Introduction to Computer Algebra

(Solving Systems of Polynomial Equations)

Anne Frühbis-Krüger
Christoph Lossen

Preliminary version

February 24, 2005

Contents

Introduction 3

1 Basic Objects and Notations **7**

1.1 Polynomial Rings and Monomial Orderings 7

1.2 Finite Fields 14

2 Factorization **17**

2.1 The Euclidean Algorithm 19

2.2 Resultants 25

2.3 Factorization in $\mathbb{F}_q[x]$ 31

2.3.1 Squarefree factorization 31

2.3.2 Distinct degree factorization 36

2.3.3 The algorithms of Berlekamp, Cantor/Zassenhaus 40

2.4 Factorization in $\mathbb{Z}[x]$, $\mathbb{Q}[x]$ 49

2.4.1 Kronecker's Principle 50

2.4.2 A Modular Approach 53

2.5 Multivariate Factorization 65

2.5.1 Kronecker's Algorithm 65

2.5.2 Multivariate Factorization in $K[x_1, \ldots, x_n]$ 67

2.6 Factorization in $K(\alpha)[x]$ 71

3 Gröbner Basis Techniques **73**

3.1 Division with Remainder 75

3.2 Gröbner Bases 78

3.3 Solving by Elimination 86

3.4 Preprocessing Steps 95

3.4.1 Gröbner Basis Conversion 96

3.4.2 Computing the Radical* 97

3.4.3 Factorizing Gröbner Bases 99

3.4.4 Primary Decomposition* 100

3.4.5 Triangular Decompositions 100

Bibliography **107**

Index **111**

Introduction

This course deals with one of the most classical problems of mathematics: **solving systems of polynomial equations**.

Probably, everybody has been taught in school how to solve systems of linear equations (by Gaussian elimination) and how to compute the solutions of a quadratic equation (by the p, q-formula or by completion of the square). But, how to solve systems of several quadratic equations or, how to find the roots of a polynomial of degree at least 5 [1], or, most generally, how to solve a system

$$\begin{array}{rcl} f_1(x_1,\ldots,x_n) & = & 0 \\ \vdots & & \vdots \\ f_k(x_1,\ldots,x_n) & = & 0 \end{array} \tag{0.0.1}$$

of arbitrary polynomial equations? The first method which a student of mathematics usually encounters is Newton's method for computing the floating point approximation of one zero of a system (0.0.1) which is defined over the field of complex numbers $\mathbb{C}$ or a subfield thereof. There are many variations of Newton's method, but all these **numerical methods** share the same weak spots: they

- are often *unstable* in an unpredictable way;
- frequently do not find all solutions, respectively the correct number of different solutions;
- have problems with *over-determined systems*, that is, systems with more equations than variables ($k > n$);
- can hardly treat *under-determined systems* ($k < n$);
- almost unavoidably get into trouble near singular points, that is, points where the solution set does not behave regularly (see also Figure 1).

Therefore, in many situations, it is worth investing more computing power to derive additional knowledge on the solution set's structure, or sometimes even to **solve** the system **symbolically** (that is, computing the exact solutions instead of floating point approximations for the solutions).

---

[1] For polynomials $f \in \mathbb{C}[x]$ of degree 3 or 4, one can find the roots of f by Cardano's formulas (which are taught by some highly motivated school teachers). Note that for polynomials of degree ≥ 5 there is no such formula as is proved in an introductory course on algebra.

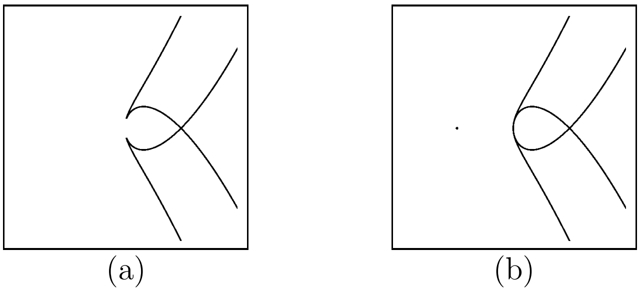

(a) (b)

Figure 1: When visualizing the real solution set of a polynomial equation $f(x,y)=0$, (a) purely numerical methods run into trouble near singular points, and they fail to visualize isolated solutions completely. Using algebraic information on the solution set as obtained by a symbolic preprocessing, together with an adapted numerical method for drawing can produce relief here (b).

For instance, if the system to be solved depends on parameters which could vary for different computations, a symbolic solution is particularly useful. It provides a *simultaneous* solution for all parameter values, and it gives perfect insight into how the solution behaves when the parameters vary.

Example. Determine the number of different solutions of the system

$$\begin{array}{rcl} x_3+x_4+2x_1 & = & 0 \\ x_1x_2^2-t^2x_3x_4 & = & 0 \\ x_1x_4-tx_2^2 & = & 0 \\ x_1x_3^2-x_2^2x_4 & = & 0 \end{array}$$

for a general choice of the parameter t, and determine how the solutions behave when the parameter t varies in $\mathbb{R}$:

Another question which requires symbolic methods for solving is the problem of counting solutions with certain properties. Such questions typically occur for finite fields in the context of coding theory:

Example. Determine the number of points $(x,y)\in\mathbb{F}_{16}^2$ satisfying $x^5+y^2+y=0$.

In this course, we give an overview of a wide range of **symbolic methods** which can be applied for solving a system of polynomial equations symbolically, or, to derive additional information on the solution set, such as the number of different solutions. After some preparation in Chapter 1, we start in Chapter 2 by discussing algorithms for the **factorization of univariate polynomials** in $K[x]$, where we concentrate on the case that K is either a finite field or $\mathbb{Q}$. In particular, we discuss a modular approach for univariate factorization over the rationals. This kind of approach applies to the **factorization of multivariate polynomials** in $\mathbb{Q}[x_1,\ldots,x_n]$, too.

Chapter 3 is then devoted to Gröbner basis techniques. The concept of a Gröbner basis generalizes the row echelon form of a system of linear equations. And, indeed, as from the row echelon form, one may read much information on the solution set from a Gröbner basis. We shall, in particular, discuss an algorithm for solving systems of polynomial equations by a mixture of symbolic and numerical methods. And, we sketch several possibilities for improving this algorithm by additional preprocessing steps. Many of these preprocessing steps cannot be discussed in detail as they require a deeper knowledge of commutative algebra and/or Algebraic Geometry.

We have to concede that, due to the high complexity, the applicability of all the symbolic methods discussed in this course to the so-called 'real life problems' is very limited. However, the continuous progress of computing facilities and algorithms for symbolic computations makes more and more systems accessible for symbolic methods.

Prerequisites. We assume that the reader is familiar with linear algebra, in particular with Gaussian elimination and with the theory of vector spaces. Moreover, we assume some basic knowledge in algebra (the concepts of finite fields, and of field extensions) and commutative algebra (the concepts of rings, ideals and quotient rings). For the convenience of the reader, we recall in Chapter 1 some of the basic notions and facts that are needed throughout this course.

Chapter 1

Basic Objects and Notations

When dealing with algorithms for solving systems of polynomial equations, for factorization of polynomials and for gcd calculations later in this course, we assume some familiarity with polynomial rings and with finite fields – on the level of knowledge of a second year student who has already followed a course in basic (commutative) algebra. To fix notation and provide the opportunity to look up statements which will be used later on, we briefly recall some definitions and theorems in this section.

1.1 Polynomial Rings and Monomial Orderings

Throughout this course, we will be working in **polynomial rings** over a commutative ring R:

$$R[x_1,\dots,x_n] := \left\{ \sum_{\boldsymbol{\alpha}\in\mathbb{N}^n} c_{\boldsymbol{\alpha}} \cdot x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n} \;\middle|\; c_{\boldsymbol{\alpha}} \in R,\ \#\{\boldsymbol{\alpha} \mid c_{\boldsymbol{\alpha}} \neq 0\} < \infty \right\}$$

($\mathbb{N} = \{0, 1, 2, 3, \dots, \}$ denotes the semigroup of non-negative integers).

Definition 1.1.1. Let $R[\boldsymbol{x}] := R[x_1,\dots,x_n]$, then we fix the following notation:

- $\boldsymbol{x}^{\boldsymbol{\alpha}} := x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n}$ is called a **monomial** in $x_1,\dots,x_n$ with **exponent vector** $\boldsymbol{\alpha} \in \mathbb{N}^n$.
- $|\boldsymbol{\alpha}| := \alpha_1 + \cdots + \alpha_n$ is called the **total degree** of the monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$.
- $\mathrm{Mon}(\boldsymbol{x}) := \{\boldsymbol{x}^{\boldsymbol{\alpha}} \mid \boldsymbol{\alpha} \in \mathbb{N}^n\}$.

Moreover, if $f = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}} \in R[\boldsymbol{x}] \setminus \{0\}$, then we call

- $c_{\boldsymbol{\alpha}}$ the **coefficient** of $\boldsymbol{x}^{\boldsymbol{\alpha}}$ in f,
- $c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$ with $c_{\boldsymbol{\alpha}} \neq 0$ a **term** of f, and
- $\deg(f) := \max\{|\boldsymbol{\alpha}| \mid c_{\boldsymbol{\alpha}} \neq 0\}$ the **total degree** of f.

Moreover, we set $\deg(0) := -\infty$.

For $n = 1$, we also refer to **univariate** polynomials, in contrast to **multivariate** polynomials for $n \geq 2$.

We shall be mainly concerned with the case that R is an integral domain. Recall that a commutative ring R is called an **integral domain** if $a \cdot b = 0$ (for $a, b \in R$) implies $a = 0$ or $b = 0$. In other words, R is an integral domain iff[1] there are no zerodivisors[2] in R. Mostly, R will either be a field, such as the field of rational numbers $\mathbb{Q}$, a finite field $\mathbb{F}_q$ (or an algebraic extension thereof), or R will be the ring of integers $\mathbb{Z}$, or in some cases a polynomial ring.

To unify notations, we make the following convention:

Convention 1.1.2. All **rings** in this lecture are commutative with a 1. K always denotes either a field or the ring of integers (unless specified otherwise).

Recall the equality of rings

$$R[x_1, \ldots, x_n] = \underbrace{R[x_1, \ldots, x_{n-1}]}_{=:\, R'}[x_n]\,. \tag{1.1.1}$$

Thus, each multivariate polynomial with coefficients in an integral domain R (e.g., $R = K$) can be viewed as a univariate polynomial with coefficients in another integral domain R'. This allows us to proceed by induction on the number of variables, when proving statements for polynomial rings $R[\boldsymbol{x}]$.

Remark 1.1.3. If $f \in R[\boldsymbol{x}]$ is not a monomial, there are many ways to write f as an (ordered) finite sum of terms. Indeed, each non-trivial permutation of the summands leads to another presentation of f. This has many disadvantages, in particular, if we want to compare two polynomial expressions.

To simplify comparisons (and other operations), we should like to specify a total ordering on the set of monomials $\mathrm{Mon}(\boldsymbol{x})$, and to require that the terms in a presentation of f (as ordered finite sum of terms) are strictly decreasing.

For *univariate polynomials*, there is a natural ordering of monomials in $R[x]$, according to their total degrees:

$$x^\alpha >_{\mathrm{nat}} x^\beta \ :\Longleftrightarrow\ \alpha > \beta\,.$$

In the case of *multivariate polynomials*, however, there is no such "natural" total ordering. Indeed, the natural extension of $>_{\mathrm{nat}}$ given by

$$x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n} >_{\mathrm{nat}} x_1^{\beta_1} \cdot \ldots \cdot x_n^{\beta_n} \ :\Longleftrightarrow\ |\alpha| > |\beta|, \text{ and } \alpha_i \geq \beta_i \text{ for all } i = 1, \ldots, n\,,$$

---

[1] "**iff**" is a short-hand for "if and only if".

[2] $a \neq 0$ is called a **zerodivisor** if there is some $b \neq 0$ such that $ab = 0$ in R. Otherwise, a is called a **non-zerodivisor**.

defines only a partial ordering on $\mathrm{Mon}(\boldsymbol{x})$ (for $n \geq 2$). It remains to extend $>_{\mathrm{nat}}$ to a total ordering. This can be done in many ways. The only (natural) requirement is that such an extension should be compatible with the semigroup structure of $\mathrm{Mon}(\boldsymbol{x})$ (w.r.t. multiplication), leading to the following definition:

Definition 1.1.4. A **monomial ordering**[3] on $R[\boldsymbol{x}] = R[x_1, \ldots, x_n]$ is any relation $>$ on $\mathrm{Mon}(\boldsymbol{x})$ (respectively on $\mathbb{N}^n$) such that

(1) $>$ is a *total ordering* on $\mathrm{Mon}(\boldsymbol{x})$ (respectively on $\mathbb{N}^n$).

(2) $>$ is *multiplicative*, that is, $\boldsymbol{x}^{\boldsymbol{\alpha}} > \boldsymbol{x}^{\boldsymbol{\beta}}$ implies $\boldsymbol{x}^{\boldsymbol{\alpha}} \cdot \boldsymbol{x}^{\boldsymbol{\gamma}} > \boldsymbol{x}^{\boldsymbol{\beta}} \cdot \boldsymbol{x}^{\boldsymbol{\gamma}}$ (respectively $\boldsymbol{\alpha} > \boldsymbol{\beta}$ implies $\boldsymbol{\alpha} + \boldsymbol{\gamma} > \boldsymbol{\beta} + \boldsymbol{\gamma}$) for all $\boldsymbol{\gamma} \in \mathbb{N}^n$.

(3) $>$ is a *well-ordering*, that is, any non-empty subset of $\mathrm{Mon}(\boldsymbol{x})$ (respectively of $\mathbb{N}^n$) has a minimal element.

Note that the third condition asserts that $>$ extends the natural partial ordering $>_{\mathrm{nat}}$. Indeed, condition (3) holds iff $x_i > 1$ for all i, and then (2) gives the implication $\boldsymbol{x}^{\boldsymbol{\alpha}} >_{\mathrm{nat}} \boldsymbol{x}^{\boldsymbol{\beta}} \Rightarrow \boldsymbol{x}^{\boldsymbol{\alpha}} > \boldsymbol{x}^{\boldsymbol{\beta}}$.

Example 1.1.5 (Monomial Orderings). The following are the most important examples of monomial orderings on $R[\boldsymbol{x}] = R[x_1, \ldots, x_n]$:

(a) **Lexicographic order** (with respect to $x_1 > x_2 > \cdots > x_n$)

$$\boldsymbol{x}^{\boldsymbol{\alpha}} >_{\mathtt{lp}} \boldsymbol{x}^{\boldsymbol{\beta}} :\Longleftrightarrow \exists\, s \text{ such that } \alpha_1 = \beta_1, \ldots, \alpha_{s-1} = \beta_{s-1}, \alpha_s > \beta_s\,.$$

(b) **Graded Lexicographic order**

$$\boldsymbol{x}^{\boldsymbol{\alpha}} >_{\mathtt{Dp}} \boldsymbol{x}^{\boldsymbol{\beta}} :\Longleftrightarrow |\boldsymbol{\alpha}| > |\boldsymbol{\beta}| \text{ or } \left(|\boldsymbol{\alpha}| = |\boldsymbol{\beta}| \text{ and } \boldsymbol{x}^{\boldsymbol{\alpha}} >_{\mathtt{lp}} \boldsymbol{x}^{\boldsymbol{\beta}}\right).$$

(c) **Graded reverse lexicographic order**

$$\begin{aligned}\boldsymbol{x}^{\boldsymbol{\alpha}} >_{\mathtt{dp}} \boldsymbol{x}^{\boldsymbol{\beta}} :\Longleftrightarrow |\boldsymbol{\alpha}| > |\boldsymbol{\beta}| \text{ or } (&|\boldsymbol{\alpha}| = |\boldsymbol{\beta}| \text{ and } \exists\, s \text{ such that} \\ &\alpha_n = \beta_n, \ldots, \alpha_{n-s+1} = \beta_{n-s+1}, \alpha_{n-s} < \beta_{n-s})\,.\end{aligned}$$

Here the indices `lp`, `Dp`, `dp` refer to the short-hand notations for these orderings in the computer algebra system SINGULAR.

Concrete examples: We consider the monomials $x_1^3 x_2 x_3^2$, $x_1^2 x_2^3 x_3$ and $x_1^2 x_2 x_3^4$. They are ordered as follows:

---

[3] In some textbooks (such as [GrP]), the notion of a monomial ordering is defined in a slightly more general way, dropping the third condition. With this generalization, computations in localized polynomial rings such as $K[\boldsymbol{x}]_{\langle \boldsymbol{x} \rangle}$ become possible. Since such computations are not in the focus of this introductory course, we restrict ourselves to well-orderings (which are also referred to as "global" orderings). This makes many statements and proofs much simpler, providing in this way an easier access to Gröbner basis techniques.

- $x_1^3x_2x_3^2 >_{\texttt{lp}} x_1^2x_2^3x_3 >_{\texttt{lp}} x_1^2x_2x_3^4$,
- $x_1^2x_2x_3^4 >_{\texttt{Dp}} x_1^3x_2x_3^2 >_{\texttt{Dp}} x_1^2x_2^3x_3$,
- $x_1^2x_2x_3^4 >_{\texttt{dp}} x_1^2x_2^3x_3 >_{\texttt{dp}} x_1^3x_2x_3^2$.

With respect to a given monomial ordering, there is a unique way of writing a polynomial f as (ordered) finite sum of strictly decreasing terms: we define (for $c_{\boldsymbol{\alpha}}, c_{\boldsymbol{\beta}} \neq 0$)

$$c_{\boldsymbol{\alpha}}x^{\boldsymbol{\alpha}} > c_{\boldsymbol{\beta}}x^{\boldsymbol{\beta}} :\Longleftrightarrow \boldsymbol{\alpha} > \boldsymbol{\beta}\,.$$

In particular, it makes sense to talk about the 'largest' term appearing in a polynomial. Since this term plays an important role, for instance, in Gröbner basis computations (as treated in Chapter 3), we introduce some additional notation:

Definition 1.1.6 (Leading Data). Let $f = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}}\boldsymbol{x}^{\boldsymbol{\alpha}} \in R[\boldsymbol{x}] \setminus \{0\}$, and let $>$ be any monomial ordering on $R[\boldsymbol{x}]$. Set $\boldsymbol{\alpha}_0 := \max\{\boldsymbol{\alpha} \in \mathbb{N}^n \mid c_{\boldsymbol{\alpha}} \neq 0\}$. Then

- $\mathrm{lc}(f) := c_{\boldsymbol{\alpha}_0}$ is called the **leading coefficient** of f.
- $\mathrm{lm}(f) := \boldsymbol{x}^{\boldsymbol{\alpha}_0}$ is called the **leading monomial** of f.
- $\mathrm{lt}(f) := c_{\boldsymbol{\alpha}_0}\boldsymbol{x}^{\boldsymbol{\alpha}_0}$ is called the **leading term** of f.

If $\mathrm{lc}(f) = 1$, the polynomial f is called **monic**.

Remark 1.1.7. If R is an integral domain, we obviously have (for $f, g \in R[\boldsymbol{x}]$):

$$\mathrm{lt}(f \cdot g) = \mathrm{lt}(f) \cdot \mathrm{lt}(g)\,, \quad \mathrm{lm}(f \cdot g) = \mathrm{lm}(f) \cdot \mathrm{lm}(g)\,, \quad \mathrm{lc}(f \cdot g) = \mathrm{lc}(f) \cdot \mathrm{lc}(g)\,.$$

As solving systems of polynomial equations is closely related to the factorization of polynomials (see Chapter 2), we now proceed by recalling some basic definitions and facts on factorization. Proofs of the statements and examples illustrating the definitions can be found in a course (or a textbook such as [AtM, Mat]) on basic commutative algebra.

Definition 1.1.8 (Unique Factorization). Let R be a ring, and let $f \in R$.

(1) f is called a **unit**, if there is an element $g \in R$ such that $f \cdot g = 1$. We also say that f is an **invertible element**.

(2) f is called **irreducible**, if the following conditions hold:

  - f is neither zero nor a unit.
  - If there are elements $g, h \in R$ such that $f = g \cdot h$, then g or h is a unit.

(3) f is called a **prime** element, if the following conditions hold:

  - f is neither zero nor a unit.

  - If f divides a product, then it divides one of the factors.

  Here, we say that $f \in R$ **divides** $g \in R$ (or that g is **divisible** by f) if there exists some $a \in R$ such that $af = g$. We also use the short-hand notation $f \mid g$.

(4) R is a called a **factorial ring**, or a **unique factorization domain** (**UFD**), if the following conditions hold:

  - R is an integral domain.
  - For every non-unit $f \in R \setminus \{0\}$ there are prime elements $f_1, \dots, f_s \in R$ such that $f = f_1 \cdot \cdots \cdot f_s$.

Facts 1.1.9 (on UFD).

(1) It follows directly from the definitions that each prime element is irreducible. If R is a UFD, then $f \in R$ is irreducible iff f is prime.

(2) Each field, as well as the ring of integers $\mathbb{Z}$, is a UFD.

(3) Each UFD R has the **unique factorization property**: if $f = f_1 \cdot \ldots \cdot f_s$ and $f = g_1 \cdot \ldots \cdot g_r$ are factorizations of f into irreducible elements, then $s = r$ and, up to a permutation of the factors, $f_i = u_i g_i$ for a suitable unit u_i in R. We also speak about "the" **irreducible factorization** of f in R

(4) The **Lemma of Gauß** states that, if R is a UFD, then the polynomial ring $R[x]$ is a unique factorization domain, too. In particular, it follows from (1.1.1) that the polynomial rings $K[\boldsymbol{x}] = K[x_1, \dots, x_n]$, $n \geq 1$, are UFDs. The set of units of $K[\boldsymbol{x}]$ is precisely the set of units of K. In particular, if K is a field, all non-zero elements of K are units; if $K = \mathbb{Z}$, only 1 and -1 are units.

(5) The quotient ring $R = \mathbb{Q}[x, y, z]/(xy - z^2)$ is not a UFD, since $z \in R$ is irreducible but not prime: z divides $xy = z^2$, but it neither divides x nor y. An alternative proof is to show the failure of the unique factorization property: $x \cdot y$ and $z \cdot z$ are two different irreducible decompositions of the same element of R (as x, y do not divide z).

For univariate polynomials with rational coefficients, there is a well-known sufficient criterion for irreducibility (which can be shown by writing f as a product of two polynomial factors and inductively comparing coefficients):

Lemma 1.1.10 (Eisenstein's Criterion). *If* $f = \sum_{i=0}^{d} a_i x^i \in \mathbb{Z}[x]$, *and if there exists a prime* $p \in \mathbb{Z}$ *such that*

- $\operatorname{lc}(f) = a_d \not\equiv 0 \bmod p$,
- $a_i \equiv 0 \bmod p$ *for all* $0 \leq i < d$,

- $a_0 \not\equiv 0 \bmod p^2$,

then f is irreducible in $\mathbb{Q}[x]$.

Example 1.1.11. $f = 2x^2+4$ is irreducible as an element of $\mathbb{Q}[x]$, but not as an element of $\mathbb{Z}[x]$: it can be written as the product $2 \cdot (x^2+2)$. In $\mathbb{Z}[x]$, none of the two factors is a unit, showing that f is not irreducible as an element of this ring. In $\mathbb{Q}[x]$, the polynomial (x^2+2) is irreducible (as follows from Eisenstein's criterion 1.1.10), and 2 is a unit in $\mathbb{Q}$. Hence, f is irreducible in $\mathbb{Q}[x]$.

Over the algebraic field extension $\mathbb{Q}(i\sqrt{2}) \supset \mathbb{Q}$ (of degree 2),

$$x^2+2 = (x+i\sqrt{2})(x-i\sqrt{2})$$

is a prime decomposition of the non-constant factor of f. In particular, f is not irreducible in $\mathbb{Q}(i\sqrt{2})[x]$.

Definition 1.1.12. Let R be an integral domain, let $f, g, h \in R$. Then h is called a **greatest common divisor** (**g.c.d.**) of f and g if h divides f and g, and if all common divisors of f and g divide h.

Remark 1.1.13 (Uniqueness of g.c.d. in a UFD). Let R be a UFD, and let h and h' be g.c.d.'s of f and g. Applying the second property of the definition to each of the g.c.d.'s, we see that h divides h' and vice versa, hence, there exist $a, b \in R$ such that $h = ah' = abh$. By the unique factorization property, the product ab is a unit. Hence, a, b are units, too, showing that h and h' coincide up to multiplication by a unit.

It is conventional to impose an additional condition to make the g.c.d. of two elements $f, g \in R$ unique. We also speak about the **normal form of a g.c.d.** and write $\gcd(f, g)$ for it:

- For $R = \mathbb{Z}$, we impose the condition $\gcd(f, g) \geq 0$.
- If R is a field, $\gcd(f, g) := 1$ (unless $f = g = 0$).
- For $R = K[\boldsymbol{x}]$, the additional condition is that $\gcd(f, g)$ is monic, if K is a field, respectively that $\operatorname{lc}\big(\gcd(f, g)\big)$ is non-negative, if $K = \mathbb{Z}$.

Note that the definition of a g.c.d. implies that $\gcd(0, 0) = 0$.

Throughout this course, we will make the following assumption:

Convention 1.1.14. All **UFD**s R are equipped with a "normal form", that is, for each $f, g \in R$, there is a well-defined $\gcd(f, g)$.

Definition 1.1.15. Let R be a UFD, and let $f_1, \ldots, f_r \in R$. Then

(1) The greatest common divisor of $f_1,\dots,f_r$ is defined recursively, by setting

$$\gcd(f_1,\dots,f_{k+1}) := \gcd\bigl(\gcd(f_1,\dots,f_k),f_{k+1}\bigr)$$

for all $k=1,\dots,r-1$. It is not difficult to show that the (final) result of this recursion is independent of the numbering of the elements $f_1,\dots,f_r$.

(2) $f_1,\dots,f_r$ are called **pairwise coprime** (or **pairwise relatively prime**) if $\gcd(f_i,f_j)=1$ for all $1\le i,j\le r$ with $i\neq j$.

A key ingredient to many of the algorithms in the subsequent sections of this course is a so-called **lifting step**, which allows results obtained over finite fields to be lifted to results over the rationals ("**modular approach**"). The most common methods to achieve such a lifting are Hensel's Lemma, which will be discussed in Section 2.4, and the Chinese Remainder Theorem, which we briefly recall here:

Theorem 1.1.16 (Chinese Remainder Theorem). *Let $R=\mathbb{Z}$ or $R=K[x]$, K a field. Moreover, let $f_1,\dots,f_r\in R$ be pairwise coprime. Then*

$$\begin{array}{rcl} \boldsymbol{c}:\ R/\langle f_1\cdot\ldots\cdot f_r\rangle & \xrightarrow{\cong} & \displaystyle\bigoplus_{i=1}^{r} R/\langle f_i\rangle \\ g & \longmapsto & (g \bmod f_1,\dots,g \bmod f_r) \end{array}$$

is an isomorphism of rings.

Exercises

Exercise 1.1.1. (a) Prove that $>_{\mathtt{dp}}$ is a monomial ordering on $K[x_1,\dots,x_n]$.

(b) Give a sketch of those elements of $\mathbb{N}^2$ which correspond to monomials in $K[x,y]$ being

- $>_{\mathtt{lp}} x^3y^4$,
- $>_{\mathtt{Dp}} x^3y^4$,
- $>_{\mathtt{dp}} x^3y^4$,

respectively.

Exercise 1.1.2 (Weight Orderings). (a) Let $L:\mathbb{R}^n\to\mathbb{R}$, $\boldsymbol{x}\mapsto\sum_{i=1}^n c_ix_i$ be a linear form with positive coefficients $c_1,\dots,c_n\in\mathbb{R}_{>0}$. Prove that

$$\underline{x}^\alpha >_L \underline{x}^\beta \ :\Longleftrightarrow\ L(\alpha)>L(\beta)$$

defines a monomial ordering on $K[x_1,\dots,x_n]$ iff $c_1,\dots,c_n$ are $\mathbb{Q}$-linearly independent.

The following part requires some knowledge in convex analysis (convex hull, separation theorems...):

(b*) Let F be a finite set of monomials in $K[x_1, \dots, x_n]$ and $>$ a monomial ordering on F. Prove that there exists a non-empty, open subset $U \subset \mathrm{Hom}(\mathbb{R}^n, \mathbb{R})$ of linear forms such that for any $L \in U$ and any $\boldsymbol{x^\alpha}, \boldsymbol{x^\beta} \in F$

$$\boldsymbol{x^\alpha} > \boldsymbol{x^\beta} \iff \boldsymbol{x^\alpha} >_L \boldsymbol{x^\beta} \;:\Longleftrightarrow\; L(\alpha) > L(\beta)\,.$$

Hint: Consider the convex hull of the set $\{\boldsymbol{\alpha} - \boldsymbol{\beta} \in \mathbb{Z}^n \mid \boldsymbol{x^\alpha}, \boldsymbol{x^\beta} \in F,\ \boldsymbol{x^\alpha} \leq \boldsymbol{x^\beta}\}$, and prove that $0 \in \mathbb{R}^n$ is one of the corners.

Exercise 1.1.3. Give a proof for Facts 1.1.9 (1) and (3).

Exercise 1.1.4. Give a proof for Eisenstein's Criterion 1.1.10.

Exercise 1.1.5. Let $K = \mathbb{F}_5 = \mathbb{Z}/5\mathbb{Z}$. Find a polynomial in $\mathbb{F}_5$ of degree at most 2 such that $f(0) = 1$, $f(1) = 1$ and $f(2) = 4$. How many polynomials in $\mathbb{F}_5[x]$ satisfy these properties? Answer the same question for polynomials of degree at most 3.

Hint. Chinese Remainder Theorem.

1.2 Finite Fields

The use of a modular approach is crucial to many of the algorithms we want to consider. Thus, large parts of the algorithms are dealing with problems over finite fields and make use of important properties thereof, which we briefly recall here. Again proofs of the statements and examples illustrating the definitions can be found in introductory courses (or textbooks such as [RSV, Lan]) on algebra:

Definition 1.2.1. Let $p \in \mathbb{Z}_{>0}$ be a prime. Then we denote the finite field $\mathbb{Z}/p\mathbb{Z}$ by $\mathbb{F}_p$ and refer to it as a **prime field**. More generally, for $r \in \mathbb{Z}_{>0}$ and $q = p^r$, we define

$$\mathbb{F}_q := (\mathbb{Z}/p\mathbb{Z})[x]/\langle f \rangle$$

with $f \in (\mathbb{Z}/p\mathbb{Z})[x]$ an irreducible polynomial of degree r. $\mathbb{F}_q$ is a field, referred to as a **Galois field**.

Example 1.2.2. As a set,

$$\mathbb{F}_4 = \mathbb{F}_2[x]/\langle x^2+x+1 \rangle = \{0,\, 1,\, x,\, x+1\}\,,$$

with $x^2 = x + 1$. Indeed, it is not difficult to see that the addition and multiplication of polynomials in $\mathbb{F}_2[x]$ induce the structure of a field on $\mathbb{F}_4$. In the same way, we get the structure of a field for

$$\mathbb{F}_8 = \mathbb{F}_2[x]/\langle x^3+x+1 \rangle = \{0,\, 1,\, x,\, x+1,\, x^2,\, x^2+1,\, x^2+x,\, x^2+x+1\}$$

with $x^3 = x + 1$.

Remark 1.2.3. (1) $\#(\mathbb{F}_q) = q$.

(2) For different irreducible polynomials $f \in (\mathbb{Z}/p\mathbb{Z})[x]$ of the same degree, we obtain (non-canonically) isomorphic fields $(\mathbb{Z}/p\mathbb{Z})[x]/\langle f \rangle$.

(3) It is important not to confuse $\mathbb{F}_{p^r}$ with $\mathbb{Z}/p^r\mathbb{Z}$ for $r \neq 1$. The latter is not an integral domain, because p is a zero-divisor in this ring!

(4) The minimal number of times the multiplicative identity element of a ring R has to be added to itself to get 0, is called the **characteristic** of R (denoted by $\mathrm{char}(R)$). In particular, the characteristic of $\mathbb{F}_{p^r}$ is p. If the sum can never lead to zero, the characteristic is defined to be zero, for example $\mathrm{char}(\mathbb{Q}) = 0$.

(5) A field $\mathbb{F}_q$ can only be embedded as a subfield in a field $\mathbb{F}_{q'}$ if the characteristic of $\mathbb{F}_q$ and $\mathbb{F}_{q'}$ coincide. Moreover, $\mathbb{F}_{p^r}$ can be embedded as a subfield in $\mathbb{F}_{p^s}$ iff r divides s. For instance, $\mathbb{F}_4$ can be embedded as a subfield in $\mathbb{F}_{16}$, but not in $\mathbb{F}_8$.

Theorem 1.2.4 (Fermat's Little Theorem[4]). *Each* $a \in \mathbb{F}_q$ *satisfies* $a^q = a$. *Hence*[5], *in* $\mathbb{F}_q[x]$ *the polynomial* $x^q - x$ *has the irreducible factorization*

$$x^q - x = \prod_{a \in \mathbb{F}_q} (x - a)\,.$$

Lemma 1.2.5. *For all* $a, b \in \mathbb{F}_q = \mathbb{F}_{p^r}$, *we have*

(1) $a^p + b^p = (a + b)^p$,

(2) $a^p = a$ *iff* $a \in \mathbb{F}_p \subset \mathbb{F}_q$.

Definition 1.2.6. Let $a \in \mathbb{F}_q$. Then we call $\operatorname{ord}(a) := \min\{s \in \mathbb{Z}_{\geq 1} \mid a^s = 1\}$ the **order** of a. If $\operatorname{ord}(a) = q - 1$, then a is called a **primitive element** of $\mathbb{F}_q$.

Remark 1.2.7. An element $a \in \mathbb{F}_q$ is primitive iff it generates $\mathbb{F}_q^*$, the multiplicative group of $\mathbb{F}_q$.

Example 1.2.8. The primitive elements of $\mathbb{F}_4 = \big\{0, 1, x, x + 1\big\}$ are precisely x and $x + 1$. The primitive elements of $\mathbb{F}_8$ are precisely x, $x+1$, x^2, x^2+1, x^2+x, x^2+x+1.

Exercises

Exercise 1.2.1. Show that x^2+x+1 is the unique irreducible polynomial of degree 2 in $\mathbb{F}_2[x]$.

Exercise 1.2.2. Prove that $\mathbb{F}_4$ cannot be embedded as a subfield in $\mathbb{F}_8$ (without using Remark 1.2.3 (5)).

Exercise 1.2.3. Give a description of all elements of $\mathbb{F}_{16}$, and show that $\mathbb{F}_4$ can be embedded as a subfield in $\mathbb{F}_{16}$ (again, without using Remark 1.2.3 (5)).

Exercise 1.2.4. Prove Lemma 1.2.5.

Exercise 1.2.5. Find all primitive elements in $\mathbb{F}_9$.

---

[4] Do not confuse Fermat's Little Theorem, which can be proved with elementary techniques, with Fermat's Last Theorem, which had been an open problem for more than three centuries, until it was proved by A. Wiles in the 1990s using very deep theoretical methods.

[5] Recall that, for a univariate polynomial $f \in K[x]$ and $a \in K$, we have $f(a) = 0$ iff $x - a$ divides f in $K[x]$.

Chapter 2

Factorization

Decomposition of a polynomial into factors is one of the most frequent subproblems in algebraic computations. A complete answer to such a factorization problem in a UFD is, of course, the decomposition into irreducible factors. But this is an expensive computational task and in some cases, weaker decompositions are sufficient to deal with a given problem.

To relate the factorization problem to the main theme of this course, solving of systems of polynomial equations, let us consider some examples:

Solutions of a Univariate Polynomial Equation.

Let $f \in K[x] \setminus \{0\}$. Then $a \in K$ is a solution of the equation $f(x) = 0$ iff $x - a$ divides f. In particular, the question whether there is a solution in K is equivalent to the question whether there is an irreducible factor of degree 1 of f in $K[x]$. We will see below that we can answer this question on the existence of a solution without computing a complete factorization into irreducible factors (see Section 2.3.2).

If K is an **algebraically closed field**[1], by definition, all irreducible polynomials are of degree one. Then, finding all solutions of $f(x) = 0$ is equivalent to a decomposition into irreducible factors.

If K is a field, but not necessarily algebraically closed[2], each irreducible factor $f_i \in K[a]$ of f with $d_i := \deg(f_i) \geq 2$ defines a finite algebraic field extension

$$K \subset K_i := K[a]/\langle f_i \rangle$$

such that $f(x) = 0$ has at least d_i different solutions in $K_i \setminus K$. Proceeding by induction, this enables us to compute a finite algebraic field extension $K \subset K'$ such that the given polynomial f decomposes into linear factors in $K'[x]$.

---

[1] The fundamental theorem of algebra says that the field C of complex numbers has this property.

[2] Note that, in practice, we can basically compute only over finite fields, over $\mathbb{Q}$, or over finite algebraic extensions thereof.

Solutions of a System of Univariate Polynomial Equations

Let $f_1, \dots, f_r \in K[x] \setminus \{0\}$. As $K[x]$ is a principal ideal domain, the ideal generated by $f_1, \dots, f_r$ can be generated by a single element, $\gcd(f_1, \dots, f_r)$. Hence, $x \in K$ is a solution to the system of polynomial equations

$$f_1(x) = \cdots = f_r(x) = 0$$

iff

$$\gcd(f_1, \dots, f_r)(x) = 0\,.$$

In this way, a gcd-computation in $K[x]$ reduces the problem of finding solutions of systems of univariate polynomial equations to the problem of finding solutions of a single polynomial equation.

Solutions of a System of Multivariate Polynomial Equations

Let $f_1, \dots, f_r \in K[x_1, \dots, x_n]$, and suppose that a decomposition of f_1 into (not necessarily irreducible) factors is known, say $f_1 = \prod_{i=1}^{s} f_{1,i}$. Then the set of solutions of the system of equations

$$f_1(\boldsymbol{x}) = \cdots = f_r(\boldsymbol{x}) = 0$$

is precisely the union of the sets of solutions of the systems

$$f_{1,i}(\boldsymbol{x}) = f_2(\boldsymbol{x}) = \cdots = f_r(\boldsymbol{x}) = 0, \quad i = 1, \dots, s\,,$$

that is, we may use the decomposition of f_1 to decompose the system of equations into several simpler systems of equations from which we can obtain the set of solutions.

Remark 2.0.1. Factorization is one of the tasks in computer algebra, where there is *no analogy between solving the problem for integers and solving it for polynomials.* Modern algorithms can attack factorization problems for polynomials (over $\mathbb{F}_2$) of degree up to ~ 200.000, whereas they can only attack factorization of integers up to ~ 150 digits[3].

