Normal forms over DEDEKIND domains, efficient implementation in the computer algebra system KANT

submitted by
Dipl.–math Andreas Hoppe

Accepted thesis (Dissertation)
of the department 3, Mathematics,
of the Technical University of Berlin
for the award of the academical degree

Doctor of the natural sciences

Commission:

Head	Prof. Dr. rer. nat. Dirk Ferus
Referee	Prof. Dr. rer. nat. Michael E. Pohst
Referee	Prof. Dr. rer. nat. Henri Cohen

Date of the academic debate: 16.11.1998

Berlin, 1998
D83

Contents

Introduction

Relative extensions of algebraic number fields are a basic concept in algebraic number theory and have been theoretically well–studied. The classic aim is to find an absolute field extension over $\mathbb{Q}$ which is isomorphic to the relative extension. If this connection is established the problem can be considered solved.

This is not the case in computational algebraic number theory. The difference between absolute and relative extension is mirrored in different representations. The pity is that a relative extension does not have a relative integral basis in general. But there is the notion of a *pseudobasis*, which goes back to the theorem [O'M63, §81:3]. The pseudobasis replaces the integral basis and is sufficient for computational purposes although quite a few problems must be solved.

A more general approach deals with finitely generated modules over integral domains which can be represented with pseudobases. There are infinitely many different pseudobases for one module, and the overall subject of this thesis is to give efficient algorithms to produce a *good* pseudobasis (which is called normal form) of a given module. As integral bases are represented by matrices, pseudobases are represented by pseudomatrices, which can be seen as a generalization. The normal form for pseudomatrices can be seen as a generalization of the Hermite normal form (HNF) for matrices.

The earliest algorithmic approach is in [BP91], which is based on the "almost" constructive proof in [O'M63]. Another algorithm is given in [Coh96], including a convention for a unique normal form. Relative normal form algorithms are implemented in KANT([Kant]) and in gp([Pari]).

In [BP91] and [Coh96], only the case of the ring of integers of an algebraic number field over $\mathbb{Q}$ is dealt with. But, for many parts of the theory, generalizations are possible. As the most general case, we will consider integral domains. Sometimes we require the existence of inverse ideals and factorizations of ideals, which is well–defined only in Dedekind rings. The main aspect of this work is the implementation of algorithms. These algorithms are given for orders in algebraic number fields over $\mathbb{Q}$ which will be called **algebraic number rings**.

The ring of integers of an algebraic number field (or the maximal order of an algebraic number field), referred to in the following as $o_{\mathcal{K}}$, is an algebraic number ring and a Dedekind ring.

The first chapter presents algorithms for ideals in algebraic number rings and Dedekind number rings. Some definitions and propositions with non–constructive proofs are given for more general rings.

The second chapter deals with the theory of reducing fractional elements with ideals over integral domains. The latter part gives reducing methods for algebraic number rings.

The third chapter deals with the theory of the representation of finitely generated modules over integral domains with matrices. A completely satisfying result (which is the proof of the unique existence of a normal form) can only be obtained for EUCLIDean rings.

The fourth chapter deals with the theory of pseudomatrices over integral domains. For DEDEKIND rings we obtain a satisfactory correlation of pseudomatrices and finitely generated modules. For maximal orders, normal form algorithms are given.

The fifth chapter introduces an important application of the normal form algorithms: the relative ideals.

The sixth chapter gives results of practical investigations with the author's implemented versions of normal form algorithms and also a comparison to the implemented version in [Pari].

I would like to refer the reader to the index if any notation or symbol is unclear.

First of all I would like to thank my supervisor Prof. Michael E. Pohst. He pointed me to the subject of this work and helped me in all stages of the work with explanations, comments, and criticisms.

I would like to thank Prof. Henri Cohen for being my second referee, for giving me an ear for questions, and for the hospitality at the Université Bordeaux.

The KANT project has given my work a solid framework. Most of all I'd like to appreciate the help of Dr. Claus Fieker for support and collaboration in many aspects, Dr. Mario Daberkow for the introduction to the relative normal forms, Florian Hess for the introduction to the KANT system, Carsten Friedrichs and Jürgen Klüners for stimulating conversations, Oliver Voigt for the technical support, and all the KANT group.

I am greatly indebted to Catherine Roegner for revising the text of the thesis to improve the English.

Last but not least I thank Brigitte Trewin for her love.

The typesetting has been done with [Knu86], using [Kop92] and [GMS94], with the aid of [Tho] and [Ned94].

Chapter 1

Ideals in algebraic number rings

This chapter contains various algorithms dealing with ideals and algebraic numbers in orders of an algebraic number field. They form the basis for the normal form algorithms for pseudomatrices in chapter 4 and for their efficient implementation.

In the first four sections a collection of efforts is given to improve the efficiency of the arithmetic and basic functions for ideals over algebraic number rings as theoretically described in [PZ93, pp. 396–408] and in [Coh95, pp. 186–193, 202–204]. The description in [vS87, pp. 23–96] is closest to the actual implementation in KANT.

The rest of the chapter consists of detailed formulations, improvements, and generalizations of algorithms for algebraic numbers and ideals which are based on either [BP91] or [Coh96].

The algorithms are supplemented with general formulations of definitions, propositions, and methods for integral domains (commutative unital rings without nontrivial zero divisors) or DEDEKIND rings, where applicable.

As usual the following definitions on ideals are used.

Definition 1.0.1:

Let $\mathcal{D}$ be an integral domain and $\mathcal{K}$ its quotient field. A $\mathcal{D}$–module $\mathfrak{a}$ contained in $\mathcal{D}$, including the zero ideal (denoted $0\mathcal{D}$), is called an **integral** $\mathcal{D}$**–ideal**.

A nonzero $\mathcal{D}$–module $\mathfrak{a} \subset \mathcal{K}$ is called a **fractional** $\mathcal{D}$**–ideal** iff there exists a $\delta \in \mathcal{D}$ such that $\delta\mathfrak{a}$ is a $\mathcal{D}$–ideal.

1.1 Representations of ideals over algebraic number rings

Let $\mathcal{K}$ be a finite algebraic field extension of $\mathbb{Q}$, which will be called an **algebraic number field**.

Let $\mathcal{O}$ be an order of $\mathcal{K}$. Let $\Omega = (\omega_1, \ldots, \omega_n)$ be a $\mathbb{Z}$–basis of $\mathcal{O}$:

$$\mathcal{O} = \sum_{i=1}^{n} \mathbb{Z}\omega_i. \tag{1.1.1}$$

This basis can be viewed as a $\mathbb{Q}$–vector space basis of $\mathcal{K}$:

$$\mathcal{K} = \sum_{i=1}^{n} \mathbb{Q}\,\omega_i.$$

Fractional $\mathcal{O}$–ideals are presented by a nonzero integral $\mathcal{O}$–ideal $\mathfrak{a}$ and a denominator $d \in \mathbb{N}$ as $\frac{\mathfrak{a}}{d}$.

There are two basic principles for representing an integral $\mathcal{O}$–ideal: either as a $\mathbb{Z}$–basis (of n elements of $\mathcal{O}$) or as an $\mathcal{O}$–generating system (where two elements of $\mathcal{O}$ suffice — therefore it is called a two–element presentation).

An integral $\mathcal{O}$–ideal $\mathfrak{a}$ can be represented as n algebraic numbers $\xi_i \in \mathcal{K}$, for $i \in \mathbb{N}_n$, or equivalently, a matrix $\mathbf{A} = (\mathsf{A}_1, \ldots, \mathsf{A}_n)$ over $\mathbb{Z}$ (where the columns $\mathsf{A}_1, \ldots, \mathsf{A}_n$ of $\mathbf{A}$ satisfy $\xi_i = \Omega \mathbf{A}_i$):

$$\mathfrak{a} = \sum_{i=1}^{n} \xi_i \mathbb{Z} = \sum_{i=1}^{n} \Omega \mathsf{A}_i \mathbb{Z}. \tag{1.1.2}$$

This is called the **basis presentation** of $\mathfrak{a}$. For practical purposes, it is very useful if $\mathbf{A}$ is in upper triangular HERMITE normal form (HNF). See the general definition 3.2.4.

An integral ideal $\mathfrak{a}$ can also be represented by a natural number a and an algebraic integer $\alpha \in \mathcal{O}$ as $\mathfrak{a} = a\mathcal{O} + \alpha\mathcal{O}$, which is called **two–element presentation**. Similarly, a principal ideal is represented with a single algebraic number α as $\mathfrak{a} = \alpha\mathcal{O}$.

It is possible to convert the basis presentation into the two–element presentation and vice–versa (described in [vS87, pp. 40–41]).

1.2 Arithmetic for ideals of algebraic numbers

Let $\mathcal{O}$ be an order of an algebraic number field. Since fractional algebraic numbers always have natural denominators, it is easy to base fractional on integral ideal arithmetic:

Algorithmic idea 1.2.1: Basing fractional arithmetic on integral arithmetic

Let $\mathfrak{a}$ and $\mathfrak{b}$ be two fractional ideals. Let d_1 be the denominator of $\mathfrak{a}$ and d_2 be the denominator of $\mathfrak{b}$. Let d_3 be the maximal natural factor of $d_1\mathfrak{a}$ (see definition 1.3.5), d_4 the maximal natural factor of $d_2\mathfrak{b}$. Let $d_5 = \gcd(d_2, d_3)$ and $d_6 = \gcd(d_1, d_4)$. Then

$$\mathfrak{a}\mathfrak{b} = \frac{(\frac{d_1}{d_5}\mathfrak{a})(\frac{d_2}{d_6}\mathfrak{b})}{\frac{d_1}{d_6}\frac{d_2}{d_5}} \quad \text{and} \quad \frac{d_1}{d_5}\mathfrak{a} \subset \mathcal{O}, \quad \frac{d_2}{d_6}\mathfrak{b} \subset \mathcal{O}, \quad \frac{d_1}{d_6} \in \mathbb{N}, \quad \frac{d_2}{d_5} \in \mathbb{N}.$$

Let $d = \operatorname{lcm}(d_1, d_2)$. Then

$$\mathfrak{a} + \mathfrak{b} = \frac{d\mathfrak{a} + d\mathfrak{b}}{d}.$$

These formulas are used to base the arithmetic of fractional ideals on integral ideals. Therefore we may confine our efforts to methods for integral ideals.

1.2.1 Special multiplication algorithms

If we analyze the computation run times (this is demonstrated in section 6.5) of the subfunctions of a normal form algorithm, we realize that a great part of the total run time is used for ideal multiplication. Therefore it is essential to optimize ideal multiplication for efficient normal form implementations.

We want to multiply two ideals. Ideals can be given in either basis presentation or two–element presentation. So we have three possible cases for the presentations of the two ideals: two basis presentations, a basis presentation and a two–element presentation, and two two–element presentations.

Since it is possible to convert the basis presentation into the two–element–presentation and vice–versa, it is sufficient to have a multiplication algorithm for one of the three cases, say to multiply two ideals in basis–presentation.

From the aspect of efficiency, this is not satisfactory, since the conversion of the ideal presentation is not an easy task. Indeed, it is possible to find special multiplication algorithms for each of the three cases for the presentations of the two ideals. Thus, for each ideal multiplication, the question arises of whether it is better to use the given presentations or if it is worth computing another presentation for one or both of the ideals.

This question is modified if we use a particular ideal in a program or a computational session not for just one multiplication, but also for other computations. Then it might be better to compute another presentation, even though this might not be preferable for one multiplication. In practice many ideals are given in more than one presentation. This raises another question: Which of the special multiplication algorithms is best, regardless of any computational costs for presentation conversion?

There is another multiplication method. It uses a special case of a two–element presentation, which is called normal presentation. The multiplication of two ideals given in two compatible normal presentations is extremely fast, but the creation of two compatible normal presentations is relatively expensive.

Below are more detailed descriptions of the different algorithms. In section 6.2 results of experiments are given and, in conclusion, the heuristics used in KANT to have a good overall performance of ideal multiplication are discussed.

Algorithmic idea 1.2.2: Basis presentations algorithm

The multiplication of two basis presentations is described in detail in [vS87, p. 35]. It involves the multiplication of each of the $\mathbb{Z}$–basis elements of the first ideal with each of the $\mathbb{Z}$–basis elements of the second ideal, resulting in n^2 algebraic numbers which form a $\mathbb{Z}$–generating set for the product. With an HNF calculation this is transformed to a $\mathbb{Z}$–basis of the product.

Many efforts have been done to improve integer HNF computations: for the modular method see [HHR93], for a formal analysis see [KB79] and [CC82], for other approaches see [Hop94], [PB74], and [Fru76], and for BLANKINSHIP's method see [Bla63], [HM94].

Algorithmic idea 1.2.3: Mixed presentations algorithm

This algorithm is suggested in [Coh95, p. 188]. Let $\mathfrak{a} = a\mathcal{O} + \alpha\mathcal{O}$ and $\mathfrak{b} = \xi_1\mathbb{Z} + \cdots + \xi_n\mathbb{Z}$. Because $\mathfrak{b}$ is an ideal we have $\mathcal{O}\mathfrak{b} = \mathfrak{b}$. Then

$$\mathfrak{ab} = a\mathcal{O}\mathfrak{b} + \alpha\mathcal{O}\mathfrak{b} = a\mathfrak{b} + \alpha\mathfrak{b} = a\xi_1\mathbb{Z} + \cdots + a\xi_n\mathbb{Z} + \alpha\xi_1\mathbb{Z} + \cdots + \alpha\xi_n\mathbb{Z}.$$

Therefore $a\xi_1, \ldots, a\xi_n, \alpha\xi_1, \ldots, \alpha\xi_n$ is a $\mathbb{Z}$–generating system for $\mathfrak{ab}$, and we compute the HNF to get a basis for $\mathfrak{ab}$. So the algorithm involves $2n$ multiplications of algebraic numbers and an HNF of an $n \times 2n$ matrix.

This algorithm is so fast that [Coh95, p. 188] prefers the determination of a two–element–presentation and a mixed presentations algorithm to the basis–presentations algorithm in case only the two $\mathbb{Z}$–bases are given. See section 6.2 for the author's results.

