........... 1
Compute GROEBNER BASIS (Criteria) ................ 2
Compute GROEBNER BASIS (Reduce All) .............. 3
Compute GROEBNER BASIS (Small Pair Set) .......... 4
EXIT ............................................. 0
Enter your choice: 3
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Current Compiled Version:
Groebner Library: Version 2.1
List Operations: Recursive
Coefficient Domain: Rational Numbers
Term Ordering: Total Degree Inverse Lexicographic Ordering
Pair Ordering: Total Degree Inverse Lexicographic Ordering
Normal Form Algorithm: Complete Normal Form
-----------------------------------------------------------
Node Type: hp9000s700
Host: hotblack
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Groebner Basis with Reduce All
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Initializing Global Variables for GROEBNER ...
Done.
Initializing Global Variables for GROEBNER-IO ...
Enter ring-variables in ascending order (? for help): z y x;
```

```
Done.
============================================================
Number of Variables: 3
Ring Variables: x > y > z
============================================================
Enter Polynomial Set (? for help):

{16*x^2 + 4*x*y^2 - 4*z + 1,
 4*x + 2*y^2*z + 1,
 2*x^2*z - x - 2*y^2}
Reading of polynomial set complete
{
   2
2 y z + 4 x + 1,
   2       2
2 x z - 2 y  - x,
     2       2
4 x y  + 16 x  - 4 z + 1
}
        New polynomial:
                [0 2 1]
        New polynomial:
                [2 0 1]
        New polynomial:
                [1 2 0]
New polynomial:
[2 0 0]
        New polynomial:
                [0 0 3]
New polynomial:
[0 4 0]
Zero-reduction
Product Criterion
Zero-reduction
Product Criterion
Zero-reduction
Product Criterion
Product Criterion
Product Criterion
```

```
Groebner Basis =
{
 2      2   1  2   3     1
x  - 2 y  + - z  - - x - - z,
            2      4     8

 3      2   3       1  2
z  + 2 y  - - x z - - z  + 9 x + 2,
            2       4

 2          1
y z + 2 x + -,
            2

   2      2      2          1     1
x y  + 8 y  - 2 z  + 3 x - - z + -,
                            2     4

 4      2       2   1          2          5     9
y  - x z  - 32 y  + - x z + 8 z  - 12 x + - z - -
                    4                     2     8
}
User Time: 0.030000

Compute GROEBNER BASIS (Crude Version) ........... 1
Compute GROEBNER BASIS (Criteria) ................ 2
Compute GROEBNER BASIS (Reduce All) .............. 3
Compute GROEBNER BASIS (Small Pair Set) .......... 4
EXIT ............................................. 0
Enter your choice: 0
```

General Remark

Never forget that you have access to the source code. In many of the source files as well as in the shell-scripts described in Section 4 you can find rich documentation. For example, every source file contains a descriprion of datastructures used and some other useful information on the respective domain. Detailed information on the different variants of the Gröbner bases algorithm available can also be found at the beginning of the respective source files. We do not duplicate all those informations in the manuals!

Acknowledgment

I want to thank the Department of Electronics and Electric Engineering at the University of Pretoria, *Prof. Walter Penzhorn* particularly, for the opportunity of my visit and the interesting applications of the GRÖBNER package in Coding Theory. Actually, implementation of the the general case of finite fields, namely $GF(p^n)$, as well as the introduction of $\mathbb{Z}_p$ as an additional choice for the base domain of rational functions is due to these applications.

I also want to thank *Cobus Nel* for spending much time in maintaining the computing environment, which helped a lot during the installation of the GRÖBNER library.

Last but not least, I want to thank *Prof. Bruno Buchberger* for all his support during the years and for giving me the opportunity of my visit at the University of Pretoria.

References

[Buchberger *et al.*, 1993] B. Buchberger, G. E. Collins, M. J. Encarnacion, H. Hong, Johnson J. R., W. Krandick, R. Loos, A. M. Mandache, A. Neubacher, and H. Vielhaber. SACLIB User's Guide. Technical Report RISC-Linz Series 93-19, Univ. Linz, RISC, Linz, Austria, 1993.

[Buchberger, 1985] B. Buchberger. Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory. In N.K. Bose, editor, *Multidimensional Systems Theory*, pages 184–232. D. Reidel Publishing Company, Dordrecht-Boston-Lancaster, 1985.

[Buchberger, 1987] B. Buchberger. Applications of Gröbner bases in Non-Linear Computational Geometry. In *Proc. Workshop on Scientific Software*, IMA, Minneapolis, USA, 1987. Springer.

[Windsteiger and Buchberger, 1993] W. Windsteiger and B. Buchberger. GRÖBNER: A Library for Computing Gröbner Bases based on SACLIB[1]. Technical Report 72, RISC-Linz, University of Linz, 1993.

[1]http://www.risc.uni-linz.ac.at:/people/wwindste/Public/Reports/GROEBNER/master.ps.gz

[Windsteiger, 1992] W. Windsteiger. Gröbner Bases: A Characterization by Syzygy Completeness and an Implementation[2]. Master's thesis, RISC-Linz, University of Linz, Austria, 1992.

[2] http://www.risc.uni-linz.ac.at:/people/wwindste/Public/Reports/master-thesis.ps.gz