From the point of view of complexity theory the problem of factorizing polynomials has never been considered to be a hard problem, because it has been known for a long time that it belongs to the complexity class 'polynomial time'. For factorization of integers, no solution in the class 'polynomial time' is known and, hence, the problem is considered to be a hard problem. These hard problems are precisely the ones which are needed for cryptographic applications[4].

---

[3] With the development of faster computers, these numbers may have increased a bit by now.

[4] Other problems used for cryptographic applications involve the discrete logarithm problem or elliptic curves. For a detailed treatment of these questions, we refer the reader to a textbook on cryptography such as [Buc, BSW].

In this chapter, the key ideas of a modular approach to factorization of (univariate and multivariate) polynomials over the rationals will be developed. To this end, we will first study univariate factorization over finite fields and, subsequently, consider a lifting method, the so-called Hensel lifting, which allows us to pass from a solution over a finite field to a solution over the rationals. At the end of the chapter, we will also briefly consider multivariate factorization. The relationship between the different factorization tasks considered here is sketched in the following diagram:

$$\begin{tikzcd}
 & & \mathbb{F}_q[x] \quad \text{(basic task)} \arrow[lld] \arrow[d] \arrow[rrd] & & \\
\mathbb{Z}/m\mathbb{Z}[x] & & \mathbb{Q}[x] \arrow[ld, dashed] \arrow[rd] & & \mathbb{F}_q[x_1, \ldots, x_n] \\
 & \mathbb{Q}(\alpha)[x] \arrow[d, dashed] & & \mathbb{Q}[x_1, \ldots, x_n] & \\
 & \mathbb{Q}(a)[x_1, \ldots, x_n] & & &
\end{tikzcd}$$

We shall not discuss ("absolute") factorization algorithms for $\overline{\mathbb{Q}}[x_1, \ldots, x_n]$, $\overline{\mathbb{Q}}$ denoting the algebraic closure[5] of $\mathbb{Q}$ (cf. [Win, Gao]).

Before we can start studying univariate factorization over finite fields, we have to consider some basic tools that will be used in the algorithms discussed below. In particular, this concerns gcd computations.

2.1 The Euclidean Algorithm

The Euclidean Algorithm is one of the oldest mathematical algorithms, dating back to Euclid's "Elements" (300 B.C.); a first application to polynomials dates back to the 16th century. Nowadays, the Euclidean Algorithm can be found in the core of a large number of algorithms - in particular, gcd computations in characteristic $p \neq 0$ are an essential tool in most of the algorithms presented in this course.

The classical Euclidean Algorithm can be used to compute a g.c.d. for two given elements of a **Euclidean domain**:

Definition 2.1.1. An integral domain R equipped with a function

$$d : R \to \mathbb{N} \cup \{-\infty\}$$

---

[5] The **algebraic closure** of a field K is the smallest extension field $\overline{K} \supset K$ which is algebraically closed, that is, such that each univariate polynomial in $K[x]$ decomposes into linear factors over $\overline{K}$. The algebraic closure of K is uniquely determined up to isomorphism, and each element $a \in \overline{K}$ is **algebraic over** K, that is, satisfies $f(a) = 0$ for some polynomial $f \in K[x]$. In particular, $\mathbb{C}$ is the algebraic closure of $\mathbb{R}$, and it contains the algebraic closure of $\mathbb{Q}$ as a strict subset ($\overline{\mathbb{Q}}$ has only countably many elements). Note that the algebraic closure of a finite field has infinitely many elements.

is called a **Euclidean domain** if for all $f, g \in R$, $g \neq 0$, there exist $q, r \in R$ such that

$$f = q \cdot g + r\,, \quad d(r) < d(g)\,. \tag{2.1.1}$$

Examples 2.1.2. The following are Euclidean domains:

(1) $R = \mathbb{Z}$ with $d(a) = |a|$ for all $a \in \mathbb{Z}$.

(2) $R = K[x]$, K a field, with $d(f) = \deg(f)$ for all $f \in K[x]$.

(3) $R = \mathbb{Z}[i] = \mathbb{Z}[x]/\langle x^2+1\rangle$, with $d(a + ib) = a^2 + b^2$ (see Exercise 2.1.3).

(4) $R = K$ a field, with $d(0) = 0$ and $d(a) = 1$ for $a \neq 0$.

Remark 2.1.3 (Division with Remainder). We also refer to (2.1.1) as ***g* divides *f* with remainder *r***. If $R = K[x]$, K a field, with $d(f) = \deg(f)$, the remainder r and the factor q are uniquely determined (see the proof of Lemma 2.1.6, below). We then use the notation

$$r =: f \ \text{rem}\ g\,, \qquad q =: f \ \text{quo}\ g\ . \tag{2.1.2}$$

The same notation is used in other Euclidean domains, even if, a priori, neither the factor q, nor the remainder r are necessarily uniquely determined. For instance, for $R = \mathbb{Z}$ and $d(a) = |a|$, 3 divides 5 with remainder 2, or with remainder -1:

$$5 = 1 \cdot 3 + 2\,, \quad 5 = 2 \cdot 3 - 1\,.$$

When writing "f rem g" in this section, we usually mean "any" remainder r of f on division by g. If, however, the need arises that the notation (2.1.2) should be given a precise meaning in a given Eucidean domain, we may impose additional conditions on the remainder r. For instance, for $R = \mathbb{Z}$ and $d(a) = |a|$, it suffices to require that the remainder is non-negative: If r_1, r_2 are two non-negative remainders of the division of f by g, then $|r_1 - r_2| = |g| \cdot |q_1 - q_2|$ with $|r_1 - r_2| \leq |r_1| < |g|$, which is only possible if $r_1 = r_2$, $q_1 = q_2$.

Algorithm 2.1.4 (Classical Euclidean Algorithm).

```
Assume: $(R, d)$ is a Euclidean domain.
Input:  $f, g \in R$
Output: a g.c.d. of $f, g$ in $R$

(1) $r_0 := f,\ r_1 := g,\ i := 1$        // Initialization
(2) WHILE $(r_i \neq 0)$ DO               // Main Loop
      • $r_{i+1} := (r_{i-1}\ \text{rem}\ r_i)$
```

```
    • $i := i + 1$
(3) RETURN ($r_{i-1}$).
```

Proof. Step 1 (Termination). Since $d(r_{k+1}) < d(r_k)$ by the definition of the remainder, and since there are only $d(g)$ possible values for d below $d(g) \in \mathbb{N} \cup \{-\infty\}$, the process terminates after at most $d(g)$ steps.

Step 2 (Correctness). By construction, the result r_{i-1} is a divisor of r_{i-2} and r_{i-3}. By iteration, we get that it is a common divisor of all r_k, $0 \leq k \leq i-1$. In particular, it is a common divisor of f and g. On the other hand, let h be a common divisor of $r_0 = f$ and $r_1 = g$. Then h is also a divisor of r_2 (which is an R-linear combination of r_0 and r_1). By iteration, the same holds for all r_k, $0 \leq k \leq i-1$; in particular, h divides the result r_{i-1} returned by the algorithm. □

Remark 2.1.5. (1) The classical Euclidean Algorithm can be modified in such a way that the normal form of the g.c.d. is returned. To this end, only the first assignment in the WHILE loop has to be modified:

$$r_{i+1} := \mathrm{normal}\,(r_{i-1} \text{ rem } r_i).$$

(2) The g.c.d. of two elements of a Euclidean domain R can be expressed as an R-linear combination of the two elements. More precisely, there exists a variant of the Euclidean algorithm, the so-called **extended Euclidean Algorithm (EEA)** which computes $s_i, t_i \in R$ such that

$$s_k g + t_k f = r_k \text{ for all } k = 1, \ldots, i-1\,.$$

(See also Exercise 2.1.2.)

Note that for $n \geq 2$ (or for $K = \mathbb{Z}$ and $n = 1$), the polynomial ring $K[x_1, \ldots, x_n]$ with the degree function is *not* a Euclidean domain. To obtain an algorithm for computing g.c.d.s in $K[x_1, \ldots, x_n] = (K[x_1, \ldots, x_{n-1}])[x_n]$, or, more generally, in $R[x]$, R any integral domain, we have to modify the requirements on the division with remainder slightly. More precisely, the problem arising from division of a polynomial $f \in R[x]$ by a polynomial $g \in R[x]$ is that the leading coefficient of g may not be a unit. Thus, f is not necessarily divisible by this leading coefficient. This problem can be avoided when multiplying f by (a suitable power of) the leading coefficient of g:

Lemma 2.1.6 (Pseudo Division with Remainder). *Let R be an integral domain, $f, g \in R[x]$ with $g \neq 0$. Then there exist unique polynomials $q, r \in R[x]$ such that*

$$\mathrm{lc}(g)^{\delta} \cdot f = q \cdot g + r\,, \quad \deg(r) < \deg(g)\,,$$

where $\delta := \max\bigl\{0,\ \deg(f) - \deg(g) + 1\bigr\}$.

Proof. Step 1 (Existence). Set $\delta_0 := \deg(f) - \deg(g) + 1$, and proceed by induction on δ_0. If $\delta_0 \leq 0$, then the degree of g exceeds the degree of f, and no division occurs, that is, we trivially have $r = f$, $q = 0$.

Now, let $\delta_0 \geq 1$, and consider

$$\widetilde{f} := \mathrm{lc}(g) \cdot f - \mathrm{lc}(f) \cdot x^{\delta_0 - 1} \cdot g\,. \tag{2.1.3}$$

In the expression on the right-hand side, the leading terms of the two summands cancel. Therefore, $\deg(\widetilde{f}) < \deg(f)$. By the induction hypothesis, there exist $\widetilde{q}, \widetilde{r} \in R[x]$ such that

$$\mathrm{lc}(g)^{\widetilde{\delta}} \cdot \widetilde{f} = \widetilde{q}g + \widetilde{r}\,, \quad \deg(\widetilde{r}) < \deg(g)\,, \quad \widetilde{\delta} = \max\Big\{0, \deg\big(\widetilde{f}\,\big) - \deg(g) + 1\Big\}\,.$$

Together with the equality (2.1.3), this implies

$$\mathrm{lc}(g)^{\widetilde{\delta}+1} \cdot f = \Big(\mathrm{lc}(f) \cdot x^{\delta_0 - 1} \cdot \mathrm{lc}(g)^{\widetilde{\delta}} + \widetilde{q}\Big) \cdot g + \widetilde{r}. \tag{2.1.4}$$

Since $\deg\big(\widetilde{f}\,\big) \leq \deg(f) - 1$, we have $m := \delta_0 - \big(\widetilde{\delta} + 1\big) \geq 0$. Multiplying both sides of (2.1.4) by $\mathrm{lc}(g)^m$ completes the proof of existence.

Step 2 (Uniqueness). Let us assume that

$$\mathrm{lc}(g)^{\delta} \cdot f = qg + r = \tilde{q}g + \tilde{r} \tag{2.1.5}$$

are two expressions satisfying the conditions of the statement of this lemma. Then

$$(q - \tilde{q}) \cdot g = \tilde{r} - r\,, \tag{2.1.6}$$

and the degree of r and $\tilde{r}$ are strictly smaller than the one of g. Thus, the degree of the difference $\tilde{r} - r$ is strictly smaller than the one of g. For an integral domain R, this is only possible if $q - \tilde{q} = 0$, since otherwise the leading term of $\tilde{r} - r$ would be the (non-zero) product of the leading terms of $q - \tilde{q}$ and of g, and this product has degree at least $\deg(g)$. It follows that both sides of (2.1.6) vanish, that is, the two expressions in (2.1.5) are identical. $\square$

Definition 2.1.7. Let R be a UFD (with "normal form"), $f = \sum_{i=0}^{d} a_i x^i \in R[x] \setminus \{0\}$. Then we define the **content** of f as

$$\mathrm{cont}(f) := \gcd(a_0, \ldots, a_d) \in R\,,$$

and we call

$$\mathrm{pp}(f) := \frac{f}{\mathrm{cont}(f)} \in R[x]$$

the **primitive part** of f. We call f a **primitive** polynomial if it coincides with its primitive part.

Example 2.1.8. Let $f = 15x^4 + 10x^2 + 5x - 20 \in \mathbb{Z}[x]$. Then

$$\operatorname{cont}(f) = \gcd(15, 10, 5, -20) = 5\,, \quad \operatorname{pp}(f) = 3x^4 + 2x^2 + x - 4\,.$$

Remark 2.1.9. Let R be a UFD. A key ingredient of the proof of the Lemma of Gauß (see Facts 1.1.9) is to show that

$$\operatorname{cont}(f \cdot g) = \operatorname{cont}(f) \cdot \operatorname{cont}(g)$$

for all $f, g \in R[x] \setminus \{0\}$. Indeed, the content of a polynomial $f \in R[x] \setminus \{0\}$, f not a unit in $R[x]$, is the product of those of its prime factors which are elements of R. It follows that $\operatorname{pp}(f \cdot g) = \operatorname{pp}(f) \cdot \operatorname{pp}(g)$ for all $f, g \in R[x] \setminus \{0\}$.

Algorithm 2.1.10 (Primitive Euclidean Algorithm, PEA).

Assume: R is a UFD with “normal form”.

Input: primitive polynomials $f, g \in R[x]$ satisfying $\deg f \geq \deg g$.

Output: $\gcd(f, g) \in R[x]$ in normal form.

```
(1) $r_0 := f\,,\ n_0 := \deg(f)\,,$                          // Initialization
    $r_1 := g\,,\ n_1 := \deg(g)\,,$
    $i := 1$
(2) WHILE ($r_i \neq 0$) DO                                     // Main Loop
      • $r_{i+1} := \operatorname{pp}\big(\operatorname{lc}(r_i)^{n_{i-1}-n_i+1} \cdot r_{i-1} \text{ rem } r_i\big)$   // Pseudo Division
      • $n_{i+1} := \deg r_{i+1}$
      • $i := i + 1$
(3) RETURN $\big(\operatorname{normal}(r_{i-1})\big)$.
```

Example 2.1.11. Consider the following two polynomials in $\mathbb{Z}[x]$:

$$\begin{aligned} f &:= x^8 + 2x^7 + 4x^6 - 2x^5 - 3x^4 + 4x^3 + 9x^2 - 2x - 5\,, \\ g &:= 5x^5 - 2x^4 - 13x^3 - 11x^2 + 7x + 6\,. \end{aligned}$$

We compute $\gcd(f, g)$ by means of the primitive Euclidean Algorithm:

$$\begin{aligned} r_2 &:= \operatorname{pp}\big(625f \text{ rem } g\big) = \operatorname{pp}\big(15201x^4 + 25424x^3 + 10223x^2 - 15201x - 10223\big) \\ &= 15201x^4 + 25424x^3 + 10223x^2 - 15201x - 10223\,, \\ r_3 &:= \operatorname{pp}\big(15201^2 g \text{ rem } r_2\big) = \operatorname{pp}\big(223925000x^3 + 223925000x^2 - 223925000\big) \\ &= x^3 + x^2 - 1\,, \\ r_4 &:= \operatorname{pp}\big(r_2 \text{ rem } r_3\big) = 0\,. \end{aligned}$$

Hence, $\gcd(f, g) = x^3 + x^2 - 1$.

Remark 2.1.12. Obviously, using pseudo division with remainder instead of division with remainder in the classical Euclidean algorithm introduces new factors from R to the intermediate polynomial in each pass through the loop. Hence, these polynomials tend to have huge coefficients. To avoid this rapid growth of the coefficients, in the primitive Euclidean algorithm the pseudo division is followed by taking the primitive part – that is, dividing out the content. But, of course, determining the content of a given polynomial can also be quite expensive, as this is again a gcd computation, this time in R.

As this is a trade-off between large coefficients and gcd computations slowing down the algorithm, intermediate strategies have been developed like the **reduced polynomial remainder sequence** or the **subresultant polynomial remainder sequence** (see, e.g., [Coh, Section 3.3]). In both of these variants, not the complete content is divided out, but just a factor of it, which can be determined without using gcd computations.

Even with the intermediate strategies, however, the direct approaches to gcd computations are still inferior to the modular approaches in characteristic zero. The first crucial step in a modular approach is the choice of suitable primes. An important *theoretical tool* for this step will be the notion of a **resultant**, which is introduced in the following section. Even though it does not enter the algorithm explicitly, it is the key tool for proving the criteria for the choice of the primes. Its use for a modular gcd is discussed in Exercises 2.2.2, 2.2.3, 2.2.4, 2.2.5.

Exercises

Exercise 2.1.1. Compute the following g.c.d.s:

(a) $\gcd(8771, 3206)$ in $\mathbb{Z}$.

(b) $\gcd(x^5 - x^3 + x^2 - 2x - 2,\ x^4 + x^3 - x^2 - 2x - 2)$ in $\mathbb{Q}[x]$.

(c) $\gcd(x^5 + x^4 + x^3 + x + 1,\ x^4 + x^3 + x^2 + x + 1)$ in $\mathbb{F}_5[x]$.

Exercise 2.1.2. (a) Formulate the extended Euclidean Algorithm explicitly.

(b) Use the extended Euclidean Algorithm to compute the inverses of all non-zero elements of $\mathbb{F}_8 = \mathbb{F}_2[x]/\langle x^3 + x + 1\rangle$ explicitly.

Exercise 2.1.3. Show that the ring of Gaussian integers $\mathbb{Z}[i] \subset \mathbb{C}$, equipped with the function $d : \mathbb{Z}[i] \to \mathbb{N}$, $a + ib \mapsto a^2 + b^2$, is a Euclidean domain. Moreover show that $a + ib$ is a unit of $\mathbb{Z}[i]$ iff $a^2 + b^2 = 1$. Finally, compute

- all g.c.d.s of 6 and $3 + i$ in $\mathbb{Z}[i]$.
- a g.c.d. of 26496 and $289 - 63i$.

Exercise 2.1.4 (Sturm's Theorem). Let $f \in \mathbb{R}[x]$ satisfy $\gcd(f, f') = 1$ (which means that f has no multiple roots, see Proposition 2.2.4 below). Then we define the sequence of polynomials $(f_k)_{k \in \mathbb{N}}$ by setting

$$f_0 := f,\ f_1 := -f',\ \ldots, f_{k+1} := -(f_{k-1} \text{ rem } f_k) \text{ for all } k \geq 1\,.$$

If ℓ is minimal such that $f_{\ell+1} = 0$ (convince yourself that such an ℓ exists), we say that $f_0, f_1, \ldots, f_\ell$ form the **Sturm chain** of f. For each $a \in \mathbb{R}$, we define

$$\sigma(a) := \#\left\{k \in \{1, \ldots, \ell\} \,\middle|\, f_k(a) \cdot f_{k-1}(a) \leq 0\right\}\,.$$

Prove the following statements:

(a) f_ℓ has no real roots.

(b) If, for $a \in \mathbb{R}$ and $1 \leq k < \ell$, we have $f_k(a) = 0$, then $f_{k-1}(a) \cdot f_{k+1}(a) < 0$.

(c) $\sigma : \mathbb{R} \setminus f^{-1}(\{0\}) \to \mathbb{N},\ a \mapsto \sigma(a),$ is a continuous function.

(d) If $f(a) = 0$, and if $\varepsilon > 0$ is sufficiently small, then $\sigma(a - \varepsilon) - \sigma(a + \varepsilon) = 1$.

Conclude Sturm's Theorem: Let $b < c \in \mathbb{R}$ with $f(b)f(c) \neq 0$, then the number of real roots of f in the open interval (b, c) equals $\sigma(b) - \sigma(c)$.

2.2 Resultants

The notion of the resultant is the outcome of an attempt to answer the question whether the gcd of two primitive polynomials in $K[x]$ is trivial or not by applying methods from linear algebra. It gains its importance (in the context of gcd computations) as a purely conceptual tool which enters the analysis of the algorithms, but not the algorithms themselves.

The key observation is the following: Given a UFD R (with normal form) and primitive polynomials $f, g \in R[x] \setminus \{0\}$, then $h := \gcd(f, g) \neq 1$ iff there exist $s, t \in R[x] \setminus \{0\}$ such that $sf + tg = 0$, where $\deg s < \deg g =: e$ and $\deg t < \deg f =: d$. This is due to the fact that $\frac{g}{h} \cdot f - \frac{f}{h} \cdot g = 0$ is a relation of smallest degree, since $\frac{f}{h}$ and $\frac{g}{h}$ are coprime.

In other words, $\gcd(f, g) = 1$ iff the linear map of finitely generated, free R-modules

$$\begin{tikzcd} \Phi : R[x]_{<e} \times R[x]_{<d} \arrow[r] & R[x]_{<d+e} \\ (s, t) \arrow[r, mapsto] & sf + tg \end{tikzcd}$$

is injective. Here $R[x]_{<k}$ denotes the R-module of polynomials of degree at most $k - 1$.

As we know from linear algebra, we may represent this linear map by a matrix with entries in R. To this end, we fix bases for the free R-modules:

$$\begin{array}{lcl} R\text{-basis of } R[x]_{<k} & : & \big(x^{k-1}, \ldots, x^1, 1\big) \\ R\text{-basis of } R[x]_{<e} \times R[x]_{<d} & : & \left(\begin{pmatrix} x^{e-1} \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ x^{d-1} \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right), \end{array}$$

and express f and g accordingly, that is,

$$f = \sum_{i=0}^{d} a_i x^i\,, \quad g = \sum_{j=0}^{e} b_j x^j\,,$$

where $a_d, b_e \neq 0$. Then the matrix of Φ is

$$S_{f,g} := \begin{array}{c} \left(\begin{array}{cccc|ccc} a_d & 0 & \cdots & 0 & b_e & & \mathbf{0} \\ \vdots & a_d & \ddots & 0 & \vdots & \ddots & b_e \\ a_0 & \vdots & \ddots & a_d & \vdots & & \vdots \\ 0 & a_0 & \ddots & \vdots & b_0 & \ddots & \\ 0 & \cdots & 0 & a_0 & & \mathbf{0} & b_0 \end{array}\right) \\ \underbrace{\qquad\qquad\qquad\qquad}_{\text{e columns}}\underbrace{\qquad\qquad\quad}_{\text{d columns}} \end{array} \in \mathrm{Mat}(d+e, R)\,.$$

Definition 2.2.1. Let R be a UFD, and let $f, g \in R[x] \setminus \{0\}$, with $\deg(f \cdot g) > 0$. Then $S_{f,g}$ is called the **Sylvester matrix**[6] of f and g, and

$$\mathrm{Res}_{f,g} := \det(S_{f,g})$$

is called the **resultant** of f and g. Moreover, we assign the following values to the resultant:

$$\begin{aligned} \mathrm{Res}_{0,g} &:= 0 \quad \text{for all } g \in (R[x] \setminus R) \cup \{0\}\,, \\ \mathrm{Res}_{a,b} &:= 1 \quad \text{for all } a, b \in R \text{ such that } (a,b) \neq (0,0)\,. \end{aligned}$$

The resultant of f and its derivative f',

$$D_f := \mathrm{Res}_{f,f'}\,,$$

is called the **discriminant** of f.

Example 2.2.2. To illustrate the notion of the discriminant of a polynomial, let us consider a (general) quadratic polynomial $f = x^2 + ax + b \in \mathbb{Q}[x]$. Its first derivative is $f' = 2x + a$ and, hence, the resultant of f and f' is

$$\mathrm{Res}_{f,f'} = \det \left(\begin{array}{c|cc} 1 & 2 & 0 \\ a & a & 2 \\ b & 0 & a \end{array}\right) = a^2 + 4b - 2a^2 = 4b - a^2\,.$$

By the considerations at the beginning of this section, this shows that f is of the form $(x+c)^2$ for a suitable $c \in \mathbb{Q}$ iff $b = \frac{a^2}{4}$.

---

[6] Note that in some textbooks, the transposed matrix of $S_{f,g}$ is called the Sylvester matrix.

Lemma 2.2.3 (Properties of the Resultant). *Let R be a ring, $a \in R \setminus \{0\}$ and let $f = \sum_{i=0}^{d} a_i x^i$ and $g = \sum_{j=0}^{e} b_j x^j$, $a_d, b_e \neq 0$, be non-constant polynomials in $R[x]$. Then the following hold:*

(1) $\mathrm{Res}_{a,f} = a^d\,.$

(2) $\mathrm{Res}_{f,f} = 0\,.$

(3) $\mathrm{Res}_{f,g} = (-1)^{de} \cdot \mathrm{Res}_{g,f}\,.$

(4) $\mathrm{Res}_{af,g} = a^e \cdot \mathrm{Res}_{f,g}\,.$

(5) $\mathrm{Res}_{x^\alpha f,g} = b_0^\alpha \cdot \mathrm{Res}_{f,g}$ *(for all $\alpha \in \mathbb{Z}_{>0}$).*

(6) Let $\varphi : R \to R'$ be a ring homomorphism, and let

$$\varphi(f) := \sum_{i=0}^{d} \varphi(a_i) x^i\,, \quad \varphi(g) := \sum_{j=0}^{e} \varphi(b_j) x^j\,.$$

Moreover, assume that $\deg\big(\varphi(f)\big) = d$, $\deg\big(\varphi(g)\big) = k \leq e$. *Then*

$$\varphi(\mathrm{Res}_{f,g}) = \varphi(a_d)^{e-k} \cdot \mathrm{Res}_{\varphi(f),\varphi(g)}\,.$$

Proof. The statements (1) – (5) are direct consequences of the properties of the determinant:

(1) $S_{a,g}$ is a diagonal matrix whose entries on the diagonal are a.

(2) $S_{f,f}$ possesses (at least) two identical columns.

(3) $S_{g,f}$ can be obtained from $S_{f,g}$ by permutation of columns.

(4) $S_{af,g}$ can be obtained from $S_{f,g}$ by multiplication of the e columns of the first block by the constant a.

(5) $S_{x^\alpha f,g}$ contains an $\alpha \times e$ zero matrix in the bottom part of the first block.

Statement (6) can be proved by using the properties of the determinant and the special structure of the matrices. It is left to the reader as Exercise 2.2.2. $\square$

Proposition 2.2.4. *Let R be a UFD, $f, g \in R[x] \setminus R$. Then the following statements hold:*

(1) $\mathrm{Res}_{f,g} = 0 \Leftrightarrow$ *f and g have a common non-constant factor.*

(2) If $\mathrm{char}(R) = 0$, *then* $\big(D_f = 0 \Leftrightarrow$ *f has a multiple non-constant factor*$\big)$.

Proof. (1) follows from the considerations at the beginning of this section. More precisely, passing to the quotient field of R, we know that the map Φ is injective iff $\det(S_{f,g}) = 0$.

With respect to statement (2), we have to show that f has a multiple factor $h \in R[x] \setminus R$ iff $\gcd(f, f') \in R[x] \setminus R$: To this end, let us first assume that f has a non-constant multiple factor, say $f = h^2 \cdot \widetilde{f}$. Then the first derivative of f is

$$f' = 2h \cdot h' \cdot \widetilde{f} + h^2 \cdot \widetilde{f}' = h \cdot \left(2h'\widetilde{f} + h\widetilde{f}'\right)$$

and, hence, h divides $\gcd(f, f')$.

For the other implication, let us assume that $f \in R[x] \setminus R$ has no non-constant multiple factor, that is, $f = a \cdot f_1 \cdot \ldots f_s$ where $a \in R \setminus \{0\}$, and where $f_1, \ldots, f_s \in R[x] \setminus R$ are pairwise coprime irreducible polynomials. Then

$$f' = a \cdot \sum_{i=1}^{s} f_1 \cdot \ldots \cdot f_i' \cdot \ldots \cdot f_s \, .$$

This and the fact that the f_i are pairwise coprime imply that f_i divides f' iff f_i divides af_i'. By comparing degrees, this is impossible in characteristic 0, because $\deg f_i' = \deg f_i - 1$. □

Remark 2.2.5 (Caution !). In characteristic $p \neq 0$, the equivalence in (2) does not hold. Indeed, let $R := \mathbb{F}_p[t]$, and let $f := x^p + t \in R[x]$. Then f is irreducible in $R[x]$, as the unique factorization of f over the algebraic closure of the quotient field $\mathbb{F}_p(t)$ of R is $f = (x + a)^p$ for a a zero of $a^p - t$, and as none of the possible factors $(x + a)^k = x^k + kax^{k-1} +$ lower terms in x, $1 \leq k < p$, exists in $R[x]$. On the other hand, $f' = p \cdot x^{p-1} = 0$, hence, $D_f = \mathrm{Res}_{f,f'} = 0$.

Corollary 2.2.6. *Let K be a field, $\overline{K}$ its algebraic closure, and let f and g be polynomials in $K[x]$. Then*

(1) f and g have a common root in $\overline{K}$ iff $\mathrm{Res}_{f,g} = 0$.

(2) f has a multiple root in $\overline{K}$ iff $D_f = 0$.

Proof. Statement (1) is an immediate consequence of Proposition 2.2.4. The same holds for statement (2) in characteristic 0.

To show the equivalence in (2) for $\mathrm{char}(K) > 0$, we have to reconsider the proof of Proposition 2.2.4: the crucial point (for $\mathrm{char}(R) = p$) was to show:

It is impossible that an irreducible factor f_i of $f \in R[x] \setminus R$ divides af_i'.

If $R = \overline{K}$, each non-constant polynomial f splits into *linear* factors $f_i = x - a_i$. Hence, $af_i' \in \overline{K} \setminus \{0\}$, and f dividing af_i' would mean that f_i is a unit in $\overline{K}[x]$, which is impossible. □

Remark 2.2.7 (Computing Resultants). Usually, resultants should only be used as a theoretical tool, but if it cannot be avoided to compute one explicitly, there are different approaches to computing resultants (cf. [GCL, Section 9.5]):

Direct evaluation of the determinant of the Sylvester matrix $\det(S_{f,g})$ is certainly the most inefficient approach, as it is of exponential time. Using the *connection between g.c.d. and resultants*, there are algorithms of polynomial time, see for instance [Coh, Section 3.3]. For characteristic zero, and for the multivariate case in positive characteristic, *modular techniques* are again the best known approach, as we have already mentioned in the section on g.c.d. computations.

Exercises

Exercise 2.2.1. Use the resultant to check whether the polynomials $f = x^3 + ay^3$, $a \in \mathbb{C}$, and $x^2 + y^2 \in \mathbb{C}[x,y]$ have more than one common complex zero.

Exercise 2.2.2. Let R be a ring, $f = \sum_{i=0}^{d} a_i x^i$, $g = \sum_{j=0}^{e} b_j x^j \in R[x]$, $a_d, b_e \neq 0$. Moreover, let $\varphi : R \to R'$ be a homomorphism of rings, such that $\varphi(a_d) \neq 0$, and let

$$\varphi(f) = \sum_{i=1}^{d} \varphi(a_i)x^i\,, \quad \varphi(g) = \sum_{j=1}^{k} \varphi(b_j)x^j\,, \quad \varphi(b_k) \neq 0\,,$$

be the images of f, g under the homomorphism $R[x] \to R'[x]$ induced by φ. Prove that

$$\varphi(\mathrm{Res}_{f,g}) = \varphi(a_d)^{e-k} \cdot \mathrm{Res}_{\varphi(f),\varphi(g)}\ .$$

Exercise 2.2.3. Let R be a Euclidean domain, let $p \in R$ be a prime, and let

$$\pi : R[x] \longrightarrow R/\langle p\rangle[x]$$

denote the projection. Moreover, let $f, g \in R[x] \setminus \{0\}$ such that p does not divide $\gcd\big(\mathrm{lc}(f), \mathrm{lc}(g)\big)$. Prove:

(a) $\deg\big(\gcd(f,g)\big) \leq \deg\big(\gcd\left(\pi(f), \pi(g)\right)\big)$.

(b) The following statements are equivalent for $h := \gcd(f,g)$:

  (i) $\deg(h) = \deg\left(\gcd\left(\pi(f), \pi(g)\right)\right)$.

  (ii) $\pi\big(\mathrm{lc}(h)\big) \cdot \gcd\big(\pi(f), \pi(g)\big) = \pi\big(h\big)$.

  (iii) p does not divide the resultant of $\frac{f}{h}$ and $\frac{g}{h}$.

Exercise 2.2.4. Let $f = x^4 - 13x^3 - 62x^2 - 78x - 408$ and $g = x^3 + 6x^2 - x - 30$.

(a) Set up the Sylvester matrix of f and g and compute $\mathrm{Res}_{f,g}$.

(b) Let $p_1 = 5$, $p_2 = 7$, $p_3 = 11$ and $p_4 = 13$. Compute $h = \gcd(f,g) \in \mathbb{Q}[x]$. For which of the primes p_i is the modular image $\overline{h} \in \mathbb{F}_{p_i}[x]$ of h equal to the gcd of the images of f and g in $\mathbb{F}_{p_i}[x]$.

Exercise 2.2.5. (a) Given pairwise different positive primes $p_1, \ldots, p_r \in \mathbb{Z}$, and $a_1, \ldots, a_r \in \mathbb{Z}$. Then, by the Chinese Remainder Theorem, the system of congruence equations

$$\begin{array}{ccc} a & \equiv & a_1 \bmod p_1\,, \\ \vdots & \vdots & \vdots \\ a & \equiv & a_r \bmod p_r\,. \end{array}$$

has a solution a which is uniquely determined modulo $p_1 \cdots p_r$. Show that

$$a = \sum_{i=1}^{r} \left(\prod_{j \neq i} a_j \right) \cdot f_i\,,$$

with $f_i \in \mathbb{Z}$ satisfying

$$\left(\prod_{j \neq i} p_j \right) f_i \equiv a_i \bmod p_i\,.$$

(b) Given primitive polynomials $f, g \in \mathbb{Z}[x]$ with $\deg(f) = d \geq \deg(g) \geq 1$, and a bound A, specify a modular approach to the computation of $\gcd(f, g)$ on the basis of the Chinese remainder theorem (see also Exercise 2.2.3).

Exercise 2.2.6. Let R be an integral domain, $f, g \in R[x] \setminus \{0\}$ of degree d, e, respectively. Prove that there exist $s, t \in R[x] \setminus \{0\}$ such that

$$sf + tg = \mathrm{Res}_{f,g}\,, \quad \deg(s) < e\,, \ \deg(t) < d\,.$$

HINT. Compute $(x^{d+e-1}, \ldots, x, 1) \cdot S_{f,g}$, and use Cramer's rule to conclude.

Exercise 2.2.7.* Let R be a UFD with field of fractions $K = \mathrm{Quot}(R)$, $f, g \in R[x]$ non-zero polynomials of degrees d, e, and $\alpha_1, \ldots, \alpha_d$, $\beta_1, \ldots, \beta_e$ the roots of f and g, respectively, in an extension field of K (counted with multiplicities). Prove that

$$\mathrm{Res}_{f,g} = \mathrm{lc}(f)^e \cdot \mathrm{lc}(g)^d \cdot \prod_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} (\alpha_i - \beta_j)\,.$$

and conclude that, for any $f, g, h \in R[x]$,

$$\mathrm{Res}_{f,gh} = \mathrm{Res}_{f,g} \cdot \mathrm{Res}_{f,h}\,.$$

HINT. Prove the claim first in the case where the roots are considered as indeterminates, by computing the determinant of the product of the Sylvester matrix with a suitable Vandermonde matrix. Recall the **Vandermonde determinant**

$$\det \begin{pmatrix} x_1^{k-1} & x_2^{k-1} & \ldots & x_k^{k-1} \\ x_1^{k-2} & x_2^{k-2} & \ldots & x_k^{k-2} \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \ldots & 1 \end{pmatrix} = \prod_{1 \leq i < j \leq k} (x_i - x_j)\,.$$

2.3 Factorization in $\mathbb{F}_q[x]$

Factorization in $\mathbb{F}_q[x]$ is the basic factorization task and reappears in virtually all (practically used) factorization algorithms, since the more efficient approaches to tackling factorization problems in $\mathbb{Q}[x]$ and $\mathbb{F}_q[x_1, \ldots, x_n]$ are modular approaches which eventually reduce the problem to factorization in $\mathbb{F}_q[x]$ (plus the problem of lifting the computed factors). On the other hand, even this basic task of determining a complete decomposition into irreducible factors over a finite field is still quite hard. Therefore, most approaches do not try to compute a complete factorization in a single step, but first apply certain preprocessing steps, which decompose the polynomials into (not yet irreducible) factors possessing some nice properties like being squarefree or being a product of irreducible factors of the same degree. In this section, we first consider two types of preprocessing steps, namely the **squarefree factorization** and the **distinct degree factorization**, and then proceed to the **algorithms of Berlekamp** and **of Cantor/Zassenhaus** for the factorization of squarefree polynomials.

2.3.1 Squarefree factorization

The main goal of the squarefree decomposition is to determine factors of a given polynomial f in such a way that

- each factor does not contain multiple non-trivial factors,
- the factors are pairwise coprime, and
- f may be recovered from the decomposition as a product of suitable powers of the factors.

Definition 2.3.1 (Squarefree factorization). Let R be a UFD, and let $g \in R[x]$ be a primitive polynomial. Then g is called **squarefree**[7] if g has no non-constant multiple factor, that is, if the irreducible factorization of g is of the type $g = g_1 \cdot \ldots \cdot g_s$, where the factors g_i are pairwise coprime.

If $f \in R[x]$ is any primitive polynomial, the **squarefree factorization** of f is

$$f = \prod_{i=1}^{k} f_{(i)}^i$$

where $f_{(1)}, \ldots, f_{(k)} \in R[x] \setminus R$ are squarefree and pairwise coprime. The product $f_{(1)} \cdot \ldots \cdot f_{(k)}$ is also called the **squarefree part** of f.

Example 2.3.2. Consider the polynomial

$$f = (x+1)^2 \cdot (x-1)^2 \cdot (x^2+1) \cdot (x^2+3)^5 \cdot x^5 \in \mathbb{Q}[x]\,.$$

---

[7] g is squarefree iff, for all $h \in R[x] \setminus R$, the square h^2 does not divide g, whence the name "squarefree".

Its squarefree decomposition is:

$$f = \underbrace{(x^2+1)}_{f_{(1)}} \cdot \underbrace{(x^2-1)}_{f_{(2)}}{}^2 \cdot \underbrace{(x^3+3x)}_{f_{(5)}}{}^5 .$$

In Proposition 2.2.4, it is stated that, over a UFD R of characteristic zero, a primitive polynomial $f \in R[x]$ is squarefree iff $\gcd(f, f') \in R$, that is, iff $D_f \neq 0$. As we saw already in Remark 2.2.5, the situation is more complicated in positive characteristic, because the first derivative of a non-constant polynomial may be zero. The following proposition gives a complete description of those cases in which the derivative of a non-constant polynomial $f \in \mathbb{F}_q[x]$ vanishes. It thus provides us with the required information for giving a (sufficient and necessary) criterion for $f \in \mathbb{F}_q[x]$ being squarefree:

Proposition 2.3.3. *Let $p \in \mathbb{N}$ be a prime number, $r \geq 1$ and $q = p^r$. Moreover, let $f \in \mathbb{F}_q[x]$. Then $f' = 0$ iff there is a polynomial $g \in \mathbb{F}_q[x]$ such that $f = g^p$.*

Proof. If f is a p-th power, then its first derivative $p \cdot g^{p-1} \cdot g'$ clearly vanishes in characteristic p.