Algorithmic idea 1.2.4: Four generators algorithm

Let $\mathfrak{a} = a\mathcal{O} + \alpha\mathcal{O}$ and $\mathfrak{b} = b\mathcal{O} + \beta\mathcal{O}$ with $a, b \in \mathbb{Z}$ and $\alpha, \beta \in \mathcal{O}$. Then $\mathfrak{ab} = ab\mathcal{O} + a\beta\mathcal{O} + b\alpha\mathcal{O} + \alpha\beta\mathcal{O}$, which gives an $\mathcal{O}$–generating set of four elements. With the representation matrices of the algebraic numbers ab, $a\beta$, $b\alpha$, and $\alpha\beta$, we get a $\mathbb{Z}$–generating system of $\mathfrak{ab}$ with $4n$ elements. We then apply the HNF algorithm to get a $\mathbb{Z}$–basis.

This algorithm requires one multiplication of algebraic numbers (the multiplication of a rational integer with an algebraic number can be neglected in complexity, since it needs only n integer multiplications as opposed to $2n^2 - n$ integer multiplications and additions for a multiplication of two integral algebraic numbers, which can be seen in [Klü97, lemma 3.5]), three representation matrix computations for algebraic numbers (the representation matrix of the rational integer ab is the identity matrix multiplied by ab and its computation can be neglected in complexity), and an HNF computation of an $n \times 4n$ matrix. But, the HNF computation behaves more like a HNF computation of an $n \times 3n$ matrix because of the simple form of the representation matrix of ab.

Algorithmic idea 1.2.5: Normal presentations algorithm

Normal presentations are only defined for the maximal order $o_\mathcal{K}$ of an algebraic number field $\mathcal{K}$. The algorithms are described in detail in [PZ93, pp. 400–406]. The main results are cited below:

Definition 1.2.6:

Let $\mathcal{P}$ be a set of prime numbers, $\mathcal{P}_\mathcal{K}$ the set of all prime $o_\mathcal{K}$–ideals dividing any of the ideals $po_\mathcal{K}$ where $p \in \mathcal{P}$. (Prime ideals are nonzero integral ideals which are not equal to any product of two nontrivial integral ideals.)

Let $\mathfrak{a}$ be an integral $o_\mathcal{K}$–ideal. The pair $(a, \alpha) \in \mathbb{N} \times \mathcal{K}^\times$ is called a $\mathcal{P}$**–normal presentation of** $\mathfrak{a}$ iff the following four conditions are satisfied:

1. $\mathfrak{a} = ao_\mathcal{K} + \alpha o_\mathcal{K}$;
2. $\mathfrak{a} = \prod_{\mathfrak{p}\in\mathcal{P}_\mathcal{K}} \mathfrak{p}^{v_\mathfrak{p}(\mathfrak{a})}$ where $v_\mathfrak{p}(\mathfrak{a}) \in \mathbb{Z}$ is the $\mathfrak{p}$-adic valuation of $\mathfrak{a}$;
3. $ao_\mathcal{K} = \prod_{\mathfrak{p}\in\mathcal{P}_\mathcal{K}} \mathfrak{p}^{v_\mathfrak{p}(a)}$ where $v_\mathfrak{p}(a) \geq 0$;
4. no $\mathfrak{p} \in \mathcal{P}_\mathcal{K}$ occurs in the prime ideal factorization of $\alpha\mathfrak{a}^{-1}$.

Proposition 1.2.7:

Let $\mathcal{P}$ be a set of prime numbers, let $\mathfrak{a}$ and $\mathfrak{b}$ be integral $o_\mathcal{K}$–ideals. If (a, α) and (b, β) are $\mathcal{P}$–normal presentations of $\mathfrak{a}$ and $\mathfrak{b}$ respectively, then $(ab, \alpha\beta)$ is a $\mathcal{P}$–normal presentation of $\mathfrak{ab}$.

Theorem 1.2.8:
Let $a, b \in \mathbb{N}$, $\alpha \in o_{\mathcal{K}}$, $\alpha \neq 0$, and $\mathfrak{a} = ao_{\mathcal{K}} + b^{-1}\alpha o_{\mathcal{K}}$ be an integral $o_{\mathcal{K}}$–ideal. Let $\mathcal{P}$ be the set of all prime numbers dividing ab. Then $\mathfrak{a}$ has a $\mathcal{P}$–normal presentation.

If it can be assumed that the two factors are both in normal presentation of the same set of prime ideals, then the multiplication is incomparably fast. A 'heuristic' algorithm to compute the normal presentation is given in [PZ93, p. 405]

Bernstein's ideas

At this point, it is worth mentioning another interesting approach, the "lazy localization" , presented in [Ber96], which requires further investigation. In the form presented the algorithms are designed for equation orders of algebraic number fields. Usually the maximal order is not an equation order. The naïve approach to calculate with non–equation orders involves switching from ideals presented in a equation order to ideals presented in the maximal order. This is time–consuming and would destroy the benefits of the method.

1.2.2 Inverse ideals

Theorem (5.6) in [PZ93, p. 269] implies the equivalence of the existence of inverse ideals for every fractional ideal (whose definition does not include the zero ideal) and the Dedekind property of a ring. Unlike addition and multiplication, we *must* insist on the order $\mathcal{O}$ being the maximal order $\mathcal{O} = o_{\mathcal{K}}$ since the maximal order is the only order which is a Dedekind ring, as proposed in [Coh95, p. 184].

Algorithmic idea 1.2.9: Inversion with the multiplicator ring
This algorithm is relatively new and is only published in [Fri97, pp. 93–98]. It is described there in the context of relative ideals, but it is valid for absolute ideals as well. The algorithm is now used in KANT since it appears to be the most efficient one.

Algorithmic idea 1.2.10: Inversion with the different
An efficient algorithm which uses the different of the algebraic number field is given in [Coh95, pp. 202–204]. The relevant statements are cited below:

Definition 1.2.11:
The **different** $\mathfrak{d}(\mathcal{K})$ of an algebraic number field $\mathcal{K}$ with the maximal order $o_{\mathcal{K}}$ is the integral $o_{\mathcal{K}}$–ideal

$$\left\{\alpha \in \mathcal{K} \mid \mathrm{Tr}_{K/\mathbb{Q}}(\alpha o_{\mathcal{K}}) \subset \mathbb{Z}\right\}^{-1} \subset o_{\mathcal{K}},$$

where $\mathrm{Tr}_{K/\mathbb{Q}}$ denotes the trace of the representation matrix of an algebraic number in $\mathcal{K}$ over $\mathbb{Q}$.

Proposition 1.2.12:
Let $(\omega_i)_{i\in\mathbb{N}_n}$ be an integral basis of the algebraic number field $\mathcal{K}$ with the maximal order $o_{\mathcal{K}}$ and the different $\mathfrak{d}$. Let the nonzero integral $\mathcal{D}$–ideal $\mathfrak{a}$ be given in basis presentation as $\mathbf{A} \in o_{\mathcal{K}}^{n\times n}$ as in (1.1.2). Let $\mathbf{T} = \left(\mathrm{Tr}_{K/\mathbb{Q}}(\omega_i\omega_j)\right)_{i,j\in\mathbb{N}_n}$. Then the columns of the matrix $(\mathbf{A}^t\mathbf{T})^{-1}$ form a $\mathbb{Z}$–basis of the ideal $\mathfrak{a}^{-1}\mathfrak{d}^{-1}$.

The algorithm to compute the different is given in [Coh95, p. 204].

Algorithmic idea 1.2.13: Ideal inversion with normal presentations

Another method uses the determination of the normal presentation. The following is a consequence of a theorem in [PZ93, p. 406]:

Theorem 1.2.14:

Let $\mathcal{P}$ be a set of prime numbers. Let $\mathfrak{a}$ be an integral $o_{\mathcal{K}}$–ideal with the $\mathcal{P}$–normal presentation (a, α). Then there exists $d \in \mathbb{N}$ such that $(1, d\alpha^{-1})$ is a $\mathcal{P}$–normal presentation of $\mathfrak{a}^{-1}$.

1.3 Minimum and norm of an ideal

The rest of this chapter will include different generalization levels. This should be clarified by the use of different symbols for the ring used: $\mathcal{D}$ refers to an integral domain or a DEDEKIND ring, $\mathcal{O}$ to any order, and $o_{\mathcal{K}}$ to the maximal order in an algebraic number field $\mathcal{K}$.

Definition 1.3.1:

Let $\mathcal{D}_c \subset \mathcal{D}$ be two integral domains, $\mathcal{K}_c$ and $\mathcal{K}$ the quotient fields of $\mathcal{D}_c$ and $\mathcal{D}$, respectively, with the properties

$$\mathcal{K}_c \cap \mathcal{D} = \mathcal{D}_c \tag{1.3.1}$$

$$\forall \delta_1 \in \mathcal{D}\ \exists \delta_2 \in \mathcal{D}, d \in \mathcal{D}_c : d = \delta_1 \delta_2. \tag{1.3.2}$$

Let $\mathfrak{a}$ be a fractional $\mathcal{D}$–ideal. Then the $\mathcal{D}_c$**–minimum ideal** of $\mathfrak{a}$ is the fractional $\mathcal{D}_c$–ideal $\mathfrak{a} \cap \mathcal{K}_c$.

The minimum of the zero ideal in $\mathcal{D}$ is the zero ideal in $\mathcal{D}_c$.

Proof that the $\mathcal{D}_c$–minimum ideal is indeed a $\mathcal{D}_c$–ideal.

Firstly, let $\mathfrak{a}$ be a nonzero integral $\mathcal{D}$–ideal. $\mathfrak{a} \cap \mathcal{K}_c$ is obviously a commutative ring since $\mathfrak{a}$ and $\mathcal{K}_c$ are. Because of $\mathcal{D}_c \subset \mathcal{K}_c$, we have $\mathcal{D}_c(\mathfrak{a} \cap \mathcal{K}_c) \subset \mathcal{K}_c$. $\mathfrak{a}$ being an ideal implies $\mathcal{D}\mathfrak{a} \subset \mathfrak{a}$, and therefore $\mathcal{D}_c\mathfrak{a} \subset \mathfrak{a}$ and $\mathcal{D}_c(\mathfrak{a} \cap \mathcal{K}_c) \subset \mathfrak{a}$. We thus get $\mathcal{D}_c(\mathfrak{a} \cap \mathcal{K}_c) \subset \mathfrak{a} \cap \mathcal{K}_c$. From property (1.3.1) and $\mathfrak{a} \subset \mathcal{D}$, we conclude that $\mathfrak{a} \cap \mathcal{K}_c \subset \mathcal{D}_c$. Therefore $\mathfrak{a} \cap \mathcal{K}_c$ is an integral $\mathcal{D}_c$–ideal.

Secondly, let $\mathfrak{a}$ be a fractional $\mathcal{D}$–ideal. By definition, there exists a $\delta \in \mathcal{D}$ such that $\delta\mathfrak{a}$ is an integral $\mathcal{D}$–ideal. Property (1.3.2) yields a $d \in \mathcal{D}_c$ such that $d\mathfrak{a}$ is an integral $\mathcal{D}$–ideal. It follows that $d\mathfrak{a} \cap \mathcal{K}_c = d(\mathfrak{a} \cap \mathcal{K}_c)$ is an integral $\mathcal{D}_c$–ideal. Hence $\mathfrak{a} \cap \mathcal{K}_c$ is a fractional $\mathcal{D}_c$–ideal, which completes the proof. □

The definition of the minimum ideal covers

- the notion of the minimum of a nonzero integral ideal in an order $\mathcal{D} = \mathcal{O}$ of an algebraic number field over $\mathbb{Q}$, which is introduced in [PZ93, p. 398]. The minimum of the integral $\mathcal{O}$–ideal $\mathfrak{a}$ is the natural number $\min(\mathfrak{a} \cap \mathbb{N})$. This is a special case of the general definition where $\mathcal{D}_c = \mathbb{Z}$ and $\mathcal{K}_c = \mathbb{Q}$. The minimum ideal of $\mathfrak{a}$ is an integral $\mathbb{Z}$–ideal and therefore a principal ideal generated by a natural number, which is the minimum.
- the generalization to fractional ideals, which is used in KANT. Again let $\mathcal{D} = \mathcal{O}$ be an order of the algebraic number field over $\mathbb{Q}$. Let $\frac{\mathfrak{a}}{d}$ be a fractional ideal, where $\mathfrak{a}$ is a nonzero integral $\mathcal{O}$–ideal and $d \in \mathbb{N}$. The minimum of $\frac{\mathfrak{a}}{d}$ is the minimum of the set $\mathfrak{a} \cap \mathbb{N}$ divided by d.

- the minimum ideal of a relative ideal, where $\mathcal{D}_c$ is the maximal order of an algebraic number field $\mathcal{K}_c$ over $\mathbb{Q}$, $\mathcal{K}$ an algebraic field extension of $\mathcal{K}_c$, and $\mathcal{D}$ an order of $\mathcal{K}$. The minimum ideal of a relative ideal is described in section 5.4.

Algorithmic idea 1.3.2: Computing the minimum of an integral ideal

Let $\mathcal{D} = \mathcal{O}$ be an order of the algebraic number field $\mathcal{K}$ and $\mathfrak{a}$ be a nonzero integral $\mathcal{O}$–ideal. We want to compute the minimum of $\mathfrak{a}$ (which is the positive generator of the $\mathbb{Z}$–minimum ideal).

Let $\mathfrak{a}$ be given in basis presentation as in (1.1.2): with a $\mathbb{Z}$–basis $(\xi_1, \ldots, \xi_n)$ which corresponds to a matrix $\boldsymbol{A}$ in upper triangular HERMITE normal form.

Since $(\xi_1, \ldots, \xi_n)$ is also a $\mathbb{Q}$–vector space basis for $\mathcal{K}$, we can find $b_i \in \mathbb{Q}$ with $i \in \mathbb{N}_n$ such that

$$1 = \sum_{i=1}^{n} b_i \xi_i.$$

Let $c \in \mathfrak{a} \cap \mathbb{Q}$. Then $c = \sum_{i=1}^{n} c b_i \xi_i$. Since $(\xi_1, \ldots, \xi_n)$ is a basis of the ideal $\mathfrak{a}$, this is equivalent to $\forall i \in \mathbb{N}_n \; cb_i \in \mathbb{Z}$. We conclude that

$$\mathfrak{a} \cap \mathbb{Q} = \bigcap_{i=1}^{n} \frac{1}{b_i}\mathbb{Z}. \tag{1.3.3}$$

Let the basis matrix of $\mathfrak{a}$ and its inverse be written as

$$\boldsymbol{A} = \left(a_{ij}\right)_{i,j \in \mathbb{N}_n}, \quad \boldsymbol{A}^{-1} = \left(\bar{a}_{ij}\right)_{i,j \in \mathbb{N}_n}.$$

If we can find $c_i \in \mathcal{K}$, where $i \in \mathbb{N}_n$, such that

$$\sum_{i=1}^{n} c_i \omega_i = 1$$

then we have

$$\sum_{j=1}^{n} \xi_j \sum_{i=1}^{n} c_i \bar{a}_{ji} = 1$$

with

$$b_j = \sum_{i=1}^{n} c_i \bar{a}_{ji} \in \mathcal{K} \quad \text{where} \quad \sum_{i=1}^{n} b_i \xi_i = 1.$$

Therefore finding a representation of the multiplicative identity can be split up in two subtasks; finding a representation of the multiplicative identity in the basis of $\mathcal{O}$ and the inversion of the representation matrix of $\mathfrak{a}$.

If the basis Ω has the property $\omega_1 = 1$, formula (1.3.3) for the minimum ideal simplifies to

$$\mathfrak{a} \cap \mathbb{Q} = \frac{1}{\bar{a}_{11}}\mathbb{Z}.$$

Since $\boldsymbol{A}$ was assumed to be in upper triangular HNF, the entry a_{11} of $\boldsymbol{A}$ satisfies $a_{11} = \bar{a}_{11}^{-1}$. Therefore the minimum ideal of $\mathfrak{a}$ is $a_{11}\mathbb{Z}$. The minimum (as a natural number) is then simply a_{11}.

Definition 1.3.3:
Let $\mathcal{D}$ be a DEDEKIND ring. The **norm of a fractional $\mathcal{D}$–ideal $\frac{\mathfrak{a}}{d}$**, where $\mathfrak{a}$ is an integral $\mathcal{D}$–ideal, and $d \in \mathbb{N}$ is the cardinality of the ring $\mathcal{D}/\mathfrak{a}$ divided by d^n.

(This definition is equivalent to the one given as a generalization to fractional ideals in [Coh95, p. 185].)

Algorithmic idea 1.3.4: Computing the norm of an integral ideal
If $\mathcal{D} = o_{\mathcal{K}}$ is a maximal order of an algebraic number field $\mathcal{K}$, the norm of an ideal is the determinant of the basis matrix $\mathbf{A}$ (see (1.1.2)) which is the product of the diagonal entries in case $\mathbf{A}$ is in HNF.

Definition 1.3.5:
Let $\mathfrak{a}$ be an integral ideal. The **maximal natural factor of** $\mathfrak{a}$ is the maximum of all $m \in \mathbb{N}$ such that $\frac{\mathfrak{a}}{m}$ is an integral ideal.

Algorithmic idea 1.3.6: Computing the maximal natural factor of an integral ideal
If $\mathcal{D} = \mathcal{O}$ is an order of an algebraic number field $\mathcal{K}$, the maximal natural factor can be determined by computing the minimum of the ideal and finding its prime factorization. Starting with $m = 1$, for each factor p of this prime factorization it is checked whether $\frac{\mathfrak{a}}{pm}$ is an integral ideal.

1.4 Modular HNF computations for addition and multiplication of ideals over algebraic number fields

This is a modification applicable to all algorithms for ideals over an order $\mathcal{D} = \mathcal{O}$ of an algebraic number field $\mathcal{K}$ which apply an HNF computation to obtain a basis of the resulting ideal, notably addition and multiplication.

Algorithmic idea 1.4.1: Modular ideal additions and multiplications
Let $\mathfrak{a}$ be a nonzero integral ideal and m its minimum as a natural number. Then $m\mathcal{O} \subset \mathfrak{a}$. Let $\xi_1, \ldots, \xi_m$ be a $\mathbb{Z}$–generating set of $\mathfrak{a}$, given as a matrix $(m_{ij})_{i\in\mathbb{N}_n, j\in\mathbb{N}_m}$, where $\xi_j = \sum_{i=1}^{n} m_{ij}\omega_i$. To transform the generating set to a basis in HNF we can use the number m for modular HNF computations.

Note that this is much better than using the gcd of rank minors of a (not HNF) matrix representing the sum and product, respectively, of the two ideals. The rank minor gcd is usually much larger because it does not profit from the fact that the matrices represent ideals.

The following proposition allows an integral multiple of the minimum of the sum and the product of two ideals to be computed.

Proposition 1.4.2:
Let $\mathfrak{a}$ and $\mathfrak{b}$ be nonzero integral $\mathcal{O}$–ideals. Then

$$\min(\mathfrak{a}+\mathfrak{b}) \mid \gcd\big(\min(\mathfrak{a}), \min(\mathfrak{b})\big) \quad \text{and} \quad \min(\mathfrak{a}\mathfrak{b}) \mid \min(\mathfrak{a})\min(\mathfrak{b}).$$

Proof.

$$\begin{aligned} \mathfrak{a}+\mathfrak{b} \supset \mathfrak{a} &\implies (\mathfrak{a}+\mathfrak{b}) \cap \mathbb{N} \supset \mathfrak{a} \cap \mathbb{N} \\ &\implies (\mathfrak{a}+\mathfrak{b}) \cap \mathbb{N} \supset (\mathfrak{a} \cap \mathbb{N}) + (\mathfrak{b} \cap \mathbb{N}) \end{aligned}$$

proves the first statement.

Let $c \in (\mathfrak{a} \cap \mathbb{N})(\mathfrak{b} \cap \mathbb{N})$. Then $\exists a \in \mathfrak{a} \cap \mathbb{N}$, $\exists b \in \mathfrak{b} \cap \mathbb{N}$ such that $c = ab$. Obviously $c \in \mathbb{N}$ and $c \in \mathfrak{ab}$, which proves the second statement. □

1.5 Primitive elements

Definition 1.5.1:

Let $\mathcal{D}$ be an integral domain. Let $\mathfrak{a}$ be a nontrivial integral ideal. The element $a \in \mathcal{D}$ is called a **primitive element of $\mathfrak{a}$** iff $a \in \mathfrak{a} \setminus \mathfrak{a}^2$.

Proposition 1.5.2:

For any invertible nontrivial integral ideal over an integral domain $\mathcal{D}$, there exists a primitive element.

Proof. $\mathfrak{a}$ is integral or equivalently $\mathfrak{a} \subset \mathcal{D}$. $\mathfrak{a}$ is not trivial; therefore $\mathfrak{a} \neq \mathcal{D}$.

Assume $\mathfrak{a} = \mathfrak{a}^2$. Since $\mathfrak{a}$ is invertible we can multiply by $\mathfrak{a}^{-1}$ on both sides and conclude $\mathfrak{a} = \mathcal{D}$ which is a contradiction. □

Algorithmic idea 1.5.3: Computing a primitive element of an integral ideal

If $\mathcal{D}$ is a DEDEKIND ring, every fractional $\mathcal{D}$–ideal is invertible. Therefore for any nontrivial integral ideal, there exists a primitive element.

In maximal orders over algebraic number fields, the following algorithmic ideas can be used to compute the primitive element of an integral ideal. If an ideal is given in two–element presentation, it is easy to see that one of the two generators of the ideal must be primitive. (Assume this is not the case. Then both generators are in $\mathfrak{a}^2$ and the whole of $\mathfrak{a}$ is in $\mathfrak{a}^2$, which is a contradiction to the fact that $\mathfrak{a}$ is integral and not trivial.)

The algorithm therefore involves the computation of $\mathfrak{a}^2$ (which is very easy) and two checks on membership of an algebraic number in an ideal. For this we have to transform the two–element presentation into an HNF–basis presentation (which involves the determination of the representation matrices of two algebraic numbers and an HNF computation of a $2n \times n$ matrix). The part with the biggest complexity is the HNF computation.

If the ideal is given in HNF–basis presentation, by the same argumentation as above it is clear that at least one of the basis elements must be primitive. So we simply choose the basis element which is not an element of $\mathfrak{a}^2$. Here the computation of $\mathfrak{a}^2$, which involves an HNF computation of an $n^2 \times n$ matrix, is the most difficult part.

The question is: Given the basis presentation of $\mathfrak{a}$, is it worth determining a two–element presentation and using the former method? Experiments show that this is almost always the case.

1.6 Idempotents for coprime ideals

Proposition 1.6.1:

Let $\mathcal{D}$ be an integral domain. Let $\mathcal{S}$ be a finite set of coprime integral ideals in $\mathcal{D}$; e.g.,

$$\sum_{\mathfrak{a} \in \mathcal{S}} \mathfrak{a} = 1\mathcal{D},$$

where $1\mathcal{D}$ denotes the trivial integral $\mathcal{D}$–ideal generated by 1.

Then, for every $\mathfrak{a} \in \mathcal{S}$, there exists $a_{\mathfrak{a}} \in \mathcal{D}$ satisfying

$$\sum_{\mathfrak{a} \in \mathcal{S}} a_{\mathfrak{a}} = 1.$$

The proof is trivial from the definition of the sum of ideals.

Let $\mathcal{D} = \mathcal{O}$ be an order of an algebraic number field $\mathcal{K}$ given with a $\mathbb{Z}$–basis $\Omega = (\omega_1, \dots, \omega_n)$ with the property $\omega_1 = 1$. The following algorithm constructs the idempotent elements:

Algorithm 1.6.2: Idempotents for integral ideals

```
Input: A set of $m$ coprime $\mathcal{O}$–ideals given in HNF–basis presentation matrices $\mathsf{A}_i$ on
    some $\mathbb{Z}$–basis $\Omega = (\omega_1, \dots, \omega_n)$ of $\mathcal{O}$ with the property $\omega_1 = 1$.
Output: Algebraic numbers $a_{\mathfrak{a}} \in \mathfrak{a}$ ($\mathfrak{a} \in \mathcal{S}$) represented in the basis $\Omega$ with $\sum_{\mathfrak{a} \in \mathcal{S}} a_{\mathfrak{a}} = 1$.
Steps:
    1: Concatenate the basis matrices of the ideals: $\mathbf{A} := \big(\mathbf{A}_1 \mid \cdots \mid \mathbf{A}_m\big)$.
    2: Apply an HNF algorithm to $\mathbf{A}$ which yields a $\mathbf{T}$ such that $\mathbf{H} = \mathbf{AT}$ is a concatenation of an identity matrix with $n(m-1)$ zero columns at the end[1].
    3: Extract the first column of $\mathbf{T}$ and split it horizontally in $m$ vectors of length $n$:
          $\mathsf{T}_1, \dots, \mathsf{T}_m$.
    4: Compute the vectors $\mathsf{U}_i := \mathbf{A}_i \mathsf{T}_i$ for $i = 1, \dots, m$. They represent elements $a_i$
          of $\mathcal{K}$ regarding the $\mathbb{Q}$–basis of $\mathcal{K}$. Because $\mathbf{T}$ is a matrix over $\mathbb{Z}$, the $a_i$
          satisfy $a_i \in \mathfrak{a}_i, i = 1, \dots, m$, and $\sum_{i=1}^{m} a_i = 1$.
    5: End.
```

Proof. It is important here that the first element of the integral basis Ω of $\mathcal{K}$ equals 1. Because of this the first canonical vector

$$\mathsf{E}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

represents indeed the 1 in $\mathcal{K}$.

We have

$$\mathbf{H} = \left(\begin{array}{c|c|c|c} \begin{matrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{matrix} & 0 & \dots & 0 \end{array}\right) = \mathbf{AT} = \big(\mathbf{A}_1 \mid \cdots \mid \mathbf{A}_m\big)\mathbf{T}.$$

The first column of $\mathbf{H}$ is E_1, so the first column B of $\mathbf{T}$ satisfies $\mathbf{A}\mathrm{B} = \mathsf{E}_1$.

$$\mathsf{E}_1 = \big(\mathbf{A}_1 \mid \cdots \mid \mathbf{A}_m\big) \begin{pmatrix} \mathsf{T}_1 \\ \vdots \\ \mathsf{T}_m \end{pmatrix} \quad \Longrightarrow \quad \mathsf{E}_1 = \sum_{i=1}^{m} \mathbf{A}_i \mathsf{T}_i$$

[1] $\mathbf{H}$ is the identity matrix since the ideals are assumed to be coprime. Moreover, this is the check if the ideals are indeed coprime.

For $i \in \mathbb{N}_m$, the vector T_i represents an element of the ideal $\mathfrak{a}_i$ because $\mathbf{A}_i$ is a basis of $\mathfrak{a}_i$. The vectors $\mathbf{A}_i\mathsf{T}_i$ represent elements a_i of $\mathcal{O}$ such that $a_i \in \mathfrak{a}_i$. Moreover the sum of the a_i is 1. So this proves the validity of the algorithm. □

The complexity of this algorithm is determined by the complexity of the HNF computation.

1.7 Approximation theorem

To formulate the approximation theorem, we need the following notion of a special class of valuations (one valuation for every prime ideal $\mathfrak{p}$ of a DEDEKIND ring), the $\mathfrak{p}$–adic valuation in [Coh95, p. 184].

Definition 1.7.1:

Let $\mathcal{D}$ be a DEDEKIND ring and $\mathfrak{p}$ a nonzero prime $\mathcal{D}$–ideal.