For the converse, let us assume that $f' = 0$. In this case, all exponents of the terms of f must be divisible by p, that is,

$$f = \sum_{j=0}^{m} a_j \cdot x^{jp} .$$

We define

$$g := \sum_{j=0}^{m} a_j^{p^{r-1}} \cdot x^j .$$

By Lemma 1.2.5 (1) and Fermat's Little Theorem, the p-th power of g is

$$g^p = \sum_{i=0}^{m} \left(a_j^{p^{r-1}}\right)^p \cdot x^{jp} = \sum_{i=0}^{m} a_j^{p^r} \cdot x^{jp} = \sum_{i=0}^{m} a_j \cdot x^{jp} = f. \qquad \square$$

Remark 2.3.4 (Caution!). In polynomial rings over transcendental extensions of $\mathbb{F}_p$, for instance, in $\big(\mathbb{F}_p(t)\big)[x]$, the above statement does not hold. Again, we may consider the polynomial $f = x^p + t$, which is irreducible (hence, squarefree), but which satisfies $f' = 0$ (see Remark 2.2.5).

Combining the proof of Proposition 2.2.4 and Proposition 2.3.3 we obtain the following corollary:

Corollary 2.3.5. *$f \in \mathbb{F}_q[x]$ squarefree iff $f' \neq 0$ and* $\gcd(f, f') = 1$.

Remark 2.3.6. This criterion provides us with the tools needed to identify nonconstant multiple factors. More precisely, let

$$f = \prod_{i=1}^{k} f_{(i)}^{i}$$

be the squarefree decomposition of f. Then the first derivative of f is

$$f' = \sum_{\substack{i=1 \\ p\nmid i}}^{k} \left(i \cdot f'_{(i)} \cdot f_{(i)}^{i-1} \cdot \prod_{j\neq i} f_{(j)}^{j} \right) = \underbrace{\prod_{\substack{j=1 \\ p\,|\,j}}^{k} f_{(j)}^{j} \cdot \prod_{\substack{i=2 \\ p\nmid i}}^{k} f_{(i)}^{i-1}}_{=:\, g} \cdot \sum_{\substack{i=1 \\ p\nmid i}}^{k} \underbrace{\left(i \cdot f'_{(i)} \cdot \prod_{\substack{j\neq i \\ p\nmid j}} f_{(j)} \right)}_{\neq 0},$$

by the chain rule. The polynomial g divides both f and f'.

We claim that *g is a g.c.d. of f and f'.* To see this, let h be an irreducible common divisor of f/g and of f'/g. Then h is a divisor of exactly one of the squarefree factors of f, say of $f_{(\ell)}$ (since the factors are pairwise coprime), and it follows that $p \nmid \ell$. As h divides f'/g and $f_{(\ell)}$, we get that h divides

$$\frac{f'}{g} - \sum_{\substack{i\neq \ell \\ p\nmid i}} \left(i \cdot f'_{(i)} \cdot \prod_{\substack{j\neq i \\ p\nmid j}} f_{(j)} \right) = \ell \cdot f'_{(\ell)} \cdot \prod_{\substack{j\neq \ell \\ p\nmid j}} f_{(j)} .$$

As $\gcd(h, f_{(j)}) = 1$ for $j \neq \ell$, this implies that h divides $f'_{(\ell)}$, too. On the other hand, $f_{(\ell)}$ is squarefree, hence $\gcd(f_{(\ell)}, f'_{(\ell)}) = 1$, which implies that $h \in \mathbb{F}_q$.

Proposition 2.3.3 and Remark 2.3.6 are the basis for the following recursive algorithm:

Algorithm 2.3.7 (`squarefree_decomposition` in $\mathbb{F}_q[x]$).

```
Input:  $f \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ monic
Output: a list of monic polynomials $f_1, \dots, f_k \in \mathbb{F}_q[x]$ such that $f = f_{(1)} f_{(2)}^2 \cdots f_{(k)}^k$
        is the squarefree factorization of $f$.

(1) IF ($f' = 0$) THEN                    // factors $f_i$, $p \nmid i$, do not exist
      • $g := \sqrt[p]{f}$
      • RETURN (squarefree_decomposition($g$)$^p$)
(2) IF ($f' \neq 0$) THEN                 // there are factors $f_i$, $p \nmid i$
      • $i := 0$ , $L :=$ empty list ,
      • $g_0 := \gcd(f, f')$ , $w_0 := \frac{f}{g_0}$ ,
```

```
• While ($w_i \neq 1$) Do                     // loop for factors $f_{(i)}$, $p \not| \, i$
    ∘ $i := i + 1$,
    ∘ $w_i := \gcd(g_{i-1}, w_{i-1})$, $g_i := \frac{g_{i-1}}{w_i}$,
    ∘ $L[i] := \frac{w_{i-1}}{w_i}$,
• If ($g_i = 1$) Then Return ($L$)           // there are no factors $f_{(i)}$, $p \mid i$
• $g := \sqrt[p]{g_i}$,                      // product of factors $f_{(i)}$, $p \mid i$
• Return ($L * \texttt{squarefree\_decomposition}(g)^p$)
```

Remark 2.3.8 (Notation). In the above algorithm, we used a short-hand notation for operations on lists:

- $(f_1, \ldots, f_r) * (g_1, \ldots, g_s) := (f_1 g_1, \ldots, f_s g_s, f_{s+1}, \ldots, f_r)$ for $r \geq s$,
- $(h_1, \ldots, h_s)^p := (\underbrace{1, \ldots, 1}_{p-1}, h_1, \underbrace{1, \ldots, 1}_{p-1}, h_2, 1, \ldots, h_s)$

Proof of Algorithm 2.3.7. Step 1 (Correctness). We show by induction that

$$w_\nu = \prod_{\substack{p \nmid i \\ i \geq \nu+1}} f_{(i)}\,, \qquad g_\nu = \prod_{p \mid i} f_{(i)}^i \cdot \prod_{\substack{p \nmid i \\ i \geq \nu+2}} f_{(i)}^{i-\nu-1}\,, \tag{2.3.1}$$

for all $\nu \geq 0$. For $\nu = 0$, our base of induction, the claim is exactly the initialization step of the algorithm according to Remark 2.3.6. The induction step $\nu \to \nu + 1$ is a direct computation:

$$w_{\nu+1} = \gcd(g_\nu, w_\nu) = \prod_{\substack{i \geq \nu+2 \\ p \nmid i}} f_{(i)}\,, \qquad g_{\nu+1} = \frac{g_\nu}{w_{\nu+1}} = \prod_{p \mid i} f_{(i)}^i \cdot \prod_{\substack{p \nmid i \\ i \geq \nu+3}} f_{(i)}^{i-\nu-2}\,.$$

It follows from (2.3.1) that the polynomials assigned to $L[i]$ in the While loop are

$$\frac{w_{i-1}}{w_i} = \begin{cases} f_{(i)}\,, & \text{if } p \nmid i\,, \\ 1\,, & \text{if } p \mid i\,. \end{cases}$$

When leaving the While loop, g_i is the product of the factors $f_{(j)}^j$, $p \mid j$. Hence, g_i is then a p-th power, and we can form the p-th root of g_i. Calling the algorithm again for this p-th root, we obtain the missing entries of the list. It can easily be checked that these entries are inserted at the correct positions of the list L.

Step 2 (Termination). We proceed by induction on the number of irreducible factors of f (counted with multiplicities). For the induction base, let f be irreducible. Then the algorithm returns $f_{(1)} = f$ after passing once through the While loop in (2). Now, let $f = f_1 \cdot f_{(2)}^2 \cdot \ldots \cdot f_{(k)}^k$, $k \geq 2$. Then there are two possibilities: either $f' = 0$ or $f' \neq 0$. In the first case, the algorithm recursively calls `squarefree_decomposition`$\big(\sqrt[p]{f}\big)$, which terminates by the induction hypothesis. If $f' \neq 0$, the algorithm passes at most k times through the WHILE loop in (2), and then calls `squarefree_decomposition`$\big(\sqrt[p]{g_i}\big)$, which terminates again by the induction hypothesis. $\square$

Example 2.3.9. Let us consider a polynomial $f = f_{(1)} \cdot f_{(2)}^2 \cdot f_{(3)}^3 \in \mathbb{F}_2[x]$, where the factors $f_{(i)}$ are irreducible and monic. We can compute directly that

$$f' = f_{(2)}^2 \cdot \big(f_{(1)}' \cdot f_{(3)}^3 + f_{(3)}' \cdot f_{(1)} f_{(3)}^2\big) \neq 0.$$

We will now pass through the steps of the algorithm:

- *Initialization.*

$$\begin{array}{l} i = 0\,, \;\; L = \texttt{empty list}\,, \\ g_0 = \gcd(f, f') = f_{(2)}^2 f_{(3)}^2\,, \quad w_0 = \dfrac{f}{g_0} = f_{(1)} f_{(3)}\,, \end{array}$$

- *Loop for factors $p \nmid i$.*

$$\begin{array}{ll} (w_0 \neq 1): & i = 1\,, \\ & w_1 = \gcd(g_0, w_0) = f_{(3)}\,, \quad g_1 = \dfrac{g_0}{w_1} = f_{(2)}^2 f_{(3)}\,, \\ & L[1] = \dfrac{w_0}{w_1} = f_{(1)}\,, \\ (w_1 \neq 1): & i = 2\,, \\ & w_2 = \gcd(g_1, w_1) = f_{(3)}\,, \quad g_2 = \dfrac{g_1}{w_2} = f_{(2)}^2\,, \\ & L[2] = \dfrac{w_1}{w_2} = 1\,, \\ (w_2 \neq 1): & i = 3\,, \\ & w_3 = \gcd(g_2, w_2) = 1\,, \quad g_3 = \dfrac{g_2}{w_3} = f_{(2)}^2\,, \\ & L[3] = \dfrac{w_2}{w_3} = f_{(3)}\,, \\ (w_3 = 1): & \text{leave the loop} \end{array}$$

- *Remaining factors (recursion).*

$$\begin{array}{rcl} g & = & \sqrt[2]{f_{(2)}^2} = f_{(2)}\,, \\ L * \texttt{squarefree\_decomposition}(g)^2 & = & (f_{(1)}, 1, f_{(3)}) * (1, f_{(2)}) \\ & = & (f_{(1)}, f_{(2)}, f_{(3)}). \end{array}$$

Remark 2.3.10. There exists a more efficient algorithm for squarefree factorization, due to Yun (cf. [GCL, 8.2]). The key idea of this algorithm is to reduce the degree of the polynomials appearing in the g.c.d. computations.

2.3.2 Distinct degree factorization

After computing a squarefree factorization of a given polynomial, it is possible to use another preprocessing step, the distinct degree decomposition. It decomposes a squarefree polynomial in such a way that all irreducible factors of degree i are collected in the factor f_i; combined with equal degree factorization (see Remark 2.3.27) it also leads to a complete factorization algorithm.

Definition 2.3.11. Let R be a UFD, and let $f \in R[x]$ be primitive. Then the **distinct degree factorization (DDF)** of f is

$$f = f_{[1]} \cdot \ldots \cdot f_{[m]}$$

where $f_{[i]}$ denotes the product of all irreducible factors of f of degree i.

Remark 2.3.12. For formulating the algorithm of the DDF later on in this section, we need the following two trivial, but important observations:

(1) The last index m in the decomposition cannot exceed $\deg(f)$.

(2) Each non-trivial factor $f_{[i]}$ with $i > \frac{\deg(f)}{2}$, is irreducible.

The following lemma generalizes Fermat's Little Theorem. It is the key ingredient for the DDF algorithm, since it allows us to compute the factors $f_{[i]}$ by means of g.c.d. computations in $\mathbb{F}_q[x]$.

Lemma 2.3.13. *Let $p \in \mathbb{N}$ be a prime, $r \geq 1$, $q = p^r$, and let $d \in \mathbb{Z}_{\geq 1}$. Then $x^{q^d} - x$ is the product of all monic, irreducible polynomials $g \in \mathbb{F}_q[x]$ whose degree is a divisor of d.*

Example 2.3.14. In $\mathbb{F}_2[x]$, we have

$$\begin{aligned}
x^2 - x &= x(x+1)\,, \\
x^4 - x &= x(x+1)(x^2+x+1)\,, \\
x^8 - x &= x(x+1)(x^3+x+1)(x^3+x^2+1)\,, \\
x^{16} - x &= x(x+1)(x^2+x+1)(x^4+x+1)(x^4+x^3+1)(x^4+x^3+x^2+x+1)\,.
\end{aligned}$$

Proof of Lemma 2.3.13. Let $g \in \mathbb{F}_q[x]$ be monic and irreducible of degree e. For proving the claim, we only need to show:

(1) $(e \mid d)$ iff $\big(g \mid x^{q^d} - x\big)$.

(2) $x^{q^d} - x$ is squarefree.

The second statement is implied by Corollary 2.3.5, as

$$\frac{d}{dx}\left(x^{q^d} - x\right) = \underbrace{q^d \cdot x^{q^d-1}}_{=0} - 1 = -1\,.$$

For statement (1), let us first suppose that $e = \deg(g)$ divides d. Then the extension field of $\mathbb{F}_q$ defined by g, $\mathbb{F}_{q^e} = \mathbb{F}_q[x]/\langle g\rangle$, can be embedded as a subfield in $\mathbb{F}_{q^d}$ (Remark 1.2.3 (5)). In particular, if $a \in \mathbb{F}_{q^e}$ is a root of g, then Fermat's Little Theorem implies that $a^{q^d} - a = 0$. Computing over the algebraic closure $\overline{\mathbb{F}_q}$ of $\mathbb{F}_q$, it follows that $x - a$ divides g and $x^{q^d} - x$, that is, the resultant of g and $x^{q^d} - x$ does not vanish (Proposition 2.2.4 (1)). Since the resultant of two polynomials with coefficients in $\mathbb{F}_q$ is again a polynomial with coefficients in $\mathbb{F}_q$, we can apply Proposition 2.2.4 (1) again to deduce that g and $x^{q^d} - x$ have a non-constant common factor in $\mathbb{F}_q$. Since g is supposed to be irreducible in $\mathbb{F}_q[x]$, this means that g divides $x^{q^d} - x$.

For the other implication, let g divide $x^{q^d} - x$. Then Fermat's Little Theorem implies that

$$g = \prod_{a\in M} (x - a)\,, \qquad M \subset \mathbb{F}_{q^d} \text{ some subset.}$$

Choose any zero a of g in the extension field $\mathbb{F}_{q^e} = \mathbb{F}_q[x]/\langle g\rangle$. Then g is the minimal polynomial for a over $\mathbb{F}_q$, which implies that $\mathbb{F}_q[x]/\langle g\rangle = \mathbb{F}_q(a)$ is the smallest extension field of $\mathbb{F}_q$ containing a. Since we can identify a with an element $a \in M \subset \mathbb{F}_{q^d}$, we get an embedding of fields

$$\mathbb{F}_{q^e} = \mathbb{F}_q[x]/\langle g\rangle = \mathbb{F}_q(a) \subset \mathbb{F}_{q^d}$$

which is only possible if e divides d (Remark 1.2.3 (5)). □

Lemma 2.3.13 implies that, for a squarefree polynomial $f \in \mathbb{F}_q[x]$ and each $d \geq 1$, we have

$$\prod_{i\,|\,d} f_{[i]} = \gcd\left(f,\, x^{q^d} - x\right)\,.$$

In particular, it identifies the factor $f_{[d_{\min}]}$ collecting the irreducible factors of f of lowest occuring degree $d_{\min}$ as

$$f_{[d_{\min}]} = \gcd\left(f,\, x^{q^{d_{\min}}} - x\right)\,.$$

Dividing f by $f_{[d_{\min}]}$, and repeating the above reasoning, we get the factor collecting the irreducible factors of f of second lowest occuring degree, and so on. We can stop the iteration as soon as we have computed all factors $f_{[d]}$ with $d \leq \frac{\deg(f)}{2}$. Then the remaining polynomial is already irreducible due to the observation in Remark 2.3.12. This is the basic idea of the following algorithm:

Algorithm 2.3.15 (`distinct_degree_factorization` in $\mathbb{F}_q[x]$).

Input: $f \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ monic, squarefree.

Output: Set of tuples $(f_{[d_j]}, d_j)$, $j = 1, \ldots, s$, such that $f_{[d_j]} \neq 1$ is a product of irreducible polynomials in $\mathbb{F}_q[x]$ of degree d_j, and $f = f_{[d_1]} \cdot \ldots \cdot f_{[d_s]}$ is a DDF of f.

```
(1) $d := 0$, Result $:= \emptyset$,                                   // Initialization
    $g_0 := f$, $h_0 := x$,
(2) WHILE $\left(d \le \frac{\deg(g_d)}{2} - 1\right)$ DO                 // Main Loop
      • $d := d + 1$,
      • $h_d := h_{d-1}^q \ \mathrm{rem}\ g_{d-1}$,
      • $f_{[d]} := \gcd(g_{d-1}, h_d - x)$,
      • IF $(f_{[d]} \neq 1)$ THEN Result := Result $\cup \ \{(f_{[d]}, d)\}$,
      • $g_d := \frac{g_{d-1}}{f_{[d]}}$,
(3) IF $(g_d \neq 1)$ THEN Result := Result $\cup \{(g_d, \deg(g_d))\}$   // Remaining Factor
(4) RETURN (Result).
```

Proof. Step 1 (Termination). We pass through the WHILE loop at most $\frac{\deg(f)}{2} - 1$ times. Hence, there is nothing to prove here.

Step 2 (Correctness). Let $f = f_{[1]} \cdot \dots \cdot f_{[m]}$ be the DDF of f. By induction on d, we show that, for all $1 \le d \le m$,

(1) $h_d = x^{q^d} \ \mathrm{rem}\ g_{d-1}$,

(2) $g_d = f_{[d+1]} \cdot \ldots \cdot f_{[m]}$,

(3) $\gcd(g_{d-1}, h_d - x) = \gcd(g_{d-1}, x^{q^d} - x) = f_{[d]}$.

The base of the induction immediately follows from the initialization step of the algorithm: $g_0 = f = f_{[1]} \cdot \dots \cdot f_{[m]}$, $h_0 = x$, and the first pass through the WHILE loop (with $d = 1$):

$$h_1 = (h_0^q \ \mathrm{rem}\ g_0) = (x^q \ \mathrm{rem}\ g_0)\,,$$

hence $h_1 \equiv x^q \bmod g_0$. It follows that

$$\gcd(g_0, h_1 - x) = \gcd(g_0, x^q - x) = f_{[1]}\,,$$

due to Lemma 2.3.13. Finally,

$$g_1 = \frac{g_0}{\gcd(g_0, h_1 - x)} = \frac{f}{f_{[1]}} = f_{[2]} \cdot \ldots \cdot f_{[m]}\,.$$

For the induction step $d \to d + 1$, we first observe that, by the induction hypothesis, $h_d \equiv x^{q^d} \bmod g_{d-1}$ and $g_{d-1} = f_{[d]} \cdot g_d$. Hence, $h_d^q \equiv (x^{q^d})^q \bmod g_d$, which proves claim (1). As the degree of all irreducible factors of g_d is at least $d + 1$ by (2) of the induction hypothesis, we can again apply Lemma 2.3.13 to prove that

$$\gcd(g_d, h_{d+1} - x) = \gcd(g_d, x^{q^{d+1}} - x) = f_{[d+1]}\,,$$

hence,

$$g_{d+1} = \frac{g_d}{\gcd(g_d, h_{d+1} - x)} = \frac{g_d}{f_{[d+1]}} = f_{[d+2]} \cdot \ldots \cdot f_{[m]}\,.$$

□

Example 2.3.16. We apply the algorithm to $f = x^{15} - 1 \in \mathbb{F}_{11}[x]$:

- *Initialization.*

$$g_0 = f = x^{15} - 1\,, \quad h_0 = x\,,$$

- *Loop for factors $f_{[d]}$.*

$$\begin{aligned}
(d = 1): \quad & h_1 := \big(h_0^{11} \text{ rem } g_0\big) = \big(x^{11} \text{ rem}(x^{15} - 1)\big) = x^{11}\,,\\
& f_{[1]} := \gcd(g_0, h_1 - x) = \gcd(x^{15} - 1, x^{11} - x) = x^5 - 1\,,\\
& g_1 := \frac{g_0}{f_{[1]}} = \frac{x^{15} - 1}{x^5 - 1} = x^{10} + x^5 + 1\,,\\
(d = 2): \quad & h_1 = \big(h_1^{11} \text{ rem } g_1\big) = \big(x^{121} \text{ rem}(x^{15} - 1)\big) = x\,,\\
& f_{[1]} = \gcd(g_0, h_1 - x) = \gcd(x^{10} + x^5 + 1, 0) = x^{10} + x^5 + 1\,,\\
& g_2 := \frac{g_1}{f_{[2]}} = \frac{x^{10} + x^5 + 1}{x^{10} + x^5 + 1} = 1\,.
\end{aligned}$$

We conclude that f has the DDF

$$\underbrace{(x^5 - 1)}_{\substack{\text{5 factors of}\\ \text{degree 1}}} \cdot \underbrace{(x^{10} + x^5 + 1)}_{\substack{\text{5 factors of}\\ \text{degree 2}}}\,.$$

Remark 2.3.17. The main disadvantage of Algorithm 2.3.15 is that it involves many g.c.d.-computations which give a trivial result, hence, do not contribute to the DDF.

To avoid this problem, modern algorithms use a preprocessing step, they compute a so-called "**coarse DDF**" in the following way: First the interval of possible degrees $I = [1, B]$ (for example, $B = \lfloor \frac{\deg(f)}{2} \rfloor$) is decomposed into pairwise disjoint intervals,

$$I = [a_1, b_1] \overset{\bullet}{\cup} \ldots \overset{\bullet}{\cup} [a_s, b_s]\,.$$

Then auxiliary polynomials

$$\widetilde{f}_j := \gcd\left(\frac{f}{\widetilde{f}_{j-1}}\,,\ \prod_{i=0}^{b_j - a_j} \left(x^{q^{b_j}} - x^{q^i}\right)\right), \qquad j = 1, \ldots, s\,,$$

are computed and, as the last step, only the non-constant polynomials among the $\widetilde{f}_j$ have to be decomposed by means of the above DDF algorithm. (See for example, the algorithm of Kaltofen and Shoup [KaS].)

The background of these algorithms is the following immediate consequence of Lemmas 2.3.13 and 1.2.5 (1):

Lemma 2.3.18. *Let $i \geq j \in \mathbb{Z}_{\geq 0}$. Then*

$$x^{q^i} - x^{q^j} = \left(x^{q^{i-j}} - x\right)^{q^j} = \prod_{\substack{g \in \mathbb{F}_q[x]\\ \textit{monic, irred.}\\ \deg(g)\,|\,i-j}} g^{q^j}\,.$$

2.3.3 The algorithms of Berlekamp, Cantor/Zassenhaus

After applying the preprocessing steps introduced in the previous sections, we still do not have a decomposition into irreducible factors; preprocessing has only transformed the problem of factorization of one polynomial to the problem of finding the irreducible factors of several squarefree polynomials (respectively polynomials whose irreducible factors are known to be all of the same degree).

The earliest algorithm for solving this remaining problem in polynomial time is the **algorithm of Berlekamp** (1967), which tackles the factorization problem by means of linear algebra, namely reducing it to Gaussian elimination via the **Frobenius map**.

As the constrution of reducing the factorization problem to a problem of linear algebra via the Frobenius map is central to the algorithms in this section, we will consider this in more detail before stating the algorithm: Let $f \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ be a squarefree, monic polynomial over a finite field $\mathbb{F}_q$, where $q = p^r$ for some prime number $p \in \mathbb{N}$ and some $r \in \mathbb{Z}_{\geq 1}$, and let $f = f_1 \cdot \ldots \cdot f_s$ be the irreducible decomposition of f. Note that, as f is squarefree, the factors f_i are pairwise coprime. The **Frobenius map**

$$\begin{array}{rccc} \mathrm{Fr}: & \mathbb{F}_q[x] & \longrightarrow & \mathbb{F}_q[x] \\ & h & \longmapsto & h^q\,, \end{array}$$

is an $\mathbb{F}_q$-linear endomorphism of $\mathbb{F}_q[x]$. It induces the ($\mathbb{F}_q$-linear) **Frobenius map modulo $\langle f \rangle$**,

$$\begin{array}{rccc} \overline{\mathrm{Fr}}: & \mathbb{F}_q[x]/\langle f\rangle & \longrightarrow & \mathbb{F}_q[x]/\langle f\rangle \\ & \overline{h} & \longmapsto & \overline{h}^q, \end{array}$$

which, in turn, gives rise to the endomorphism

$$\begin{array}{rccc} \Phi_f := \overline{\mathrm{Fr}} - \overline{\mathrm{id}}: & \mathbb{F}_q[x]/\langle f\rangle & \longrightarrow & \mathbb{F}_q[x]/\langle f\rangle \\ & \overline{h} & \longmapsto & \overline{h}^q - \overline{h} \end{array}$$

of the finite dimensional $\mathbb{F}_q$-vector space $\mathbb{F}_q[x]/\langle f\rangle$. The properties of Φ_f (stated in the subsequent Lemma 2.3.19) are the key ingredients to the algorithms of Berlekamp and Cantor/Zassenhaus.

Lemma 2.3.19. *Let $\Phi_f = \overline{\mathrm{Fr}} - \overline{\mathrm{id}}$ be the above endomorphism of the finite dimensional $\mathbb{F}_q$-vector space $\mathbb{F}_q[x]/\langle f\rangle$, and let $f = f_1 \cdot \ldots \cdot f_s$ be the irreducible factorization of f. Then the following holds:*

(1) The dimension of the kernel of Φ_f equals the number of irreducible factors of f, that is,

$$\dim_{\mathbb{F}_q}\big(\mathrm{Ker}(\Phi_f)\big) = s\,.$$

(2) Let $h \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ *represent an element* $\overline{h} \in \operatorname{Ker}(\Phi_f)$*. Then*

$$f = \prod_{a \in \mathbb{F}_q} \gcd(f,\, h - a)\,.$$

(3) Let $1 =: h_1, h_2, \dots, h_s$ *be elements of* $\mathbb{F}_q[x]$ *representing a basis for* $\operatorname{Ker}(\Phi_f)$*. Then, for all* $1 \le i < j \le s$*, there exist* $k \in \{2, \dots, s\}$ *and* $a \neq b \in \mathbb{F}_q$ *satisfying*

$$f_i \mid h_k - a\,, \quad \textit{and} \quad f_j \mid h_k - b\,.$$

Remark 2.3.20. The most important observation of statement (3) is that $\gcd(f, h_k - a)$ and $\gcd(f, h_k - b)$ are two *proper* factors of f (that is, factors of lower degree than f). This implies, in particular, that for the chosen element h_k the decomposition in statement (2) is non-trivial.

We call such a polynomial $h \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$, for which the decomposition in (2) is non-trivial, an ***f*-reducing polynomial**.

Now, the basic idea for the factorization algorithm should be clear: One computes the dimension of the kernel of Φ_f (e.g., by Gaussian elimination). If its dimension is larger than 1, one looks for an f-reducing polynomial, leading (via g.c.d.-computations) to a non-trivial factorization $f = g_1 \cdot \ldots \cdot g_k$. Then, for each of the factors g_i, one repeats this procedure, until one has found a factorization of f in $s = \dim_{\mathbb{F}_q}\bigl(\operatorname{Ker}(\Phi_f)\bigr)$ non-constant factors.

Of course, one of the key points for the efficiency of such a recursive algorithm is to have a good strategy for selecting the f-reducing polynomials. Here, an easy observation is helpful: If $h \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ represents an element of $\operatorname{Ker}(\Phi_f)$ and $\deg(h) < \deg(f)$, then h is f-reducing (compare degrees on both sides of the equality in (2)).

Proof of Lemma 2.3.19. (1) Via the Chinese remainder isomorphism $\boldsymbol{c}$ (see Theorem 1.1.16), we may identify $\mathbb{F}_q[x]/\langle f \rangle$ with a direct sum of fields:

$$\boldsymbol{c} = (c_1, \dots, c_s) :\ \mathbb{F}_q[x]/\langle f \rangle \xrightarrow{\cong} \underbrace{\mathbb{F}_q[x]/\langle f_1 \rangle}_{\cong\, \mathbb{F}_{q^{d_1}}} \oplus \dots \oplus \underbrace{\mathbb{F}_q[x]/\langle f_s \rangle}_{\cong\, \mathbb{F}_{q^{d_s}}}\,,$$

where $d_i := \deg(f_i)$. In particular, we get the equivalences

$$\overline{h} \in \operatorname{Ker}(\Phi) \iff \overline{h} = \overline{h}^q \iff \forall\, 1 \le i \le s :\ c_i\bigl(\overline{h}\bigr) = c_i\bigl(\overline{h}^q\bigr) = c_i\bigl(\overline{h}\bigr)^q\,.$$

On the other hand, Lemma 1.2.5 (2) gives that $c_i\bigl(\overline{h}\bigr) \in \mathbb{F}_{q^{d_i}}$ satisfies $c_i\bigl(\overline{h}\bigr)^q = c_i\bigl(\overline{h}\bigr)$ iff $c_i\bigl(\overline{h}\bigr) \in \mathbb{F}_q$. Hence,

$$\operatorname{Ker}(\Phi_f) = \boldsymbol{c}^{-1}\bigl(\underbrace{\mathbb{F}_q \oplus \dots \oplus \mathbb{F}_q}_{s \text{ factors}}\bigr)\,,$$

which is an s-dimensional $\mathbb{F}_q$-vector space as claimed.

(3) Let $i < j$. Since $(\overline{h}_1, \dots, \overline{h}_s)$ is a basis of $\mathrm{Ker}(\Phi_f) = \boldsymbol{c}^{-1}\big(\mathbb{F}_q \oplus \cdots \oplus \mathbb{F}_q\big)$, and since $\boldsymbol{c}$ is surjective, there exists a $k \geq 2$ such that

$$a := \underbrace{c_i\big(\overline{h}_k\big)}_{\in\, \mathbb{F}_q} \neq \underbrace{c_j\big(\overline{h}_k\big)}_{\in\, \mathbb{F}_q} =: b\,.$$

On the other hand, we know that

$$\begin{array}{lclcl} c_i\big(\overline{h}_k\big) = a & \Longleftrightarrow & c_i\big(\overline{h}_k\big) - a = 0 & \Longleftrightarrow & f_i \mid \big(\overline{h}_k - a\big) \\ c_j\big(\overline{h}_k\big) = b & \Longleftrightarrow & c_j\big(\overline{h}_k\big) - b = 0 & \Longleftrightarrow & f_j \mid \big(\overline{h}_k - b\big)\,, \end{array}$$

which proves (3).

(2) Let $h \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ represent an element $\overline{h} \in \mathrm{Ker}(\Phi_f)$. Then, the same arguments as in the proof of (3) give that

$$f_i \mid \big(\overline{h} - c_i\big(\overline{h}\big)\big)\,, \tag{2.3.2}$$

for all $1 \leq i \leq s$. As we have seen in the proof of (1), $a_i := c_i\big(\overline{h}\big) \in \mathbb{F}_q$. Since f_i divides f, we obtain from (2.3.2) that f_i divides $h - a_i$. Hence,

$$f_i \;\Bigg|\; \prod_{a \in \mathbb{F}_q} (h - a)$$

for all $1 \leq i \leq s$. As the irreducible polynomials f_i and f_j cannot have a common non-constant factor for $i \neq j$, we can conclude that also $f = f_1 \cdot \ldots \cdot f_s$ divides the product of all $h - a$, $a \in \mathbb{F}_q$. On the other hand, the elements $h - a$, $a \in \mathbb{F}_q$, are pairwise coprime. Hence, each irreducible factor f_i occurs in precisely one of the factors $h - a$. Forming the g.c.d. of f and $h - a$ for each $a \in F_q$, we get rid of unnecessary factors and, finally, obtain the desired expression

$$f = \prod_{a \in \mathbb{F}_q} \gcd\,(f,\, h - a)\,.$$

□

For the convenience of the reader, we present the factorization algorithm proceeding along the lines of Remark 2.3.20 by decomposing it into three parts. We start by discussing two auxiliary procedures: the `Berlekamp_Splitting` Algorithm, and the computation of $\mathrm{Ker}(\Phi_f)$. The main algorithm (Algorithm 2.3.24) is then presented as an iteration based on these two procedures.

Algorithm 2.3.21 (`Berlekamp_Splitting`).

Input: $f \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ squarefree and monic,
$h_2, \dots, h_s \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ such that $1, \overline{h}_2, \dots, \overline{h}_s$ form a basis of $\mathrm{Ker}(\Phi_f)$

```
Output: If $f$ is reducible: a non-trivial factor of $f$;
        Otherwise: $f$

(1) IF ($s = 1$) THEN RETURN ($f$)
(2) $i := 2$,
(3) WHILE ($i \leq s$)
      • FOR ($a \in \mathbb{F}_q$) DO
          ◦ $g := \gcd(f,\, h_i - a)$,
          ◦ IF (($g \neq 1$) AND ($g \neq f$)) THEN RETURN ($g$)
      • $i := i + 1$,
```

Remark 2.3.22 (Computing $\operatorname{Ker}(\Phi)$**).** The basic tool for computing the kernel of the endomorphism Φ_f is Gaussian elimination. To this end, we first need to compute the matrix of Φ_f with respect to a fixed basis of $\mathbb{F}_q[x]/\langle f\rangle$, say $\overline{1}, \overline{x}, \ldots, \overline{x}^{d-1}$, $d := \deg(f)$. With this choice, computing the matrix of Φ_f boils down to determining the (unique) expression of the elements $\overline{x}^{iq} \in \mathbb{F}_q[x]/\langle f\rangle$, $1 \leq i \leq d-1$, as an $\mathbb{F}_q$-linear combination of the basis elements $\overline{1}, \overline{x}, \ldots, \overline{x}^{d-1}$.

A direct approach for this task is a successive computation of $x^i \bmod f$ for all $d \leq i \leq (d-1)\cdot q$ in the following way: Let

$$f = x^d + \sum_{j=0}^{d-1} a_j x^j \,,$$

and let $\overline{x}^i = \sum_{j=0}^{d-1} b_{i,j}\overline{x}^j$. Then $\overline{x}^{i+1}$ can be computed as

$$\begin{aligned}
\overline{x}^{i+1} &= \sum_{j=0}^{d-1} b_{i,j} \cdot \overline{x}^{j+1} = \left(\sum_{j=1}^{d-1} b_{i,j-1} \cdot \overline{x}^j\right) - b_{i,d-1} \sum_{j=0}^{d-1} a_j \overline{x}^j \\
&= \big(\underbrace{-b_{i,d-1} \cdot a_0}_{=:\, b_{i+1,0}}\big) \cdot \overline{1} + \sum_{j=1}^{d-1} \big(\underbrace{b_{i,j-1} - b_{i,d-1} \cdot a_j}_{=:\, b_{i+1,j}}\big) \cdot \overline{x}^j \,.
\end{aligned}$$

Let $B := (b_{iq,j})_{i,j=0..d-1}$. Then the matrix of Φ_f is given by $B - \mathbf{1}_d$, with $\mathbf{1}_d$ the $d \times d$ unity matrix.

A more efficient way of computing exponentiation is the **Repeated Squaring Algorithm**.

Example 2.3.23 (Computing $\operatorname{Ker}(\mathbf{\Phi_f})$ **and Berlekamp Splitting).** Consider the finite field $\mathbb{F}_9$, specified as $\mathbb{F}_3[a]/\langle a^2 - a - 1\rangle$. Over this field, the task is to find a non-trivial factor of the polynomial

$$f = x^3 + x^2 + x - a \in \mathbb{F}_9[x]$$

(which has the irreducible decomposition $\big(x+a\big)\cdot\big(x^2-(a-1)x-1\big)$). To this end, we first compute the kernel of Φ_f and then apply the Berlekamp splitting: The matrix of the $\mathbb{F}_9$-linear map Φ_f with respect to the basis $\big(\overline{1},\overline{x},\overline{x}^2\big)$ of $\mathbb{F}_9[x]/\langle f\rangle$ is

$$\begin{pmatrix} 0 & -a & -1 \\ 0 & a+1 & a \\ 0 & -a-1 & -a \end{pmatrix}$$

as can be checked by a straightforward computation. By Gaussian elimination, we obtain the matrix

$$\begin{pmatrix} 0 & 1 & a-1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

which shows that $\mathrm{Ker}(\Phi_f)$ has dimension $s=2$, and that

$$\big\{\overline{1},\ \overline{x}^2-(a-1)\cdot\overline{x}\big\}$$

is a basis of $\mathrm{Ker}(\Phi_f)$. We set $h_2 := x^2-(a-1)x$.

Given these data, we can now enter the Berlekamp Splitting Algorithm. We then compute the following g.c.d.s (according to the algorithm):

$$\begin{aligned} \gcd\big(f,\, h_2+0\big) &= 1\,,\\ \gcd\big(f,\, h_2+1\big) &= 1\,,\\ \gcd\big(f,\, h_2-1\big) &= x^2-(a-1)x-1\,, \end{aligned}$$

and obtain the non-trivial factor $x^2-(a-1)x-1$. Note that

$$\gcd\big(f,\, h_2-a+1\big) = x+a \qquad ,$$

that is, the output of the Berlekamp Splitting Algorithm depends on the ordering in which the elements of $\mathbb{F}_q$ are treated in the For loop.

Algorithm 2.3.24 (Berlekamp Factorization Algorithm, BFA).

Input: $f \in \mathbb{F}_q[x]\setminus\mathbb{F}_q$ squarefree, monic

Output: $f_1,\dots,f_s \in \mathbb{F}_q[x]$ monic, irreducible such that $f=f_1\cdot\ldots\cdot f_s$

```
(1) Compute a basis $\overline{1},\overline{h}_2,\dots,\overline{h}_s$ of $\mathrm{Ker}(\Phi_f)$ in $F_q[x]/\langle f\rangle$, and choose representatives $h_1 := 1,\, h_2,\dots,h_s \in \mathbb{F}_q[x]\setminus\mathbb{F}_q$ thereof;[8]
(2) Factors := $\{f\}$;                          // Initialization
    Irred := $\emptyset$;
(3) WHILE $\big(\#(\mathrm{Factors})+\#(\mathrm{Irred}) < s\big)$ DO
```

---

[8] For example, we can use the canonical representation such that $\deg(h_i) < \deg(f)$.

```
    • Choose $g \in$ Factors ;
    • $h :=$ Berlekamp_Splitting $\big(g,\ \{1, h_2, \dots, h_s\}\big)$ ;
    • If $(h \neq g)$ Then                      // $g$ is reducible
        ∘ Factors := (Factors \ $\{g\}$) ∪ $\left\{h,\ \frac{g}{h}\right\}$ ;
      Else                                      // $g$ is irreducible
        ∘ Factors := Factors \ $\{g\}$ ;
        ∘ Irred := Irred ∪ $\{g\}$ ;
(4) Return (Factors ∪ Irred)
```

In this formulation of the Berlekamp Factorization Algorithm, no effort has been made to *reduce the number of g.c.d.-computations* involved. But, even if we drop all those g.c.d.-computations which are obviously (a priori) unnecessary, we are still dealing with a large number of g.c.d.-computations most of which return 1.

Making further efforts to reduce the number of g.c.d.s (cf. [Zas]), then leads to a tradeoff: what will be saved by avoiding one kind of computation, has to be spent on a different kind. Thus, avoiding unproductive g.c.d.-computations at all cost does not lead to a significant improvement of Berlekamp's algorithm.

The practical solution to this dilemma is a variant of the algorithm which has a probabilistic running time: the **Cantor/Zassenhaus** approach (cf. [CaZ]). Although the general idea is the same for all positive characteristics, the case of characteristic 2 needs to be treated with special care and will hence be postponed; we will assume for now that *q is odd.*

The key ingredient for the Cantor/Zassenhaus variant of Berlekamp's algorithm is the rather simple observation that the polynomial $x^q - x \in \mathbb{F}_q[x]$ can be factorized in the following way:

$$x^q - x \;=\; x \cdot \big(x^{(q-1)/2} + 1\big)\,\big(x^{(q-1)/2} - 1\big)\,.$$

Hence, any $h \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ mapping to $\overline{h} \in \operatorname{Ker}(\Phi_f) \setminus \{0\}$ satisfies

$$h \cdot \big(h^{(q-1)/2} + 1\big)\,\big(h^{(q-1)/2} - 1\big) \;=\; h^q - h \;\equiv\; 0 \mod f\,,$$

and, therefore,

$$f = \gcd(f,\, h^q - h) = \gcd(f, h) \cdot \gcd\big(f, h^{(q-1)/2} + 1\big) \cdot \gcd\big(f, h^{(q-1)/2} - 1\big). \quad (2.3.3)$$

It remains to determine the probability that the decomposition (2.3.3) of a monic squarefree polynomial f is non-trivial, that is, that at least two of the factors on the right-hand side are non-constant. Let us assume that the contrary holds. Then, as $\overline{h} \neq 0$, we have that either f divides $h^{(q-1)/2} + 1$, or that f divides $h^{(q-1)/2} - 1$. Via the Chinese remainder isomorphism $\boldsymbol{c} = (c_1, \dots, c_s)$, this means that the images

$c_i(h) \in \mathbb{F}_q$ either satisfy $c_i(h)^{(q-1)/2} = -1$, for all $i = 1, \ldots, s$, or $c_i(h)^{(q-1)/2} = 1$, for all $i = 1, \ldots, s$. It is not difficult to see that precisely $\frac{q-1}{2}$ elements a of $\mathbb{F}_q$ satisfy $a^{(q-1)/2} = 1$ (see Exercise 2.3.5), and that the same number of elements satisfies $a^{(q-1)/2} = -1$. Thus, the decomposition (2.3.3) is non-trivial with probability at least

$$1 - \left(\frac{q-1}{2q}\right)^s - \left(\frac{q-1}{2q}\right)^s \geq 1 - \frac{1}{2^{s-1}} \geq \frac{1}{2}$$

for $s \geq 2$. In the same way, we get that the third factor, $\gcd\bigl(f, h^{(q-1)/2} - 1\bigr)$, is a non-trivial factor of f with probability at least

$$1 - \left(\frac{q-1}{2q}\right)^s - \left(\frac{q+1}{2q}\right)^s \geq \frac{4}{9}$$

for $s \geq 2$.

Using these ideas, we obtain the following **Las Vegas algorithm**[9]:

Algorithm 2.3.25 (Berlekamp/Cantor/Zassenhaus).

```
Assume: $p$ is an odd prime, $q = p^r$, $r \geq 1$.
Input:  $f \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ squarefree, monic
Output: $f_1, \ldots, f_s \in \mathbb{F}_q[x]$ monic, irreducible, such that $f = f_1 \cdot \ldots \cdot f_s$

(1) Compute a basis $\overline{1}, \overline{h}_2, \ldots, \overline{h}_s$ of $\mathrm{Ker}(\Phi_f)$ in $F_q[x]/\langle f \rangle$, and choose representatives $h_1 := 1$, $h_2, \ldots, h_s \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$ thereof,
(2) Result := $\{f\}$,
(3) WHILE $\bigl(\#(\mathrm{Result}) < s\bigr)$ DO
      • choose $g \in$ Result, $\deg g > 1$, by random,
      • choose $(c_1, \ldots, c_s) \in \mathbb{F}_q^s \setminus \{0\}$ by random,
      • set $h := c_1 + c_2 h_2 + \ldots + c_s h_s$,    // random element of $\mathrm{Ker}(\Phi_f)$
      • set $w := \gcd\bigl(g,\, h^{(q-1)/2} - 1\bigr)$,
      • IF $(w \notin \{1, g\})$ THEN Result := $\bigl(\mathrm{Result} \setminus \{g\}\bigr) \cup \left\{w, \frac{g}{w}\right\}$,
(4) RETURN (Result)
```

---

[9] A **Las Vegas algorithm** is a probabilistic algorithm for which the error probability is 0, and only the running time is probabilistic. Another kind of probabilistic algorithms are **Monte Carlo algorithms**, for which the result is obtained by a majority vote, that is, the error probability is ε^k, $\varepsilon \in (0, \frac{1}{2})$.

Example 2.3.23 revisited. To illustrate the Berlekamp/Cantor/Zassenhaus Algorithm, we can use the same example as for the Berlekamp Splitting Algorithm. In particular, we can use the basis of $\mathrm{Ker}(\Phi_f)$ which we already computed. We know that the dimension of the kernel is 2. Since, for a given linear combination h of 1 and h_2, we have $(ah)^4 = a^4h^4 = -h^4$, only 18 different choices for h have to be considered: $b + h_2$ and $b + ah_2$, $b \in \mathbb{F}_9$. Indeed, 10 of these choices lead to non-trivial g.c.d.s as a direct computation shows:

$$\begin{aligned}
x^2-(a-1)x-1 &= \gcd\bigl(f, h_2^4-1\bigr) = \gcd\bigl(f,(1+h_2)^4-1\bigr)\\
&= \gcd\bigl(f,(-a+1+h_2)^4-1\bigr) = \gcd\bigl(f,(a-1+ah_2)^4-1\bigr)\\
&= \gcd\bigl(f,(-a-1+ah_2)^4-1\bigr)\,,\\
x+a &= \gcd\bigl(f,(-1+h_2)^4-1\bigr) = \gcd\bigl(f,(-a+h_2)^4-1\bigr)\\
&= \gcd\bigl(f,(-a-1+h_2)^4-1\bigr) = \gcd\bigl(f,(a+ah_2)^4-1\bigr)\\
&= \gcd\bigl(f,(ah_2)^4-1\bigr)\,.
\end{aligned}$$

(Note that only 2 of the 9 possible g.c.d. computations $\gcd(f, h_2-b)$, as applied in the Berlekamp Splitting Algorithm, have a non-trivial result.)

A comparison of this algorithm with the Berlekamp factorization algorithm shows that we have actually only replaced the application of `Berlekamp_Splitting` by a loop over g.c.d.-computations for random elements $h \in \mathrm{Ker}(\Phi_f)$ in a suitable way. It is the significantly higher probability of a g.c.d.-computation being productive in the course of the algorithm that makes the Berlekamp/Cantor/Zassenhaus Algorithm superior to the Berlekamp Algorithm.

Remark 2.3.26. If $q = 2^r$ for some $r \geq 1$, the factors of $x^q - x \in \mathbb{F}_q[x]$ have to be chosen in a different way, namely by using the **trace polynomial**:

$$x^q - x = \underbrace{\left(\sum_{i=0}^{r-1} x^{2^i}\right)}_{=:\,\mathrm{Tr}(x)} \cdot \underbrace{\left(1 + \sum_{i=0}^{r-1} x^{2^i}\right)}_{=\,1+\mathrm{Tr}(x)}.$$

For a monic squarefree polynomial f with $s \geq 2$ irreducible factors, the probability of $\gcd\bigl(f, \mathrm{Tr}(h)\bigr)$ being non-trivial is

$$1 - \frac{1}{2^{s-1}} \geq \frac{1}{2}\,.$$

Remark 2.3.27 (Equal degree factorization). The idea of Cantor/Zassenhaus can also be used to compute a factorization of $f \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$, if all irreducible factors f_i are known to have the same degree e. If q is odd, we use the following decomposition of f (see Exercise 2.3.7):

$$f = \gcd(f, h) \cdot \gcd\left(f,\, h^{(q^e-1)/2}\right) \cdot \gcd\left(f,\, h^{(q^e-1)/2}+1\right)\,,$$

for $h \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$. The probability that the decomposition does not return a nontrivial factor is less than $2^{1-\deg(f)/e} \leq \frac{1}{2}$.

If $q = 2^r$, the $(r \cdot e)$-th trace polynomial is used to obtain an analogous statement.

Resumé 2.3.28 (Algorithms for irreducible factorization). We can combine the algorithms discussed so far in Section 2.3 to obtain two different approaches to the (complete) factorization problem:

(1) Apply the squarefree factorization, followed by a factorization of the squarefree factors via Berlekamp/Cantor/Zassenhaus.

(2) Apply the squarefree factorization, followed by a distinct degree factorization, and then by an equal degree factorization for each of the factors arising from the DDF.

Exercises

Exercise 2.3.1. Let p be a prime, $f \in \mathbb{F}_p[x]$ a polynomial of degree 4. Suppose that $\gcd(x^p - x, f) = \gcd(x^{p^2} - x, f) = 1$. What can you say about the factorization of f in $\mathbb{F}_p[x]$?

Exercise 2.3.2. Trace Algorithm 2.3.7 on computing the squarefree decomposition in $\mathbb{F}_3[x]$ of the (images of the) following polynomials:

(a) $f = x^6 - x^5 - 4x^4 + 2x^3 + 5x^2 - x - 2\,,$

(b) $g = x^6 + x^5 + x^4 + x^3 + x^2 + x\,.$

Use the same procedure to compute the squarefree decomposition of f in $\mathbb{Q}[x]$.

Exercise 2.3.3. Trace Algorithm 2.3.15 on computing the distinct-degree factorization of the squarefree polynomial

$$f = x^{17} + 2x^{15} + 4x^{13} + x^{12} + 2x^{11} + 2x^{10} + 3x^9 + 4x^8 + 4x^4 + 3x^3 + 2x^2 + 4x$$

in $\mathbb{F}_5[x]$. Read from the output how many irreducible factors of degree i, $1 \leq i \leq 17$, the polynomial f has.

Exercise 2.3.4. Let p be a prime number, $q = p^r$, $r \geq 1$, and $d \geq 2$. Let s_d denote the number of monic squarefree polynomials of degree d in $\mathbb{F}_q[x]$.

(a) Prove the recursive formula

$$q^d = \sum_{k=0}^{\lfloor d/2 \rfloor} s_{d-2k} \cdot q^k\,.$$

Hint. Every polynomial $f \in \mathbb{F}_q[x]$ can be uniquely written as $f = g^2 h$ with h being squarefree.

(b) Conclude that $s_d = q^d - q^{d-1}$, and that the probability for a random polynomial in $\mathbb{F}_q[x]$ of degree d to be squarefree is $1 - \frac{1}{q}$.

Exercise 2.3.5. Let p be a prime number, $q = p^r$, $r \geq 1$, and let k be a divisor of $q - 1$. Prove that

$$S := \left\{ b^k \mid b \in \mathbb{F}_q^* \right\} \subset \mathbb{F}_q^*$$

is a subgroup of order $(q-1)/k$. Moreover, prove that

$$S = \left\{ a \in \mathbb{F}_q^* \mid a^{(q-1)/k} = 1 \right\}.$$

Hint. Consider the endomorphisms of groups $\sigma_m : \mathbb{F}_q^* \to \mathbb{F}_q^*$, $a \mapsto a^m$, and use that each element of $\operatorname{Ker}(\sigma_m)$ is a zero of the polynomial $x^m - 1$.

Exercise 2.3.6. Let $q = 2^r$, $r \geq 1$, and let $f \in \mathbb{F}_q[x]$ be squarefree and monic, with the irreducible decomposition $f = f_1 \cdot \ldots \cdot f_s$. Let

$$\Phi_f : \mathbb{F}_q[x]/\langle f \rangle \to \mathbb{F}_q[x]/\langle f \rangle, \quad \overline{h} \mapsto \overline{h}^q - \overline{h},$$

let $h \in \mathbb{F}_q[x]$ be a representative of a randomly chosen $\overline{h} \in \operatorname{Ker}(\Phi_f) \setminus \{0\}$, and let $\operatorname{Tr} \in \mathbb{F}_q[x]$ denote the trace polynomial (see Remark 2.3.26). Prove that $\gcd\big(f, \operatorname{Tr}(h)\big)$ is a non-trivial factor of f with probability at least

$$1 - \left(\frac{1}{2}\right)^{s-1}.$$

Exercise 2.3.7. Let p be an odd prime, $q = p^r$, $r \geq 1$, and let $f \in \mathbb{F}_q[x]$ be squarefree and monic, with the irreducible decomposition $f = f_1 \cdot \ldots \cdot f_s$. Moreover, suppose that $\deg(f_i) = e$ for all $i = 1, \ldots, s$. Prove that

$$f = \gcd(f, h) \cdot \gcd\big(f, h^{(q^e-1)/2} - 1\big) \cdot \gcd\big(f, h^{(q^e-1)/2} + 1\big).$$

2.4 Factorization in $\mathbb{Z}[x]$, $\mathbb{Q}[x]$

The factorization problems for univariate polynomials in $\mathbb{Z}[x]$ and in $\mathbb{Q}[x]$ are closely related. We observe, first of all, that for a *primitive* polynomial $f \in \mathbb{Z}[x]$, we have the following equivalence

$$f \in \mathbb{Z}[x] \text{ irreducible} \iff f \in \mathbb{Q}[x] \text{ irreducible}.$$

Proof. The non-trivial of the two implications follows from the fact that every factorization of f in $\mathbb{Q}[x]$ gives rise to one in $\mathbb{Z}[x]$ in the following way: Let $f = f_1 f_2$ for some $f_1, f_2 \in \mathbb{Q}[x] \setminus \mathbb{Q}$, then we can find $n_1, n_2 \in \mathbb{Q}_{>0}$ such that $n_1 f_1$, $n_2 f_2 \in \mathbb{Z}[x]$ are primitive polynomials (by multiplying with the common denominator and then dividing by the content). We consider the following equation in $\mathbb{Q}[x]$:

$$(n_1 n_2) \cdot f = (n_1 f_1) \cdot (n_2 f_2). \tag{2.4.4}$$

Since $f \in \mathbb{Z}[x]$ is primitive, and since the two factors on the right-hand side are integer polynomials, we necessarily have $n_1 n_2 \in \mathbb{Z}$. Hence, both sides of (2.4.4) are decompositions in $\mathbb{Z}[x]$. As $\mathbb{Z}[x]$ is a UFD, and as $n_1 f_1$, $n_2 f_2$ are primitive, equality can only hold if $n_1 n_2 = 1$. It follows that (2.4.4) is a (non-trivial) factorization of f in $\mathbb{Z}[x]$. □

In the above argument, we used another important fact linking the two factorization problems: Every polynomial $g \in \mathbb{Q}[x]$ can be written as $g = \frac{a}{b} \cdot f$, where $f \in \mathbb{Z}[x]$ is a primitive polynomial, and where $\frac{a}{b} \in \mathbb{Q}$.

Combining these two observations, the problem of factorizing univariate polynomials f with integer coefficients turns out to be equivalent to a combination of the problem of factorizing the primitive part of f as a polynomial in $\mathbb{Q}[x]$ and of the problem of factorizing the content of f as a number in $\mathbb{Z}$:

$$\boxed{\text{factorization in } \mathbb{Z}[x] \longleftrightarrow \left\{ \begin{array}{c} \text{factorization in } \mathbb{Q}[x] \;\; \text{(of the primitive part)} \\ \& \;\; \text{factorization in } \mathbb{Z} \;\; \text{(of the content)}\,. \end{array} \right.}$$

Although a lot of research has been done on the problem of factorization in $\mathbb{Z}$, the algorithms developed are by far not as powerful as the algorithms developed for factorization of polynomials. In particular, no algorithm in “polynomial time” is known for this problem (see also Remark 2.0.1). Studying algorithms for factorization in $\mathbb{Z}$ is beyond the scope of this lecture, and the interested reader is referred to [vGG, Chapter 19] for an overview on this field.

From now on, we restrict our considerations to the problem of *factorizing primitive polynomials* $f = \sum_{i=0}^{d} a_i x^i \in \mathbb{Z}[x]$, $a_d \neq 0$, that is, we basically only treat the factorization problem in $\mathbb{Q}[x]$. Note that, in this context, Eisenstein's Criterion (Lemma 1.1.10) provides an (a priori) criterion for irreducibility: If there exists a prime $p \in \mathbb{Z}$ such that $f \equiv a_d x^d \bmod p$, and $a_0 \not\equiv 0 \bmod p^2$, then f is irreducible in $\mathbb{Q}[x]$.

2.4.1 Kronecker's Principle

The first algorithm for factorization in $\mathbb{Q}[x]$ is due to von Schubert (1793), respectively Kronecker (1882, 1883) (see [vSc, Kro, Kro1]). The idea behind this algorithm is quite simple: Let f be a non-constant polynomial in $\mathbb{Z}[x]$ which is the product of two non-trivial factors g and h, which we want to determine. Say

$$f = \underbrace{\left(\sum_{i=0}^{e} b_i x^i \right)}_{=\,g} \cdot h\,, \qquad b_e \neq 0\,.$$

Then, in particular, $g(a)$ divides $f(a)$, for all $a \in \mathbb{Z}$. Hence, for each $a \in \mathbb{Z}$, there must be an integer $c = c(a) \in \mathbb{Z}$ with $c \mid f(a)$, such that $\boldsymbol{b} = (b_e, \dots, b_0)^T \in \mathbb{Z}^{e+1}$

is a solution to the linear equation

$$\sum_{i=0}^{e} a^i \cdot b_i = c\,.$$

By choosing $e+1$ pairwise distinct integers $a_0, \ldots, a_e \in \mathbb{Z}$, we get a system of $e+1$ linear equations in $e+1$ indeterminates:

$$\underbrace{\begin{pmatrix} a_0^e & a_0^{e-1} & \ldots & \alpha_0 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ a_e^e & a_e^{e-1} & \ldots & \alpha_e & 1 \end{pmatrix}}_{=:\,A} \begin{pmatrix} b_e \\ \vdots \\ b_0 \end{pmatrix} = \begin{pmatrix} c_0 \\ \vdots \\ c_e \end{pmatrix} \tag{2.4.5}$$

with $c_i := c(a_i)$ for $i = 0, \ldots, e$. This system has a unique solution $\boldsymbol{b} \in \mathbb{Q}^{e+1}$, as the matrix A is a Vandermonde matrix with determinant $\prod_{i<j}(a_i - a_j)$ (see also Exercise 2.2.7). In this way, determining $\boldsymbol{b}$ comes basically down to solving a system of linear equations.

There are mainly two problems, when trying to turn this idea into an (efficient) algorithm: We do not know a priori the value of e , and we do not know which of the (not necessarily irreducible) factors c_i of $f(a_i)$ has to be chosen for setting up the system of equations. Hence, we have to try out all possibilities until we have either found a non-trivial factor or run out of possibilities. In the latter case, we then know that f is irreducible.

Algorithm 2.4.1 (Kronecker's Principle).

```
Input:  $f \in \mathbb{Z}[x] \setminus \mathbb{Z}$
Output: a non-trivial factor of $f$ in $\mathbb{Q}[x]$, or "$f$ is irreducible in $\mathbb{Q}[x]$"

(1) Choose $a_0 \neq a_1 \in \mathbb{Z}$ ;
(2) FOR $(i = 0, 1)$ DO
      • IF $\big(f(a_i) = 0\big)$ THEN RETURN $(x - a_i)$ ;
(3) Set $M := \big\{c \in \mathbb{Z}_{>0} \mid c \text{ divides } f(a_0)\big\}$ ;
(4) FOR $\big(e = 1, \ldots, \lfloor \frac{\deg f}{2} \rfloor\big)$ DO
      • Set $M_e := \big\{c \in \mathbb{Z} \mid c \text{ divides } f(a_e)\big\}$, and set $M' := M := M \times M_e$ ;
      • WHILE $(M' \neq \emptyset)$ DO
          ◦ Choose $\boldsymbol{c} = (c_0, \ldots, c_e)^T \in M'$ ;
          ◦ Compute the unique solution $\boldsymbol{b} = (b_e, \ldots, b_0) \in \mathbb{Q}^{e+1}$ of the system
            of linear equations (2.4.5) ;
```

```
            ◦ Set $g := \sum_{i=0}^{e} b_i x^i \in \mathbb{Q}[x]$ ;
            ◦ IF $\big((b_e \neq 0)$ AND $(g$ divides $f$ in $\mathbb{Q}[x])\big)$ THEN RETURN $(g)$ ;
            ◦ Set $M' := M' \setminus \{\boldsymbol{c}\}$ ;
        • Choose $a_{e+1} \in \mathbb{Z} \setminus \{a_0, \ldots, a_e\}$ ;
    (5) RETURN $\big($"$f$ is irreducible in $\mathbb{Q}[x]$"$\big)$.
```

Due to the fact that so many cases have to be tried out explicitly, this is an extremely expensive algorithm, in particular if f is irreducible (see the example below). Therefore, it is more of theoretical, respectively historical, than of practical relevance.

Example 2.4.2. We apply Kronecker's Principle to compute the irreducible factorization of $f := x^7 + 2x^5 + x^4 - 2x^3 + x^2 - 3x - 3 \in \mathbb{Q}[x]$:

Step 1. set $e := 1$, and choose $a_0 := 0$, $a_1 := 1$.

- Test: $f(a_0) = -3 \neq 0$, $f(a_1) = -3 \neq 0$.
- $M_0 := \{c \in \mathbb{Z}_{>0} \mid c \text{ divides } -3\} = \{1, 3\}$
- $M_1 := \{c \in \mathbb{Z} \mid c \text{ divides } -3\} = \{1, -1, 3, -3\}$.
- $M = M_0 \times M_1 = \left\{\binom{1}{1}, \binom{3}{1}, \binom{1}{-1}, \binom{3}{-1}, \binom{1}{3}, \binom{3}{3}, \binom{1}{-3}, \binom{3}{-3}\right\}$
- Solve the 8 systems of linear equations $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}\right) \cdot \binom{b_1}{b_0} = \binom{c_0}{c_1} \in M$:

$$\begin{pmatrix} b_1 \\ b_0 \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -2 \\ 3 \end{pmatrix}, \begin{pmatrix} -2 \\ 1 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} -4 \\ 1 \end{pmatrix}, \begin{pmatrix} -6 \\ 3 \end{pmatrix},$$

- Test: for all solutions $\binom{b_1}{b_0}$ with $b_1 \neq 0$, we have $(b_1 x + b_0) \nmid f$.

So far, we know that f has no irreducible factor of degree 1 in $\mathbb{Q}[x]$.

Step 2. set $e := 2$, and choose $a_2 := -1$.

- $f(a_2) = f(-1) = 1$.
- $M_2 := \{c \in \mathbb{Z} \mid c \text{ divides } 1\} = \{1, -1\}$.
- $M = M_0 \times M_1 \times M_2 = \left\{ \left(\begin{smallmatrix} 1 \\ 1 \\ 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 3 \\ 1 \\ 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 \\ -1 \\ 1 \end{smallmatrix}\right), \ldots, \left(\begin{smallmatrix} 1 \\ -3 \\ -1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 3 \\ -3 \\ -1 \end{smallmatrix}\right) \right\}$.
- Solve the 16 systems of linear equations $\left(\begin{smallmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \end{smallmatrix}\right) \cdot \left(\begin{smallmatrix} b_2 \\ b_1 \\ b_0 \end{smallmatrix}\right) = \left(\begin{smallmatrix} c_2 \\ c_1 \\ c_0 \end{smallmatrix}\right) \in M$:

$$\begin{pmatrix} b_2 \\ b_1 \\ b_0 \end{pmatrix} = \begin{pmatrix} -1 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & -\frac{1}{2} \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} c_2 \\ c_1 \\ c_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -2 \\ 0 \\ 3 \end{pmatrix}, \ldots, \begin{pmatrix} -3 \\ -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -5 \\ -1 \\ 3 \end{pmatrix}.$$

- Test: for all solutions $\left(\begin{smallmatrix} b_2 \\ b_1 \\ b_0 \end{smallmatrix}\right)$ with $b_2 \neq 0$, we have $(b_2x^2 + b_1x + b_0) \nmid f$.

So far, we know that f has no irreducible factor of degree ≤ 2 in $\mathbb{Q}[x]$.

Step 3. Set $e := 3$, and choose $a_3 := 2$.

- $f(a_3) = f(2) = 187 = 17 \cdot 11$.
- $M_3 := \{c \in \mathbb{Z} \mid c \text{ divides } 1\} = \{1, -1, 11, -11, 17, -17, 187, -187\}$.
- $M = M_0 \times \ldots \times M_3 = \left\{ \left(\begin{smallmatrix} 1 \\ 1 \\ 1 \\ 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 \\ 1 \\ -1 \\ 1 \end{smallmatrix}\right), \ldots, \left(\begin{smallmatrix} -3 \\ -3 \\ 1 \\ -187 \end{smallmatrix}\right), \left(\begin{smallmatrix} -3 \\ -3 \\ -1 \\ -187 \end{smallmatrix}\right) \right\}$.
- Solve the 128 systems of linear equations $\left(\begin{smallmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 8 & 4 & 2 & 1 \end{smallmatrix}\right) \cdot \left(\begin{smallmatrix} b_3 \\ b_2 \\ b_1 \\ b_0 \end{smallmatrix}\right) = \left(\begin{smallmatrix} c_3 \\ c_2 \\ c_1 \\ c_0 \end{smallmatrix}\right) \in M$:

$$\begin{pmatrix} b_3 \\ b_2 \\ b_1 \\ b_0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & -\frac{1}{6} & \frac{1}{6} \\ -1 & \frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{3} & -\frac{1}{6} \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} c_3 \\ c_2 \\ c_1 \\ c_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -\frac{1}{3} \\ -1 \\ \frac{2}{3} \\ 1 \end{pmatrix}, \ldots, \begin{pmatrix} -31 \\ 1 \\ 30 \\ -3 \end{pmatrix}.$$

Exactly one of these systems leads to an irreducible factor of f:

$$\begin{pmatrix} b_3 \\ b_2 \\ b_1 \\ b_0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & -\frac{1}{6} & \frac{1}{6} \\ -1 & \frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{3} & -\frac{1}{6} \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 3 \\ -1 \\ 11 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

We get $f = (x^3 + x + 1)(x^4 + x^2 - 3)$, which is the irreducible factorization of f in $\mathbb{Q}[x]$, due to the results of the previous steps of the algorithm.

2.4.2 A Modular Approach

A much more efficient approach to the factorization problem over the rationals is given by the modular algorithm due to Berlekamp and Zassenhaus which can be sketched as follows:

Algorithm 2.4.3 (Factorization in $\mathbb{Z}[x]$ using Hensel lifting).

Input: $f \in \mathbb{Z}[x] \smallsetminus \mathbb{Z}$

Assume: f is primitive and squarefree

Output: $f_1, \ldots, f_s \in \mathbb{Z}[x] \setminus \mathbb{Z}$ irreducible polynomials, such that $f = f_1 \cdot \ldots \cdot f_s$

(1) Choose a suitable prime $p \in \mathbb{N}$, such that

  - $p \nmid \mathrm{lc}(f)$,

- $\overline{f} \in \mathbb{F}_p[x]$ is squarefree (that is, $\overline{f}' \neq 0$ and $p \nmid D_f$).

  Note that a random choice of primes yields such p.

(2) Compute the irreducible factorization of $\overline{f} \in \mathbb{F}_p[x]$.

(3) Choose a positive integer N such that the coefficients of each (not necessarily irreducible) factor $g \in \mathbb{Z}[x]$ of f are integers in $\left(-\frac{1}{2}p^N, \frac{1}{2}p^N\right]$.

(4) *Hensel Lifting*: Lift the irreducible factorization in $\mathbb{F}_p[x]$ to $\overline{f} = \overline{g}_1 \cdot \ldots \cdot \overline{g}_\ell$ in $(\mathbb{Z}/p^N)[x]$.

(5) Combine the (unique) representatives $g_1, \ldots, g_\ell \in \mathbb{Z}[x]$ of $\overline{g}_1, \ldots, \overline{g}_\ell$ with coefficients in $(-\frac{1}{2}p^N, \frac{1}{2}p^N]$ in an appropriate way to obtain the *true factors* of f (see below).

The best way to get an overview of this algorithm is to consider the following diagram illustrating the steps of the algorithm:

$$\begin{tikzcd}
\mathbb{Z}[x] \arrow[rr, dashed, "\substack{\textit{Factorization}\\ \textit{Problem}}"] \arrow[dd, "(1)"'] & & \mathbb{Z}[x] \\
& & \mathbb{Z}/(p^N\mathbb{Z})[x] \arrow[u, "(5)"'] \\
\mathbb{F}_p[x] \arrow[rr, "(2)"'] & & \mathbb{F}_p[x] \arrow[u, "{(3), (4)}"']
\end{tikzcd}$$

For Steps (1) and (2) we refer to the result of the previous section (see Corollary 2.3.5 and Resumé 2.3.28.). Therefore, the only steps which need to be discussed here are Steps (3), (4) and (5). Before starting a detailed discussion of these steps, we should like to comment on the difficulties in Steps (3) and (5):

ad (3) There exists no (a priori) bound for the coefficients of a factor $g \in \mathbb{Z}[x]$ of an polynomial $f \in \mathbb{Z}[x]$ *which only depends on the size of the coefficients of* f. This becomes obvious when considering the following example:

$$f := x^n - 1 = \prod_{k=0}^{d-1}\left(x - e^{\pi i \frac{k}{d}}\right) = \prod_{d \mid n} \Phi_d$$

where $\Phi_1 := (x-1)$ and, for $d > 1$,

$$\Phi_d := \prod_{\substack{k=1 \\ \gcd(k,d)=1}}^{d-1} (x - e^{2\pi i \frac{k}{d}}) \in \mathbb{Z}[x]$$

(the Φ_d are called **cyclotomic polynomials**, see Exercise 2.4.3. Writing the cyclotomic polynomials as $\Phi_d = \sum_k a_{k,d} \cdot x^k$, there exists a sequence of coefficients $(a_{k(d),d})_{d\in\mathbb{N}}$ such that $|a_{k(d),d}|$ tends to infinity[10] for $d \to \infty$.

ad (5) Consider the primitive polynomial $f = x^4 + 2 \in \mathbb{Z}[x]$. By the criterion of Eisenstein, it is irreducible in $\mathbb{Q}[x]$ (and thus in $\mathbb{Z}[x]$). But, there are prime numbers p for which it is squarefree and reducible in $\mathbb{F}_p[x]$, for example:

$$\begin{array}{rcl} p = 3 & : & (x+1)(x-1)(x^2+1)\,, \\ p = 7 & : & (x^2-x-3)\cdot(x^2+x-3)\,, \\ p = 11 & : & (x+5)(x-5)(x^2+3)\,, \\ p = 17 & : & (x^2-7)(x^2+7)\,, \\ p = 19 & : & (x+5)(x-5)(x^2+6)\,. \end{array}$$

Having considered the possible difficulties which can arise in Steps (3) and (5), we now look at each of the Steps (3), (4), and (5) separetely:

ad Step (3): The Mignotte Bounds

The task in this step is to find an a priori bound for the possible coefficients of factors of a given polynomial $f = \sum_{i=0}^m a_i x^i \in \mathbb{C}[x]$. The central notions for formulating and proving such a bound are the **quadratic norm** of f,

$$\|f\|_2 := \sqrt{\sum_{i=0}^m |a_i|^2}\,,$$

and the **measure** of f,

$$M(f) := \bigl|\operatorname{lc}(f)\bigr| \cdot \prod_{j=1}^m \max\{1, |z_j|\}\,,$$

where $z_1, \dots, z_m \in \mathbb{C}$ are the complex roots of f. These notions are related to each other by the following lemma:

Lemma 2.4.4 (Landau). $M(f) \le \|f\|_2$.

Proof. Denote by $z_1, \dots, z_m$ the roots of f, and assume that they are numbered in such a way that the first k of them are the ones with $|z_j| > 1$. Therefore,

$$M(f) = \bigl|\operatorname{lc}(f)\bigr| \cdot \prod_{j=1}^k |z_j|\,.$$

---

[10] When computing the first few cyclotomic polynomials explicitly, one might get the impression that this statement is not correct, because only coefficients 0 and ±1 appear in those polynomials where $d < 105$. The first d for which there is a coefficient $a_{k(d),d} \notin \{0, \pm 1\}$ is $d = 3\cdot 5\cdot 7 = 105$, the smallest product of 3 odd prime numbers. In fact, it can be shown by more involved arguments that any natural number can appear as a coefficient of a cyclotomic polynomial.

Now, consider the polynomial

$$g := \operatorname{lc}(f) \cdot \prod_{j=1}^{k} \bigl(\overline{z_j} x - 1\bigr) \cdot \prod_{j=k+1}^{m} \bigl(x - z_j\bigr) .$$

It satisfies

$$\bigl|\operatorname{lc}(g)\bigr| = \bigl|\operatorname{lc}(f)\bigr| \cdot \prod_{j=1}^{k} \bigl|\overline{z}_j\bigr| = M(f) ,$$

whence $\|g\|_2 \geq \bigl|\operatorname{lc}(g)\bigr| = M(f)$. On the other hand, it is a rather direct observation that for all $P = \sum_{i=0}^{m} p_i x^i \in \mathbb{C}[x]$, and for all $z \in \mathbb{C}$, the following holds (with $p_{-1} := p_{m+1} := 0$):

$$\begin{aligned} \bigl\|(\overline{z}x - 1) \cdot P\bigr\|_2^2 &= \sum_{i=0}^{m+1} \bigl(\overline{z} p_{i-1} - p_i\bigr) \cdot \bigl(z \overline{p}_{i-1} - \overline{p}_i\bigr) \\ &= \sum_{i=0}^{m} \overline{z} z \overline{p}_i p_i + \sum_{i=0}^{m} \overline{p}_i p_i - \sum_{i=0}^{m+1} \bigl(\overline{z} p_{i-1} \overline{p}_i + z \overline{p}_{i-1} p_i\bigr) \\ &= \sum_{i=0}^{m+1} \bigl(z p_i - p_{i-1}\bigr) \cdot \bigl(\overline{z p}_i - \overline{p}_{i-1}\bigr) \;=\; \bigl\|(z - x) \cdot P\bigr\|_2^2 . \end{aligned}$$

Iterating the application of this process k times proves that $\|g\|_2^2 = \|f\|_2^2$. Combined with the previous bound $\|g\|_2 \geq M(f)$, this proves the assertion of the lemma. ☐

The previous lemma allows us to formulate the desired bound for the possible coefficients of a factor of the given polynomial f:

Corollary 2.4.5. *Let* $g = \sum_{i=0}^{k} b_i x^i \in \mathbb{Z}[x]$ *be a factor of* f*. Then,*

$$|b_i| \leq \binom{k}{\lfloor \frac{k}{2} \rfloor} \cdot \|f\|_2 , \quad \textit{for all } i = 0, \ldots, k .$$

Proof. The coefficients of g are sums of products of the complex roots of f multiplied by $\operatorname{lc}(f)$. Hence,

$$|b_i| \;\leq\; M(f) \cdot \#(\text{products of roots contributing to the term } b_i x^i \text{ in } g) .$$

The number of non-zero summands contributing to the term $b_i x^i$ of g is bounded by the number of subsets of cardinality i in a set of k elements, that is, by $\binom{k}{i}$. Combining this with the well-known inequality

$$\binom{k}{i} \leq \binom{k}{\lfloor \frac{k}{2} \rfloor} ,$$

we obtain the desired bound. ☐

This allows us to formulate the condition on the choice of the exponent N in the modular factorization algorithm explicitly[11]:

N has to be chosen sufficiently large to satisfy $p^N \geq 2^{\deg(f)+1} \cdot \|f\|_2$.

ad Step (4): Hensel Lifting

This step is the central part of the modular approach for factorization in characteristic zero. It allows us to transfer the result of the factorization in $\mathbb{F}_p[x]$ to $(\mathbb{Z}/p^N\mathbb{Z})[x]$, where N is chosen according to the bound in Step (3). Due to this condition on N, the coefficients of the irreducible factors of the polynomial f lie in the range $(-\frac{p^N}{2}, \frac{p^N}{2}]$. On the other hand, the liftings of the irreducible factors of the homomorphic image $\overline{f} \in \mathbb{F}_p[x]$ have unique representatives $g_1, \dots, g_s \in \mathbb{Z}[x]$ with coefficients in this range. Thus, this lifting step provides us with a finite list of candidates for the irreducible factors of f (each irreducible factor has to be the product of some of the g_i, $i = 1, \dots, s$, see Step (5)).