The $\mathfrak{p}$–**adic valuation** is the map

$$v_{\mathfrak{p}} : \mathrm{I}_{\mathcal{K}} \cup \{0\mathcal{D}\}^{\dagger} \to \mathbb{Z} \cup \{\infty\}$$

which satisfies

$$\begin{aligned}&\forall \mathfrak{a} \in \mathrm{I}_{\mathcal{K}} : \mathfrak{a} \subset \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{a})}, \mathfrak{a} \not\subset \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{a})+1} \quad \text{and}\\ &v_{\mathfrak{p}}(0\mathcal{D}) = \infty.\end{aligned}$$

The $\mathfrak{p}$–adic valuation is defined for elements of $\mathcal{K}$ as:

$$\begin{aligned}v_{\mathfrak{p}} :& \mathcal{K} \to \mathbb{Z} \cup \{\infty\}\\ &\alpha \mapsto v_{\mathfrak{p}}(\alpha\mathcal{D}).\end{aligned} \tag{1.7.1}$$

Proposition 1.7.2:

Let $\mathcal{D}$ be a DEDEKIND ring. Let $\mathcal{P}$ be the set of all nonzero prime ideals in $\mathcal{D}$ and $\mathcal{S}$ be a finite subset of $\mathcal{P}$. Let $(e_{\mathfrak{p}})_{\mathfrak{p}\in\mathcal{S}} \in \mathbb{Z}^{\mathcal{S}}$ be integral exponents. Then there exists an $a \in \mathcal{K}$ such that

$$v_{\mathfrak{p}}(a) \begin{cases} = e_{\mathfrak{p}} & \text{if} \quad \mathfrak{p} \in \mathcal{S} \\ \geq 0 & \text{if} \quad \mathfrak{p} \in \mathcal{P} \setminus \mathcal{S}. \end{cases}$$

a is called the approximation of the fractional $\mathcal{D}$–ideal $\mathfrak{a} = \prod_{\mathfrak{p}\in\mathcal{S}} \mathfrak{p}^{e_{\mathfrak{p}}}$ which satisfies

$$v_{\mathfrak{p}}(\mathfrak{a}) = \begin{cases} e_{\mathfrak{p}} & \text{if} \quad \mathfrak{p} \in \mathcal{S} \\ 0 & \text{if} \quad \mathfrak{p} \in \mathcal{P} \setminus \mathcal{S}. \end{cases}$$

Both the proof for DEDEKIND rings and the algorithm for maximal orders of algebraic number fields split naturally in two parts: to solve the problem for nonnegative exponents and to base the general problem on the solution for positive exponents.

†. $0\mathcal{D}$ denotes the zero ideal (the ideal only containing zero), and $\mathrm{I}_{\mathcal{K}}$ denotes the group of fractional $\mathcal{D}$–ideals, which excludes the zero ideal

1.7.1 Approximation for nonnegative exponents

The approximation theorem for nonnegative exponents can be formulated in the more general context of integral domains with the following modification of the above notion of $v_{\mathfrak{p}}$:

Definition 1.7.3:

Let $\mathcal{D}$ be an integral domain and $\mathfrak{p}$ an invertible prime $\mathcal{D}$–ideal.

The $\mathfrak{p}$**–adic valuation** is the map

$$v_{\mathfrak{p}} : \mathrm{I}_{\mathcal{D}} \to \mathbb{Z}^{\geq 0} \cup \{\infty\}$$

with $v_{\mathfrak{p}}(0\mathcal{D}) = \infty$[‡]. If $\mathfrak{a} \in \mathrm{I}_{\mathcal{D}} \setminus \{0\mathcal{D}\}$, then $v_{\mathfrak{p}}(\mathfrak{a})$ is the integer which satisfies

$$\mathfrak{a} \subset \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{a})} \quad \text{and} \quad \mathfrak{a} \not\subset \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{a})+1}.$$

Since $\mathfrak{p}$ was assumed to be invertible, we have for all $v \in \mathbb{Z}^{\geq 0}$, $\mathfrak{p}^{v+1} \subset \mathfrak{p}^{v}$. A nonzero integral $\mathcal{D}$–ideal $\mathfrak{a}$ satisfies $\mathfrak{a} \subset \mathcal{D} = \mathfrak{p}^0$. Therefore we defined a proper map $v_{\mathfrak{p}}$ for every invertible prime ideal $\mathfrak{p}$. With the properties given in [Coh95, p. 184] we see that the map $v_{\mathfrak{p}}$ is an exponential valuation of $\mathrm{I}_{\mathcal{D}}$ in the sense of [PZ93, p. 248].

This definition is equivalent to the definition 1.7.1 if $\mathcal{D}$ is a DEDEKIND ring. Again the $\mathfrak{p}$–adic valuation is defined for elements of $\mathcal{D}$ according to formula (1.7.1).

Proposition 1.7.4:

Let $\mathcal{D}$ be a DEDEKIND domain. Let $\mathcal{P}$ be the set of all invertible prime ideals in $\mathcal{D}$ and $\mathcal{S}$ be a finite subset of $\mathcal{P}$. Let $(e_{\mathfrak{p}})_{\mathfrak{p}\in\mathcal{S}} \in (\mathbb{Z}^{\geq 0})^{\mathcal{S}}$. Then there exists an $a \in \mathcal{K}$ such that

$$v_{\mathfrak{p}}(a) \begin{cases} = e_{\mathfrak{p}} & \text{if} \quad \mathfrak{p} \in \mathcal{S} \\ \geq 0 & \text{if} \quad \mathfrak{p} \in \mathcal{P} \setminus \mathcal{S}. \end{cases}$$

Proof. The idea is to consider for each $\mathfrak{p} \in \mathcal{S}$ the ideal product

$$\mathfrak{a}_{\mathfrak{p}} = \prod_{\mathfrak{q}\in\mathcal{S}\setminus\{\mathfrak{p}\}} \mathfrak{q}^{e_{\mathfrak{q}}+1}.$$

Then the $\mathfrak{a}_{\mathfrak{p}}$ are nonzero integral ideals which sum to $1\mathcal{D}$. By proposition 1.6.1, there exist $a_{\mathfrak{p}} \in \mathfrak{a}_{\mathfrak{p}}$ whose sum is equal to 1.

By proposition 1.5.2, and since all $\mathfrak{p} \in \mathcal{P}$ are assumed to be invertible, $\mathfrak{p}$ contains a primitive element b. We set $b_{\mathfrak{p}} = b^{e_{\mathfrak{p}}} \in \mathfrak{p}^{e_{\mathfrak{p}}} \setminus \mathfrak{p}^{e_{\mathfrak{p}}+1}$.

For all nonzero integral ideals $\mathfrak{a}$ and $\mathfrak{b}$ we have $v_{\mathfrak{p}}(\mathfrak{a}\mathfrak{b}) = v_{\mathfrak{p}}(\mathfrak{a}) + v_{\mathfrak{p}}(\mathfrak{b})$. Moreover $v_{\mathfrak{p}}(b_{\mathfrak{p}}) = 1$ yields

$$v_{\mathfrak{p}}(b_{\mathfrak{p}}^{e_{\mathfrak{p}}}) = e_{\mathfrak{p}};$$

hence, the element

$$a = \sum_{\mathfrak{p}\in\mathcal{S}} a_{\mathfrak{p}} b_{\mathfrak{p}}$$

[‡]. $\mathrm{I}_{\mathcal{D}}$ denotes the $\mathcal{D}$–module of integral $\mathcal{D}$–ideals including the zero ideal

satisfies

$$v_{\mathfrak{p}}(a)\begin{cases}= e_{\mathfrak{p}} & \text{if } \mathfrak{p}\in\mathcal{S}\\ \geq 0 & \text{if } \mathfrak{p}\in\mathcal{P}\setminus\mathcal{S}.\end{cases}$$

□

Algorithm 1.7.5: Non–negative approximation

```
Input: $(e_{\mathfrak{p}})_{\mathfrak{p}\in\mathcal{S}} \in (\mathbb{Z}^{\geq 0})^{\mathcal{S}}$.
Output: $a \in \mathcal{K}$ such that $v_{\mathfrak{p}}(a) = e_{\mathfrak{p}}$ for $\mathfrak{p}\in\mathcal{S}$ and $v_{\mathfrak{p}}(a) \geq 0$ for $\mathfrak{p}\in\mathcal{P}\setminus\mathcal{S}$.
Steps:
    1: Set $\mathfrak{a}_{\mathfrak{p}} := \prod_{\mathfrak{q}\in\mathcal{S}\setminus\{\mathfrak{p}\}} \mathfrak{q}^{e_{\mathfrak{q}}+1}$ for all $\mathfrak{p}\in\mathcal{S}$ [2].
    2: Apply algorithm 1.6.2 to $\mathfrak{a}_{\mathfrak{p}}, \mathfrak{p}\in\mathcal{S}$, obtain $a_{\mathfrak{p}}, \mathfrak{p}\in\mathcal{S}$ with $\prod_{\mathfrak{p}\in\mathcal{S}} a_{\mathfrak{p}} = 1$.
    3: Find primitive elements $c_{\mathfrak{p}}, \mathfrak{p}\in\mathcal{S}$, for the ideals $\mathfrak{a}_{\mathfrak{p}}, \mathfrak{p}\in\mathcal{S}$, as described in 1.5.2 .
    4: Set $b_{\mathfrak{p}} := c_{\mathfrak{p}}^{e_{\mathfrak{p}}}, \mathfrak{p}\in\mathcal{S}$.
    5: Set $a := \sum_{\mathfrak{p}\in\mathcal{S}} a_{\mathfrak{p}} b_{\mathfrak{p}}$.
    6: End.
```

The complexity of this algorithm is determined by the complexity of finding the idempotents (see section 1.6).

1.7.2 Simple assembling

Let $\mathcal{D}$ be a DEDEKIND ring. Let $\mathcal{S}$ be a finite set of prime ideals of $\mathcal{D}$, and let $(e_{\mathfrak{p}})_{\mathfrak{p}\in\mathcal{S}} \in \mathbb{Z}^{\mathcal{S}}$. The following algorithm finds an $a\in\mathcal{K}$ such that $v_{\mathfrak{p}}(a) = e_{\mathfrak{p}}$ for $\mathfrak{p}\in\mathcal{S}$. However, it does not guarantee that $v_{\mathfrak{p}}(a)\geq 0$ for $\mathfrak{p}\notin\mathcal{S}$. The idea of the algorithm is to split the positive and negative values of e_p, to compute a separate a_{pos} and a_{neg} for the positive and negative values, respectively, and to divide a_{pos} by a_{neg}.

Algorithm 1.7.6: Simple assembling

```
Input: $(e_{\mathfrak{p}})_{\mathfrak{p}\in\mathcal{S}} \in \mathbb{Z}^{\mathcal{S}}$.
Output: $a \in \mathcal{K}$ such that $v_{\mathfrak{p}}(a) = e_{\mathfrak{p}}$ for $\mathfrak{p}\in\mathcal{S}$.
Steps:
    1: Set $\mathcal{S}_{\mathsf{pos}} := \{\mathfrak{p}\in\mathcal{S} | e_{\mathfrak{p}} \geq 0\}$
           $\mathcal{S}_{\mathsf{neg}} := \mathcal{S}\setminus\mathcal{S}_{\mathsf{pos}}$.
    2: Set $f_{\mathfrak{p}} := \begin{cases} e_{\mathfrak{p}} & \text{if } \mathfrak{p}\in\mathcal{S}_{\mathsf{pos}} \\ 0 & \text{if } \mathfrak{p}\in\mathcal{S}_{\mathsf{neg}}. \end{cases}$
    3: Set $g_{\mathfrak{p}} := \begin{cases} 0 & \text{if } \mathfrak{p}\in\mathcal{S}_{\mathsf{pos}} \\ -e_{\mathfrak{p}} & \text{if } \mathfrak{p}\in\mathcal{S}_{\mathsf{neg}}. \end{cases}$
    4: Apply algorithm 1.7.5 to $\mathcal{S}$ together with $f_{\mathfrak{p}}, \mathfrak{p}\in\mathcal{S}$ to obtain $a_{\mathsf{pos}}$ and to $\mathcal{S}$
           together with $g_{\mathfrak{p}}, \mathfrak{p}\in\mathcal{S}$ to obtain $a_{\mathsf{neg}}$.
    5: Set $a := a_{\mathsf{pos}}/a_{\mathsf{neg}}$.
    6: End.
```

[2] If $\mathcal{D}$ is a DEDEKIND ring, then we can use ideal inversions to save computation time: set $\mathfrak{a} := \prod_{\mathfrak{q}\in\mathcal{S}} \mathfrak{q}^{e_{\mathfrak{q}}+1}$ and $\mathfrak{a}_{\mathfrak{p}} := \mathfrak{a}\mathfrak{p}^{-e_{\mathfrak{p}}-1}$ for all $\mathfrak{p}\in\mathcal{S}$

It is clear that if there is any $e_{\mathfrak{p}} < 0$, we can find a prime ideal $\mathfrak{p} \in \mathcal{P} \setminus \mathcal{S}$ with $v_{\mathfrak{p}}(a_{\text{neg}}) > v_{\mathfrak{p}}(a_{\text{pos}})$. This $\mathfrak{p}$ satisfies $v_{\mathfrak{p}}(a) < 0$.

1.7.3 Corrected assembling

Let $\mathcal{D}$ be a Dedekind ring. On the basis of algorithm 1.7.6, it is possible to modify the result a to guarantee $v_{\mathfrak{p}}(a) \geq 0$ for all prime ideals $\mathfrak{p} \notin \mathcal{S}$.

This is done by multiplying a number $c \in \mathbb{N} \subset \mathcal{D}$ to a with the property that

$$v_{\mathfrak{p}}(c) \begin{cases} = 0 & \mathfrak{p} \in \mathcal{S} \\ \geq -v_{\mathfrak{p}}(a) & \mathfrak{p} \in \mathcal{P} \setminus \mathcal{S}. \end{cases}$$

Definition 1.7.7:
Let $\mathcal{D}$ be a Dedekind ring. The set $\mathcal{S}$ of prime $\mathcal{D}$–ideals is called **complete** iff for every prime number p such that there is a prime ideal $\mathfrak{p} \in \mathcal{S}$ over p, it follows that every prime ideal over p is in $\mathcal{S}$.

Algorithm 1.7.8: Correction of the simple assembling

```
Input: $(e_{\mathfrak{p}})_{\mathfrak{p}\in\mathcal{S}} \in \mathbb{Z}^{\mathcal{S}}$, and $\mathcal{S}$ forms a complete set of prime ideals.
Output: $a \in \mathcal{K}$ such that $v_{\mathfrak{p}}(a) = e_{\mathfrak{p}}$ for $\mathfrak{p} \in \mathcal{S}$ and $v_{\mathfrak{p}}(a) \geq 0$ for $\mathfrak{p} \notin \mathcal{S}$.
Steps:
    1: Apply algorithm 1.7.6 to $\mathcal{S}$ and $(e_{\mathfrak{p}})_{\mathfrak{p}\in\mathcal{S}}$ to obtain $a_{\text{pos}}$ and $a_{\text{neg}}$.
    2: Set $c := |N(a_{\text{neg}})| \prod_{\mathfrak{p}\in\mathcal{S}_{\text{neg}}} N(\mathfrak{p})^{e_{\mathfrak{p}}}$ †.
    3: The result is $a := c a_{\text{pos}} a_{\text{neg}}^{-1}$.
    4: End.
```

Proof. First we assure that c has indeed zero valuations on the prime ideals in $\mathcal{S}$. a_{neg} can be written as $\hat{\mathfrak{a}} \prod_{\mathfrak{p}\in\mathcal{S}_{\text{neg}}} \mathfrak{p}^{-e_{\mathfrak{p}}}$, where $\hat{\mathfrak{a}}$ is an integral ideal, because of the properties of algorithm 1.7.5. Moreover $v_{\mathfrak{p}}(\hat{\mathfrak{a}}) = 0$ for $\mathfrak{p} \in \mathcal{S}$.