For specifying the algorithm, let us consider the following situation: Let f be a polynomial in $\mathbb{Z}[x]$ and let $\overline{f}$ be its homomorphic image in $\mathbb{F}_p[x]$. Let us assume that we have found a non-trivial decomposition of $\overline{f}$ into two factors $\overline{f} = \overline{g} \cdot \overline{h} \in \mathbb{F}_p[x]$ where $\overline{g}$ and $\overline{h}$ are coprime, that is, $\gcd(\overline{g}, \overline{h}) = 1$. Applying the extended Euclidean algorithm (see Remark 2.1.5) leads to polynomials $\overline{s}, \overline{t} \in \mathbb{F}_p[x]$ such that

$$\overline{s} \cdot \overline{g} + \overline{t} \cdot \overline{h} = 1 \in \mathbb{F}_p[x], \qquad \deg \overline{s} < \deg \overline{h}\,, \;\; \deg \overline{t} < \deg \overline{g}\,.$$

Now, these data are lifted successively from $\mathbb{F}_p[x]$ to $(\mathbb{Z}/p^{2^k}\mathbb{Z})[x]$, $k = 1, 2, 3, \dots$:

Algorithm 2.4.6 (Hensel Lifting).

```
Assume: $R$ is an integral domain and $q \in R$.
Input:  $f, g, h, s, t \in R[x]$, where $q \nmid \mathrm{lc}(f)$, $q \nmid \mathrm{lc}(g)$, $q \nmid \mathrm{lc}(s)$, $q \nmid \mathrm{lc}(t)$, and
        (a) $h$ is monic,
        (b) $\deg(s \bmod q) < \deg(h \bmod q)$, $\deg(t \bmod q) < \deg(g \bmod q)$,
        (c) $f \equiv g \cdot h \bmod q$,
        (d) $sg + th \equiv 1 \bmod q$.
Output: $g^*, h^*, s^*, t^* \in R[x]$, satisfying
        (a) $h^*$ is monic,
        (b) $\deg(s^* \bmod q^2) < \deg(h^* \bmod q^2)$, $\deg(t^* \bmod q^2) < \deg(g^* \bmod q^2)$,
        (c) $f \equiv g^* h^* \bmod q^2$,
        (d) $s^* g^* + t^* h^* \equiv 1 \bmod q^2$,
        (e) $g^* \equiv g \bmod q$, $h^* \equiv h \bmod q$.

 (1) Set $\Delta := f - gh$,
         $g^* := g \cdot \big(1 + (s\Delta \operatorname{quo} h)\big) + t\Delta$,
         $h^* := h + (s\Delta \operatorname{rem} h)$;
```

---

[11] When formulating this inequality, we also used the fact that $2^k \geq \binom{k}{\lfloor k/2 \rfloor}$.

```
(2) Set $\delta := sg^* + th^* - 1$,
        $s^* := s - (s\delta \operatorname{rem} h^*)$,
        $t^* := (1-\delta)\cdot t - g^* \cdot (s\delta \operatorname{quo} h^*)$;

(3) RETURN $(g^*, h^*, s^*, t^*)$.
```

Remark 2.4.7. We are usually dealing with $R = \mathbb{Z}$, respectively with $R = K[y]$, K a field. The element q will then be some power p^r, $r \geq 1$, of a prime number p, respectively of an irreducible polynomial $p \in K[y]$. The condition $q \nmid \operatorname{lc}(f)$ is always satisfied by the choice of p (see Step (1) of Algorithm 2.4.3). Moreover, as the requirements (b) – (d) on the input polynomials are only conditions on the residue class $\bmod\ q$, it is natural to assume $q \nmid \operatorname{lc}(g)$, $q \nmid \operatorname{lc}(s)$, and $q \nmid \operatorname{lc}(t)$. If one of these conditions is not satisfied, we may simply pass to another representative of the class of the polynomial $\bmod\ q$.

Note that in practical computations, one is keen to have a unique way of choosing representatives in $R[x]$ for the classes $\bmod\ q$, keeping the coefficients as small as possible:

- For $R = \mathbb{Z}$, the elements of $R/\langle q\rangle$ are uniquely represented by elements inside the range $\left(-\frac{q}{2}, \frac{q}{2}\right]$.

- For $R = K[y]$, the elements $\overline{f} \in K[y]/\langle q\rangle$ are uniquely represented by the remainder of $f \in K[y]$ on division by q. [12]

These choices of representatives perfectly fit with the above, as they provide us with polynomials in $R[x]$ whose leading coefficients are not divisible by q.

Proof of Algorithm 2.4.6. First of all, the steps of the algorithm are well-defined, since division with remainder by a *monic* polynomial $h \in R[x]$ is well-defined (see Lemma 2.1.6). Moreover, we observe that h^* is also a monic polynomial, of degree $\deg(h^*) = \deg(h)$, as $\deg(s\Delta \operatorname{rem} h) < \deg(h)$ by definition of the remainder. The remaining properties are proved by several explicit calculations.

As $\Delta \equiv 0 \bmod q$, we see right away that (e) holds: $g^* \equiv g \bmod q$, $h^* \equiv h \bmod q$. Another consequence is that there exists a $\widetilde{\Delta} \in R[x]$ such that $\Delta = q \cdot \widetilde{\Delta}$. Equipped

---

[12] Note that this remainder is independent of the choice of the representative $f \in K[y]$ for $\overline{f}$.

with this last observation, we can show that the output data satisfy property (c):

$$\begin{aligned}
g^*h^* &= \Big(g\cdot\big(1+(s\Delta \operatorname{quo} h)\big)+t\Delta\Big)\cdot\big(h+(s\Delta \operatorname{rem} h)\big)\\
&= gh+t\Delta h+\underbrace{gh(s\Delta \operatorname{quo} h)+g(s\Delta \operatorname{rem} h)}_{=\,\Delta sg}\\
&\qquad +t\Delta(s\Delta \operatorname{rem} h)+g(s\Delta \operatorname{quo} h)(s\Delta \operatorname{rem} h)\\
&= gh+\Delta\underbrace{(th+sg)}_{=\,1}\\
&\qquad +tq\widetilde{\Delta}\big(sq\widetilde{\Delta} \operatorname{rem} h\big)+g\big(sq\widetilde{\Delta} \operatorname{quo} h\big)\big(sq\widetilde{\Delta} \operatorname{rem} h\big)\\
&\equiv \underbrace{gh+\Delta}_{=\,f} \bmod q^2\,.
\end{aligned}$$

Here, we used that $sq\widetilde{\Delta} \operatorname{quo} h = q\cdot(s\widetilde{\Delta} \operatorname{quo} h)$ and $sq\widetilde{\Delta} \operatorname{rem} h = q\cdot(s\widetilde{\Delta} \operatorname{rem} h)$.

In the same way as before, $\delta\equiv 0 \bmod q$ (which holds due to (e) and the assumption (d) on the input data) implies that there is a $\widetilde{\delta}\in R[x]$ such that $\delta = q\cdot\widetilde{\delta}$. We use this to deduce condition (d):

$$\begin{aligned}
s^*g^*+t^*h^* &= \Big(s-(s\delta \operatorname{rem} h^*)\Big)\cdot g^*+\Big((1-\delta)t-g^*(s\delta \operatorname{quo} h^*)\Big)\cdot h^*\\
&= sg^*+(1-\delta)\cdot th^*-\underbrace{\big((s\delta \operatorname{rem} h^*)\cdot g^*+h^*\cdot(s\delta \operatorname{quo} h^*)g^*\big)}_{=\,s\delta g^*}\\
&= \big(1-\delta\big)\cdot\underbrace{(sg^*+th^*)}_{=\,1+\delta} \;=\; 1-q^2\cdot\widetilde{\delta}^2\\
&\equiv 1 \bmod q^2\,.
\end{aligned}$$

What remains to be shown is property (b) for the output data. First of all, we recall that $\deg(h^*)=\deg(h)$, as we observed at the beginning of the proof. Hence, s^* differs from s only by terms of degree at most $\deg(h)-1$. Together with the assumption (b) for the input data, this implies that $\deg(s^*)<\deg(h)=\deg(h^*)$.

For the comparison of the degrees of t^* and g^*, we need to consider these polynomials more closely: As $\operatorname{lc}(g)$ is not divisible by q, and as $g^*\equiv g \bmod q$, we know that $\deg(g^* \bmod q^2)\geq\deg(g)$. On the other hand,

$$g^* \;=\; g+\Big(\underbrace{(\Delta th+\Delta sg)}_{\equiv\,\Delta \bmod q^2} \operatorname{quo} h\Big) \;\equiv\; g+(\Delta \operatorname{quo} h) \bmod q^2, \tag{2.4.6}$$

and assumption (c) on the input data gives $\deg(\Delta)\leq\deg(f)=\deg(gh)$, hence, $\deg(\Delta \operatorname{quo} h)\leq\deg(g)$. Together with the above, this implies

$$\deg(g^* \bmod q^2)=\deg(g)\,. \tag{2.4.7}$$

For t^*, we compute:

$$t^* \;=\; t - \Big(\underbrace{(\delta t h^* + \delta s g^*)}_{=\,\delta\cdot(1+\delta)} \operatorname{quo} h^*\Big) \;=\; t - (\delta \operatorname{quo} h^*) - q^2 \cdot \Big(\widetilde{\delta}^2 \operatorname{quo} h^*\Big).$$

The equality $\delta \equiv 0 \bmod q$, together with (2.4.7) and the assumptions on the leading coefficients, implies that $\deg(\delta \bmod q^2) \le \deg(s) + \deg(g) = \deg(t) + \deg(h^*)$. Thus, $\deg\big((\delta \operatorname{quo} h^*) \bmod q^2\big) \le \deg(t)$, that is,

$$\deg(t^* \bmod q^2) \le \deg(t \bmod q^2)\,. \qquad \square$$

Remark 2.4.8. The equality (2.4.6) shows that by always removing all terms divisible by q^2 in the Hensel Lifting Algorithm we may assume additionally:

- $\operatorname{lc}(g^*) = \operatorname{lc}(g) = \operatorname{lc}(f)$.
- $\deg(g^*) = \deg(g)\,,$

Proposition 2.4.9 (Uniqueness of Hensel Lifting). *Let R be a ring, $q \in R$ a non-zerodivisor, $k \ge 2$, and let $g, h, g^*, h^*, s, t \in R[x] \setminus \{0\}$ such that*

(1) $g \equiv g^* \bmod q$, $\deg(g) = \deg(g^*)$, *and* $\operatorname{lc}(g) = \operatorname{lc}(g^*)\,,$
$h \equiv h^* \bmod q$, $\deg(h) = \deg(h^*)$, *and* $\operatorname{lc}(h) = \operatorname{lc}(h^*)\,,$

(2) $\operatorname{lc}(g)$, $\operatorname{lc}(h)$ *are non-zerodivisors in* $R/\langle q\rangle\,,$

(3) $sg + th \equiv 1 \bmod q\,,$

(4) $gh \equiv g^*h^* \bmod q^k\,.$

Then $g \equiv g^* \bmod q^k$ *and* $h \equiv h^* \bmod q^k$.

Proof. To prove this assertion, let us assume the converse, that is, that g and g^* (or h and h^*) do not coincide modulo q^k. Then there is a maximal integer $1 \le j < k$ with the property $g^* - g,\ h^* - h \in \langle q^j\rangle$. We can find suitable polynomials $u, v \in R[x]$ such that $g^* - g = u \cdot q^j$, $h^* - h = v \cdot q^j$, and either $q \nmid u$ or $q \nmid v$. By the symmetry of the statements in g and h, we may assume that $q \nmid u$.

The conditions (1) and (2) imply that the leading terms cancel in $g^* - g$ and in $h^* - h$. Hence, $\deg(u) < \deg(g)$ and $\deg(v) < \deg(h)$. Moreover, the assumption (4) implies that

$$(g^*v + hu) \cdot q^j \;=\; g^*(h^* - h) + h(g^* - g) \;=\; g^*h^* - gh \;\equiv\; 0 \;\bmod\; q^k\,.$$

Knowing that q is a non-zerodivisor, we can immediately deduce from this calculation that $q^{k-j} \mid (g^*v + hu)$. In particular, this implies that $0 \equiv gv + hu \bmod q$. Thus,

$$u + g(tv - su) \;=\; gtv + u(1 - gs) \;\overset{(3)}{\equiv}\; gtv + uth \;=\; t(gv + hu) \;\equiv\; 0 \;\bmod\; q\,.$$

The last statement obviously implies that g divides u in $R/\langle q\rangle$. On the other hand, we had previously observed that $\deg(g \bmod q) = \deg(g) > \deg(u) \ge \deg(u \bmod q)$. Therefore, u can only be in the class of 0 in $R/\langle q\rangle$, which contradicts the assumption that $q \nmid u$. $\square$

ad Step (5): True Factors

The last step in the modular factorization algorithm which has not been discussed yet is the recombination step, whose importance has already been illustrated by the example right after the sketch of the modular algorithm. The basic idea behind the recombination step is quite simple: Try out all possible combinations of the lifted factors obtained in Step (4). However, the explicit formulation as an algorithm is a bit technical:

Algorithm 2.4.10 (True Factors).

```
Input:   $f \in \mathbb{Z}[x]$, $p, N \in \mathbb{N}$, and $g_1, \dots, g_\ell \in \mathbb{Z}[x]$.
Assume:  $f$ is squarefree and primitive, $p$ and $N$ are chosen as in Steps (1) and (3)
         of Algorithm 2.4.3, and $g_1, \dots, g_\ell$ are monic and satisfy
           • $f \equiv \mathrm{lc}(f) \cdot g_1 \cdot \ldots \cdot g_\ell \mod p^N$,
           • $\sum_{i=1}^{\ell} \deg(g_i) = \deg(f)$.
Output:  irreducible polynomials $f_1, \dots, f_r \in \mathbb{Z}[x]$ such that $f = f_1 \cdot \ldots \cdot f_r$.

(1) $h := f$, $L := \mathrm{lc}(f)$, Result $:= \emptyset$,            // Initialization
    $d := 1$, $I := \{1, \dots, \ell\}$;
(2) WHILE $(2d \leq \#I)$ DO
      • Set $\mathscr{S} := \{S \subset I \mid \#S = d\}$;
      • WHILE $\big((\mathscr{S} \neq \emptyset)$ AND $(2d \leq \#I)\big)$ DO
          ○ Choose $S \in \mathscr{S}$;
          ○ Set $\mathscr{S} := \mathscr{S} \setminus \{S\}$;
          ○ Compute $\overline{g} := L \cdot \prod_{i \in S} g_i \mod p^N$, and represent $\overline{g}$ by its unique
            representative $g \in \mathbb{Z}[x]$ with coefficients in $\left(-\frac{1}{2}p^N, \frac{1}{2}p^N\right]$;
          ○ IF $(g \mid L \cdot h)$ THEN
              □ Result := Result $\cup \{\mathrm{pp}(g)\}$;
              □ $h := \dfrac{h}{\mathrm{pp}(g)}$;
              □ $I := I \setminus S$;
              □ $\mathscr{S} := \mathscr{S} \setminus \{T \subset I \mid T \cap S \neq \emptyset\}$;
          ○ $d := d + 1$;
(3) IF $\big(\deg(h) > 0\big)$ THEN Result := Result $\cup \{h\}$;
(4) RETURN (Result).
```

Proof. Step 1. We show that each factor g of f satisfies $g \equiv a \cdot \prod_{i\in S} g_i \bmod p^N$, where $a \in \mathbb{Z}$ is some divisor of $L = \mathrm{lc}(f)$.

Let $f = g \cdot h \in \mathbb{Z}[x]$ be a decomposition of f into non-trivial factors g and h. Then

$$\overline{g} \cdot \overline{h} = \overline{\mathrm{lc}(f)} \cdot \overline{g}_1 \cdot \ldots \cdot \overline{g}_\ell \in \mathbb{F}_p[x]\,.$$

As $\mathbb{F}_p[x]$ is a UFD and as $h, g_1, \ldots, g_\ell$ are monic, it follows that

$$\mathrm{lc}(g) \cdot h \;\equiv\; \mathrm{lc}(f) \cdot \prod_{i\in S} g_i \mod p\,,$$

for a suitable subset $S \subset \{1, \ldots, \ell\}$. On the other hand, we know

$$\big(\mathrm{lc}(g)\cdot h\big) \cdot \big(\mathrm{lc}(h)\cdot g\big) \;=\; \mathrm{lc}(f) \cdot f \;\equiv\; \Big(\mathrm{lc}(f) \cdot \prod_{i\in S} g_i\Big) \cdot \Big(\mathrm{lc}(f) \cdot \prod_{i\notin S} g_i\Big) \mod p^N.$$

By the uniqueness of the Hensel lifting (see Proposition 2.4.9), this implies

$$\mathrm{lc}(g) \cdot h \;\equiv\; \mathrm{lc}(f) \cdot \prod_{i\in S} g_i \bmod p^N. \tag{2.4.8}$$

Step 2. We show that precisely the irreducible factors of f are returned.

First of all, we observe that each polynomial included into the set Result during the inner WHILE loop is an irreducible factor of h (which itself is primitive, squarefree and a factor of f): Indeed, each such polynomial is primitive and divides $\mathrm{lc}(f) \cdot h$. Hence, it is a factor of h. Moreover, as h is squarefree, Step 1 shows that, for d fixed, all irreducible factors of h which result from the combination of d of the factors g_i, $i \in I$, are found. Thus, already having passed $d - 1$ times through the outer WHILE loop, we know that h has no factor resulting from the combination of less than d of the g_i, $i = 1, \ldots, \ell$. In particular, the polynomials included into Result have no non-trivial factors.

Finally, we leave the outer WHILE loop with a primitive factor h of f which has to be irreducible. Indeed, at this stage, h could only factorize into two non-trivial factors which result from the combination of $d > \frac{\#I}{2}$ of the g_i, $i \in I$. But, this is impossible due to the condition for leaving the loop. □

Example 2.4.11. We apply Algorithm 2.4.3 to compute the irreducible factorization of $f := x^7 + 2x^5 + x^4 - 2x^3 + x^2 - 3x - 3 \in \mathbb{Q}[x]$ (see Example 2.4.2):

Steps (1) - (3). We choose $p = 7$, and compute the irreducible factorization

$$\overline{f} = (x^3 + x + 1) \cdot (x^2 + 3x - 2) \cdot (x^2 - 3x - 2) \in \mathbb{F}_7[x]\,. \tag{2.4.9}$$

Moreover, we choose $N := 4$, as $7^4 \geq 2^8 \cdot \sqrt{29} = 2^{\deg(f)+1} \cdot \|f\|_2$.

Step (4). We lift the factorization (2.4.9) to $(\mathbb{Z}/7^4\mathbb{Z})[x]$ by applying two successive Hensel lifting steps:

$\boxed{\mathbb{F}_7[x] \to (\mathbb{Z}/7^2\mathbb{Z})[x]}$ Let

$$g := x^3 + x + 1\,, \quad h := x^4 + x^2 - 3 \equiv (x^2 + 3x - 2)(x^2 - 3x - 2) \mod 7\,.$$

Then the extended Euclidean algorithm provides us with

$$s := x^3 - 3x^2 + 3x - 1\,, \qquad t := -x^2 + 3x - 3\,,$$

satisfying $st + gh \equiv 1 \mod 7$. Now, we compute (via Hensel Lifting):

$$g^* := x^3 + x + 1\,, \qquad h^* := x^4 + x^2 - 3\,,$$

(satisfying $g^*h^* \equiv f \mod 49$), and

$$s^* = 22x^3 - 17x^2 + 24x - 22\,, \qquad t^* = -22x^2 + 17x - 24\,,$$

satisfying $s^*t^* + g^*h^* \equiv 1 \mod 49$.

In the same way, we lift the factorization

$$x^4 + x^2 - 3 \;\equiv\; \underbrace{(x^2 + 3x - 2)}_{=:\,h_1}\underbrace{(x^2 - 3x - 2)}_{=:\,h_2} \mod 7\,.$$

Here, we get

$$h_1^* = x^2 - 11x + 12\,, \quad h_2^* = x^2 + 11x + 12\,, \quad t_1^* = -18x - 2\,, \quad t_2^* = 18x - 2\,,$$

satisfying $h_1^*h_2^* \equiv h \mod 49$ and $t_1^*h_1^* + t_2^*h_2^* \equiv 1 \mod 49$. Altogether, we have the lifted factorization

$$f \equiv (x^3 + x + 1)(x^2 - 11x + 12)(x^2 + 11x + 12) \mod 49\,.$$

$\boxed{(\mathbb{Z}/7^2\mathbb{Z})[x] \to (\mathbb{Z}/7^4\mathbb{Z})[x]}$ Proceeding as before, we get the lifted factorization

$$f \equiv (x^3 + x + 1)(x^2 + 185x + 306)(x^2 - 185x + 306) \mod 2401\,.$$

Step (5). We compute the true factors of f along the lines of Algorithm 2.4.10. First, we have to check combinations of $d = 1$ of the lifted factors:

- $g = x^3 + x + 1$ divides f (and is primitive). Thus, we set

$$\text{Result} := \left\{x^3 + x + 1\right\}, \qquad h := \frac{f}{x^3 + x + 1} = x^4 + x^2 - 3\,.$$

- $x^2 + 185x + 306$ and $x^2 - 185x + 306$ do not divide h.

Setting $d := 2$, the condition for leaving the outer WHILE-loop is satisfied. Thus, we get

$$\text{Result} = \left\{x^3+x+1\,,\ x^4+x^2-3\right\}.$$

Of course, we should mention that with another choice of prime number (say, $p = 5$) we could immediately get the correct number 2 of irreducible factors. Thus, sometimes, it may be worthwhile to compute several factorizations of a given polynomial modulo different primes before turning to the Hensel lifting step (see also Exercise 2.4.1).

With this example, we close our discussion of the univariate factorization problem over the integers, respectively over the rational numbers. More refined algorithms for these tasks, and a discussion of the practical relevance, advantages and disadvantages of these algorithms can be found, for example, in [vGG].

Exercises

Exercise 2.4.1. Suppose that $f \in \mathbb{Z}[x]$ has degree 12, and that $p \in \mathbb{N}$ is a prime such that the homomorphic image $\overline{f}$ of f in $\mathbb{F}_p[x]$ decomposes as $\overline{f} = \overline{g}_1 \cdot \overline{g}_2 \cdot \overline{g}_3$, where $\overline{g}_1, \overline{g}_2, \overline{g}_3 \in \mathbb{F}_p[x]$ are monic and irreducible with $\deg(g_1) = 2$, $\deg(g_2) = 3$, $\deg(g_3) = 7$.

(a) What can you say about the irreducible factorization of the image of f in $(\mathbb{Z}/p^{50}\mathbb{Z})[x]$?

(b) What can you say about the irreducible factorization of f in $\mathbb{Q}[x]$?

(c) Suppose there is another prime p' such that the homomorphic image of f in $\mathbb{F}_{p'}[x]$ decomposes into two irreducible factors of degree 6. What can you say then about the irreducible factorization of f in $\mathbb{Q}[x]$?

Exercise 2.4.2. Trace Algorithm 2.4.3 with input the primitive polynomials

(a) $f = x^4 + 3x^3 - 13x^2 + 6x - 30 \in \mathbb{Z}[x]$,

(b) $f = 12x^3 + 10x^2 - 36x + 35 \in \mathbb{Z}[x]$.

Exercise 2.4.3 (Cyclotomic Polynomials). Let $d, \ell \in \mathbb{N}$, $d \geq 2$ and $\ell \geq 1$. Moreover, let

$$\Phi_d := \prod_{\substack{k=1\\ \gcd(k,d)=1}}^{d-1} (x - e^{2\pi i \frac{k}{d}}) \in \mathbb{C}[x]$$

denote the d-th cyclotomic polynomial. Prove the following statements:

(a) If d is a prime number, then $\Phi_d = \sum\limits_{i=0}^{d-1} x^i$.

(b) If $d = 2\ell$ with $\ell \geq 3$ odd, then $\Phi_d = \Phi_\ell(-x)$.

(c) If $\gcd(\ell, d) = 1$, then $\Phi_{\ell d} = \dfrac{\Phi_d(x^\ell)}{\Phi_d}$.

(d) If every prime divisor of ℓ divides d, then $\Phi_{\ell d} = \Phi_d(x^\ell)$.

Deduce that the cyclotomic polynomials are monic polynomials with integer coefficients. Moreover, compute Φ_{105}.

2.5 Multivariate Factorization

Our next goal is the factorization of multivariate polynomials. Indeed, the solution for this problem is similar to the solution for the factorization problem in $\mathbb{Z}[x]$. We will discuss a direct (but non-efficient) approach and a modular approach to the multivariate factorization problem. For this modular approach, factorization in $\mathbb{Z}[x]$ will be the basic task (as factorization in $\mathbb{F}_p[x]$ was in Algorithm 2.4.3).

2.5.1 Kronecker's Algorithm

Also the direct approach reduces the multivariate factorization problem to a univariate one. The basic idea is simple: we just have to substitute each variable by a sufficiently high power of a new variable. More precisely, consider the following ring homomorphism:

$$\Psi : \mathbb{Z}[\boldsymbol{x}] = \mathbb{Z}[x_1, \ldots, x_n] \longrightarrow \mathbb{Z}[y] , \quad f(x_1, \ldots, x_n) \longmapsto f(y, y^e, \ldots, y^{e^{n-1}}) ,$$

which induces an isomorphism

$$\Psi : \big\{ f \in \mathbb{Z}[\boldsymbol{x}] \bigm| \deg_{x_i}(f) < e \text{ for all } i \big\} \xrightarrow{\cong} \mathbb{Z}[y]_{<e^n} .$$

In this way, it is possible to compute a univariate factorization of the polynomial $\Psi(f)$ and deduce a factorization of the original polynomial f from it.

Instead of $\mathbb{Z}$, this idea applies to each UFD R. We formulate the resulting algorithm in this more general setting:

Algorithm 2.5.1 (Kronecker).

Assume: R is a UFD, and there exists an efficient algorithm for univariate factorization over R.

Input: $f \in R[x_1, \ldots, x_n]$.

Output: A list of polynomials f_i, $i = 1, \ldots, s$, such that $f = f_1 \cdot \ldots \cdot f_s$ is an irreducible factorization of f.

(1) Choose $e \in \mathbb{Z}$ such that $e > \deg_{x_i}(f)$ for all $i = 1, \ldots, n$;

```
(2) Set $g := \Psi(f) := f\big(y, y^e, \dots, y^{e^{n-1}}\big)$.
(3) Compute an irreducible factorization $g = g_1 \cdot \ldots \cdot g_s$ in $R[y]$; [13]
(4) $h := f$,  Result $:= \emptyset$,  $I := \{1, \dots, s\}$,  $d := 1$;
(5) WHILE $(2 \cdot d \leq \#I)$ DO                    // True factors step
      • Set $\mathscr{S} := \big\{S \subset I \,\big|\, \#S = d\big\}$;
      • WHILE $\big((\mathscr{S} \neq \emptyset)$ AND $(2d \leq \#I)\big)$ DO
          ◦ Choose $S \in \mathscr{S}$;
          ◦ Set $\mathscr{S} := \mathscr{S} \setminus \{S\}$;
          ◦ Compute $g := \Psi^{-1}\left(\prod_{i \in S} g_i\right) \in R[x_1, \dots, x_n]$;
          ◦ IF $(g \mid h)$ THEN
              □ Result := Result, $g$;
              □ $f := \dfrac{f}{g}$;
              □ $I := I \setminus S$;
              □ $\mathscr{S} := \mathscr{S} \setminus \{T \subset I \,\big|\, T \cap S \neq \emptyset\}$;
      • $d := d + 1$;
(6) IF $\big(\deg(f) > 0\big)$ THEN Result := Result, $\{\Psi^{-1}(f)\}$;
(7) RETURN (Result).
```

Remark 2.5.2. (1) The correctness of the algorithm follows from the considerations in Section 2.4 and from the fact that all possible factors f_i of f satisfy $\deg_{x_j}(f_i) < e$, since there is no (non-trivial) zerodivisor in R. Hence, no ambiguity concerning the preimages of the factors under the map Ψ can occur.

(2) The main drawback of this algorithm is its running time: First of all, the algorithm depends very much on the efficiency of the underlying univariate factorization algorithm, since the degree of the univariate polynomials which have to be dealt with is $\deg\big(\Psi(f)\big) \sim e^n$, which obviously tends to be quite large.

In addition to this, the "true factors" step may take exponential time in $\deg(f)$; an example of a particularly bad situation is that f is irreducible, but $\Phi(f)$ decomposes completely into linear factors.[14]

---

[13] Note that the polynomials $g_1, \dots, g_s$ are irreducible, but not necessarily pairwise coprime.

[14] For multivariate *complex* polynomials, the corresponding univariate polynomial $\Phi(f)$ *always decomposes* into linear factors, while the multivariate polynomial may well be irreducible. Even for multivariate factorization over the integers, this situation occurs quite frequently.

2.5.2 Multivariate Factorization in $\mathbf{K}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\mathbf{n}]$

Due to the lack of efficiency, the previous approach to the reduction of multivariate factorization to univariate factorization is not of practical relevance. The usual way to do this reduction in practice is a modular approach, which is, in some sense, very close to Algorithm 2.4.3 for factorization in $\mathbb{Z}[x]$. Our brief sketch of an algorithm follows Wang and Preiss Rothschild [WaR]. For improvements and newer algorithms, we refer the interested reader to [Kal].

Algorithm 2.5.3 (Extended Zassenhaus for factorization in $\mathbf{K}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\mathbf{n}]$).

Input: $f \in K[x_1, \ldots, x_n]$.

Assume: $K = \mathbb{Z}$ or K a field, f is primitive and squarefree[15], and $\deg_{x_i}(f) \geq 1$ for all $i = 1, \ldots, n$.

Output: A list of polynomials f_i, $i = 1, \ldots, s$, such that $f = f_1 \cdot \ldots \cdot f_s$ is an irreducible factorization of f.

(1) Choose a *main variable*, say $x = x_n$, and consider f as a univariate polynomial in x, with coefficients in $R := K[x_1, \ldots, x_{n-1}]$.

(2) Choose a suitable point $\boldsymbol{a} := (a_1, \ldots, a_{n-1}) \in K^{n-1}$ such that[16]

- $\operatorname{lc}(f)(\boldsymbol{a}) \neq 0$,
- $f(\boldsymbol{a}, x) \in K[x_n]$ is squarefree $\bigl($that is, $\frac{\partial f}{\partial x}(\boldsymbol{a}, x) \neq 0$ and $D_f(\boldsymbol{a}) \neq 0\bigr)$.

(3) Choose $N \in \mathbb{N}$ to be the total degree of $F := f(y_1 + a_1, \ldots, y_{n-1} + a_{n-1}, x)$ as a polynomial in $\boldsymbol{y} = (y_1, \ldots, y_{n-1})$, with coefficients in $K[x]$.

(4) Compute the irreducible factorization of the univariate polynomial

$$\overline{F} := F(\mathbf{0}, x) = f(\boldsymbol{a}, x) \in K[x] = \bigl(K[\boldsymbol{y}]/\langle \boldsymbol{y} \rangle\bigr)\bigl[x\bigr] .$$

(5) *Hensel Lifting*: Lift the irreducible factorization in $\bigl(K[\boldsymbol{y}]/\langle \boldsymbol{y} \rangle\bigr)\bigl[x\bigr]$ to

$$\overline{F} = \overline{G}_1 \cdot \ldots \cdot \overline{G}_\ell \text{ in } \bigl(K[\boldsymbol{y}]/\langle \boldsymbol{y} \rangle^{N+1}\bigr)\bigl[x\bigr] .$$

(6) Combine the (unique) representatives $G_1, \ldots, G_\ell \in (K[x])[\boldsymbol{y}]$ of $\overline{G}_1, \ldots, \overline{G}_\ell$ of total degree $\leq N$ in an appropriate way to obtain the *true factors* of F. Then, obtain the irreducible factors of f by resubstituting y_i by $x_i - a_i$, $i = 1, \ldots, n-1$, in the (true) factors of F.

---

[15] If f is not primitive, $\operatorname{cont}(f)$ and $\operatorname{pp}(f)$ may be factorized independently. If f is not squarefree, we may divide f by $\gcd\bigl(f, \frac{\partial f}{\partial x_n}\bigr)$ to obtain a squarefree polynomial (see also Step (1) of the algorithm).

[16] For $K = \mathbb{Z}$ or K an *infinite* field, a random choice of point yields such $\boldsymbol{a}$. For a *finite* field K, however, it may happen that there is no $\boldsymbol{a} \in K^{n-1}$ such that the conditions are satisfied. In this situation, a way to proceed is passing to an extension field F of K which contains sufficiently many elements to find such an $\boldsymbol{a} \in F^{n-1}$.

In the further discussion of the algorithm, we focus on Steps (1) and (5), referring to the previous section for a discussion of Steps (4), (6) (see also Remark 2.5.7, below).

ad Step (1): Choice of Main Variable

The choice of the main variable may seem to be a rather trivial step, but it can be of extraordinary importance for the time consumption of the whole multivariate factorization algorithm. For example, if a polynomial happens to have low degree with respect to one of the variables, say x, but significantly higher degree with respect to the others, choosing the variable x as the main variable will be likely to reduce the number of factors arising in Step (4) and, hence, to drastically reduce the number of combinations to be tried in Step (6). On the other hand, it is often a good choice to use a variable which occurs in a pure power instead of one which occurs only in terms (of high degree) involving several other variables.

Unfortunately, there is no general rule for the (best) choice of the main variable. Our considerations on this step provide a rough outline of the starting point for a heuristic for this task. Further ingredients to the heuristic may be deduced from knowledge about the univariate factorization algorithm and the favourable, respectivly less favourable, effects of certain properties of its input data on the running time.

ad Step (5): Hensel Lifting

Similar to the modular approach to univariate factorization in characteristic zero, the central step in the modular algorithm for multivariate factorization is the lifting step. As the multivariate Hensel lifting algorithm proceeds along the lines of the Hensel lifting algorithm in Section 2.4.2 (with a few changes due to the changes to the underlying task and theorem, and due to some additional technical details), we will not formulate the algorithm explicitly, but focus on the underlying theorem. To this end, we first need to consider an auxiliary lemma:

Lemma 2.5.4. *Let K be a field, and let $g, h \in K[x]$ with $\gcd(g,h) = 1$. Then, for all $i = 0, \ldots, \deg(g) + \deg(h)$, there exist $s_i, t_i \in K[x]$, such that*

$$s_i g + t_i h = x^i \,,$$

subject to the additional conditions $\deg(s_i) < \deg(h)$ and $\deg(t_i) \leq \deg(g)$.

Proof. The polynomials s_i and t_i can be constructed explicitly: We start by determining s_0 and t_0 in $K[x]$ (subject to the condition $s_0 g + t_0 h = 1$) using the extended Euclidean algorithm[17]. Then we define

$$s_i := x^i s_0 \operatorname{rem} h \,, \quad t_i := x^i t_0 + (x^i s_0 \operatorname{quo} h) \cdot g \,,$$

---

[17] Here, we need $K[x]$ to be a Euclidean domain. Indeed, the statement of the lemma does *not* hold for $K = \mathbb{Z}$. For instance, 1 is no $\mathbb{Z}[x]$-linear combination of $g := x$, $h := 2$.

Then the degree condition on s_i is obviously satisfied, and

$$s_i g + t_i h = \underbrace{(x^i s_0 \operatorname{rem} h) \cdot g + (x^i s_0 \operatorname{quo} h) \cdot g \cdot h}_{= x^i s_0 g} + x^i t_0 h = x^i \, .$$

Only the degree condition on t_i remains to be checked: But $t_i h = x^i - s_i g$, where the degree of x^i is at most $\deg(g) + \deg(h)$, and where $\deg(s_i g)$ cannot exceed this bound either. Hence, $\deg(t_i h) \leq \deg(g) + \deg(h)$, proving the remaining inequality (since K is an integral domain). □

Proposition 2.5.5 (Hensel Lifting). *Let* $R := K[\boldsymbol{y}] = K[y_1, \ldots, y_m]$*, where* K *is a field or* $K = \mathbb{Z}$*, let* $k \geq 1$*, and let* $f, g_k, h_k \in R[x]$ *such that* $\operatorname{lc}(f) \notin \langle \boldsymbol{y} \rangle$*, and*

(1_k) $f \equiv g_k h_k \mod \langle \boldsymbol{y} \rangle^k$,

(2_k) $\deg(f) = \deg(g_k) + \deg(h_k)$.

Moreover, assume that for all $i \in \{0, \ldots, \deg(f)\}$ *there exist* $s_i, t_i \in K[x]$ *such that*

(3_k) $s_i g_k + t_i h_k \equiv x^i \mod \langle \boldsymbol{y} \rangle$,

(4_k) $\deg(s_i) < \deg(h_k)\,, \quad \deg(t_i) \leq \deg(g_k)$.

Then there are $g_{k+1}, h_{k+1} \in R[x]$ *subject to the analogous conditions* (1_{k+1}) – (4_{k+1})*, additionally satisfying*

(5) $g_{k+1} \equiv g_k \mod \langle \boldsymbol{y} \rangle^k, \quad h_{k+1} \equiv h_k \mod \langle \boldsymbol{y} \rangle^k$.

Moreover, these liftings g_{k+1} *and* h_{k+1} *are uniquely determined up to units in the ring* $K[\boldsymbol{y}]/\langle \boldsymbol{y} \rangle^{k+1}$.

Proof. As in the Hensel Lifting Algorithm 2.4.6, we set

$$\Delta := g_k h_k - f =: \sum_{j=0}^{m} b_j x^j \, .$$

Then we define

$$g_{k+1} := g_k - \sum_{j=0}^{m} b_j t_j \, , \qquad h_{k+1} := h_k - \sum_{j=0}^{m} b_j s_j \, .$$

Condition (1_k) implies $b_i \in \langle \boldsymbol{y} \rangle^k$, for all $i = 0, \ldots, m$, hence, (5) is satisfied. Condition (4_k) ensures that $\deg(h_k) = \deg(h_{k+1})$, and $\deg(g_{k+1}) \leq \deg(g_k)$. Equipped with these observations, we now start checking properties (1_{k+1}) – (4_{k+1}) explicitly: (1_{k+1}) holds as

$$g_{k+1} h_{k+1} \equiv g_k h_k - \sum_{j=0}^{m} b_j \cdot (\underbrace{s_j g_k + t_j h_k}_{\equiv x^j \bmod \langle \boldsymbol{y} \rangle}) \equiv \underbrace{g_k h_k - \Delta}_{= f} \mod \langle \boldsymbol{y} \rangle^{k+1} \, .$$

Combining this with the facts that $\deg(h_{k+1}) = \deg(h_k)$, that $\deg(g_{k+1}) \leq \deg(g_k)$, that (2_k) holds, and that $\mathrm{lc}(f) \notin \langle \boldsymbol{y} \rangle$, we can deduce that (2_{k+1}) holds, and that $\deg(g_{k+1}) = \deg(g_k)$, hence (4_{k+1}). The remaining property (3_{k+1}) can be shown by an explicit computation:

$$s_i g_{k+1} + t_i h_{k+1} \;=\; s_i g_k + t_i h_k - \sum_{j=0}^{m} b_j \cdot \big(s_i t_j + t_i s_j\big) \;\equiv\; x^i \mod \langle \boldsymbol{y} \rangle\,.$$

Note that this equality even holds mod $\langle \boldsymbol{y} \rangle^k$. The proof of the uniqueness statement is left to the reader as Exercise 2.5.2. □

Remark 2.5.6. For K being a field, the existence of polynomials s_i, t_i satisfying the initial conditions (3_1) and (4_1) is given by Lemma 2.5.4. For $K = \mathbb{Z}$, however, such polynomials do not necessarily exist (see Footnote 17 on Page 68). If the existence fails, a priori, we are forced to compute over $\mathbb{Q}$ instead of computing over $\mathbb{Z}$. But, there is another solution, avoiding the expensive computations over the rationals: from the practical point of view, it is advisable to compute over $\mathbb{Z}/p^M\mathbb{Z}$ for a sufficiently large $M \gg 1$ (see Exercises 2.5.3, 2.5.4 for the theoretical basis of such an approach).