Let $\mathfrak{p} \in \mathcal{S}$, p be the prime number such that $p\mathcal{D} \subset \mathfrak{p}$. Then the norm of a prime ideal $\mathfrak{p}$ is always a power of p. It follows $v_{\mathfrak{p}}(N(\hat{\mathfrak{a}})) = 0$ for $\mathfrak{p} \in \mathcal{S}$ since $\mathcal{S}$ is a complete set of prime ideals. By construction, we have

$$c = \frac{N(a_{\text{neg}})}{\prod_{\mathfrak{p}\in\mathcal{S}_{\text{neg}}} N(\mathfrak{p})^{-e_{\mathfrak{p}}}} = N\left(\frac{a_{\text{neg}}}{\prod_{\mathfrak{p}\in\mathcal{S}_{\text{neg}}} \mathfrak{p}^{-e_{\mathfrak{p}}}}\right) = N(\hat{\mathfrak{a}}).$$

(See [PZ93, p. 381] for properties of the norm of an ideal in a Dedekind ring.)

Let $\mathfrak{q} \in \mathcal{P} \setminus \mathcal{S}$. Then

$$v_{\mathfrak{q}}\big(\prod_{\mathfrak{p}\in\mathcal{S}_{\text{neg}}} \mathfrak{p}^{e_{\mathfrak{p}}}\big) = 0$$

since $\mathfrak{q}$ is a prime ideal. This yields

$$v_{\mathfrak{q}}\big(\prod_{\mathfrak{p}\in\mathcal{S}_{\text{neg}}} N(\mathfrak{p})^{e_{\mathfrak{p}}}\big) = 0$$

†. $N()$ denotes the norm of an ideal, see definition 1.3.3.

because $\mathcal{S}$ is a complete set of prime ideals. It follows that

$$v_{\mathfrak{q}}(c) = v_{\mathfrak{q}}(N(a_{\text{neg}})) \quad \text{and} \quad v_{\mathfrak{q}}(\frac{c}{a_{\text{neg}}}) = 0,$$

and hence

$$v_{\mathfrak{q}}(a) = v_{\mathfrak{q}}(c\frac{a_{\text{pos}}}{a_{\text{neg}}}) \geq 0.$$

This completes the proof of the validity of the algorithm. □

1.7.4 Complete algorithm

To finish the algorithm for the approximation theorem, it is necessary to extend a set of prime ideals to a complete set of prime ideals:

Algorithm 1.7.9: Approximation

```
Input: a DEDEKIND ring $\mathcal{D}$, a set $\mathcal{S}$ of prime ideals, and $(e_{\mathfrak{p}})_{\mathfrak{p}\in\mathcal{S}} \in \mathbb{Z}^{\mathcal{S}}$.
Output: $a \in \mathcal{K}$ such that $v_{\mathfrak{p}}(a) = e_{\mathfrak{p}}$ for $\mathfrak{p} \in \mathcal{S}$ and $v_{\mathfrak{p}}(a) \geq 0$ for $\mathfrak{p} \notin \mathcal{S}$.
Steps:
    1: Collect all the prime numbers that the prime ideals of $\mathcal{S}$ are over in a list $L$.
    2: Set: $\mathcal{S}_{\mathsf{comp}} := \mathcal{S}$.
    3: Loop: $p \in L$.
    4:     Check if $\mathcal{S}$ contains all prime ideals over $p$. If so go to the next loop cycle.
    5:     Factorize the $\mathcal{D}$–ideal $p\mathcal{D}$. Add every prime ideal $\mathfrak{p}$ which is not in $\mathcal{S}$ to $\mathcal{S}_{\mathsf{comp}}$
               together with $e_{\mathfrak{p}} := 0$.
    6: Apply algorithm 1.7.8 to $\mathcal{S}_{\mathsf{comp}}$ and $(e_{\mathfrak{p}})_{\mathfrak{p}\in\mathcal{S}_{\mathsf{comp}}}$ to obtain $a$, which is the result.
    7: End.
```

1.8 Existence proofs and algorithms for other problems for ideals

Proposition 1.8.1 ([Coh96, Corollary 1.8]):
Let $\mathcal{D}$ be a DEDEKIND ring. Let $\mathfrak{a}$ and $\mathfrak{b}$ be two nonzero integral $\mathcal{D}$–ideals. Then there exist

- an $\alpha \in \mathcal{K}$ such that $\alpha\mathfrak{a}$ is an integral ideal coprime to $\mathfrak{b}$, and
- another $\alpha \in \mathcal{K}$ such that $\alpha\mathfrak{a}^{-1}$ is an integral ideal coprime to $\mathfrak{b}$.

The proof can be found in the mentioned article.

Algorithm 1.8.2: Solving $\alpha\mathfrak{a} \subset 1\mathcal{D}$, $\alpha\mathfrak{a} + \mathfrak{b} = 1\mathcal{D}$ in α

```
Input: Non zero integral ideals $\mathfrak{a}$ and $\mathfrak{b}$.
Output: $\alpha \in \mathcal{K}$ such that $\alpha\mathfrak{a} \subset 1\mathcal{D}$ and $\alpha\mathfrak{a} + \mathfrak{b} = 1\mathcal{D}$.
Steps:
    1: Prime factorize $\mathfrak{b}$, let the prime ideals dividing $\mathfrak{b}$ be $\mathcal{S}$.
    2: Apply algorithm 1.7.9 to obtain $\alpha \in \mathcal{K}$ with $v_{\mathfrak{p}}(\alpha) = -v_{\mathfrak{p}}(\mathfrak{a})$ for $\mathfrak{p} \in \mathcal{S}$ and
           $v_{\mathfrak{p}}(\alpha) \geq 0$ for all prime ideals $\mathfrak{p}$ not in $\mathcal{S}$.
    3: End.
```

```
Algorithm 1.8.3: Solving $\alpha\mathfrak{a} \subset 1\mathcal{D}$, $\alpha\mathfrak{a}^{-1} + \mathfrak{b} = 1\mathcal{D}$ in $\alpha$
  Input: Non zero integral ideals $\mathfrak{a}$ and $\mathfrak{b}$.
  Output: $\alpha \in \mathcal{K}$ such that $\alpha\mathfrak{a} \subset 1\mathcal{D}$ and $\alpha\mathfrak{a} + \mathfrak{b} = 1\mathcal{D}$.
  Steps:
    1: Prime factorize $\mathfrak{a}\mathfrak{b}$, let the prime ideals dividing $\mathfrak{a}\mathfrak{b}$ be $\mathcal{S}$.
    2: Apply algorithm 1.7.5 to obtain $\alpha \in \mathcal{D}$ with $v_{\mathfrak{p}}(\alpha) = v_{\mathfrak{p}}(\mathfrak{a})$ for $\mathfrak{p} \in \mathcal{S}$ and
         $v_{\mathfrak{p}}(\alpha) \geq 0$ for all prime ideals $\mathfrak{p}$ not in $\mathcal{S}$.
    3: End.
```

The complexity is determined by the complexity of the factorization of the ideals $\mathfrak{b}$ for algorithm 1.8.2 resp. $\mathfrak{a}\mathfrak{b}$, for algorithm 1.8.3. Apart from the factorizations, the algorithms are polynomial. Algorithm 1.8.3 needs only the nonnegative approximation, which is much easier than the general approximation needed for algorithm 1.8.2.

Proposition 1.8.4 ([Coh96, Proposition 1.11]):
Let $\mathcal{D}$ be a Dedekind ring, and let $\mathfrak{a}$ and $\mathfrak{b}$ be two fractional $\mathcal{D}$–ideals. Then there exist $\mu_1 \in \mathfrak{a}, \mu_2 \in \mathfrak{b}$ and $\nu_1 \in \mathfrak{a}^{-1}, \nu_2 \in \mathfrak{b}^{-1}$ such that

$$\mu_1\nu_1 - \mu_2\nu_2 = 1.$$

The proof is immediate with the construction of the following algorithm and propositions 1.6.1 and 1.8.1.

The following algorithm computes the required elements in a maximal order of an algebraic number field.

```
Algorithm 1.8.5: Finding $\mu_1 \in \mathfrak{a}, \mu_2 \in \mathfrak{b}, \nu_1 \in \mathfrak{a}^{-1}, \nu_2 \in \mathfrak{b}^{-1}$ with $\mu_1\nu_1 - \mu_2\nu_2 = 1$
  Input: Fractional ideals $\mathfrak{a}$ and $\mathfrak{b}$.
  Output: Algebraic numbers $\mu_1 \in \mathfrak{a}, \mu_2 \in \mathfrak{b}$ and $\nu_1 \in \mathfrak{a}^{-1}, \nu_2 \in \mathfrak{b}^{-1}$ such that $\mu_1\nu_1 -$
      $\mu_2\nu_2 = 1$.
  Steps:
    1: If either $\mathfrak{a}$ or $\mathfrak{b}$ is not integral, set $d = \operatorname{lcm}(\operatorname{den}(\mathfrak{a}), \operatorname{den}(\mathfrak{b}))$. We execute the
         algorithm with the ideals $d\mathfrak{a}$ and $d\mathfrak{b}$ to obtain $\mu_1' \in d\mathfrak{a}, \mu_2' \in d\mathfrak{b}$ and
         $\nu_1' \in \frac{\mathfrak{a}^{-1}}{d}, \nu_2' \in \frac{\mathfrak{b}^{-1}}{d}$ such that $\mu_1'\nu_1' - \mu_2'\nu_2' = 1$. Return with $\mu_1 = \frac{\mu_1'}{d}$,
         $\mu_2 = \frac{\mu_2'}{d}$, $\nu_1 = d\nu_1'$, $\nu_2 = d\nu_2'$.
    2: Apply algorithm 1.8.3 to obtain $\alpha \in \mathcal{D}$ such that $\alpha\mathfrak{a}^{-1} \subset \mathcal{D}$ and $\alpha\mathfrak{a}^{-1} + \mathfrak{b} = \mathcal{D}$.
    3: Apply algorithm 1.6.2 to obtain $\gamma \in \alpha\mathfrak{a}^{-1}$ and $\beta \in \mathfrak{b}$ such that $\beta + \gamma = 1$.
    4: Return with $\mu_1 = \alpha$, $\mu_2 = \beta$, $\nu_1 = \frac{\gamma}{\alpha}$, $\nu_2 = -1$.
    5: End.
```

The complexity of this algorithm is determined by the complexity of the algorithm to compute idempotents of coprime ideals and the complexity to find elements which make ideals coprime, which was described above.

Another algorithmic idea for the case $\mathfrak{a} = \mathfrak{b}$ that uses the normal presentation of an ideal was given in [BP91] implicitly.

Algorithm 1.8.6: Finding $\mu_1, \mu_2 \in \mathfrak{a}$, $\nu_1, \nu_2 \in \mathfrak{a}^{-1}$ with $\mu_1\nu_1 - \mu_2\nu_2 = 1$

```
Input: A fractional ideal $\mathfrak{a}$.
Output: Algebraic numbers $\mu_1, \mu_2 \in \mathfrak{a}$ and $\nu_1, \nu_2 \in \mathfrak{a}^{-1}$ such that $\mu_1\nu_1 - \mu_2\nu_2 = 1$.
Steps:
    1: If $\mathfrak{a}$ is not integral, set $d = \mathrm{den}(\mathfrak{a})$. Execute the algorithm with the ideal $d\mathfrak{a}$ to
           obtain $\mu_1', \mu_2' \in d\mathfrak{a}$ and $\nu_1', \nu_2' \in \frac{\mathfrak{a}^{-1}}{e}$ such that $\mu_1'\nu_1' - \mu_2'\nu_2' = 1$. Return
           with $\mu_1 = \frac{\mu_1'}{d}$, $\mu_2 = \frac{\mu_2'}{d}$, $\nu_1 = d\nu_1'$, $\nu_2 = d\nu_2'$.
    2: Compute the two normal presentation of $\mathfrak{a}$ as $\mathfrak{a} = a\mathcal{D} + \alpha\mathcal{D}$.
    3: Compute $\mathfrak{a}^{-1}$ with the two normal presentation as $\mathfrak{a}^{-1} = \mathcal{D} + b\alpha^{-1}$, where
           $b \in \mathbb{Z}, (a, b) = 1$.
    4: Apply the extended EUCLIDean algorithm to obtain $r, s \in \mathbb{Z}$ such that $ar+bs = 1$.
    5: Return with $\mu_1 = ar$, $\mu_2 = \alpha$, $\nu_1 = -1$, $\nu_2 = sb\alpha^{-1}$.
    6: End.
```

The methods to compute and invert a normal presentation are described in [PZ93, pp.400–406]. Apart from the properties of the presentation of the inverse ideal, the proof of this algorithm is trivial. The most difficult part of this algorithm is the determination of the normal presentation (see definition 1.2.6).

Proposition 1.8.7 ([Coh96, Theorem 1.2]):
Let $\mathcal{D}$ be a DEDEKIND ring. Let $\mathfrak{a}$ and $\mathfrak{b}$ be two fractional $\mathcal{D}$–ideals, and let $a, b \in \mathcal{K}$ not both equal to zero. Set $\mathfrak{d} = a\mathfrak{a} + b\mathfrak{b}$. Then there exist $u \in \mathfrak{a}\mathfrak{d}^{-1}$ and $v \in \mathfrak{b}\mathfrak{d}^{-1}$ such that $au + bv = 1$.

The proof is easy using proposition 1.6.1 and the constructions of the following algorithm.