Remark 2.5.7. Various difficulties have been identified with Algorithm 2.5.3, respectively with similar algorithms for multivariate factorization:

(1) **Leading coefficient problem.** If F is *monic*, each irreducible factor of F corresponds to some product $\prod_{i \in S} G_i$.

But, if F is *not monic*, this is not necessarily the case. The weaker uniqueness statement for the Hensel lifting (as compared to Proposition 2.4.9) leads to a weaker equality (as compared to (2.4.8) in the proof of Algorithm 2.4.10): for each factor G of F, there exists a set $S \subset \{1, \ldots, \ell\}$ such that

$$G = \frac{\mathrm{lc}(G)}{\prod_{i \in S} \mathrm{lc}(G_i)} \cdot \prod_{i \in S} G_i\,,$$

where $\mathrm{lc}(G)$ is unknown. Wang solved this problem by giving an extra algorithm to predetermine the leading coefficients of the factors.

(2) **Bad zero problem**: Choosing a point $\boldsymbol{a}$ with several non-zero components a_i often leads to a dense polynomial F.

(3) **Bad reduction problem**: It may occur that $f(\boldsymbol{a}, x) \in K[x]$ has considerably more factors than $f \in K[x_1, \ldots, x_n]$. To get rid of this garbage as soon as possible, the solution is to consider different tuples $\boldsymbol{a}$, and to compare the resulting factorizations.

Exercises

Exercise 2.5.1. Trace Algorithm 2.5.3 with input the multivariate polynomial

$$\begin{aligned} f \;=\; & z^4 + (-y+3)\cdot z^3 + \big(y^3+(x-3)y-x^2-13\big)\cdot z^2 \\ & + \big(-y^4+(x^2+3x+15)y+6\big)\cdot z \\ & + xy^4+2y^3-(x^3+15x)y-2x^2-30 \;\in\; \mathbb{Z}[x,y,z]\,. \end{aligned}$$

Exercise 2.5.2. Prove the uniqueness statement in Proposition 2.5.5.

Exercise 2.5.3. Let p be a prime number, $M \geq 2$, and let $g, h \in \mathbb{Z}[x]$ such that $p \nmid \mathrm{lc}(h)$. Moreover, assume that there are polynomials $\widetilde{s}_0, \widetilde{t}_0 \in \mathbb{Z}[x]$ such that

$$\widetilde{s}_0 g + \widetilde{t}_0 h \equiv 1 \bmod p^M\,.$$

Prove that there exist $s_i, t_i \in K[x]$, $i = 0, \dots, \deg(g) + \deg(h)$ such that

$$s_i g + t_i h \equiv x^i \bmod p^M\,, \quad \deg(s_i) < \deg(h)\,, \quad \deg(t_i) \leq \deg(g)\,.$$

Exercise 2.5.4. Verify that Proposition 2.5.5 generalizes to $K = \mathbb{Z}/p^M\mathbb{Z}$, where p is a prime number and $M \geq 2$.

2.6 Factorization in $\mathbf{K}(\boldsymbol{\alpha})[\mathbf{x}]$

A solution to the factorization problem for univariate polynomials over algebraic extensions of finite fields is already given by the Berlekamp/Cantor/Zassenhaus approach (Algorithm 2.3.25). A more difficult task is to factorize (squarefree) polynomials over algebraic extensions of $\mathbb{Q}$, respectively of other fields of characteristic zero. We do not want to discuss this problem in detail, but to sketch the basic idea for a solution (due to Trager). A more detailed description of this algorithm (and of more sophisticated algorithms) can be found in [GCL, Sect. 8.8].

Let K be a field of characteristic zero (e.g., $\mathbb{Q}$ or some algebraic extension thereof), and let $K \subset K(\alpha)$ be a (primitive) algebraic extension. Assume we want to compute the irreducible factorization in $K(\alpha)[x]$ of a polynomial $f \in K(\alpha)[x]$. Then, the core step is to reduce the factorization problem over $K(\alpha)$ to a factorization problem over K. This is achieved by introducing a new concept:

Definition 2.6.1. Let $\mu_\alpha \in K[t]$ be the minimal polynomial of the field extension $K \subset K(\alpha)$, let d denote its degree, and let $\alpha_1, \dots, \alpha_d \in \overline{K}$ be the zeros of the minimal polynomial in the algebraic closure of K. Then, under the isomorphism $K(\alpha) \xrightarrow{\cong} K[t]/\mu_\alpha$, the coefficients of $f \in K(\alpha)[x]$ are uniquely represented by polynomials in $K[t]$ of degree less than d. Hence, f can be identified with a polynomial $f(t,x) \in K[t,x] = (K[x])[t]$ (of degree less than d in t). We set

$$\mathrm{Norm}(f) := \prod_{i=1}^{d} f(\alpha_i, x) = R_{f(t,x),\mu_\alpha} \in K[x]\,.$$

Indeed, the following facts on $\mathrm{Norm}(f)$ allow the reduction of the factorization problem to a problem over K:

Fact 1. If $\mathrm{Norm}(f) \in K[x]$ is squarefree, and if it has the irreducible factorization $\mathrm{Norm}(f) = g_1 \cdot \ldots \cdot g_r$ in $K[x]$, then $f \in K(\alpha)[x]$ has the irreducible factorization $f = \gcd(f, g_1) \cdot \ldots \cdot \gcd(f, g_r)$ in $K(\alpha)[x]$.

Fact 2. If $f \in K(\alpha)[x]$ is squarefree, then $\mathrm{Norm}\big(f(x - \alpha s)\big) \in K[x]$ is squarefree for all but finitely many $s \in K$.

Fact 3. If f is irreducible in $K(\alpha)[x]$, then $\mathrm{Norm}(f)$ is some power of an irreducible polynomial in $K[x]$.

Chapter 3

Gröbner Basis Techniques

In the previous chapter, we (implicitly) dealt with the problem of solving univariate polynomial equations and systems thereof. Moreover, we sketched an approach to the problem of multivariate factorization. As we observed in the introductory remarks to Chapter 2, multivariate factorization may be used as a preprocessing step for solving a given system of multivariate polynomial equations: the problem of finding solutions of the given system may be reduced to the problem of finding solutions of several simpler systems.

What remains to be given is an algorithm for actually solving a given system of multivariate polynomial equations. In the following, we present two different approaches, the first one being based on the theory of Gröbner bases, and the second one being based on the theory of resultants (which historically preceded the approach based on Gröbner bases).

To this end, let us recall the general computational task we are facing: Let $f_1, \ldots, f_k$ be polynomials in $K[\boldsymbol{x}] = K[x_1, \ldots, x_n]$, where K is a field. The goal is to *solve* the system of polynomial equations

$$f_1(\boldsymbol{x}) = \cdots = f_k(\boldsymbol{x}) = 0.$$

In other words, we are interested in determining the set of points $\boldsymbol{p}$ at which the polynomials $f_1, \ldots, f_k$ vanish simultaneously. More formally, we want to determine the set

$$V_L(f_1, \ldots, f_k) := \{\boldsymbol{p} \in L^n \mid f_1(\boldsymbol{p}) = \ldots = f_k(\boldsymbol{p}) = 0\},$$

where L is either K or some algebraic extension field of K. This set of points can also be described as the vanishing locus[1]

$$V_L(I) := \{\boldsymbol{p} \in L^n \mid f(\boldsymbol{p}) = 0 \ \ \forall f \in I\}$$

---

[1] If $L = \overline{K}$, we simply write $V(I)$ instead of $V_{\overline{K}}(I)$. A typical situation is that the polynomials $f_1, \ldots, f_k$ have *rational coefficients*, and we want to determine the set of *complex solutions* by computations over $\mathbb{Q}$, respectively over finite algebraic extensions of $\mathbb{Q}$.

of the ideal $I := \langle f_1, \dots, f_k \rangle \subset K[\boldsymbol{x}]$ generated by the given polynomials. Indeed, all elements of an ideal vanish at a point $\boldsymbol{p} \in L^n$ iff all generators vanish at this point.

Before discussing the general case, we shortly recall the well-known solution of the problem in two particular cases:

Example 3.0.1 (Systems of Univariate Polynomial Equations).
If $f_1, \dots, f_k \in K[x]$, then $I = \langle f_1, \dots, f_k \rangle \subset K[x]$ is a principal ideal, generated by $f := \gcd(f_1, \dots, f_k)$. As we have seen in Section 2.1, this generator can be computed by applying the Euclidean algorithm $k-1$ times. Determining $V(I)$ then boils down to factorizing f.

Example 3.0.2 (Systems of Linear Equations). Let $f_1, \dots, f_k$ be linear polynomials, then the system of equations can be written as

$$A \cdot \boldsymbol{x} = \boldsymbol{b}\,,$$

for a suitable $k \times n$ matrix A with entries from K, and a suitable vector $\boldsymbol{b} \in K^k$. Applying **Gaussian Elimination** to the extended matrix $(A \mid \boldsymbol{b})$ leads to a matrix $(A' \mid \boldsymbol{b}')$ in (reduced) row echelon form, that is,

$$(A' \mid \boldsymbol{b}') = \left(\begin{array}{cccccccc|c} 1 & * & 0 & * & 0 & & 0 & & \\ & & 1 & & 0 & * & 0 & * & \\ & & & & 1 & & 0 & & * \\ & & & & & & 1 & & \\ & & \mathbf{0} & & & & & & \end{array}\right).$$

We can then immediately deduce the number of solutions of the given system of linear equations: First of all, there cannot be any solutions if there is a row in $(A' \mid \boldsymbol{b}')$ whose only non-zero entry is in the last column. If $(A' \mid \boldsymbol{b}')$ consists of a $n \times n$ unit matrix in the upper left-hand corner and contains only zeros in the rows $n+1, \dots, k$, then there is a unique solution. If $(A' \mid \boldsymbol{b}')$ is of any other structure, then the set of solutions is a positive dimensional affine space (in particular, if the field K is infinite, this implies that there are infinitely many solutions).

In addition to that, there is a well-known procedure for reading off the solution(s) of $A\boldsymbol{x} = \boldsymbol{b}$ explicitly from the matrix $(A' \mid \boldsymbol{b}')$. In particular, a system of linear equations with coefficients in K has a solution over the algebraic closure $\overline{K}$ iff it has a solution over K. Even more, if K is an infinite field, the number of solutions over K and $\overline{K}$ coincides.

In both examples, we replace the given system of generators $f_1, \dots, f_k$ for the ideal I by another system of generators for I which allows us to determine $V(I)$ more easily.

The approach to solving systems of polynomial equations, which we are going to consider in this chapter, is based on the same idea. The new system of generators should form a Gröbner basis (which generalizes the row echelon form of linear equations), and the method to obtain such a Gröbner basis is **Buchberger's algorithm** (a generalization of both, the Euclidean algorithm, and the Gaussian elimination algorithm).

3.1 Division with Remainder

Throughout this section, let K be a field, and let $K[\boldsymbol{x}] = K[x_1, \dots, x_n]$ be equipped with a fixed monomial ordering $>$. The following theorem generalizes division with remainder in $K[x]$ (see Remark 2.1.3) to **simultaneous division with remainder** in $K[\boldsymbol{x}] = K[x_1, \dots, x_n]$ by a finite set of polynomials:

Theorem 3.1.1 (Division with Remainder). *Let* $f_1, \dots, f_k \in K[\boldsymbol{x}] \setminus \{0\}$. *For every* $f \in K[\boldsymbol{x}]$ *there exist* $a_1, \dots, a_k \in K[\boldsymbol{x}]$ *and* $h \in K[\boldsymbol{x}]$ *such that*

(a) $f = a_1 f_1 + \ldots + a_k f_k + h$,

(b) $\mathrm{lm}(f) \geq \mathrm{lm}(a_i f_i)$ *if* f *and* $a_i f_i$ *are non-zero, and*

(c) if $h \neq 0$, *then* $\mathrm{lt}(h)$ *is not divisible by any of* $\mathrm{lt}(f_1), \dots, \mathrm{lt}(f_k)$.

Any such expression is called a **standard expression** *for* f *in terms of the* f_i *(with* **remainder** h*).*

The proof of Theorem 3.1.1 consists of a **division algorithm** for computing a standard expression:

Algorithm 3.1.2 (Division with Remainder).

```
Input:  $f, f_1, \dots, f_k \in K[\boldsymbol{x}]$.
Output: $a_1, \dots, a_k, h \in K[\boldsymbol{x}]$, such that $f = a_1 f_1 + \ldots + a_k f_k + h$ is a standard expression for $f$ in terms of the $f_i$.

(1) $h := f,\ a_1 := \ldots = a_k := 0$;                     // Initialization
(2) WHILE $\big((h \neq 0)$ and $(\exists\, i : \mathrm{lt}(f_i) \mid \mathrm{lt}(h))\big)$ DO
      • Choose $\ell$ such that $\mathrm{lt}(f_\ell) \mid \mathrm{lt}(h)$;
      • Set $a_\ell := a_\ell + \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_\ell)}$, and $h := h - \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_\ell)} \cdot f_\ell$;
(3) RETURN $(a_1, \dots, a_k, h)$.
```

Proof. Step 1 (Termination). Since in the second step of the WHILE loop the leading terms of h and $\frac{\mathrm{lt}(h)}{\mathrm{lt}(f_\ell)} \cdot f_\ell$ coincide, each passing through the loop replaces h by a polynomial with smaller leading monomial. As $>$ is a well-ordering on the set of monomials, this implies that we pass through the loop only finitely many times.

Step 2 (Correctness). From the condition for leaving the While loop, it is obvious that the return value of h satisfies condition (c) for (the remainder of) a standard expression. Moreover, it is not difficult to see that the equality

$$f = \sum_{i=1}^{k} a_i f_i + h$$

holds at each stage of the algorithm. Indeed, in the initialization step, we have the equality

$$f = \sum_{i=1}^{k} 0 \cdot f_i + h \,,$$

and each passing through the While loop replaces $f = \sum_{i=1}^{k} a_i f_i + h$ by

$$f = \sum_{i \neq \ell} a_i f_i + \left(a_\ell + \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_\ell)} \right) \cdot f_\ell + \left(h - \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_\ell)} \cdot f_\ell \right) .$$

In the same way, we see that, at each stage, property (b) of a standard expression is satisfied. Indeed, the leading monomial of $\frac{\mathrm{lt}(h)}{\mathrm{lt}(f_\ell)} \cdot f_\ell$ equals $\mathrm{lm}(h)$, which is at most $\mathrm{lm}(f)$ (in the initialization step, we have $h = f$, and upon each passing through the While loop $\mathrm{lm}(h)$ decreases as observed in Step 1 of the proof). $\square$

The division algorithm as described here is indeterminate, in that we allow choices to be made in the computational process, and in that the computed remainder depends on these choices:

Example 3.1.3. Consider the polynomials

$$f := x_1^3 x_2, \quad f_1 := x_1^3, \quad f_2 := x_1^2 x_2 - x_2^4 \in K[x_1, x_2] \,.$$

With respect to the lexicographic ordering $>_{\mathtt{lp}}$, we get

$$\mathrm{lt}(f) = x_1^3 x_2 \,, \quad \mathrm{lt}(f_1) = x_1^3 \,, \quad \mathrm{lt}(f_2) = x_1^2 x_2 \,.$$

Then, the following are possible standard expressions for f in terms of f_1, f_2 (as returned by some determinate version of the division algorithm):

Version 1. Choosing ℓ minimal such that $\mathrm{lt}(f_\ell) \mid \mathrm{lt}(h)$ leads to the standard expression $f = x_2 \cdot f_1 + 0 \cdot f_2 + 0$ with remainder 0.

Version 2. Choosing ℓ maximal such that $\mathrm{lt}(f_\ell) \mid \mathrm{lt}(h)$ leads to the standard expression $f = 0 \cdot f_1 + x_1 \cdot f_2 + x_1 x_2^4$ with remainder $x_1 x_2^4$.

To simplify and generalize notation, we introduce the concept of a normal form:

Definition 3.1.4 (Normal Form). Let $\mathcal{FOS}(K[\boldsymbol{x}])$ be the set of all finite, ordered subsets of $K[\boldsymbol{x}]$. Then a **normal form** is a map

$$\begin{aligned} \mathrm{NF} : K[\boldsymbol{x}] \times \mathcal{FOS}(K[\boldsymbol{x}]) &\longrightarrow K[\boldsymbol{x}]\,, \\ (f, \mathcal{F}) &\longmapsto \mathrm{NF}(f \mid \mathcal{F})\,, \end{aligned}$$

such that, for each tuple $(f, \mathcal{F})$, the image $\mathrm{NF}(f \mid \mathcal{F})$ is the remainder of some standard expression for f in terms of the elements of $\mathcal{F}$. In the same way, we define a **normal form with respect to $\mathcal{F} \in \mathcal{FOS}(K[\boldsymbol{x}])$** to be a map

$$\begin{aligned} \mathrm{NF}(\_ \mid \mathcal{F}) : K[\boldsymbol{x}] &\longrightarrow K[\boldsymbol{x}]\,, \\ f &\longmapsto \mathrm{NF}(f \mid \mathcal{F})\,, \end{aligned}$$

such that, for each f, the image $\mathrm{NF}(f \mid \mathcal{F})$ is the remainder of some standard expression for f in terms of the elements of $\mathcal{F}$. We also refer to $\mathrm{NF}(f \mid \mathcal{F})$ as a **normal form for f with respect to $\mathcal{F}$**.

In particular, each determinate version of the division algorithm leads to a particular normal form NF, taking the tuple $\big(f, \{f_1, \dots, f_k\}\big)$ to the remainder h as returned by the algorithm.

Remark 3.1.5 (Reduced Normal Form). We may easily modify the division algorithm such that it returns a standard expression whose remainder h satisfies the following stronger condition:

(c') If $h \neq 0$, then no term of h is divisible by any of the $\mathrm{lt}(f_i)$.

Indeed, it suffices to replace the WHILE loop in step (2) of Algorithm 3.1.2 by

```
(2') $r := 0$;
     WHILE ($h \neq 0$) DO
       • WHILE $\big(\exists\, i : \mathrm{lt}(f_i) \mid \mathrm{lt}(h)\big)$ DO
           ◦ Choose $\ell$ such that $\mathrm{lt}(f_\ell) \mid \mathrm{lt}(h)$;
           ◦ Set $a_\ell := a_\ell + \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_\ell)}$, and $h := h - \frac{\mathrm{lt}(h)}{\mathrm{lt}(f_\ell)} \cdot f_\ell$;
       • Set $r := r + \mathrm{lt}(h)$, and $h := h - \mathrm{lt}(h)$;
     $h := r$;
```

If a polynomial h satisfies the condition (c'), we also say that h is **reduced** with respect to $\mathcal{F} = \{f_1, \dots, f_k\}$.

In the same way, a map NF as above is called a **reduced normal form**, if the image of each tuple $(f, \mathcal{F})$ is the remainder of some standard expression for f

in terms of the elements of $\mathcal{F}$ satisfying the stronger condition (c'). Thus, each determinate version of the modified division algorithm leads to a particular reduced normal form.

Again, Example 3.1.3 shows that there are different reduced normal forms $\mathrm{NF}_{\mathrm{red}}$. This is different if we consider reduced normal forms with respect to a Gröbner basis $\mathcal{G} \subset K[\boldsymbol{x}]$ (see Lemma 3.2.6).

3.2 Gröbner Bases

Analogous to the special case of systems of linear equations, where the key step is achieving a special structure for the system of equations (the row echelon form), the key task is transforming the system to a special structure (which we describe in this section).

Throughout the following, let K be a field, and let $I \subset K[\boldsymbol{x}] = K[x_1, \ldots, x_n]$ be an ideal. Moreover, we fix a monomial ordering $>$ on $K[\boldsymbol{x}]$.

Definition 3.2.1. (1) The **leading ideal** of I (**with respect to** $>$) is the ideal

$$L(I) := L_>(I) := \big\langle \mathrm{lt}_>(f) \bigm| f \in I \big\rangle \subset K[\boldsymbol{x}] \,.$$

That is, $L(I)$ is the (monomial) ideal generated by the leading terms of the elements of I.

(2) A finite subset $\mathcal{G} = \{g_1, \ldots, g_r\}$ of I is called a **Gröbner basis for $\boldsymbol{I}$** (with respect to $>$) if

$$L(I) = \big\langle \mathrm{lt}(g_1), \ldots, \mathrm{lt}(g_r) \big\rangle \,.$$

That is, the leading ideal of I is generated by the leading monomials of the elements of $\mathcal{G}$. Or, equivalently, for each $f \in I \setminus \{0\}$, there is an element $g \in G$ such that $\mathrm{lt}(g) \mid \mathrm{lt}(f)$.

Whenever it is stated that a set $\mathcal{G}$ 'is a Gröbner basis', without reference to an ideal, this is a short-hand notation for $\mathcal{G}$ being a Gröbner basis for the ideal generated by the elements of $\mathcal{G}$. The following lemma shows that this convention (and the name Gröbner 'basis' [2]) is justified:

Lemma 3.2.2. *Let $I \subset K[\boldsymbol{x}]$ be an ideal, and let $\mathcal{G} \subset I$ be a Gröbner basis for I (with respect to $>$). Then the following hold:*

(1) If $J \subset K[\boldsymbol{x}]$ is an ideal such that $I \subset J$, then $\big(I = J \Longleftrightarrow L(I) = L(J)\big)$.

(2) $\langle \mathcal{G} \rangle = I$.

---

[2] Historically, the word 'basis' was used as another name for a 'finite set of generators'.

Proof. (1) If the ideals coincide, they clearly have the same leading ideal. So we only need to prove the other implication. To this end, let $f \in J \setminus I$, and let

$$f = \sum_{i=1}^{r} a_i g_i + h$$

be a standard expression for f in terms of the elements $g_1, \dots, g_r$ of $\mathcal{G}$. Then, $h \notin I$, and property (c) of a standard expression implies that $\mathrm{lt}(h)$ is not divisible by any of $\mathrm{lt}(g_1), \dots, \mathrm{lt}(g_r)$. But this means that $\mathrm{lt}(h) \notin \langle \mathrm{lt}(g_1), \dots, \mathrm{lt}(g_r) \rangle = L(I)$. As $h \in J$, this implies that $L(I) \subsetneq L(J)$.

(2) is a direct consequence of (1), applied to the inclusion $\langle \mathcal{G} \rangle \subset I$ (which is possible, as $\mathcal{G}$ is a Gröbner basis for $\langle \mathcal{G} \rangle$). □

Remark 3.2.3. Leading ideals depend on the choice of the monomial order. Also, a Gröbner basis with respect to one monomial ordering may not be a Gröbner basis with respect to another monomial order:

Example 3.2.4. Consider again the polynomials

$$f_1 := x_1^3, \quad f_2 := x_1^2 x_2 - x_2^4 \in K[x_1, x_2] .$$

With respect to the lexicographic ordering $>_{\texttt{lp}}$, we get

$$\mathrm{lm}(f_1) = x_1^3 , \quad \mathrm{lm}(f_2) = x_1^2 x_2 .$$

In particular, $\mathcal{G} = \{f_1, f_2\}$ is *not* a Gröbner basis with respect to $>_{\texttt{lp}}$, as

$$f := x_2 \cdot f_1 - x_1 \cdot f_2 = x_1 x_2^4 \in \langle f_1, f_2 \rangle ,$$

and as $\mathrm{lm}(f) = x_1 x_2^4$ is neither divisible by x_1^3 nor by $x_1^2 x_2$.

On the other hand, with respect to the degree reverse lexicographic ordering $>_{\texttt{dp}}$, we get $\mathrm{lm}(f_1) = x_1^3$, $\mathrm{lm}(f_2) = x_2^4$, and it is not difficult to see, that $\mathcal{G} = \{f_1, f_2\}$ is a Gröbner basis with respect to $>_{\texttt{dp}}$.

The original motivation for introducing the concept of Gröbner bases was to develop tools for computing in quotient rings $K[\boldsymbol{x}]/I$, $I \subset K[\boldsymbol{x}]$ some ideal. The following lemma shows that this task may be solved by using Gröbner bases:

Lemma 3.2.5. *Let $I \subset K[\boldsymbol{x}]$ be an ideal, let $\mathcal{G} \subset I$ be a Gröbner basis for I (with respect to $>$), and let $\mathrm{NF}(\_ \mid \mathcal{G})$ be a normal form with respect to $\mathcal{G}$. Moreover, let $f \in K[\boldsymbol{x}]$. Then the following are equivalent:*

(1) $f \in I$.

(2) $\mathrm{NF}(f \mid \mathcal{G}) = 0$.

Proof. Let

$$f = \sum_{i=1}^{r} a_i g_i + h$$

be a standard expression for f in terms of the elements $g_1, \dots, g_r$ of $\mathcal{G}$. Then $h = 0$ implies $f \in I$, which proves (2) $\Rightarrow$ (1). To show the opposite implication, assume that $h \neq 0$. Then, property (c) of a standard expression implies that

$$\operatorname{lt}(h) \notin \langle \operatorname{lt}(g_1), \dots, \operatorname{lt}(g_r) \rangle = L(I)\,.$$

Thus, $h \notin I$, which implies $f \notin I$. □

This lemma shows that $g, \widetilde{g} \in K[\boldsymbol{x}]$ satisfy $g \equiv \widetilde{g} \bmod I$ iff $\mathrm{NF}(g \mid \mathcal{G}) = \mathrm{NF}(\widetilde{g} \mid \mathcal{G})$, where $\mathrm{NF}(\_ \mid \mathcal{G})$ is any normal form with respect to a Gröbner basis $\mathcal{G}$ for I. In particular, if we know some Gröbner basis $\mathcal{G}$ for I, then applying division with remainder by the elements of $\mathcal{G}$ allows us to represent computations in $K[\boldsymbol{x}]/I$ (in a unique way) by polynomial computations in $K[\boldsymbol{x}]$. Even more:

Lemma 3.2.6. *Let $I \subset K[\boldsymbol{x}]$ be an ideal, $\mathcal{G} \subset I$ a Gröbner basis for I (with respect to $>$). Then the reduced normal form with respect to $\mathcal{G}$ is uniquely defined.*

Proof. Let us suppose that $h \neq \widetilde{h}$ are two reduced normal forms of f with respect to $\mathcal{G} = \{g_1, \dots, g_r\}$. Then no monomial appearing in h and $\widetilde{h}$ is divisible by any of $\operatorname{lt}(g_1), \dots, \operatorname{lt}(g_r)$. On the other hand, $f - h$, $f - \widetilde{h} \in \langle \mathcal{G} \rangle$, and, hence,

$$\big(f - \widetilde{h}\big) - (f - h) = h - \widetilde{h} \in \langle \mathcal{G} \rangle = I\,.$$

Thus, $\operatorname{lt}\big(h - \widetilde{h}\big) \in L(\mathcal{G})$, which is impossible as no monomial in $h, \widetilde{h}$ is divisible by any of $\operatorname{lt}(g_1), \dots, \operatorname{lt}(g_r)$. □

It is quite obvious that a Gröbner basis for a given ideal (with respect to a given ordering) is by no means unique. Nevertheless, it is desirable not to have redundant elements in a Gröbner basis, and to impose additional conditions which finally ensure uniqueness. To this end, we need to introduce two more notions which generalize the row echelon form, respectively the reduced row echelon form, of a system of linear polynomials:

Definition 3.2.7. Let $\mathcal{F} \subset K[\boldsymbol{x}]$ be any finite subset.

(1) $\mathcal{F}$ is called **interreduced** (or **minimal**) if $\mathcal{F}$ does not contain zero, and if any two distinct elements f and g of $\mathcal{F}$ satisfy $\operatorname{lm}(f) \nmid \operatorname{lm}(g)$.

(2) $\mathcal{F}$ is called (**completely**) **reduced** if the following three conditions hold:

  (a) $\mathcal{F}$ is interreduced,
  (b) all elements $f \in \mathcal{F}$ are monic, and,

(c) for all $f \in \mathcal{F}$, the **tail of $\boldsymbol{f}$**,

$$\mathrm{tail}(f) := f - \mathrm{lt}(f)\,,$$

is reduced with respect to $\mathcal{F}$.

It is a first important observation that any finite set $\mathcal{F} \subset K[\boldsymbol{x}]$ can be transformed into an interreduced set $\mathcal{F}'$ such that $\langle \mathcal{F} \rangle = \langle \mathcal{F}' \rangle$. Indeed, we may proceed along the lines of the following algorithm:

Algorithm 3.2.8 (Interreduction).

Input: A finite set $\mathcal{F} \subset K[\boldsymbol{x}]$.

Output: An interreduced set $\mathcal{F}' \subset K[\boldsymbol{x}]$ satisfying $\langle \mathcal{F} \rangle = \langle \mathcal{F}' \rangle$.

```
(1) $\mathcal{F} := \mathcal{F} \setminus \{0\}$,
(2) WHILE ($\mathcal{F}$ is not interreduced) DO
        • Choose $f \neq g \in \mathcal{F}$ such that $\mathrm{lm}(f) \mid \mathrm{lm}(g)$;
        • Set $g := g - \frac{\mathrm{lt}(g)}{\mathrm{lt}(f)} \cdot f$;
        • IF ($g = 0$) THEN $\mathcal{F} := \mathcal{F} \setminus \{g\}$;
(3) RETURN($\mathcal{F}$).
```

To see that this algorithm terminates, we may apply the same reasoning as for the termination of the division algorithm 3.1.2. The correctness follows directly from the condition for leaving the WHILE loop.

Remark 3.2.9. Let $\mathcal{G} \subset K[\boldsymbol{x}]$ be a Gröbner basis. Then the following hold:

(1) $\mathcal{G}$ can be transformed into an *interreduced Gröbner basis* simply by deleting appropriate elements[3] of $\mathcal{G}$. To see this, we only need a simple observation: if the input $\mathcal{G}$ of the above algorithm is a Gröbner basis, we know that the leading term of each new polynomial, say g_{new}, arising in the WHILE loop is again in $L(\mathcal{G})$, that is, there exists an $f \in \mathcal{G}$, $f \neq g_{\text{new}}$, such that $\mathrm{lt}(f)$ divides $\mathrm{lt}(g_{\text{new}})$. Hence, we can successively reduce g_{new} until we eventually obtain 0.

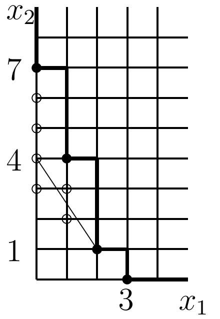
(2) $\mathcal{G}$ is *reduced* iff all elements of $\mathcal{G}$ are monic and for all $f \neq g \in \mathcal{G}$ no term of f is divisible by the leading term of g.

Below, we shall see that each ideal $I \subset K[\boldsymbol{x}]$ has a Gröbner basis $\mathcal{G} = \{g_1, \dots, g_r\}$. We may assume that $g_1, \dots, g_r$ are monic (dividing g_i by $\mathrm{lc}(g_i)$ if necessary) and that $\mathcal{G}$ is interreduced. If we then replace each g_i by the (unique) reduced normal form of g_i with respect to $\mathcal{G} \setminus \{g_i\}$, we obtain a reduced Gröbner basis for I. Such a reduced Gröbner basis is unique (see Exercise 3.2.2)[4].

Example 3.2.10. (Example 3.2.4 revisited) We have seen above that the two polynomials $f_1 = x_1^3$, $f_2 = x_1^2 x_2 - x_2^4$ do not form a Gröbner basis for $I = \langle f_1, f_2 \rangle$ with respect to $>_{\mathtt{lp}}$. On the other hand, we see immediately that

$$\begin{aligned} f_3 &:= \quad x_2 f_1 - x_1 f_2 &&= x_1 x_2^4 \\ f_4 &:= -x_2^3 f_2 + x_1 f_3 &&= x_2^7 \end{aligned}$$

are elements of I. Hence, we know that $\langle x_1^3, x_1^2 x_2, x_1 x_2^4, x_2^7 \rangle \subset L(I)$. Since all terms of the two generators f_1 and f_2 of I are of degree at least 3, no monomials of degree less than 3 can occur in $L(I)$. This leaves us with only six monomials in question: $x_1 x_2^2$, $x_1 x_2^3$, x_2^3, x_2^4, x_2^5 and x_2^6; all of these can be excluded by an explicit computation:



Therefore, we know that $L(I) = \langle x_1^3, x_1^2 x_2, x_1 x_2^4, x_2^7 \rangle$, and that $\mathcal{G} = \{f_1, f_2, f_3, f_4\}$ is a Gröbner basis for $\mathcal{G}$. We can also use this example to illustrate the notions of interreduced and reduced Gröbner basis:

---

[3] This shows that for Gröbner bases the property of being interreduced is a minimality condition.

[4] Note that the reduced Gröbner basis for the ideal $I = 0$ is $\mathcal{G} = \emptyset$.

(1) $\{f_1, f_2, f_3, f_4\}$ is a reduced Gröbner basis for I.

(2) $\{f_1, f_2, f_3, f_4, f_1+f_4\}$ is a Gröbner basis, but neither interreduced nor reduced.

(3) $\{f_1+f_4, f_2, f_3, f_4\}$ is an interreduced Gröbner basis, but it is not reduced.

Up to now, we have discussed many properties of Gröbner bases. What is still missing is a proof of the existence, or an algorithm for computing a Gröbner basis for each ideal $I \subset K[\boldsymbol{x}]$.

The first algorithm for computing Gröbner bases was given by B. Buchberger in his thesis [Buc]. Before we can state this algorithms in a compact form, however, we have to introduce one more notion: the **s-polynomial** of two given polynomials.

Definition 3.2.11. Let $f, g \in K[\boldsymbol{x}] \setminus \{0\}$, and denote their leading monomials by $\boldsymbol{x}^{\boldsymbol{\alpha}} := \mathrm{lm}(f)$ and $\boldsymbol{x}^{\boldsymbol{\beta}} := \mathrm{lm}(g)$. Setting

$$\boldsymbol{x}^{\boldsymbol{\gamma}} := \mathrm{lcm}(\boldsymbol{x}^{\boldsymbol{\alpha}}, \boldsymbol{x}^{\boldsymbol{\beta}})\,,$$

that is, $\gamma_i := \max\{\alpha_i, \beta_i\}$ for each i, the **s-polynomial** of f and g is defined as

$$\mathrm{spoly}(f, g) := \mathrm{lc}(g) \cdot \boldsymbol{x}^{\boldsymbol{\gamma}-\boldsymbol{\alpha}} \cdot f - \mathrm{lc}(f) \cdot \boldsymbol{x}^{\boldsymbol{\gamma}-\boldsymbol{\beta}} \cdot g.$$

Remark 3.2.12. The importance of this notion becomes clear when we consider the following special cases:

(1) If $\mathrm{lm}(g)$ divides $\mathrm{lm}(f)$, the exponent $\boldsymbol{\gamma}$ of the least common multiple coincides with $\boldsymbol{\alpha}$ and, hence,

$$\mathrm{spoly}(f, g) = \mathrm{lc}(g) \cdot \left(f - \frac{\mathrm{lt}(f)}{\mathrm{lt}(g)} \cdot g \right)\,,$$

which coincides with the operation in the WHILE loop of the division algorithm up to the constant factor $\mathrm{lc}(g)$.[5]

(2) The s-polynomial can be regarded as a generalization of the elementary row operations used in the Gaussian elimination algorithm. More precisely, canceling the first non-zero entry of the i-th row by subtracting a suitable (K-) multiple of another row, say the j-th row, is just a substitution of the i-th equation by its s-polynomial with the j-th equation in the system of linear equations.

---

[5] The symmetric form of the s-polynomial as introduced here has the advantage that the s-polynomial of two polynomials in $\mathbb{Z}[\boldsymbol{x}]$ is again a polynomial with coefficients in $\mathbb{Z}$. In practice, this is an important point, since the arithmetic operations over $\mathbb{Q}$ are much more expensive than those over $\mathbb{Z}$. Hence, given polynomials with rational coefficients, one usually clears all denominators before entering an algorithm for computing (reduced) normal forms or Gröbner bases, and one wants to perform all calculations in such an algorithm over the integers (of course, division by the content should always be allowed in order to keep the integer coefficients as small as possible).

Equipped with the notion of an s-polynomial, we are now able to formulate Buchberger's algorithm for computing Gröbner bases:

Algorithm 3.2.13 (Buchberger's algorithm).

Input: $f_1, \dots, f_k \in K[\boldsymbol{x}]$, and a normal form NF.

Output: $\mathcal{G} \subset K[\boldsymbol{x}]$, a Gröbner basis for $\langle f_1, \dots, f_k\rangle \subset K[\boldsymbol{x}]$ with respect to $>$.

```
(1) Set $\mathcal{G} := \{f_1, \dots, f_k\}$ ;                // Initialization
        $\mathscr{P} := \{(f_i, f_j) \mid 1 \leq i < j \leq k\}$ ;
(2) WHILE ($\mathscr{P} \neq \emptyset$) DO
        • Choose $(f, g) \in \mathscr{P}$ ;
        • Set $\mathscr{P} := \mathscr{P} \setminus \{(f, g)\}$ ;
        • Set $h := \mathrm{NF}\big(\mathrm{spoly}(f, g) \mid \mathcal{G}\big)$ ;
        • IF ($h \neq 0$) THEN
            ◦ $\mathscr{P} := \mathscr{P} \cup \{(h, f) \mid f \in \mathcal{G}\}$ ;
            ◦ $\mathcal{G} := \mathcal{G} \cup \{h\}$ ;
(3) RETURN ($\mathcal{G}$).
```

Remark 3.2.14. If $L \subset K$ is a subfield, and if I is an ideal of $K[\boldsymbol{x}]$ generated by polynomials $f_1, \dots, f_k$ with coefficients in L, then Buchberger's algorithm applied to $f_1, \dots, f_k$ yields Gröbner basis elements for I which are also defined over L.

The *termination* of Algorithm 3.2.13 is implied by Dickson's lemma (which itself easily can be proved by applying induction on n):

Lemma 3.2.15 (Dickson). *Let $M \subset \mathbb{N}^n$ be any subset. Then there is a finite set $B \subset M$ (called a* ***Dickson basis*** *for M), satisfying*

$$\forall\, \boldsymbol{\alpha} \in M \;\; \exists\, \boldsymbol{\beta} \in B \text{ such that } \boldsymbol{\beta} \leq_{nat} \boldsymbol{\alpha}\,.$$

Considering the exponent vectors of monomials, Dickson's lemma applies to **monomial ideals**, that is, ideals which are generated by monomials. It implies:

(1) Each monomial ideal is finitely generated.