Algorithm 1.8.8: Solving $au + bv = 1$ with $\mathfrak{d} = a\mathfrak{a} + b\mathfrak{b}$ in $u \in \frac{\mathfrak{a}}{\mathfrak{d}}, v \in \frac{\mathfrak{b}}{\mathfrak{d}}$

```
Input: Fractional $\mathcal{D}$–ideals $\mathfrak{a}$,$\mathfrak{b}$; $a, b \in \mathcal{K}$.
Output: $\mathfrak{d} = a\mathfrak{a} + b\mathfrak{b}$, $u \in \mathfrak{a}\mathfrak{d}^{-1}$, $v \in \mathfrak{b}\mathfrak{d}^{-1}$ such that $au + bv = 1$.
Steps:
    1: If $a = 0$ return $u = 0$, $v = \frac{1}{b}$.
    2: If $b = 0$ return $u = \frac{1}{a}$, $v = 0$.
    3: Compute $\mathfrak{d} = a\mathfrak{a} + b\mathfrak{b}$, $\mathfrak{c}_1 = a\mathfrak{a}\mathfrak{d}^{-1}$, and $\mathfrak{c}_2 = b\mathfrak{b}\mathfrak{d}^{-1}$.
    4: Apply algorithm 1.6.2 to obtain $e \in \mathfrak{c}_1$, $f \in \mathfrak{c}_2$ such that $e + f = 1$.
    5: Return $u = \frac{e}{a}$ and $v = \frac{f}{b}$.
    6: End.
```

Proof. (Validity of the algorithm) Let $a, b \neq 0$. Since $a\mathfrak{a} \subset \mathfrak{d}$, the ideal $\mathfrak{c}_1$ is integral, and the same applies to $\mathfrak{c}_2$. From the construction, it follows that $\mathfrak{c}_1 + \mathfrak{c}_2 = 1\mathcal{D}$, and therefore the algorithm 1.6.2 is applicable. □

This algorithm is polynomial since algorithm 1.6.2 is.

Let $\mathcal{O}$ be an order of an algebraic number field $\mathcal{K}$ of degree n. Let Ω be a $\mathbb{Z}$–basis of $\mathcal{O}$ as in formula 1.1.1. The following algorithm gives a solution of an equation if it exists.

```
Algorithm 1.8.9: Solving $\sum_{i=1}^{k} \alpha_i\beta_i = \alpha$ in $\beta_i$
  Input: $\alpha, \alpha_1, \ldots, \alpha_k \in \mathcal{K}, k \in \mathbb{N}$.
  Output: $\beta_1, \ldots, \beta_n \in \mathcal{D}$ such that $\sum_{i=1}^{k} \alpha_i\beta_i = \alpha$.
  Steps:
    1: Compute $d = \operatorname{lcm}\big(\operatorname{den}(\alpha), \operatorname{den}(\alpha_1), \ldots, \operatorname{den}(\alpha_k)\big)$.
    2: Compute the representation matrices of the algebraic numbers $d\alpha_1, \ldots, d\alpha_k$,
         and concatenate them to a matrix $\mathbf{M}$.
    3: Compute the upper column HNF of $\mathbf{M}$ with transformation matrix $\mathbf{T}$. Let the
         first $n$ columns, the nontrivial part, of $\mathbf{MT}$ be $\mathbf{H}$. Let the first $n$ columns
         of $\mathbf{T}$ be denoted with $\mathbf{T}' \in \mathcal{K}^{kn\times n}$.
    4: Let $\mathsf{A}$ be the representation vector of $d\alpha$: $d\alpha = \Omega\mathsf{A}$. Solve the matrix equation
         $\mathbf{H}\mathsf{C} = \mathsf{A}$ in $\mathsf{C} \in \mathbb{Q}^n$.
    5: If $\mathsf{C} \notin \mathbb{Z}^n$, return with the message that the equation cannot be solved.
    6: Compute the vector $\mathsf{B} = \mathbf{T}'\mathsf{C}$.
    7: Split $\mathsf{B} = \begin{pmatrix} b_1 \\ \vdots \\ b_k n \end{pmatrix} \in \mathcal{D}^{kn}$ into vectors $\mathsf{B}_i = \begin{pmatrix} b_{n(i-1)+1} \\ \vdots \\ b_{ni} \end{pmatrix} \in \mathcal{D}^n$ vertically, for
         $i \in \mathbb{N}_k$, representing the algebraic numbers $\beta_i = \Omega\mathsf{B}_i$.
    8: End.
```

Proof. Consider the ideal $\mathfrak{a} = d\alpha_1\mathcal{D} + \cdots + d\alpha_k\mathcal{D}$. A solution to the equation exists iff $d\alpha \in \mathfrak{a}$. Using the $\mathbb{Z}$–basis presentation matrix $\mathbf{H}$ of $\mathfrak{a}$, it is easy to decide whether $d\alpha \in \mathfrak{a}$. The rest of the proof deals with the transformation of a linear combination in the $\mathbb{Z}$–basis of $\mathfrak{a}$ to a linear combination in the generating set $\alpha_1, \ldots, \alpha_n$ of $\mathfrak{a}$.

The matrix $\mathbf{M}$ represents the ideal $\mathfrak{a}$. $\mathbf{H}$ is another representation of $\mathfrak{a}$ which allows the linear combination $\mathbf{H}\mathsf{C} = \mathsf{A}$ to be computed easily because of its triangular shape. Consequently,

$$d\alpha = \Omega\mathsf{A} = \Omega\mathbf{H}\mathsf{C}. \tag{1.8.1}$$

The algorithm constructed $\mathbf{H}$ and $\mathbf{T}$ with

$$\mathbf{MT} = (\mathbf{H} \mid 0 \mid \cdots \mid 0) \quad \text{and} \quad \mathbf{MT}' = \mathbf{H}. \tag{1.8.2}$$

$\mathbf{M} = (\mathbf{M}_1 \mid \cdots \mid \mathbf{M}_k)$ consists of k $n \times n$–matrices $\mathbf{M}_i$ satisfying $\Omega\mathbf{M}_i\mathbb{Z}^n = d\alpha_i\mathcal{D}$. Since $\beta_i = \Omega\mathsf{B}_i$, this gives

$$d\alpha_i\beta_i = \Omega\mathbf{M}_i\mathsf{B}_i. \tag{1.8.3}$$

We conclude that

$$\begin{aligned}
\sum_{i=1}^{k} d\alpha_i\beta_i &= \sum_{i=1}^{k} \Omega\mathbf{M}_i\mathsf{B}_i \quad \text{(formula (1.8.3))} \\
&= \Omega\left(\sum_{i=1}^{k} \mathbf{M}_i\mathsf{B}_i\right) \\
&= \Omega\mathbf{M}\mathsf{B} \\
&= \Omega\mathbf{MT}'\mathsf{C} \\
&= \Omega\mathbf{H}\mathsf{C} \quad \text{(formula (1.8.2))} \\
&= d\alpha \quad \text{(formula (1.8.1))}.
\end{aligned}$$

□

Chapter 2

Reducing algebraic numbers with ideals

This chapter deals with the following general task: Let $\mathcal{D}$ be an integral domain (a commutative unital ring without nontrivial zero divisors), $\mathcal{K}$ its quotient field, and $\mathfrak{a}$ be a fractional $\mathcal{D}$–ideal. Assume the statement: "if the element $\alpha \in \mathcal{K}$ has a certain property so has $\alpha + \beta$ for any $\beta \in \mathfrak{a}$". The task is to find a $\alpha + \beta$ which is "small" regarding a certain notion of size.

The modulo calculus is a well–known theory addressing a similar task — in elementary number theory, $\mathcal{D} = \mathbb{Z}$. The basic steps can be applied to more general rings as well.

The problem is to extend the theory of modulo calculus to *fractional* ideals and to *fractional* elements, but there is no canonical way to do this. In fact the usual fractional extension is *not* what we need for the general task mentioned above. In the sequel a distinction will be made between the *modulo calculus*, which is the usual fractional extension, and the *reduce calculus*.

The first section deals with the case $\mathcal{D} = \mathbb{Z}$ to clarify the difference between the modulo and reduce calculi. The second section states the basic definitions and propositions for the general case of an integral domain $\mathcal{D}$. The third section deals with the special case of orders of algebraic number fields, including detailed algorithms most of which are implemented in KANT.

2.1 Reduce calculus for the rational numbers

This section goes back to elementary number theory. The well–known modulo calculus deals with a relation defined by

$$a \equiv_m b \iff_{\text{Def}} \exists c \in \mathbb{Z}\colon a - b = cm \quad \text{where} \quad a, b \in \mathbb{Z}, m \in \mathbb{N}. \tag{2.1.1}$$

This is an equivalence relation, and for $m > 1$ the classes form the finite unital ring $\mathbb{Z}_m$. For the units of this ring (which are the rational integers coprime to m) the multiplication has an inverse operation. From now on let $m > 1$.

It makes sense to write fractions with denominators coprime to m in the modulo calculus. We have FERMAT's proposition [1]

$$\frac{1}{c} \equiv_m c^{\varphi(m)-1}, \text{ where } \quad c \in \mathbb{Z}, \quad (c, m) = 1. \tag{2.1.2}$$

[1] φ denotes the EULER phi–function

From a different standpoint it is possible to say that the relation $\equiv_m$ is *extended* to the set of rational numbers whose denominators are coprime to m, which will be denoted by

$$\mathbb{Z}^{(m)} \quad =_{\text{Def}} \quad \left\{ \tfrac{a}{n} \mid a \in \mathbb{Z}, n \in \mathbb{N}, (a, n) = 1, (n, m) = 1 \right\}. \tag{2.1.3}$$

With $\mathcal{S} = \left\{ n \in \mathbb{Z} \mid (n, m) = 1 \right\}$, in [PZ93, p. 226] this set is called the $\mathcal{S}$–localization of $\mathbb{Z}$, in [Lan94, ch. II,§4] it is called quotient ring of $\mathbb{Z}$ by $\mathcal{S}$.

Together with formulas (2.1.1) and (2.1.2), we have defined a new relation which will be denoted by $\equiv_m^{\mathrm{M}}$. The superscript only refers to the behavior of the relation on fractional numbers. The superscript **M** should indicate that it refers to the extension of the **M**odulo calculus towards fractional elements. Another superscript **R** refers to the **R**educe calculus, which will be defined later.

The relation $\equiv_m^{\mathrm{M}}$ is an equivalence relation on the elements of $\mathbb{Z}^{(m)}$. It has exactly m equivalence classes which form the finite unital commutative ring $\mathbb{Z}^{(m)}/\equiv_m^{\mathrm{M}}$, which is isomorphic to the ring $\mathbb{Z}_m$.

But this is not the only possible extension of the relation $\equiv_m$ towards $\mathbb{Q}$:

$$a \equiv_m^{\mathrm{R}} b \iff_{\text{Def}} \exists c \in \mathbb{Z} \quad \text{where} \quad a - b = cm \quad \text{where} \quad a, b, m \in \mathbb{Q}\,. \tag{2.1.4}$$

The relations $\equiv_m^{\mathrm{R}}$ and $\equiv_m^{\mathrm{M}}$ are identical for integral numbers. But, if denominators occur, they are very different: $\frac{1}{2} \equiv_5^{\mathrm{M}} 3$ but $\frac{1}{2} \not\equiv_5^{\mathrm{R}} 3$. Although $\frac{12}{5} \equiv_5^{\mathrm{R}} -\frac{13}{5}$, we cannot write $\frac{12}{5}$ in connection with $\equiv_5^{\mathrm{M}}$ because it is not well–defined. And $\frac{1}{3} \equiv_{\frac{3}{4}}^{\mathrm{R}} \frac{10}{3}$ is true, but $\equiv_{\frac{3}{4}}^{\mathrm{M}}$ is not defined at all.

If $a, b, m \in \mathbb{Q}$ and d is the least common denominator of a, b, and m then

$$a \equiv_m^{\mathrm{R}} b \iff ad \equiv_{md}^{\mathrm{M}} bd. \tag{2.1.5}$$

2.1.1 Modulo and reduce functions

Integral modulo functions fix representatives of the classes of $\mathbb{Z}/\equiv_m$. They come in two flavors (as smallest positive and smallest absolute) and can be assumed to be known.

To fix representatives of the classes of $\mathbb{Z}^{(m)}/\equiv_m^{\mathrm{M}}$ we use the following definition, which is a simple extension from the integral modulo functions.

Definition 2.1.1 (Modulo function):
Let $m \in \mathbb{N}$. Then we define two functions

$$\overset{+}{\mathrm{mod}}{}_m^{\mathrm{M}} : \mathbb{Z}^{(m)} \to \mathbb{Z}^{\geq 0} \tag{2.1.6}$$

$$a \mapsto b \quad \text{with} \quad a \equiv_m^{\mathrm{M}} b \quad \text{and} \quad 0 \leq b < m \tag{2.1.7}$$

(referring to the residue system of $\mathbb{Z}_m$ with the smallest nonnegative values) and

$$\overset{\pm}{\mathrm{mod}}{}_m^{\mathrm{M}} : \mathbb{Z}^{(m)} \to \mathbb{Z} \tag{2.1.8}$$

$$a \mapsto b \quad \text{with} \quad a \equiv_m^{\mathrm{M}} b \quad \text{and} \quad -\frac{m}{2} < b \leq \frac{m}{2} \tag{2.1.9}$$

(referring to the residue system of $\mathbb{Z}_m$ with the smallest absolute values). The notation $\mathrm{mod}_m^{\mathrm{M}}$ refers to either $\overset{+}{\mathrm{mod}}{}_m^{\mathrm{M}}$ or to $\overset{\pm}{\mathrm{mod}}{}_m^{\mathrm{M}}$ by convention.

We extend $\operatorname{mod}^{\mathrm{M}}_m(a)$ to a nonpositive m by

$$\operatorname{mod}^{\mathrm{M}}_m(a) \quad =_{\mathrm{Def}} \quad \begin{cases} \operatorname{mod}^{\mathrm{M}}_{-m}(a) & \text{if} \quad m < 0 \\ a & \text{if} \quad m = 0. \end{cases}$$

A **modulo function** $\operatorname{mod}^{\mathrm{M}}$ is a family

$$\begin{aligned} \operatorname{mod}^{\mathrm{M}} : \mathbb{Z} &\to \mathbb{Q}^{\mathbb{Z}} \\ m &\mapsto \operatorname{mod}^{\mathrm{M}}_m \end{aligned}$$

of such functions.