(2) Each ascending chain $I_1 \subset I_2 \subset I_3 \subset \dots$ of monomial ideals becomes stationary, that is, there is some $m \in \mathbb{N}$ such that $I_i = I_{i+1}$ for all $i \geq m$.

Statement (2) provides a proof for the *termination* of Algorithm 3.2.13: Each time the set of pairs $\mathscr{P}$ grows during the algorithm, we add a new polynomial h to the set $\mathcal{G}$ which satisfies $\mathrm{lt}(h) \notin \langle \mathrm{lt}(g) \mid g \in \mathcal{G} \rangle$. As the monomial ideal generated by $\{\mathrm{lt}(g) \mid g \in \mathcal{G}\}$ strictly increases when h is added to $\mathcal{G}$, and as this may only happen finitely many times due to Dickson's lemma, the set of pairs $\mathscr{P}$ is enlarged only finitely many times when passing through the algorithm.

To show the *correctness* of Algorithm 3.2.13 is much more difficult. It is based on Buchberger's criterion, which was the essential new result of Buchberger providing the computational breakthrough in the theory of Gröbner bases:

Theorem 3.2.16 (Buchberger's Criterion). *Let $I \subset K[\boldsymbol{x}]$ be an ideal, let $\mathcal{G} = \{g_1, \dots, g_r\} \subset I$, and let NF be a normal form with respect to $\mathcal{G}$. Then the following are equivalent:*

(1) $\mathcal{G}$ is a Gröbner basis for I.

(2) $\mathrm{NF}(f \mid \mathcal{G}) = 0$ for all $f \in I$.

(3) I is generated by $\mathcal{G}$, and, for each $i, j = 1, \dots, r$, we have

$$\mathrm{NF}\big(\mathrm{spoly}(g_i, g_j) \mid \mathcal{G}_{ij}\big) = 0$$

for some subset $\mathcal{G}_{ij}$ of $\mathcal{G}$.

Proof. The implication (1) ⇒ (2) holds due to Lemma 3.2.5. To show the implication (2) ⇒ (3), observe that (2) implies that each $f \in I$ is a linear combination of $g_1, \dots, g_r$, in particular, that $f \in \langle \mathcal{G} \rangle$. Moreover, note that the second statement of (3) is obviously implied by (2) as $\mathrm{spoly}(g_i, g_j) \in I$.

The implication (3) ⇒ (1) is the core of Buchberger's criterion. Its proof requires tools which are beyond the scope of this course, in particular the concept of syzygies. We refer the interested reader to a textbook such as [CLO] or [GrP]. ☐

Example 3.2.17. (Example 3.2.4 revisited) We now use Algorithm 3.2.13 to compute a Gröbner basis with respect to $>_{\mathrm{lp}}$ for the ideal generated by the polynomials

$$f_1 := x_1^3, \quad f_2 := x_1^2 x_2 - x_2^4 \in K[x_1, x_2]\,.$$

Then the initialization step provides us with

$$\mathscr{P} = \{(f_2, f_1)\}\,, \quad \mathcal{G} = \{f_1, f_2\}\,.$$

Hence, we first consider the s-polynomial of f_1 and f_2:

$$\mathrm{spoly}(f_2, f_1) = x_1 \cdot f_2 - x_2 \cdot f_1 = -x_1 x_2^4 =: f_3$$

which is a reduced normal form for f_3 with respect to $\mathcal{G}$. Updating $\mathcal{G}$ and $\mathscr{P}$ accordingly yields:

$$\mathscr{P} = \{(f_3, f_2), (f_3, f_1)\}\,, \quad \mathcal{G} = \{f_1, f_2, f_3\}.$$

In the next passing through the WHILE loop, we obtain:

$$\mathrm{spoly}(f_3, f_2) = x_1 \cdot f_3 + x_2^3 \cdot f_2 = -x_2^7 =: f_4\,,$$

which is a reduced normal form for f_4 with respect to $\mathcal{G}$. Thus, $\mathcal{G}$ and $\mathscr{P}$ are updated as

$$\mathscr{P} = \{(f_3, f_1), (f_4, f_3), (f_4, f_2), (f_4, f_1)\}\,, \quad \mathcal{G} = \{f_1, f_2, f_3, f_4\}.$$

As the s-polynomial of two monomials is always 0, the only non-trivial s-polynomial to be considered in the subsequent steps of the algorithm is

$$\mathrm{spoly}(f_4, f_2) = x_1^2 \cdot f_4 + x_2^6 \cdot f_2 = -x_2^{10} = x_2^3 \cdot f_4\,,$$

which has remainder 0 on division by $f_1, \dots, f_4$. Thus, the algorithm returns the set $\{f_1, f_2, f_3, f_4\}$ as we expected.

Exercises

Exercise 3.2.1. Let $f = x^4 + y^4 + z^4 + xyz$. Give a direct argument to show that the set

$$\mathcal{G} = \left\{ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right\}$$

is a Gröbner basis with respect to $>_{\mathtt{dp}}$.

Exercise 3.2.2. Let $I \subset K[\boldsymbol{x}]$ be an ideal. Show that the reduced Gröbner basis for I is uniquely determined.

Exercise 3.2.3. Let $I \subset K[\boldsymbol{x}]$ be a **binomial ideal**, that is, an ideal generated by **binomials** (polynomials of the form $c_{\boldsymbol{\alpha}}\boldsymbol{x}^{\boldsymbol{\alpha}} + c_{\boldsymbol{\beta}}\boldsymbol{x}^{\boldsymbol{\beta}}$). Prove that the reduced Gröbner basis for I consists only of binomials.

Exercise 3.2.4. Prove Dickson's Lemma 3.2.15.

3.3 Solving by Elimination

We are now in the position to describe a method for solving systems of polynomial equations with coefficients in a field K. This method applies to any system of polynomial equations which has only finitely many solutions over the algebraic closure $\overline{K}$. It only requires an effective algorithm for determining the zeros[6] of a univariate polynomial in $K[x]$.

---

[6] If we aim at calculating the exact solutions in K^n (respectively in $\overline{K}^n$), this should be an algorithm for univariate factorization in $K[x]$ (respectively in $\overline{K}[x]$). If K is a subfield of $\mathbb{C}$, and if we aim at calculating floating point approximations for the complex solutions of the system, the algorithm for determining the zeros should be a numerical algorithm such as Newton's algorithm or the algorithm of Laguerre.

Before presenting the general method, we discuss some more of the problems arising when trying to generalize the well-known method for solving systems of linear equations: For systems of linear equations, the dimension of the set of solutions (which is an affine space) can be easily read from the row echelon form: if $A \in \mathrm{Mat}(k \times n, K)$, is in row echelon form, and if r is the number of non-zero rows of A, the set of solutions of the homogeneous system $A\boldsymbol{x} = 0$ has dimension $n - r$. In particular, the homogeneous system has only finitely many solutions over $\overline{K}$ iff $n = r$, and in this case the solution is defined over K and uniquely determined.

While a homogeneous system of linear equations always has the trivial solution $\boldsymbol{x} = \boldsymbol{0}$, an inhomogeneous system $A\boldsymbol{x} = \boldsymbol{b}$ may have an empty set of solutions. If the extended matrix $(A \mid \boldsymbol{b})$ is in row echelon form, this is the case iff the last non-zero row of $(A \mid \boldsymbol{b})$ has only one non-zero entry and this non-zero entry is in the last column.

Recall that systems of linear equations have a particularly nice property: if K is an infinite field, the number of solutions over K coincides with the number of solutions over $\overline{K}$. Already for univariate polynomials of degree 2 this is different: the equation $x^2 = -1$ has no real solution, but 2 complex (conjugate) solutions. Hence, turning to systems of arbitrary polynomial equations, there is no hope to get a general criterion for determining the number of solutions over the field of definition K. Thus, we restrict ourselves to the following list of problems concerning the set of solutions of an ideal $I \subset K[\boldsymbol{x}]$ over the algebraic closure $\overline{K}$:

Problem 1. *How can we decide whether* $V(I)$ *is non-empty?*

Problem 2. *How can we decide whether* $V(I)$ *is a finite set?*

Or, more generally,

Problem 2'. *How can we determine the dimension of* $V(I)$ *?*

Of course, as $V(I)$ is not a linear space, before giving an answer to Problem 2', we have to specify what is meant by the dimension of $V(I)$. This is beyond the scope of this course (see also the footnote on Page 89).

As discussed above, when looking for an answer to these problems, we should replace the row echelon form of a system of linear equations by $f_1, \ldots, f_r \in K[\boldsymbol{x}]$ forming a *Gröbner basis* $\mathcal{G}$ for I. And, indeed, the following important theorem of Algebraic Geometry provides the basis for giving an answer to Problems 1 and 2 by means of properties of $\mathcal{G}$:

Theorem 3.3.1 (Hilbert's Nullstellensatz). *Let* $I \subset K[\boldsymbol{x}]$ *be an ideal, and let* $f \in K[\boldsymbol{x}]$. *Then* f *vanishes on* $V(I)$ *iff* $f^m \in I$ *for some* $m \in \mathbb{N}$.

We also introduce the notion of the **radical** of I,

$$\sqrt{I} := \left\{ f \in K[\boldsymbol{x}] \;\middle|\; f^m \in I \text{ for some } m \in \mathbb{N} \right\},$$

which is an ideal of $K[\boldsymbol{x}]$, generalizing the squarefree part of a polynomial $g \in K[\boldsymbol{x}]$. Then the Hilbert Nullstellensatz may be reformulated as:

$$f\big|_{V(I)} = 0 \iff f \in \sqrt{I}\,.$$

Corollary 3.3.2. *Let $\mathcal{G}$ be a Gröbner basis for the ideal $I \subset K[\boldsymbol{x}]$ (with respect to any monomial ordering). Then the following are equivalent:*

(a) $V(I) = \emptyset$.

(b) $1 \in I$.

(c) $\exists\ g \in \mathcal{G}$ *such that* $\operatorname{lm}(g) = 1$.

Proof. The equivalence of (a) and (b)[7] follows from Hilbert's Nullstellensatz. The equivalence of (b) and (c) follows from Lemma 3.2.5, respectively from the fact that 1 is the smallest element of the set of monomials. □

Corollary 3.3.3 (Finiteness Theorem). *Let $\mathcal{G}$ be a Gröbner basis for the ideal $I \subset K[\boldsymbol{x}]$ (with respect to any monomial ordering). Then the following are equivalent:*

(a) $V(I)$ *is a finite set.*

(b) $\forall\ 1 \leq i \leq n\ \ \exists\ \alpha_i \in \mathbb{N}$ *such that* $x_i^{\alpha_i} \in L(I)$.

(c) $\forall\ 1 \leq i \leq n\ \ \exists\ g_i \in \mathcal{G}$ *such that* $\operatorname{lm}(g_i) = x_i^{\alpha_i}$, $\alpha_i \in \mathbb{N}$.

(d) The K-vector space $K[\boldsymbol{x}]/L(I)$ is finite dimensional.

(e) The K-vector space $K[\boldsymbol{x}]/I$ is finite dimensional.

Proof. If there are only finitely many points in $V(I)$, say

$$V(I) = \{(a_{j,1}, \ldots, a_{j,n}) \mid 1 \leq j \leq s\}\,,$$

we know that the products

$$f_i := \prod_{j=1}^{s} (x_i - a_{j,i})\,, \quad i = 1, \ldots, n\,,$$

vanish along $V(I)$. By Hilbert's Nullstellensatz, this implies $f_i^{m_i} \in I$ for some $m_i \in \mathbb{N}$. As $\operatorname{lm}(f_i^{m_i}) = x_i^{sm_i}$, we get $(a) \Rightarrow (b)$.

The implication $(b) \Rightarrow (c)$ immediately follows from the definition of a Gröbner basis. Now, (c) implies that $K[\boldsymbol{x}]/L(I)$ is spanned by the (finitely many) monomials $\boldsymbol{x}^{\boldsymbol{\beta}}$ with $\beta_i < \alpha_i$ for each i, hence (d). Moreover, each reduced normal form with respect to $\mathcal{G}$ induces an isomorphism

$$K[\boldsymbol{x}]/L(I) \xrightarrow{\cong} K[\boldsymbol{x}]/I\,, \quad \overline{f} \mapsto \operatorname{NF}(f \mid \mathcal{G})\,.$$

---

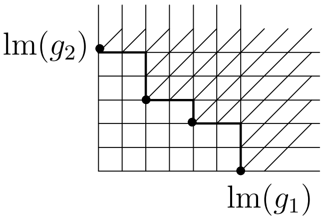
[7] Sometimes, one refers to this equivalence as the **weak Hilbert Nullstellensatz**.

Thus, it remains only to show the implication $(e) \Rightarrow (a)$: Let $i \in \{1, \ldots, n\}$ be fixed and suppose that $\dim_K K[\boldsymbol{x}]/I < \infty$. Then the classes mod I of x_i^j, $j \in \mathbb{N}$, must be linearly dependent. That is, there is some relation

$$-c_{i,m} x_i^m \equiv \sum_{j=0}^{m-1} c_{i,j} x_i^{m-1} \bmod I\,.$$

But, this means that $h_i := \sum_{j=0}^{m} c_{i,j} x_i^j \in I \cap K[x_i]$. As each such h_i has at most $\deg(h_i)$ zeros, there can be at most $\deg(h_1) \cdot \ldots \cdot \deg(h_n) < \infty$ points in $V(I)$. $\square$

In two variables, we can illustrate the situation of the finiteness theorem by the following example of a monomial diagram of a Gröbner basis $\mathcal{G}$ (each point (α_1, α_2) of the diagram corresponds to the monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$):



From now on, we restrict our considerations to the case that $V(I)$ is *finite*. In this case, we also say that I is a **zero-dimensional ideal**[8].

Remark 3.3.4 (Multiplicities of solutions*). As we count the roots of a univariate polynomial with multiplicities, we may introduce a multiplicity for each point p in the solution set $V(I) \subset \overline{K}^n$ of a zero-dimensional ideal I. We define the multiplicity of a solution $p \in V(I)$ to be the dimension of the $\overline{K}$-vector space $\overline{K}[\boldsymbol{x}]_{\langle \boldsymbol{x}-p\rangle} / I\overline{K}[\boldsymbol{x}]_{\langle \boldsymbol{x}-p\rangle}$. Here,

$$\overline{K}[\boldsymbol{x}]_{\langle \boldsymbol{x}-p\rangle} := \left\{ \frac{f}{g} \;\middle|\; f, g \in \overline{K}[\boldsymbol{x}],\ g(p) \neq 0 \right\}$$

denotes the localization of $\overline{K}[\boldsymbol{x}]$ at the maximal ideal $\langle \boldsymbol{x} - p\rangle$ corresponding to p. Note that for a univariate polynomial $f \in K[x]$, the dimension of $\overline{K}[x]_{\langle x-p\rangle}/\langle f\rangle$ equals the multiplicity of the factor $x - p$ in the irreducible decomposition of f.

With this definition, we may specify the statement of the finiteness theorem: if one of the conditions of the finiteness theorem holds, then, counted with multiplicities, $V(I)$ has precisely $\dim_K(K[\boldsymbol{x}]/I) = \dim_K(K[\boldsymbol{x}]/L(I))$ elements.

---

[8] This notion refers to the Krull dimension of $K[\boldsymbol{x}]/I$, which is zero. More generally, in Algebraic Geometry one introduces the dimension of I, respectively of $V(I)$, as the **Krull dimension** of the affine ring $K[\boldsymbol{x}]/I$, which is the maximal length d of an ascending chain of prime ideals $\mathfrak{p}_0 \subsetneq \mathfrak{p}_1 \subsetneq \ldots \subsetneq \mathfrak{p}_d$ in $K[\boldsymbol{x}]/I$. This dimension can be read from a Gröbner basis $\mathcal{G}$ for I by pure combinatorics: it equals the maximal cardinality of an **independent set** with respect to $L(I)$, that is, a subset $\boldsymbol{u} \subset \{x_1, \ldots, x_n\}$ such that no leading term of $\mathcal{G}$ is in $K[\boldsymbol{u}]$.

The proof of Corollary 3.3.3, $(e) \Rightarrow (a)$ already provides some **naive idea of how solving might work**: just compute generators h_i for $I \cap K[x_i]$, $i = 1, \dots, n$. Then solve the univariate equations $h_i(x_i) = 0$ (either algebraically by factorization over a suitable algebraic field extension or numerically) to obtain all possible values for each single coordinate entry of a solution. Form all possible tuples thereof and then discard non-solutions.

Thus, what is missing (from the theoretical point of view) is an algorithm for computing a generator for the ideal $I \cap K[x_i]$. We discuss this problem in a more general context:

The General Elimination Problem. Let $\boldsymbol{s} \subset \boldsymbol{x} = \{x_1, \dots, x_n\}$ be a subset of the set of variables, and let I be an ideal of $K[\boldsymbol{x}]$. Then the task is to eliminate the variables in $\boldsymbol{s}$ from I, that is, to compute a set of generators for the **elimination ideal** $I \cap K[\boldsymbol{x} \setminus \boldsymbol{s}]$.

A method to deal with this task is based on computing Gröbner bases with respect to orderings satisfying a particular property:

Definition 3.3.5. A monomial ordering $>$ on $K[\boldsymbol{x}]$ is called an **elimination ordering** with respect to $\boldsymbol{s}$ (the variables in $\boldsymbol{s}$) if the following implication holds for all $f \in K[\boldsymbol{x}]$:

$$\mathrm{lt}(f) \in K[\boldsymbol{x} \setminus \boldsymbol{s}] \implies f \in K[\boldsymbol{x} \setminus \boldsymbol{s}]\,.$$

In this case, we also say that $>$ has the **elimination property** with respect to $\boldsymbol{s}$ (the variables in $\boldsymbol{s}$).

Example 3.3.6. Let $\boldsymbol{s} \subset \boldsymbol{x}$, and let $\boldsymbol{t} := \boldsymbol{x} \setminus \boldsymbol{s}$. Moreover, let $>_{\boldsymbol{s}}$ on $K[\boldsymbol{s}]$ and $>_{\boldsymbol{t}}$ on $K[\boldsymbol{t}]$ be monomial orderings. The **product order** (or **block order**) $> = (>_{\boldsymbol{s}}, >_{\boldsymbol{t}})$ on $K[\boldsymbol{x}]$ is defined by

$$\boldsymbol{s}^\alpha \boldsymbol{t}^\gamma > \boldsymbol{s}^\beta \boldsymbol{t}^\delta \;:\Longleftrightarrow\; \boldsymbol{s}^\alpha >_{\boldsymbol{s}} \boldsymbol{s}^\beta \;\text{ or }\; (\boldsymbol{s}^\alpha = \boldsymbol{s}^\beta \;\text{ and }\; \boldsymbol{t}^\gamma >_{\boldsymbol{t}} \boldsymbol{t}^\delta).$$

It is a monomial ordering which has the elimination property with respect to $\boldsymbol{s}$. A particular example of a product ordering is the **lexicographic ordering** $>_{\mathtt{lp}}$ on $K[\boldsymbol{x}]$ which is an elimination ordering with respect to *each initial set of variables* $\boldsymbol{s} = \{x_1, \dots, x_k\}$, $k = 1, \dots, n$.

On the other hand, note that the degree reverse lexicographic ordering $>_{\mathtt{dp}}$ has *no* elimination property.

The following proposition relates the elimination property of a monomial ordering to a method for computing a set of generators for the elimination ideal $I \cap K[\boldsymbol{x} \setminus \boldsymbol{s}]$:

Proposition 3.3.7. *Let $>$ be an elimination ordering on $K[\boldsymbol{x}]$ with respect to $\boldsymbol{s} \subset \boldsymbol{x}$, and let $\mathcal{G}$ be a Gröbner basis for I with respect to $>$. Then the set*

$$\mathcal{G}' := \{g \in \mathcal{G} \mid \mathrm{lm}(g) \in K[\boldsymbol{x} \setminus \boldsymbol{s}]\}$$

is a Gröbner basis for $I \cap K[\boldsymbol{x} \setminus \boldsymbol{s}]$ with respect to the restriction of $>$ to $K[\boldsymbol{x} \setminus \boldsymbol{s}]$.

Proof. By the elimination property of $>$, we get $\mathcal{G}' \subset I'$, and it remains to show that, for each $f \in I \cap K[\boldsymbol{x} \setminus \boldsymbol{s}]$, there is some element of $\mathcal{G}'$ whose leading term divides $\mathrm{lt}(f)$.

But this is not difficult to show: As $\mathcal{G}$ is a Gröbner basis for I, we find an element $h \in \mathcal{G}$ whose leading monomial divides $\mathrm{lm}(f)$. As $\mathrm{lm}(f) \in K[\boldsymbol{x} \setminus \boldsymbol{s}]$, we also have $\mathrm{lm}(h) \in K[\boldsymbol{x} \setminus \boldsymbol{s}]$, that is, indeed, $h \in \mathcal{G}'$. □

Thus, the missing ingredient for the above naive algorithm for solving a system of polynomial equations could be n different Gröbner basis computations, with respect to orderings $>_i$, where $>_i$ has the elimination property for $\boldsymbol{s} := \boldsymbol{x} \setminus \{x_i\}$, $i = 1, \ldots, n$.

In practice, it turns out that the computation of one Gröbner basis is sufficient. But, we need a Gröbner basis with respect to the lexicographic ordering. Indeed, using the elimination property of $>_{\mathtt{lp}}$ with respect to any initial set of variables, we obtain the following proposition:

Proposition 3.3.8. *Let $I \subset K[\boldsymbol{x}]$ be a zero-dimensional ideal, and let $\mathcal{G}$ be a Gröbner basis for I with respect to $>_{\mathtt{lp}}$. Then there are elements $g_1, \ldots, g_n \in \mathcal{G}$ such that*

$$\begin{array}{ll} g_1 \in K[x_n]\,, & \mathrm{lt}(g_1) = c_1 x_n^{m_1}\,, \\ g_2 \in K[x_{n-1}, x_n]\,, & \mathrm{lt}(g_2) = c_2 x_{n-1}^{m_2}\,, \\ \quad\vdots & \qquad\vdots \\ g_n \in K[x_1, \ldots, x_n]\,, & \mathrm{lt}(g_n) = c_n x_1^{m_n}\,. \end{array} \tag{3.3.1}$$

Thus, having computed a lexicographic Gröbner basis $\mathcal{G} = \{g_1, \ldots, g_r\}$ for I (with $g_1, \ldots, g_n$ as in Proposition 3.3.8), computing $V(I)$ is mainly reduced to successive univariate solving: we solve $g_1(x_n) = 0$, substitute x_n by the obtained solutions $a_n^{(i)}$, then solve $g_2(x_{n-1}, a_n^{(i)}) = 0$, and so on.

What remains is to discard those elements of $V(g_1, \ldots, g_n)$ for which the remaining polynomials $g_{n+1}, \ldots, g_r$ in $\mathcal{G}$ do not vanish. The resulting algorithm may be formulated as follows:

Algorithm 3.3.9 (Zero-dimensional solving by elimination).

```
Input:  $f_1, \ldots, f_k \in K[x_1, \ldots, x_n]$
Assume: There is an algorithm for calculating the roots in $\overline{K}$ of a univariate poly-
        nomial with coefficients in $K$.
Output: If $I := \langle f_1, \ldots, f_k \rangle$ is zero-dimensional: the set $V(I)$.
        Otherwise: 'V(I) is an infinite set'.

  (1) Compute a reduced Gröbner basis $\mathcal{G}$ for $I$ with respect to $>_{\mathtt{lp}}$.

  (2) Set $\mathcal{G}_0 := \emptyset$;
      FOR $\ell = 1, \ldots, n$ DO
```

```
        • set $\mathcal{G}_\ell := \big(\mathcal{G} \cap K[x_{n+1-\ell},\dots,x_n]\big) \setminus \mathcal{G}_{\ell-1}$ ; [9]
        • set $\mathcal{G}'_\ell := \big\{g \in \mathcal{G}_\ell \mid \mathrm{lm}(g) \in K[x_{n+1-\ell}]\big\}$ ;
        • If $(\mathcal{G}'_\ell = \emptyset)$ Then Return ('V(I) is an infinite set') ;
        • choose $g_\ell \in G'_\ell$ ; [10]
(3) Compute $\mathscr{S}_1 := \big\{x \in \overline{K} \bigm| g_1(x) = 0\big\}$          // Univariate Solving
(4) Set $\ell := 1$ ;
    While $(\ell \leq n-1)$ Do                                     // Extending to Solutions
        • set $\ell := \ell + 1$ ;
        • set $\mathscr{S}_\ell := \bigcup_{\boldsymbol{a} \in \mathscr{S}_{\ell-1}} \{(x,\boldsymbol{a}) \in \overline{K}^\ell \mid g_\ell(x,\boldsymbol{a}) = 0\}$ ;
        • For each $g \in \mathcal{G}_\ell \setminus \{g_\ell\}$ Do           // Discarding Non-Solutions
            ◦ For each $\boldsymbol{b} \in \mathscr{S}_\ell$ Do
                If $g(\boldsymbol{b}) \neq 0$ Then $\mathscr{S}_\ell := \mathscr{S}_\ell \setminus \{\boldsymbol{b}\}$ ;
(5) Return $(\mathscr{S}_n)$.
```

Remark 3.3.10 (Numerical versus symbolic solving). As mentioned before, if K is a subfield of $\mathbb{C}$, the univariate solving step in the above algorithm could be performed numerically or by factorization. What approach needs to be chosen very much depends on the interpretation of the problem of solving itself.

If the task is to find an **approximation** of the complex solutions $\boldsymbol{x} \in V(I)$ up to N digits, we talk about **numerical solving**. It's main drawbacks are well-known to us (see the introduction to this course on Page 3).

On the other hand, the task may also be understood as the task of finding a **finite algebraic field extension** $K \subset L \subset \mathbb{C}$ such that all points of $V(I)$ are elements of L^n. This is usually referred to as **symbolic solving**. In the course of the above algorithm, all univariate solving steps would be implemented as factorization steps. More precisely, given the task of solving the univariate polynomial g symbolically, we would first compute an irreducible decomposition $g = g_1^{e_1} \cdot \ldots \cdot g_s^{e_s}$ over the given ground field, say K_0, and then pass to the extension field $K(\alpha_j) := K_0[t]/g_j$. All subsequent steps of the zero-dimensional solving algorithm involving the solution $a_j \in K(\alpha_j)$ of g then take place over $K(\alpha_j)$ or a suitable field extension thereof. This leads to chains of field extensions

$$K \subset K(\alpha_j) \subset K(\alpha_j, \beta_k) \subset \dots .$$

---

[9] Due to the elimination property of $>_{\mathtt{lp}}$, it suffices to consider the leading monomial of a polynomial $g \in \mathcal{G}$ to decide whether it belongs to $\mathcal{G}_\ell$ or not.

[10] Note that g_ℓ is a polynomial in ℓ variables: $x_{n+1-\ell}, \dots, x_n$.

The central drawback of this approach is that factorization over primitive algebraic extensions of $\mathbb{Q}$ is already a difficult task, and working with either several extension parameters or applying the theorem of the primitive element makes the occurring calculations even worse.

Besides the purely numerical solving and the symbolic solving, there is another, symbiotic approach to the solving problem, called **symbolic-numerical solving**. If we are interested in reliable numerical solutions, we may use some kind of symbolic preprocessing before applying numerical methods. The algorithm stated above may be regarded as a symbolic-numerical approach if the univariate solving steps in (3), (4) are implemented using a numerical solver.

Even if the symbolic-numerical version of Algorithm 3.3.9 is more efficient than the purely symbolic one, and even if it is more stable than a purely numerical method, there are still several computational problems to be discussed:

Problem A. The high complexity of Buchberger's Algorithm (combined with the rather difficult lexicographic ordering) may give rise to computations which do not terminate within a reasonable amount of time.

Problem B. The early use of a univariate numerical solver introduces rounding errors which tend to grow significantly in the course of the subsequent calculations. This is particularly nasty, if there are *multiple roots*, or *equally spaced roots*. If there is a multiple roots, introducing a rounding error usually leads to the splitting of this multiple root into several simple roots, thus, to the wrong number of different solutions. If the roots of f are equally spaced, a small perturbation of f may lead to a considerable perturbation of the roots: consider, for example, the polynomial

$$f := \prod_{j=1}^{20} (x+j)^{20} = x^{20} + 210 \cdot x^{19} + \ldots$$

which has 20 equally spaced real roots: $-20, \ldots, -1$. If we introduce a small error, say $f_{\text{approx}} = f + 10^{-9} \cdot x^{19}$, and compute the roots of f_{approx}, we get 12 real and 8 complex roots, among them $x = -16.57... + i \cdot 0.88...$, thus an error of almost 10^9 times the error in the input data.

Problem C. Discarding the non-solutions in Step (4) of the algorithm may be a serious problem if we have only a floating point approximation of the solutions. Indeed, the growing numerical errors may make such a decision hard and in some cases not feasable.

In the next section, we discuss some improvements of the basic algorithm for solving by elimination. In particular, we present the concept of triangular decompositions of zero-dimensional ideals, which provides a way out of Problem C.

Remark 3.3.11 (Geometric Interpretation of Elimination). Let K be an algebraically closed field, and let I be an ideal of $K[\boldsymbol{x}]$. Then the vanishing locus of the elimination ideal $I \cap K[\boldsymbol{x} \setminus \boldsymbol{s}]$ is obtained from the vanishing locus of I by projecting via $\pi : K^n = K^s \times K^{n-s} \to K^{n-s}$, $(\boldsymbol{s}, \boldsymbol{t}) \mapsto \boldsymbol{t}$. More precisely, the **Zariski closure theorem** states:

> *Let K be an algebraically closed field. Then $V(I \cap K[\boldsymbol{x} \setminus \boldsymbol{s}])$ is the smallest vanishing locus $V(J) \subset K^{n-s}$, J an ideal of $K[\boldsymbol{x} \setminus \boldsymbol{s}]$, such that $V(J) \supset \pi\big(V(I)\big)$.*

For a proof of the closure theorem, we refer to a course in Algebraic Geometry.

Example 3.3.12. Given the following system of polynomial equations in $\mathbb{Q}[x, y, z]$:

$$\begin{aligned} f_1 &= xy - 1 \\ f_2 &= xz - 1, \end{aligned}$$

find necessary conditions involving only the variables y and z for a point in $\mathbb{C}^3$ to be a solution of this system.

Setting $I := \langle f_1, f_2 \rangle \subset \mathbb{Q}[x, y, z]$, this task may be reformulated as an elimination problem: find generators for the elimination ideal $I \cap \mathbb{Q}[y, z]$. To this end, we choose the ordering $>_{\texttt{lp}}$ on $\mathbb{Q}[x, y, z]$ and compute a Gröbner basis by Buchberger's algorithm, obtaining

$$\mathcal{G} = \{f_1, f_2, f_3 := z \cdot (xy - 1) - y \cdot (xz - 1) = y - z\}.$$

Applying Proposition 3.3.7, we obtain that $\{f_3\}$ is a Gröbner basis for $I \cap \mathbb{Q}[y, z]$. In particular, $I \cap \mathbb{Q}[y, z] = \langle y - z \rangle$, and our desired condition is $y - z = 0$.

Geometrically, we have:

- $V(f_1) = \left\{ \left(x, \frac{1}{x}, z\right) \in \mathbb{C}^3 \mid x \neq 0 \right\}$,
- $V(f_2) = \left\{ \left(x, y, \frac{1}{x}\right) \in \mathbb{C}^3 \mid x \neq 0 \right\}$,

hence

- $V(I) = V(f_1) \cap V(f_2) = \left\{ \left(x, \frac{1}{x}, \frac{1}{x}\right) \mid x \in \mathbb{C} \setminus \{0\} \right\}$.

Projecting onto the yz-plane, we obtain the image

$$\pi\big(V(I)\big) = \left\{ (y, z) \in \mathbb{C}^2 \mid y = z \neq 0 \right\}.$$

In particular, this example shows that the image of $V(I)$ under the projection π may be a proper subset of the vanishing locus of the elimination ideal.

Caution! Again, we *need that K is algebraically closed* for this kind of geometric interpretation.

If we consider, for instance, the set of real solutions, it may well be that the image of $V_{\mathbb{R}}(I)$ under the projection π is the empty set, while the elimination ideal has a non-empty vanishing locus: Consider $I := \langle x^2 + y^2 - 1,\, y \rangle \subset \mathbb{Q}[x, y]$. Then I has no real solution, that is, $V_{\mathbb{R}}(I) = \emptyset$. Hence, $\pi\big(V_{\mathbb{R}}(I)\big) = \emptyset$. But, $I \cap \mathbb{Q}[y] = \langle y \rangle$ and $V_{\mathbb{R}}(y) = \{0\}$.

Exercises

Exercise 3.3.1. Consider the system of equations

$$\begin{aligned} x^2+2y^2 &= 3\,, \\ x^2+xy+y^2 &= 3\,. \end{aligned}$$

(1) If I is the ideal generated by these equations, find systems of generators for the elimination ideals $I \cap K[x]$ and $J \cap K[y]$.

(2) Determine $V(I)$!

(3) Which of the solutions are rational, that is, in $\mathbb{Q}^2$?

(4) What is the smallest extension field L of $\mathbb{Q}$ such that all complex solutions of the above system are in L^2.

Exercise 3.3.2. Let $>$ be a monomial ordering on $K[\boldsymbol{x}]$, and let $J \subset \{1,\dots,n\}$. Moreover, let $\boldsymbol{w} \in \mathbb{N}^n$ be a ('weight') vector such that $w_j > 0$ iff $j \in J$, and define

$$\boldsymbol{x}^{\boldsymbol{\alpha}} >_{\boldsymbol{w}} \boldsymbol{x}^{\boldsymbol{\beta}} :\Longleftrightarrow \boldsymbol{w}^T\boldsymbol{\alpha} > \boldsymbol{w}^T\boldsymbol{\beta} \text{ or } \big(\boldsymbol{w}^T\boldsymbol{\alpha} = \boldsymbol{w}^T\boldsymbol{\beta} \text{ and } \boldsymbol{x}^{\boldsymbol{\alpha}} > \boldsymbol{x}^{\boldsymbol{\beta}}\big)\,.$$

Prove that $>_{\boldsymbol{w}}$ defines an elimination ordering with respect to $\boldsymbol{s} := \{x_j \mid j \in J\}$.

Exercise 3.3.3. Let $I = \langle f, g\rangle \subset K[\boldsymbol{x}]$, and assume that f is of the form

$$f = cx_1^N + \text{ terms in which } x_1 \text{ has degree} < N\,,$$

where $c \in K \setminus \{0\}$ and $N > 0$. Moreover, let $I' = I \cap K[x_2,\dots,x_n]$. Prove that each solution $(a_2,\dots,a_n) \in V(I')$ extends to a solution $(a_1,\dots,a_n) \in V(I)$.

HINT: Consider f, g as univariate polynomials in $R[x_1]$, where $R = K[x_2,\dots,x_n]$, and use the theory of resultants (Proposition 2.2.4). Note that one can modify the given set of generators such that the leading coefficient of the second generator g does not vanish at any point $(a_2,\dots,a_n) \in V(I')$.

3.4 Preprocessing Steps

As explained in the previous section, there are some serious limitations for solving zero-dimensional systems of polynomial equations by applying Algorithm 3.3.9. This observation applies to both tasks: symbolic solving and symbolic-numerical solving.

This section is devoted to introducing different ideas and algorithms for a symbolic preprocessing which considerably enlarge the range of applicability of Algorithm 3.3.9, in particular, with respect to a symbolic-numerical approach. Actually, the preprocessing algorithms may be classified as follows:

Gröbner Basis Conversion Techniques. The performance of Buchberger's algorithm is sensitive to the choice of monomial ordering. A Gröbner basis computation with respect to a less favorable ordering such as $>_{\texttt{lp}}$ may easily run out of time or memory even in cases where a Gröbner basis computation with respect to a more efficient ordering such as $>_{\texttt{dp}}$ is very well feasible. Gröbner basis conversion algorithms are based on this observation.

Removing Multiplicities. If one is not interested in computing the multiplicities of the solutions, one possible goal for symbolic preprocessing may be to find an 'easier' system of polynomials defining an ideal which has the same radical as the ideal I under consideration. In particular, one may aim at computing generators for the radical of I.

Decomposition Techniques. These algorithms aim at decomposing a given system of polynomial equations into several smaller systems of polynomial equations. The set of solutions of the original system is then obtained by taking the union of the sets of solutions of the new systems. We discuss three rather different decomposition techniques: the factorizing Buchberger algorithm, primary decomposition, and triangular decompositions.

The following table illustrates the benefit of these preprocessing techniques with respect to the list of problems (A – C) in Remark 3.3.10:

	Problem A	Problem B	Problem C
Gröbner Basis Conversion	X		
Removing Multiplicities	((X))	X	(X)
Decomposition Techniques:			
- Factorizing Buchberger	X		(X)
- Primary Decomposition	(X)		X
- Triangular Decomposition			X

To discuss most of these preprocessing steps in detail is far beyond the scope of this introductory course. However, we want to present a rough idea of how such a preprocessing might work, and refer to the literature cited below for details and examples. The only approach that we inspect more closely is the decomposition into triangular bases. This is mainly due to the fact that it cures the problem of discarding non-solutions in the symbolic-numerical version of Algorithm 3.3.9.

3.4.1 Gröbner Basis Conversion

A Gröbner basis conversion algorithm proceeds along the following lines:

Step 1. Given an ideal $I \subset K[\boldsymbol{x}]$ and a 'slow' monomial ordering $>$ (e.g., $>_{\texttt{lp}}$), compute a Gröbner basis for I with respect to an appropriately chosen 'fast' ordering $>'$ (e.g., $>_{\texttt{dp}}$).

Step 2. Convert the result to a Gröbner basis with respect to $>$.

There are different types of conversion algorithms: for instance,

- the **FGLM algorithm**[11] converts Gröbner bases for zero-dimensional ideals by means of linear algebra. It is based on the fact that for a zero-dimensional ideal $I \subset K[\boldsymbol{x}]$ the quotient $K[\boldsymbol{x}]/I$ is a finite dimensional K-vector space, whence we may apply Gaussian elimination in Step 2.[12]
- the **Gröbner walk** algorithm[13], which works for arbitrary ideals, and which makes use of the so-called Gröbner fan.

Besides Gröbner basis conversion algorithms, there is another important approach to the computation of lexicographic Gröbner bases. This approach makes use of the additional information on a homogeneous ideal I provided by the Hilbert function[14] of $K[\boldsymbol{x}]/I$. The resulting algorithm is called the **Hilbert driven Buchberger algorithm** (see [Tra1]). Applying it usually means to proceed in two steps as well:

Step 1. Given a homogeneous ideal $I \subset K[\boldsymbol{x}]$ and a slow ordering $>$, compute a Gröbner basis for I with respect to an appropriately chosen fast ordering $>'$. From the result, compute the Hilbert function of $K[\boldsymbol{x}]/I$.

Step 2. Compute a Gröbner basis for I with respect to $>$. On your way, make use of the Hilbert function to further reduce the number of S-polynomials which actually need to be considered in Buchberger's test.

Here, the first step is of course superfluous if the Hilbert function of $K[\boldsymbol{x}]/I$ is already known.

3.4.2 Computing the Radical*

In view of Hilbert's Nullstellensatz, the set of solutions of a system of polynomial equations $f_1(\boldsymbol{x}) = \ldots = f_r(\boldsymbol{x}) = 0$ only depends on the radical of the ideal I generated by $f_1, \ldots, f_r$. Thus, if we are not interested in the multiplicities of the

---

[11] FGLM refers to the initials of Faugére, Gianni, Lazard, and Mora who designed the algorithm in [FGLM]. See [Wic] for a discussion of implementation issues.

[12] A trimmed-down version of Step 2 of the FGLM algorithm also provides a method for (simultaneously) computing polynomials $h_1, \ldots, h_n$ such that $I \cap K[x_i] = h_i$ (as required by the 'naive' algorithm for solving by elimination).

[13] See [CKM, AmG, Tra].

[14] For the definition of the Hilbert function and properties thereof, we refer to a course on commutative algebra, or to the textbook [GrP]. The latter textbook provides also an algorithm for computing the Hilbert function by means of Gröbner bases.

solutions, it does not matter whether we apply Algorithm 3.3.9 to the given set of polynomials $f_1, \ldots, f_r$ or to a set of generators for the radical of I.

Passing to a set of generators for $\sqrt{I}$ may have several advantages: first of all, it usually raises the numerical stability of the symbolic-numerical version of Algorithm 3.3.9[15]. Moreover, a Gröbner basis for $\sqrt{I}$ usually consists of fewer polynomials than a Gröbner basis for I, and these polynomials are of lower degree. In particular, computing a lexicographic Gröbner basis for a **radical ideal**, that is, an ideal J with $J = \sqrt{J}$, is usually considerably easier than computing a Gröbner basis for a general ideal with radical J.

The following lemma reduces the problem of computing the radical of a zero-dimensional ideal over $\mathbb{C}$ (or a subfield thereof) to a problem of eliminating[16] and of computing the squarefree part of a polynomial:

Lemma 3.4.1. *Let K be a perfect field*[17]*. If $I \subset K[\boldsymbol{x}]$ is a zero-dimensional ideal, we have*

$$\sqrt{I} = I + \left\langle h_{\text{red}}^{(1)}, \ldots, h_{\text{red}}^{(n)} \right\rangle, \tag{3.4.2}$$

where $h_{\text{red}}^{(i)}$ is the squarefree part of any polynomial $h^{(i)}$ such that $\langle h^{(i)} \rangle = I \cap K[x_i]$.

Indeed, Hilbert's Nullstellensatz shows that the ideal J on the right-hand side of (3.4.2) is contained in $\sqrt{I}$. The other inclusion holds since J is a radical ideal, which can be seen by considering the ideal generated by J in $\overline{K}[\boldsymbol{x}]$. This ideal turns out to be an intersection of maximal (hence, radical) ideals.

Remark 3.4.2 (Algorithms). There are different algorithms for computing the radical of an arbitrary ideal $I \subset K[\boldsymbol{x}]$. We mention here

- the algorithm by Krick and Logar [KrL] which is based on Lemma 3.4.1, and which is designed for coefficient fields of characteristic zero,
- the algorithm by Kemper [Kem], which can be considered as the analogue for coefficient fields of positive characteristic, and
- the algorithm by Eisenbud, Huneke and Vasconcelos [EHV], which relies on syzygy methods.

Also the algorithm for computing minimal associated primes due to Gianni, Trager and Zacharias [GTZ] can be used to compute the radical. A detailed survey of different methods for computing the radical of an ideal can be found in [DGP].

---

[15] This should be clear from the case of a principal ideal I, where passing from I to its radical means to pass from the generating polynomial f to the squarefree part of f.

[16] Here, we may use again the trimmed-down version of the FGLM algorithm mentioned in the foonote on Page 97.

[17] Note that each field of characteristic 0 and each finite field are perfect. If K is not a perfect field, it may happen that a polynomial is squarefree in $K[x]$ but not in $\overline{K}[x]$. Consider, for instance, $x^2 - t \in K[x]$, where $K = \mathbb{F}_2(t)$.

3.4.3 Factorizing Gröbner Bases

As the name suggests, the factorizing Buchberger algorithm combines Buchberger's algorithm 3.2.13 with multivariate factorization. When applied to an ideal $I \subset K[\boldsymbol{x}]$, it computes a list of Gröbner bases $\mathcal{G}_1, \ldots, \mathcal{G}_N$, consisting of *irreducible* polynomials only and satisfying

$$V(I) = V(\mathcal{G}_1) \cup \ldots \cup V(\mathcal{G}_N)\,. \tag{3.4.3}$$

In the case of a zero-dimensional ideal I, it is even possible to achieve a disjoint union in this decomposition.

The basic idea can be described as follows: If an s-polynomial in the main loop of Algorithm 3.2.13 has a non-zero normal form which decomposes as $h = g_1 g_2$, we can modify the algorithm so that the computation branches by means of a decomposition

$$V(\mathcal{G}, g_1 g_2) = V(\mathcal{G}, g_1) \cup V(\mathcal{G}, g_2).$$

Since the two sets on the right-hand side usually contain common points, it would be preferable to consider

$$V(\mathcal{G}, g_1 g_2) = V(\mathcal{G}, g_1) \cup (V(\mathcal{G}, g_2) \setminus V(g_1))$$

instead. If $\langle \mathcal{G}, g_2 \rangle$ is a zero-dimensional ideal, the set $V(\mathcal{G}, g_2) \setminus V(g_1)$ can be computed by elimination. Indeed, it equals the vanishing locus of the ideal

$$I' := \langle \mathcal{G}, g_2, tg_1 + 1 \rangle_{K[\boldsymbol{x},t]} \cap K[\boldsymbol{x}]\,.$$

The decomposition (3.4.3) of I into 'simpler' systems reduces, of course, the difficulties of computing each of the lexicographic Gröbner bases; but this is a tradeoff, because the total number of Gröbner bases to be computed increases, and we additionally have to deal with the problem of factorizing multivariate polynomials. Nevertheless, if the occurring polynomials can easily be factorized, this enhancement usually results in an improvement of performance. If, however, many of the polynomials turn out to be irreducible after expensive calculations, we have not reached any improvement. Hence, the factorizing Gröbner basis approach is most suitable if it uses a factorization algorithm which does not necessarily return a complete factorization, but some factorization which can be computed in a cheap way.

Another advantage of this approach is that each of the new systems is usually 'simpler' than the original one and, hence, the number of subsequent calculation steps (in particular, the discarding of non-solutions) after the first use of a univariate solver often drops.

For more details on the factorizing Buchberger algorithm, we refer to [Cza] and [Grä].

3.4.4 Primary Decomposition*

Another possibility to decompose the system of polynomial equations into several simpler systems is the concept of primary decomposition. As we are only interested in the set of solutions, it is sufficient for our purposes to construct a decomposition of the radical of our ideal (as we already observed in Section 3.4.2). This can be viewed as the task of specifying a given variety in terms of a union of irreducible varieties.[18]

There are several constructions for primary decomposition whose different concepts (e.g. [GTZ], [EHV], [Wan2, Wan]) are all beyond the scope of this course. A detailed overview of this field may be found in [DGP]. As each of these constructions involves rather expensive calculations itself, this method should be considered as an improvement to problems arising from the use of numerical solutions rather than as a speed-up.

3.4.5 Triangular Decompositions

The last decomposition method which we are going to mention in this course is specifically designed to reduce the problems arising from the subsequent use of numerical solvers: It decomposes a zero-dimensional system into several systems each of which consists of exactly n polynomials $g_1, \dots, g_n \in K[x_1, \dots, x_n]$ as in Proposition 3.3.8. The triangular structure of these systems makes discarding of non-solutions unnecessary.

Definition 3.4.3. A set of polynomials $\mathcal{T} := \{f_1, \dots, f_n\} \subset K[x_1, \dots, x_n]$ is called a **triangular basis** iff, for each $i = 1, \dots, n$,

(1) $f_i \in K[x_{n-i+1}, \dots, x_n]$, and

(2) the leading monomial of f_i with respect to $>_{\mathtt{lp}}$ is $x_{n-i+1}^{m_i}$.

We call a list of triangular bases $\mathcal{T}_1, \dots, \mathcal{T}_s$ a **triangular decomposition** for a zero-dimensional ideal I iff $V(I) = V(\mathcal{T}_1) \cup \ldots \cup V(\mathcal{T}_s)$.

Triangular decompositions of zero-dimensional ideals can be computed effectively. There are two different approaches based on Gröbner basis techniques, due to [Laz], respectively [Möl].

We present here a modification of the algorithm by Möller (see [Hil]). It has the additional advantage that the solution sets of the computed triangular bases $\mathcal{T}_1, \dots, \mathcal{T}_s$ are always pairwise disjoint, that is,

$$V(I) = V(\mathcal{T}_1) \,\dot{\cup}\, \ldots \,\dot{\cup}\, V(\mathcal{T}_s)\,.$$

---

[18] A variety $V \subset K^n$ is called **irreducible**, if it cannot be written as a union $V = V_1 \cup V_2$ of two varieties unless at least one of them is again V.

If $\{g_1, \ldots, g_r\}$ is a *reduced* lexicographic Gröbner basis for I, sorted such that $\operatorname{lt}(g_1) < \ldots < \operatorname{lt}(g_r)$, this algorithm recursively applies a decomposition of the form

$$V(\mathcal{G}) = \dot{\bigcup_{i=1}^{r-1}} \Big(V\big(\mathcal{G}, a_1, \ldots, a_{i-1}\big) \setminus V\big(a_i\big) \Big) \ \dot{\cup}\ \Big(V\big(\mathcal{G}, a_1, \ldots, a_{r-1}\big) \Big) , \tag{3.4.4}$$

where $a_j := \operatorname{lc}_{x_1}(g_j) \in K[x_2, \ldots, x_n]$ denotes the leading coefficient of g_j considered as a univariate polynomial in x_1. The key point for this approach are the equalities

$$\operatorname{lm}(g_r) = x_1^m , \tag{3.4.5}$$

for some $m \in \mathbb{N}$, and

$$V(\mathcal{G}, a_1, \ldots, a_{r-1}) = V(\underbrace{a_1, \ldots, a_{r-1}}_{\subset K[x_2, \ldots, x_n]}, g_r) \tag{3.4.6}$$

(see Lemma 3.4.6 below), which enables us to proceed by recursion on the number n of variables: If the ideal $\langle a_1, \ldots, a_{r-1} \rangle \subset K[x_2, \ldots, x_n]$ has a decomposition into triangular bases $\mathcal{T}_1', \ldots, \mathcal{T}_\ell'$, then $\mathcal{T}_1' \cup \{g_r\}, \ldots, \mathcal{T}_\ell' \cup \{g_r\}$ defines a triangular decomposition for the zero-dimensional ideal $\langle a_1, \ldots, a_{r-1}, g_r \rangle$.

Another foundation for Möller's approach is that, in the case of a zero-dimensional ideal $I \subset K[\boldsymbol{x}]$, the set $V\big(I\big) \setminus V\big(a\big)$, $a \in K[\boldsymbol{x}]$, is the vanishing locus of some ideal, which can be computed from I by saturating with respect to a (see Lemma 3.4.7).

Remark 3.4.4 (Saturation). Let $I \subset K[\boldsymbol{x}]$ be an ideal, and $a \in K[\boldsymbol{x}]$. Then

$$(I : a) := \{ f \in K[\boldsymbol{x}] \mid af \in I \}$$

is an ideal in $K[\boldsymbol{x}]$, called the **ideal quotient** of I by a. Iterating ideal quotients by the same polynomial a, we obtain an ascending chain of ideals:

$$I \subset (I : a) \subset (I : a^2) \subset (I : a^3) \ldots$$

Since the union of all these ideals is an ideal J of $K[\boldsymbol{x}]$. As J has a Gröbner basis, it is finitely generated. Since this finite set of generators is contained in some of the ideals $(I : a^m)$, $m \in \mathbb{N}$, the chain becomes **stationary**, that is, satisfies

$$(I : f^m) = (I : f^{m+k}) \ \text{ for all } k \in \mathbb{N}. \tag{3.4.7}$$

In this case, we call $(I : a^\infty) := (I : a^m)$ the **saturation** of I with respect to a. If m is chosen minimal with the property (3.4.7), we call it the **saturating exponent** of $(I : a^\infty)$.

The resulting algorithm for computing a triangular decomposition of a zero-dimensional ideal reads as follows:

Algorithm 3.4.5 (`triangular_decomposition`).

Input: $\mathcal{G} = \{g_1, \ldots, g_r\} \subset K[\boldsymbol{x}]$, a reduced Gröbner basis with respect to $>_{\mathtt{lp}}$.

Assume: $\mathcal{G}$ generates a zero-dimensional ideal of $K[\boldsymbol{x}] = K[x_1, \ldots, x_n]$, and $g_1, \ldots, g_r$ are sorted in such a way that $\mathrm{lm}(g_1) < \ldots < \mathrm{lm}(g_r)$.

Output: $\mathcal{T}_1, \ldots, \mathcal{T}_r$, a list of triangular bases such that $V(\mathcal{G}) = V(\mathcal{T}_1) \uplus \ldots \uplus V(\mathcal{T}_r)$.

```
(1) • Set $\mathcal{G}_1 := \mathcal{G}$ ;                                  // Initialization
    • FOR ($i = 1, \ldots, r-1$) DO
         ◦ set $a_i := \mathrm{lc}_{x_1}(g_i) \in K[x_2, \ldots, x_n]$ ;
    • Compute a reduced Gröbner basis $\mathcal{G}'$ for $\langle a_1, \ldots, a_{r-1}\rangle \subset K[x_2, \ldots, x_n]$
      with respect to $>_{\mathtt{lp}}$ ;
(2) • Set $L'$ := triangular_decomposition($\mathcal{G}'$) ;   // Recursion in dimension
    • Set $L := \{\mathcal{T}' \cup \{g_r\} \mid \mathcal{T}' \in L'\}$ ;
(3) FOR ($i = 1, \ldots, r-1$) DO
      • IF $a_i \notin \langle \mathcal{G}_i \rangle$ THEN
           ◦ Compute the reduced Gröbner basis $\mathcal{G}_i'$ with respect to $>_{\mathtt{lp}}$ for the
             saturation $(\langle \mathcal{G}_i \rangle : a_i^{\infty}) \subset K[\boldsymbol{x}]$ ;
           ◦ Set $L := L \cup$ triangular_decomposition($\mathcal{G}_i'$) ;
           ◦ Set $\mathcal{G}_{i+1} := \mathcal{G}_i \cup \{a_i\}$ ;
        ELSE                                               // $V(\mathcal{G}_i) \setminus V(a_i) = \emptyset$
           ◦ Set $\mathcal{G}_{i+1} := \mathcal{G}_i$ ;
(4) RETURN ($L$) .
```

As explained above, the correctness and termination of this algorithm is based on two lemmas, which we are going to formulate (and prove) next:

Lemma 3.4.6. *Let* $\mathcal{G} = \{g_1, \ldots, g_r\}$ *be a reduced lexicographic Gröbner basis for the zero-dimensional ideal* $I \subset K[x_1, \ldots, x_n]$*, and assume that* $\mathrm{lt}(g_1) < \ldots < \mathrm{lt}(g_r)$*. Then* $\mathrm{lt}(g_r) = x_1^m$ *for some* $m \in \mathbb{N}$*, and the set* $\mathcal{G}' := \{\mathrm{lc}_{x_1}(g_1), \ldots, \mathrm{lc}_{x_1}(g_{r-1})\}$ *is a Gröbner basis for the quotient* $(\langle g_1, \ldots, g_{r-1}\rangle : g_r)$ *with respect to* $>_{\mathtt{lp}}$*. In particular,* $V(\mathcal{G}', g_r) \subset V(\mathcal{G})$.

Proof. The first statement holds since for a zero-dimensional ideal I, there must be a pure power $x_1^m \in L(I)$. Choosing m minimally, the Gröbner basis property of $\mathcal{G}$ gives that for some i we have $\mathrm{lm}(g_i) = x_1^m$. As $\mathcal{G}$ is reduced, $\mathrm{lc}(g_i) = 1$, and x_1^m is the largest possible leading term for some g_i with respect to $>_{\mathtt{lp}}$. Hence, $i = r$ according to the way the g_i are ordered.

The second statement is not as obvious as the first one. To prove it, we first observe that the set $\mathcal{G}_1 := \{g_1, \ldots, g_{r-1}\}$ is a lexicographic Gröbner basis, because every s-polynomial of elements thereof has a standard expression in terms of $g_1, \ldots, g_r$ by the Gröbner basis property of $\mathcal{G}$, and as g_r cannot appear in a non-trivial summand of this standard expression (as the leading monomial of g_r is too large). The same kind of reasoning applies to the elements

$$S(g_i, g_r) := \mathrm{lc}_{x_1}(g_i) \cdot g_r - x_1^{m-\deg_{x_1}(g_i)} \cdot g_i \,,$$

$i = 1, \ldots, r-1$, of the ideal $\langle \mathcal{G} \rangle$, which have a standard expression in terms of $g_1, \ldots, g_r$ with remainder 0. Hence, $S(g_i, g_r) \in \langle g_1, \ldots, g_{r-1} \rangle$, which implies that

$$\mathrm{lc}_{x_1}(g_i) \in Q := \big(\langle g_1, \ldots, g_{r-1} \rangle : g_r \big) \,.$$

Moreover, given an element $f \in Q$, we know that $f \cdot g_r$ has a standard expression in terms of $g_1, \ldots, g_{r-1}$ with remainder 0 (since $\mathcal{G}_1$ is a Gröbner basis). This implies, in particular, that $\mathrm{lc}_{x_1}(f \cdot g_r) = \mathrm{lc}_{x_1}(f)$ is divisible by $\mathrm{lc}_{x_1}(g_i)$ for some $1 \leq i \leq r-1$, which proves the claim. □

Lemma 3.4.7. *Let $I \subset K[\boldsymbol{x}]$ be an ideal, and let $a \in K[\boldsymbol{x}]$. Then, for all $k \in \mathbb{N}$,*

$$V(I) = V(I, a) \cup V(I : a^k) \,.$$

If I is a zero-dimensional ideal, then

$$V(I : a^\infty) = V(I) \setminus V(a) \,.$$

In particular, we have then the disjoint decomposition

$$V(I) = V(I, a) \mathbin{\dot\cup} V(I : a^\infty) \,.$$

Proof. Both statements are a consequence of Hilbert's Nullstellensatz. Indeed, the first statement follows from the inclusions of ideals

$$I \subset \langle I, a \rangle \cap (I : a^k) \subset \sqrt{I} \,.$$

Here, the inclusion on the left-hand side is obvious. For the inclusion on the right-hand side, it suffices to observe that $f \in \langle I, a \rangle \cap (I : a^k)$ implies that $f^k \in \langle I, a^k \rangle$ and that $a^k f \in I$. Hence, $f^{k+1} = f^k \cdot f \in I$.

Now, let I be a zero-dimensional ideal. To see that $V(I : a^\infty) \subset V(I) \setminus V(a)$, we have to show that for each point $p \in V(I, a)$ there is an element $f \in (I : a^\infty)$ such that $f(p) \neq 0$. To see this, consider some polynomial g which vanishes on the finite set $V(I) \setminus \{p\} = \{q_1, \ldots, q_\ell\}$ (such a g can be choosen as a product $\prod_{i=1}^{\ell} h_i$ of linear polynomials h_i vanishing at q_i but not at p, $i = 1, \ldots, \ell$). Then ag vanishes on $V(I)$, that is, $ag \in \sqrt{I}$ by Hilbert's Nullstellensatz. It follows that $a^k g^k \in I$ for some $k \in \mathbb{N}$, that is, $f := g^k \in (I : a^\infty)$ and $f(p) \neq 0$.

On the other hand, let $p \in V(I)$ and $a(p) \neq 0$. Then each $f \in (I : a^m)$ satisfies $a(p)^m f(p) = 0$, hence $f(p) = 0$. Thus, $V(I : a^m) \supset V(I) \setminus V(a)$. □

Example 3.4.8. We now construct a triangular system for the zero-dimensional ideal generated by the polynomials

$$\begin{aligned} g_1 &= z^2 - 2\,, \\ g_2 &= y^2 + 2y - 1\,, \\ g_3 &= (y+z+1)\cdot x + yz + z + 2\,, \\ g_4 &= x^2 + x + y - 1\,, \end{aligned}$$

in $\mathbb{Q}[x,y,z]$. By a direct calculation, it is easy to check that $\mathcal{G} = \{g_1, \dots, g_4\}$ is a reduced Gröbner basis with respect to $>_{\mathtt{lp}}$ (where $x > y > z$). Moreover, we have $\mathrm{lm}(g_1) < \mathrm{lm}(g_2) < \mathrm{lm}(g_3) < \mathrm{lm}(g_4)$. Hence, we can use $\mathcal{G}$ as input for Algorithm 3.4.5. In Step (1), we initialize

$$\mathcal{G}_1 := \{g_1, g_2, g_3, g_4\}\,, \quad a_1 := z^2 - 2\,, \quad a_2 := y^2 + 2y - 1\,, \quad a_3 := y + z + 1\,.$$

Computing the reduced Gröbner basis $\mathcal{G}'$ for $\langle a_1, a_2, a_3\rangle$, we obtain $\mathcal{G}' := \{a_1, a_3\}$. In Step (2), we observe that $\mathcal{G}'$ is already a triangular set in $\mathbb{Q}[x,y]$. Hence, we can proceed directly to initializing L:

$$L := \{(a_1, a_3, f_4)\} = \{(z^2 - 2,\ y + x + 1,\ x^2 + x + y - 1)\}\,.$$

As $a_1 = g_1$ and $a_2 = g_2$, there is only one productive pass through the For loop in Step (3), namely the pass for $i = 2$:

$$\begin{aligned} \mathcal{G}_2' &:= \texttt{reducedGB}\left(\langle \mathcal{G}_1\rangle : a_2^\infty\right) = \{z^2 - 2,\ y - z + 1,\ x + z\} \\ L &:= L \cup \texttt{triangular\_decomposition}\,(\mathcal{G}_2') \\ &= \{(z^2 - 2,\ y + z + 1,\ x^2 + x + y - 1),\ (z^2 - 2,\ y - z + 1,\ x + z)\} \end{aligned}$$

This set L is then returned as the triangular decomposition. Hence, we get the disjoint decomposition

$$\begin{aligned} V\big(g_1, g_2, g_3, g_4\big) &= V\big(z^2 - 2,\ y + z + 1,\ x^2 + x + y - 1\big) \\ &\quad \mathbin{\dot\cup}\ V\big(z^2 - 2,\ y - z + 1,\ x + z\big). \end{aligned}$$

where the first of the two sets in the decomposition obviously contains 4 points in $\mathbb{C}^3$, the other one 2 points. We can determine the coordinates of these 6 points along the lines of Algorithm 3.3.9 symbolically:

Solutions corresponding to the first triangular basis:

- $z^2 - 2$ decomposes into linear factors over $\mathbb{Q}(\sqrt{2}) = \mathbb{Q}[a]/\langle a^2 - 2\rangle$. The corresponding roots are $z = \pm\sqrt{2}$.
- Substituting z by $\pm\sqrt{2}$ in the second generator, we get the univariate polynomial $y \pm \sqrt{2} + 1$, which has the unique zero $y = \mp\sqrt{2} - 1$.
- Substituting z by $\pm\sqrt{2}$ and accordingly y by $\mp\sqrt{2} - 1$ in the third generator, we get the univariate polynomial $x^2 + x \mp \sqrt{2} - 2$, which factorizes as $(x \mp \sqrt{2})(x \pm \sqrt{2} + 1)$ over $\mathbb{Q}(\sqrt{2})$.

We conclude that the solutions corresponding to the first triangular basis are

$$\left(\sqrt{2},\ -\sqrt{2} - 1,\ \sqrt{2}\right), \quad \left(-\sqrt{2} - 1,\ -\sqrt{2} - 1,\ \sqrt{2}\right),$$

$$\left(-\sqrt{2},\ \sqrt{2} - 1,\ -\sqrt{2}\right), \quad \left(\sqrt{2} - 1,\ \sqrt{2} - 1,\ -\sqrt{2}\right),$$

Solutions corresponding to the second triangular basis: Proceeding as before, we calculate the two solutions

$$\left(-\sqrt{2},\ \sqrt{2} - 1,\ \sqrt{2}\right), \quad \left(\sqrt{2},\ -\sqrt{2} - 1,\ -\sqrt{2}\right).$$

Bibliography

Textbooks on Computer Algebra

[CLO] Cox, D.; Little, J.; O'Shea, D.: *Ideals, Varieties, and Algorithms.* 2nd edition, Springer Verlag (1997).

[CLO1] Cox, D.; Little, J.; O'Shea, D.: *Using Algebraic Geometry.* Springer Verlag (1998).

[GCL] Geddes, K.; Czapor, S.R.; Labahn, G.: *Algorithms for computer algebra.* Kluwer Academic Publishers, Boston (1992).

[GrP] Greuel, G.-M.; Pfister, G.: *A* SINGULAR *introduction to commutative algebra.* Springer Verlag (2002).

[Mig] Mignotte, M.: *Mathematics for computer algebra.* Springer Verlag (1992).

[Stu] Sturmfels, B.: *Solving Systems of Polynomial Equations.* CBMS Regional Conference Series in Mathematics, American Mathematical Society, Providence, RI (2002).

[vGG] von zur Gathen, J.; Gerhard, J.: *Modern Computer Algebra.* Cambridge Univ. Press, Cambridge (1999).

[Wan] Wang, D.: *Elimination methods.* Springer Verlag (2001).

[Win] Winkler, F.: *Polynomial Algorithms in Computer Algebra.* Springer Verlag (1996).

Further Publications on Computer Algebra

[AmG] Amrhein, B.; Gloor, O.: *The Fractal Walk.* In: B. Buchberger and F. Winkler (Eds.): *Gröbner Bases and Applications,* 305–322, LNS 251, CUP, Cambridge (1998).

[Buc] Buchberger, B.: *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings.* PhD thesis, Innsbruck (1965).

[Buc1] Buchberger, B.: *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems. Aequationes mathematicae* **4**, 374–383 (1970).

[CaZ] Cantor, D.G.; Zassenhaus, H.: *A New Algorithm for Factoring Polynomials over Finite Fields.* Math. Comp. **36**, 587–592 (1981).

[CKM] Collart, S.; Kalkbrener, M.; Mall, D.: *Converting Bases with the Gröbner Walk.* J. Symb. Comput. **24**, 465–470 (1997).

[Cza] Czapor, S.R.: *Solving algebraic equations: Combining Buchberger's algorithm with multivariate factorization.* J. Symb. Comput. **7**(1), 49–53 (1989).

[DGP] Decker, W., Greuel, G.-M., Pfister, G.: *Primary decomposition: algorithms and comparisons*, In: B.H. Matzat et al (Eds.), *Algorithmic algebra and number theory, Heidelberg 1997*, 187–220, Springer Verlag (1999).

[EHV] Eisenbud, D.; Huneke, C.; Vasconcelos, W.: *Direct methods for primary decomposition.* Invent. Math. **110**, 207–235 (1992).

[FGLM] Faugère, C.; Gianni, P.; Lazard, D.; Mora, T.: *Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering.* J. Symb. Comput. **16**, 329–344 (1993).

[Gao] Gao, S.: *Factoring multivariate polynomials via partial differential equations.* Math. Comp. **72**, 801–822 (2003).

[Grä] Gräbe, H.-G.: *On Factorized Gröbner Bases.* In: Computer Algebra in Science and Engineering (ed. Fleischer, Grabmeier, Hehl, Küchlin). World Scientific Singapore, 77–89 (1995).

[Grä1] Gräbe, H.-G.: *Triangular Systems and Factorized Gröbner Bases.* In Proceedings AAECC Paris, LNCS **948**, Springer Verlag , 248–261 (1995).

[GTZ] Gianni, P.; Trager, B.; Zacharias, G.: *Gröbner bases and primary decomposition of polynomial ideals.* J. Symb. Comput. **6**, 149–167 (1988).

[Hil] Hillebrand, D.: *Triangulierung nulldimensionaler Ideale – Implementierung und Vergleich zweier Algorithmen.* Diplomarbeit, Universität Dortmund (1999).

[Kal] Kaltofen, E.: *Polynomial factorization.* In: B. Buchberger et al (Eds.), *Computer algebra*, 95–113, Springer Verlag (1982).

[Kal1] Kaltofen, E.: *Polynomial factorization 1982-1986.* In: I. Simon (ed.), *Computers in mathematics*, 285–309. Marcel Dekker, New York (1990).

[Kal2] Kaltofen, E.: *Polynomial factorization 1987-1991.* In: D.V. Chudnovsky and R.D. Jenks (Eds.), *Proceedings of LATIN'92, Sao Paulo*, 294–313. Springer Verlag (1992).

[KaS] Kaltofen, E.; Shoup, V.: *Subquadratic-Time Factoring of Polynomials over Finite Fields.* Math. Comp. **67** (223), 1179–1197 (1998).

[Kem] Kemper, G.: *The Calculation of Radical Ideals in Positive Characteristic.* J. Symb. Comput. **23**, 229–238 (2002).

[KrL] Krick, T.; Logar, A.: *An algorithm for the computation of the radical of an ideal in the ring of polynomials.* In: H.F. Mattson et al (Eds.), *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 9th International Symposium, AAECC-9, New Orleans, LA, USA, October 7-11, 1991, Proceedings*, 195–205, LNCS 539, Springer Verlag (1991).

[Laz] Lazard, D.: *Solving Zero-dimensional Algebraic Systems*, J. Symb. Comput. **13**, 117–131 (1992).

[MaM] Mayr, E.; Meyer, A.: *The complexity of the word problem for commutative semigroups and polynomial ideals.* Adv. in Math. **46**, 305–329 (1982).

[Möl] Möller, H.M.: *On decomposing systems of polynomial equations with finitely many solutions.* Appl. Algebra Eng. Commun. Comput. **4**, 217–230, (1993).

[Möl1] Möller, H.M.: *Gröbner Bases and Numerical Analysis.* In: B. Buchberger and F. Winkler (Eds.): *Gröbner Bases and Applications,* 159–178, LNS 251, CUP, Cambridge (1998).

[Sch] Schmidt, J.: *Algorithmen zur Berechnung größter gemeinsamer Teiler im univariaten Polynomring ber den ganzen Zahlen und deren effiziente Implementierung in Factory.* Diplomarbeit Univ. Kaiserslautern (1999).

[Tra] Tran, Q.-N.: *A Fast Algorithm for Gröbner Basis Conversion and its Applications.* J. Symb. Comput. **30**, 451–467 (2000).

[Tra1] Traverso, C.: *Hilbert functions and the Buchberger algorithm.* J. Symb. Comput. **22**, No. 4, 355–376 (1996).

[Wan1] Wang, P.S.: *An Improved Multivariate Polynomial Factoring Algorithm.* Math. Comp. **32**, 1215–1231 (1978).

[Wan2] Wang, D.: *Characteristic sets and zero structures of polynomial sets.* Preprint RISC-LINZ, Linz, Austria (1989).

[WaR] Wang, P.S.; Preiss Rothschild, L.: *Factoring Multivariate Polynomials Over the Integers.* Math. Comp. **29**, 935–950 (1975).

[Wen] Wenk, M.: *Resultantenmethoden zur Lösung algebraischer Gleichungssysteme implementiert in* SINGULAR. Diplomarbeit Univ. Kaiserslautern (1999).

[Wic] Wichmann, T.: *Der FGLM-Algorithmus: verallgemeinert und implementiert in* SINGULAR. Diplomarbeit Univ. Kaiserslautern (1997).

[Zas] Zassenhaus, H.: *On Hensel factorization.* J. Number Theory **1**, 291–311 (1969).

Literature on Related Fields

[AtM] Atiyah, M.F.; McDonald, I.G.: *Introduction to commutative algebra.* Addison-Wesley (1969).

[BSW] Beutelspacher, A.; Schwenk, J.; Wolfenstetter, K.-D.: *Moderne Verfahren der Kryptographie.* Vieweg (2001).

[Buc] Buchmann, J.: *Introduction to cryptography.* Springer Verlag (2001).

[Coh] Cohen, H.: *A course in computational algebraic number theory.* Springer Verlag (1993).

[Kro] Kronecker, L.: *Grundzüge einer arithmetischen Theorie der algebraischen Grössen.* Journal für reine und angew. Math. **92**, 1–122 (1882).

[Kro1] Kronecker, L.: *Die Zerlegung der ganzen Grössen eines natürlichen Rationalitäts-Bereichs in ihre irreductiblen Factoren.* Journal für reine und angew. Math. **94**, 344–348 (1883).

[Lan] Lang, S.: *Algebra.* Rev. 3rd ed., Springer Verlag (2002).

[Mat] Matsumura, H.: *Commutative ring theory.* Cambridge Univ. Press, Cambridge (1986).

[RSV] Reiffen, H.-J.; Scheja, G.; Vetter, U.: *Algebra.* 2., durchges. Aufl., unveränd. Nachdr., B-I-Wissenschaftsverlag (1991).

[vSc] von Schubert, F.T.: *De inventione divisorum.* Nova Acta Acad. Sci. Imp. Petropolitanae **11**, 172–182 (1793).

Index

Symbols

$(I : a^\infty)$, 101
$||f||_2$, 55
$\mathrm{lt}(f)$, 10
$f \mid g$, 11
$\mathrm{cont}(f)$, 22
D_f, 26
$\mathbb{F}_p$, 14
$\mathbb{F}_q$, 14
gcd, 12
$\mathrm{lc}_x(f)$, 101
$\mathrm{lc}(f)$, 10
$\mathrm{lm}(f)$, 10
$M(f)$, 55
$\mathrm{Mon}(\boldsymbol{x})$, 7
$\mathbb{N}$, 7
$\mathrm{Norm}(f)$, 71
$\mathrm{pp}(f)$, 22
$\mathrm{Res}_{f,g}$, 26
$R[\boldsymbol{x}]$, 7
$R[x]_{<k}$, 25
$S_{f,g}$, 26
$\mathrm{spoly}(f, g)$, 83
$V(I)$, 73

A

absolute factorization, 19
algebraic closure, 19
algebraically closed, 17
algorithm
  Berlekamp factorization, 44
  Berlekamp splitting, 42
  Berlekamp/Cantor/Zassenhaus, 46
  Buchberger, 84
  distinct degree factorization, 37
  division with remainder, 75
  interreduction, 81
  Kronecker's principle, 51
  Las Vegas, 46
  Monte Carlo, 46
  repeated squaring, 43
  squarefree decomposition in $\mathbb{F}_q[x]$, 33
  triangular decomposition, 102
  zero-dimensional solving, 91

B

bad
  reduction problem, 70
  zero problem, 70
Berlekamp, 40, 53
  factorization, 44
  Splitting, 43
  splitting, 42
binomial, 86
  ideal, 86
block order, 90
Buchberger algorithm, 75, 84
  factorizing, 99
  Hilbert driven, 97
Buchberger's criterion, 85

C

Cantor, 40, 46
characteristic, 15
Chinese Remainder Theorem, 13
coefficient, 7
content, 22
coprime, 13
cyclotomic polynomial, 55

D
DDF, 36
  coarse, 39
degree
  total, 7
Dickson's Lemma, 84
discriminant, 26
distinct degree factorization, 36
  coarse, 39
divide, 11
divisible, 11
division
  algorithm, 75
  theorem, 75
  with remainder, 20, 75
    pseudo, 21

E
EEA, 21
Eisenstein, 11
elimination
  geometric interpretation, 94
  ideal, 90
  order, 90
  property, 90
equal degree factorization, 47
Euclidean
  Algorithm, 19
    classical, 20
    extended, 21
    primitive, 23
  domain, 19, 20
exponent vector, 7
extended Euclidean algorithm, 21

F
factorial ring, 11
factorization
  distinct degree, 36
  equal degree, 47
  irreducible, 11, 18
    in $\mathbb{F}_q[x]$, 31
    in $\mathbb{Z}[x]$, 49
    in $\mathbb{Z}[x_1, \ldots, x_n]$, 65
  multivariate, 65
  squarefree, 31
  unique, 11
FGLM algorithm, 97
f-reducing polynomial, 41
Frobenius map, 40

G
g.c.d., 12
Galois field, 14
Gaussian Elimination, 74
Gauß
  Lemma of, 11
Gröbner basis, 78

H
Hilbert Nullstellensatz, 87
  weak, 88

I
ideal quotient, 101
iff, 8
independent set, 89
integral domain, 8
  factorial, 11
interreduced, 80
interreduction, 81
invertible, 10
irreducible, 10
  factorization, 11
    in $\mathbb{F}_q[x]$, 31
    in $\mathbb{Z}[x]$, 49
    in $\mathbb{Z}[x_1, \ldots, x_n]$, 65
  variety, 100

K
Kronecker, 50, 65
  Principle, 51
Krull dimension, 89

L
Landau, 55
leading
  coefficient, 10
  ideal, 78

  monomial, 10
  term, 10
leading coefficient problem, 70
lexicopraphic ordering, 90

M
measure, 55
minimal, 80
monic, 10
monomial, 7
  ideal, 84
  ordering, 9
multivariate, 8

N
non-zerodivisor, 8
normal form, 12, 77
numerical solving, 3, 92

O
order, 16
  graded lex, 9
  graded reverse lex, 9
  lexicographic, 9

P
polynomial
  multivariate, 8
  univariate, 8
polynomial ring, 7
prime, 10
  relatively, 13
prime field, 14
primitive, 16, 22
  part, 22
product order, 90
pseudo division with remainder, 21

Q
quadratic norm, 55

R
radical, 87
  computation of, 97
  ideal, 98
real roots, 25
reduced, 77, 80
  normal form, 77
reducing polynomial, 41
remainder, 75
resultant, 26
ring, 8

S
s-polynomial, 83
saturation, 101
squarefree, 31
  decomposition, 33
  factorization, 31
  part, 31, 88
standard expression, 75
stationary, 101
Sturm chain, 25
Sylvester matrix, 26
symbolic solving, 3, 92

T
tail, 81
term, 7
total degree, 7
triangular
  basis, 100
  decomposition, 100
    computation of, 100
true factors, 61, 66

U
UFD, 11, 12
unique factorization domain, 11
unit, 10
univariate, 8

V
Vandermonde determinant, 30

W
well-ordering, 9

Z
Zariski closure theorem, 94

Zassenhaus, 40, 46, 53
zero-dimensional ideal, 89
zerodivisor, 8