Definition 2.1.2 (Reduce function)**:**
Let $m \in \mathbb{Q}$ and let $\operatorname{mod}^{\mathrm{M}}$ be a fixed modulo function.

$$\begin{aligned} \operatorname{mod}^{\mathrm{R}}_m : \mathbb{Q} &\to \mathbb{Q} \\ a &\mapsto \frac{\operatorname{mod}^{\mathrm{M}}_{md}(ad)}{d}, \end{aligned}$$

where d is the least common denominator of a and m. A **reduce function** is a family of functions

$$\begin{aligned} \operatorname{mod}^{\mathrm{R}} : \mathbb{Q} &\to \mathbb{Q}^{\mathbb{Q}} \\ m &\mapsto \operatorname{mod}^{\mathrm{R}}_m \end{aligned}$$

A reduce function $\operatorname{mod}^{\mathrm{R}}$ satisfies

$$a \equiv^{\mathrm{R}}_m b \iff \operatorname{mod}^{\mathrm{R}}_m(a) = \operatorname{mod}^{\mathrm{R}}_m(b), \text{ where } \quad a, b, m \in \mathbb{Q}\,.$$

To investigate the structure of $\mathbb{Q}/{\equiv^{\mathrm{R}}_m}$, we first let $a, m \in \mathbb{Q}$ and $\overline{a}$ the class of a regarding $\equiv^{\mathrm{R}}_m$. We can define addition with

$$\overline{a} + \overline{b} \quad =_{\mathrm{Def}} \quad \overline{a+b}$$

This definition is independent of the choice of the representatives: Let $a_1, a_2, b_1, b_2 \in \mathbb{Q}$, $a_1 \equiv^{\mathrm{R}}_m a_2$, $b_1 \equiv^{\mathrm{R}}_m b_2$. There exist $c_1, c_2 \in \mathbb{Z}$ such that $a_1 - a_2 = c_1 m$ and $b_1 - b_2 = c_2 m$. Therefore $(a_1 + b_1) - (b_1 + b_2) = (c_1 + c_2)m$, and thus $a_1 + b_1 \equiv^{\mathrm{R}}_m a_2 + b_2$.

The usual construction of multiplication

$$\overline{a} \cdot \overline{b} \quad =_{\mathrm{Def}} \quad \overline{a \cdot b},$$

however, depends upon the choice of representatives:

For $m = 7$, $\overline{\left(\frac{3}{2}\right)}\,\overline{\left(\frac{2}{3}\right)} = \overline{\left(\frac{3}{2}\frac{2}{3}\right)} = \overline{1}$. While, on the other hand, $\frac{3}{2} \equiv^{\mathrm{R}}_7 \frac{17}{2}$ and $\overline{\left(\frac{17}{2}\right)}\,\overline{\left(\frac{2}{3}\right)} = \overline{\left(\frac{17}{2}\frac{2}{3}\right)} = \overline{\left(\frac{17}{3}\right)} \neq \overline{1}$.

Thus, this definition of multiplication with the usual construction does not lead to a multiplication with the usual properties, so that $\mathbb{Q}/{\equiv^{\mathrm{R}}_m}$ can only be considered as an additive Abelean commutative group.

2.2 Reduce calculus in integral domains

Let $\mathcal{D}$ be an integral domain (a commutative unital ring without nontrivial zero divisors). Let $\mathcal{K}$ be the field of fractions of $\mathcal{D}$ as defined in [Lan94, ch. II, §4] . $\mathcal{D}$ can be embedded in $\mathcal{K}$, e.g. an element of $\mathcal{K}$ which can be represented as $\frac{\alpha}{1}$ is identified with $\alpha \in \mathcal{D}$. These elements will be called integral elements, as opposed to all elements of $\mathcal{K}$ which are called fractional elements.

An element $\alpha \in \mathcal{K}$ can be represented by (usually infinitely) many symbolic fractions. To pick one of them, the general notion of denominator will be needed.

Definition 2.2.1:

A denominator mapping is a function: $\mathrm{den}\colon \mathcal{K} \to \mathcal{D} \setminus \{0\}$ with the property $\forall \alpha \in \mathcal{K},\ \mathrm{den}(\alpha)\alpha \in \mathcal{D}$.

There exists at least one denominator mapping — this is a simple consequence of the selection axiom of set theory.

Every element $\alpha \in \mathcal{K}$ can be represented by the fraction $\frac{\mathrm{den}(\alpha)\alpha}{\mathrm{den}(\alpha)}$, where $\mathrm{den}(\alpha)\alpha \in \mathcal{D}$ and $\mathrm{den}(\alpha) \in \mathcal{D}$.

Example 2.2.2. *Let $\mathcal{D}$ be an algebraic number ring. $\mathbb{N}$ can be embedded in $\mathcal{D}$. For every $\alpha \in \mathcal{K}$, it is possible to find a natural number d such that $d\alpha \in \mathcal{D}$. Consequently, the denominator should be defined as the least natural number d which satisfies $d\alpha \in \mathcal{D}$.*

2.2.1 Modulo calculus

Definition 2.2.3:

Let $\mathcal{D}$ be an integral domain and $\mathcal{K}$ its quotient field. Let a denominator mapping den be fixed according to definition 2.2.1.

Let $\alpha, \beta \in \mathcal{K}$ and $\mathfrak{a}$ a nonzero integral $\mathcal{D}$–ideal. α is called congruent β modulo $\mathfrak{a}$, denoted $\alpha \equiv^{\mathrm{M}}_{\mathfrak{a}} \beta$, iff

- $\mathrm{den}(\alpha)\mathcal{D} + \mathfrak{a} = 1\mathcal{D}$, [2]
- $\mathrm{den}(\beta)\mathcal{D} + \mathfrak{a} = 1\mathcal{D}$, and
- $\mathrm{den}(\alpha)\,\mathrm{den}(\beta)\alpha - \mathrm{den}(\alpha)\,\mathrm{den}(\beta)\beta \in \mathfrak{a}$.

Example 2.2.4. *Let $\mathcal{D} = \mathbb{Z}$. This definition of the modulo relation is consistent with the modulo calculus in $\mathbb{Z}$. Every nonzero integral ideal is generated by a natural number, let $\mathfrak{a}$ be the $\mathbb{Z}$–ideal generated by $m \in \mathbb{N}$: $\mathfrak{a} = m\mathbb{Z}$.*

Example 2.2.5. *Let $\mathcal{D}$ be a ring of algebraic numbers over $\mathbb{Z}$, let the denominator be fixed as in example 2.2.2. We want to construct an element in $\mathcal{D}$ which is equivalent to $\frac{\gamma}{d}$ where $\gamma \in \mathcal{D}, d \in \mathbb{N}$. We can invert the denominator d modulo the nonzero integral $\mathcal{D}$–ideal $\mathfrak{a}$, e.g. find a positive integer n that $nd \equiv^{\mathrm{M}}_{\mathfrak{a}} 1$. This last property is equivalent to $nd \equiv^{\mathrm{M}}_{\mathfrak{a}\cap\mathbb{Z}} 1$. The ideal $\mathfrak{a} \cap \mathbb{Z}$ is principal and generated by a natural number m, so we have $n := d^{\varphi(m)-1}$. As in the integer case this is only possible if d and m are coprime which is implied by $\mathfrak{a} + d\mathcal{D} = 1\mathcal{D}$.*

Proposition 2.2.6:

Let $\mathcal{D}$ be an integral domain and $\mathfrak{a}$ a nonzero integral $\mathcal{D}$–ideal. Let $\mathcal{D}^{(\mathfrak{a})}$ be the set of fractional elements whose denominator is coprime to $\mathfrak{a}$:

$$\mathcal{D}^{(\mathfrak{a})} := \left\{ \tfrac{\alpha}{\delta} \in \mathcal{K} \mid \alpha, \delta \in \mathcal{D}, \delta \neq 0, \delta\mathcal{D} + \mathfrak{a} = 1\mathcal{D} \right\}.$$

[2] $\alpha\mathcal{D}$ denotes the fractional $\mathcal{D}$–ideal generated by $\alpha \in \mathcal{K}$; $1\mathcal{D}$ is the trivial ideal generated by 1.

With $\mathcal{S} = \{\delta \in \mathcal{K} \mid \delta\mathcal{D} + \mathfrak{a} = 1\mathcal{D}\}$, in [PZ93, p. 226] this set is called the $\mathcal{S}$–localization of $\mathcal{D}$, in [Lan94, ch. II, §4] it is called quotient ring of $\mathcal{D}$ by $\mathcal{S}$.

Then $\equiv^{\mathrm{M}}_{\mathfrak{a}}$ is an equivalence relation on $\mathcal{D}^{(\mathfrak{a})}$.

The quotient $\mathcal{D}^{(\mathfrak{a})}/\equiv^{\mathrm{M}}_{\mathfrak{a}}$ is a unital commutative ring isomorphic to $\mathcal{D}/\equiv^{\mathrm{M}}_{\mathfrak{a}}$.

Proof. Reflexivity is trivial because 0 is an element of every ideal. Symmetry is trivial because -1 is always a unit in $\mathcal{D}$. Transitivity is easy because, for $\alpha_1 - \alpha_2 \in \mathfrak{a}$ and $\alpha_2 - \alpha_3 \in \mathfrak{a}$, by the definition of an ideal, $\alpha_1 - \alpha_3 = (\alpha_1 - \alpha_2) + (\alpha_2 - \alpha_3) \in \mathfrak{a}$.

The last statement uses the fact that $\mathcal{D}/\mathfrak{a}$ is a unital commutative ring. In every equivalence class there are integral elements. This clearly defines a 1–1 correspondence which is an isomorphism. □

Definition 2.2.7:

Let $\mathcal{D}$ be an integral domain. A **modulo function** is a family of functions

$$\begin{aligned} \mathrm{mod}^{\mathrm{M}} : \mathrm{I}_{\mathcal{D}} \setminus \{0\mathcal{D}\}^{\dagger} &\to {\mathcal{D}^{(\mathfrak{a})}}^{\mathcal{D}} \\ \mathfrak{a} &\mapsto \mathrm{mod}^{\mathrm{M}}_{\mathfrak{a}}, \end{aligned}$$

where for any integral $\mathcal{D}$–ideal $\mathfrak{a}$ the function $\mathrm{mod}^{\mathrm{M}}_{\mathfrak{a}} : \mathcal{D}^{(\mathfrak{a})} \to \mathcal{D}$ satisfies the properties

$$\forall \alpha, \beta \in \mathcal{D}^{(\mathfrak{a})},\ \alpha \equiv^{\mathrm{M}}_{\mathfrak{a}} \beta \iff \mathrm{mod}^{\mathrm{M}}_{\mathfrak{a}}(\alpha) = \mathrm{mod}^{\mathrm{M}}_{\mathfrak{a}}(\beta) \quad \text{and}$$

$$\mathrm{mod}^{\mathrm{M}}_{\mathfrak{a}}(\alpha) \quad \text{lies in the same class of } \mathcal{D}^{(\mathfrak{a})}/\equiv^{\mathrm{M}}_{\mathfrak{a}} \text{ as } \alpha.$$

For any integral domain $\mathcal{D}$, there exists such a modulo function — this is an immediate consequence of the selection axiom of set theory and the fact that $\equiv^{\mathrm{M}}_{\mathfrak{a}}$ is an equivalence relation and splits $\mathcal{D}^{(\mathfrak{a})}$ in disjoint classes.

2.2.2 Reduce calculus

Again the relation $\equiv^{\mathrm{M}}_{\mathfrak{a}}$ can be seen as an extension of the integral modulo calculus to $\mathcal{K}$, and again this is not the only possible extension.

Definition 2.2.8:

Let $\alpha, \beta \in \mathcal{K}$ and let $\mathfrak{a}$ be a fractional $\mathcal{D}$–ideal or the zero ideal. Then $\alpha \equiv^{\mathrm{R}}_{\mathfrak{a}} \beta$ iff $\alpha - \beta \in \mathfrak{a}$.

Remark:

Let $\alpha, \beta \in \mathcal{D}$ and let $\mathfrak{a}$ be a nonzero integral $\mathcal{D}$–ideal. Then $\alpha \equiv^{\mathrm{R}}_{\mathfrak{a}} \beta \iff \alpha \equiv^{\mathrm{M}}_{\mathfrak{a}} \beta$.

The $\equiv^{\mathrm{M}}_{\mathfrak{a}}$ relation can be defined in terms of the $\equiv^{\mathrm{R}}_{\mathfrak{a}}$ relation:

Lemma 2.2.9:

Let α, $\beta \in \mathcal{K}$ and $\mathfrak{a}$ be a fractional $\mathcal{D}$–ideal. Let δ be a common multiple of the denominators of α, β, and $\mathfrak{a}$. Then

$$\alpha \equiv^{\mathrm{R}}_{\mathfrak{a}} \beta \iff \delta\alpha \equiv^{\mathrm{M}}_{\delta\mathfrak{a}} \delta\beta.$$

[†] $\{0\mathcal{D}\}$ denotes the zero ideal (the ideal only containing zero) and $\mathrm{I}_{\mathcal{D}}$ denotes the set of integral $\mathcal{D}$–ideals with the zero ideal.

Proposition 2.2.10:

$\equiv^{\mathrm{R}}_{\mathfrak{a}}$ is an equivalence relation for any fractional $\mathcal{D}$–ideal $\mathfrak{a}$.

Proof. Reflexivity is trivial because 0 is an element of every ideal. Symmetry is trivial because -1 is always a unit in $\mathcal{D}$. Transitivity can be shown with the ideal properties. □

Definition 2.2.11:

Let $\mathcal{D}$ be an integral domain and $\mathcal{K}$ its quotient field. A **reduce function** is a family of functions

$$\begin{aligned} \mathrm{mod}^{\mathrm{R}} : \mathrm{I}_{\mathcal{K}}{}^{\ddagger} &\to \mathcal{K}^{\mathcal{K}} \\ \mathfrak{a} &\mapsto \mathrm{mod}^{\mathrm{R}}_{\mathfrak{a}}, \end{aligned}$$

where for any $\mathfrak{a} \in \mathrm{I}_{\mathcal{K}}$ the function $\mathrm{mod}^{\mathrm{R}}_{\mathfrak{a}} : \mathcal{K} \to \mathcal{K}$ satisfies the properties

$$\forall \alpha, \beta \in \mathcal{K},\ \alpha \equiv^{\mathrm{R}}_{\mathfrak{a}} \beta \iff \mathrm{mod}^{\mathrm{R}}_{\mathfrak{a}}(\alpha) = \mathrm{mod}^{\mathrm{R}}_{\mathfrak{a}}(\beta) \quad \text{and} \tag{2.2.1}$$

$$\mathrm{mod}^{\mathrm{R}}_{\mathfrak{a}}(\alpha) \equiv^{\mathrm{R}}_{\mathfrak{a}} \alpha. \tag{2.2.2}$$

An element $\alpha \in \mathcal{K}$ is called **reduced** (modulo the fractional $\mathcal{D}$–ideal $\mathfrak{a}$) iff $\mathrm{mod}^{\mathrm{R}}_{\mathfrak{a}}(\alpha) = \alpha$.

Proposition 2.2.12:

Let $\mathcal{D}$ be any integral domain and $\mathcal{K}$ its quotient field.

- There exists a reduce function $\mathrm{mod}^{\mathrm{R}}$.
- For any reduce function $\mathrm{mod}^{\mathrm{R}}$, any fractional $\mathcal{D}$–ideal, and any $\alpha \in \mathcal{K}$, the element $\mathrm{mod}^{\mathrm{R}}_{\mathfrak{a}}(\alpha) \in \mathcal{K}$ is reduced.
- Let $\mathfrak{a}$ be a fractional $\mathcal{D}$–ideal and $\mathrm{mod}^{\mathrm{R}}$ a reduce function. In every class of $\mathcal{K}/{\equiv^{\mathrm{R}}_{\mathfrak{a}}}$ there exists exactly one reduced element.

Proof. The first statement is a consequence of the selection axiom of set theory and the fact that $\equiv^{\mathrm{R}}_{\mathfrak{a}}$ is a equivalence relation and splits $\mathcal{K}$ in disjoint classes.

For the second statement, let $\beta = \mathrm{mod}^{\mathrm{R}}_{\mathfrak{a}}(\alpha)$. From property (2.2.2) we know $\beta \in \alpha \left(\mathcal{K}/{\equiv^{\mathrm{R}}_{\mathfrak{a}}}\right)$ and $\alpha \equiv^{\mathrm{R}}_{\mathfrak{a}} \beta$. By property (2.2.1), $\mathrm{mod}^{\mathrm{R}}_{\mathfrak{a}}(\alpha) = \mathrm{mod}^{\mathrm{R}}_{\mathfrak{a}}(\beta)$, which completes the proof.

The third statement follows from the second statement and property (2.2.1). □

Reduce functions are very important for computational applications. If there is an efficient algorithm to compute the reduce function, it can be used to decide the relation $\equiv^{\mathrm{R}}_{\mathfrak{a}}$.

If $\mathcal{K}$ has a strict ordering $<$ and any subset of $\mathcal{K}$ contains a minimum regarding $<$, a reduce function can be defined using this ordering

$$\mathrm{red}_{\mathfrak{a}}(\alpha) \quad =_{\mathrm{Def}} \quad \min_{<} \{\beta \in \mathcal{K} \mid \alpha \equiv^{\mathrm{R}}_{\mathfrak{a}} \beta\}.$$

If we have a modulo function for integral ideals and elements, we can construct a reduce function with the following lemma.

‡. $\mathrm{I}_{\mathcal{K}}$ denotes the group of fractional $\mathcal{D}$–ideals without the zero ideal.

Lemma 2.2.13:

Let $\mathcal{D}$ be an integral domain and $\mathcal{K}$ its quotient field. Let $\mathrm{mod}^{\mathrm{M}}$ be a modulo function as in definition 2.2.7 . Let $\mathfrak{a}$ be a fractional $\mathcal{D}$–ideal and $\alpha \in \mathcal{K}$. Letting $\delta \in \mathcal{D}$ be the product of the denominators of $\mathfrak{a}$ and α,

$$\mathrm{mod}^{\mathrm{R}}_{\mathfrak{a}}(\alpha) \;=_{\mathrm{Def}}\; \frac{\mathrm{mod}^{\mathrm{M}}_{\delta\mathfrak{a}}(\delta\alpha)}{\delta} \in \mathcal{K},$$

is a reduce function.

2.3 Representatives in algebraic number rings

Let $\mathcal{K}$ be a finite algebraic field extension of $\mathbb{Q}$. Let $\mathcal{O}$ be an order of $\mathcal{K}$. This is a special case of the previous section since $\mathcal{K}$ is the quotient field of $\mathcal{O}$.

Let $\Omega = (\omega_1, \ldots, \omega_n)$ be a $\mathbb{Z}$–basis of $\mathcal{O}$. An $\alpha \in \mathcal{O}$ is represented by a vector

$$\mathsf{A} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \quad \text{with} \quad a_1, \ldots, a_n \in \mathbb{Z}$$

as

$$\alpha = \sum_{i=1}^{n} a_i \omega_i = \Omega \mathsf{A}. \tag{2.3.1}$$

The problem of finding reduce functions can be dealt with in two steps:

- Finding modulo functions for integral algebraic numbers and integral ideals. (This is the main subject of this section.)
- Constructing a reduce function from an integral modulo function. (This is simply done with lemma 2.2.13.)

There is a variety of possible modulo functions to choose from in algebraic number rings, the classification of which is dealt with in the next subsection. The key for this classification is the notion of quality.

2.3.1 Quality measurements of representations of algebraic numbers

It is important to stress that we are not talking about the quality of algebraic numbers — the reduce functions are not independent of the basis that the algebraic numbers are presented with.

A quality measurement can be either

- a relation $>_\chi$, where the quality of the representations of two algebraic numbers (and not the numbers themselves) is compared or
- a function

  $$\chi : \mathbb{Z}^n \to \mathbb{R}^{\geq 0}.$$

  If we have a quality function χ, we can define a quality relation $>_\chi$ as

  $$\mathsf{A} >_\chi \mathsf{B} \iff_{\mathrm{Def}} \chi(\mathsf{A}) > \chi(\mathsf{B}).$$

The aim of the quality relation/function is that small values of χ resp. small representations of algebraic numbers regarding $>_\chi$ should correspond to computational "desirable" algebraic numbers.

What do we expect from a good quality function?

It should

1. measure how much memory space is required to store the representation, with small values of χ corresponding to little required memory space;
2. give a prediction on the time and memory efficiency of computations with the representation, with small values of χ corresponding to fast and efficient computations;
3. give a prediction on the quality of the results of arithmetic with the representations; i.e., the representation of the sum/product of algebraic numbers with small values of χ of their representations should also have a small value of χ;
4. give results independent from the basis Ω; even an "odd" basis should not destroy the usefulness of the quality function, in which case we can speak of the quality of an algebraic number and not only of the quality of the representation;
5. be computationally inexpensive to decide $>_\chi$ resp. compute χ;
6. be easy to select (for a representation A of a given algebraic number) another representation B of an algebraic number with $\mathsf{B} <_\chi \mathsf{A}$.

In the sequel different possible quality functions/relations are introduced, none of which is perfect. For each of them comments are provided on how each of the criteria for a quality function/relation given above is satisfied by the particular quality function/relation. The comments are based on both practical observations and theoretical considerations.

Vector norm of the representation

This quality function is either the 1–norm ($\sum\limits_{i=1}^{n} |a_i|$), 2–norm ($\sqrt{\sum\limits_{i=1}^{n} a_i^2}$), or the ∞–norm ($\max\{|a_i|\}$) of the representation vector A.

Evaluation of the criteria:

1. Perfect. The vector norm of the representation is a good estimate of the memory space required, in particular the 1–norm.
2. Perfect. Addition of algebraic numbers involves vector addition of the representation. Multiplication is usually done with a multiplication table — the actual multiplication involves a matrix multiplication with the representation vector.
3. Perfect for addition. Good for multiplication if the basis Ω is such that the entries of the multiplication table are small.
4. Basis dependent.
5. Very easy for the 1–norm and the ∞–norm; relatively easy for the 2–norm.
6. Trivial since the norms are monotone in each of the coefficients of the representation.

Lexicographic ordering of the representation

This is a quality relation, only, defined as:

$$\begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} >_\chi \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \quad \Longleftrightarrow_{\text{Def}} \quad \begin{array}{r} \exists i \in \mathbb{N}_n : \Big(\big(|a_i| > |b_i| \text{ or } a_i = -b_i > 0 \big) \\ \text{and } \forall j \in \mathbb{N}_n,\ j > i \to a_j = b_j \Big) \end{array}$$

Evaluation of the criteria:

1. Bad. There are infinitely many algebraic numbers with a lower quality than e.g. any basis element of Ω which is not a rational number.
2. Bad for the same reason as criteria 1. On the other hand, there is a nice feature. If the first basis element ω_1 is 1 (every order in a number field has a basis with this property,) then the representations with the lowest qualities are representations which represent the rational integers.
3. Good for addition, bad for multiplication.
4. Basis dependent.
5. Very easy to decide.
6. Trivial since the norms are monotone in each of the coefficients of the representation.

Complex absolute value/T_1*–norm/rooted* T_2*–norm*

These two quality functions are based on the field embeddings of the algebraic number field $\mathcal{K}$ into the complex numbers $\mathbb{C}$ via a monomorphism

$$\phi : \mathcal{K} \hookrightarrow \mathbb{C}.$$

The complex absolute value

$$\chi(\mathsf{A}) = \big|\phi(\Omega\mathsf{A})\big|$$

can serve as a quality function.

χ depends on the choice of a particular embedding ϕ and usually there exist different embeddings.

An approach to overcome the dependency is to consider the conjugate fields of $\mathcal{K}$. [Coh95, Theorem 4.1.8] states that there are exactly n field embeddings, where n is the degree of $\mathcal{K}$. Let Φ be the set of all field embeddings $\phi : \mathcal{K} \hookrightarrow \mathbb{C}$. Then we have two quality functions which depend neither on a field embedding nor on the basis Ω:

T_1–norm $\chi(\mathsf{A}) = \mathrm{T}_1(\Omega\mathsf{A}) = \sum_{\phi\in\Phi} \big|\phi(\Omega\mathsf{A})\big|$ and

rooted T_2–norm $\chi(\mathsf{A}) = \sqrt{\mathrm{T}_2(\Omega\mathsf{A})} = \sqrt{\sum_{\phi\in\Phi} \big|\phi(\Omega\mathsf{A})\big|^2}$.

Evaluation of the criteria for the above quality functions:

1. Good if the basis is relatively “well–behaved”, but still reasonable if not.
2. Good, in particular for multiplication.
3. Excellent for the multiplication because we have the multiplicativity for both χ. For addition we have the triangle inequality. This is a bad estimate if the algebraic numbers are close to being orthogonal, but reasonable in practice.

4. Basis independent because the value is not based on the representation, but on the algebraic number itself. But the complex absolute value is dependent of the choice of the field embedding of $\mathcal{K}$ in $\mathbb{C}$. The T_1/rooted T_2–norm is independent on the choice of the embedding.
5. More expensive than the representation vector norm and the lexicographic ordering.

   We need the approximated complex values of the basis elements $\omega_1, \ldots, \omega_n$, which are not difficult to obtain and also important for other algorithms with $\mathcal{O}$, so that they are likely to be given anyway. Thus, the complex absolute value is quite easy to compute.

   For the T_1/rooted T_2–norm we need the complex values of the conjugates and some real number computations. Because the precision is not required to be high, this should not be much more expensive than the complex absolute value.
6. Difficult.

Norm

The norm of an algebraic number is the determinant of its representation matrix, which is the basis matrix of the principal $\mathcal{O}$–ideal generated by this algebraic number.

Evaluation of the criteria:

1. Bad. Units of $\mathcal{O}$ have norm 1, but usually there are infinitely many of them. Of course almost all of them can be said to be incredibly huge. But the problem with units is only the tip of the iceberg. In general there is only a weak correlation of the norm and the memory space.
2. Bad, as above.
3. Perfect for the multiplication because the norm is multiplicative. Bad for addition: the sum of two units might have a huge norm.
4. Basis independent.
5. The computation of the representation matrix involves a matrix multiplication and a determinant calculation. This is more expensive than all other quality functions considered in this section.
6. Difficult. Probably as difficult as unit computation (see [Coh95, Algorithm 4.9.9]).

Comparison

In most cases the vector norm of the representation is the reasonable choice for a quality function regarding the criteria given. Because of the efficient algorithm 2.3.1, the lexicographic order is a good alternative for the task of reducing algebraic numbers modulo ideals. If we require the quality function to be independent of the representation of the algebraic number, then the T_1/rooted T_2–norm should be preferred.

2.3.2 HNF basis reduction

This subsection describes one important modulo function — the HNF basis reduction.

Let $\mathfrak{a}$ be an integral ideal. We use the $\mathbb{Z}$–basis (as in equation 1.1.2) $\xi_1, \dots, \xi_n$ for $\mathfrak{a}$, where $\xi_i \in \mathcal{O}$. The ξ_i have representations in the basis Ω of $\mathcal{O}$ such that

$$\mathfrak{a} = \sum_{i=1}^{n} \mathbb{Z}\xi_i = \sum_{i=1}^{n} \mathbb{Z} \sum_{j=1}^{n} a_{ij}\omega_j. \tag{2.3.2}$$

The ideal $\mathfrak{a}$ is said to be represented by the $\mathbb{Z}$–matrix $\mathbf{A} = (a_{ij})_{i,j \in \mathbb{N}_n}$.

Because of the elementary algebraic properties of an ideal, this matrix can be transformed to an upper triangular HNF which still represents a basis for $\mathfrak{a}$.

Denote the projection of the i-th component of the vector $\begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$, which is an epimorphism from $\mathcal{K}^n$ to $\mathcal{K}$, by $\mathrm{pr}_i : \mathcal{K}^n \twoheadrightarrow \mathcal{K}$, i.e.

$$\mathrm{pr}_i \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = a_i.$$

Algorithm 2.3.1: HNF reduction of an algebraic number modulo an ideal

```
Input: An HNF basis of an integral ideal $\xi_1 \dots \xi_n$ and an integral algebraic number $\alpha$
    represented as a vector.
Output: A canonical representative $\beta$ of the class of $\alpha$ in $\mathcal{O}/\equiv^{\mathrm{M}}_{\mathfrak{a}}$.
Steps:
    Init $\beta := \alpha$.
    Loop $i = n, \dots, 1$.
        Find $q \in \mathbb{Z}$ that $|\mathrm{pr}_i(\beta - q\xi_i)|$ is minimal[3].
        Assign $\beta := \beta - q\xi_i$.
    The representative of $\alpha$'s class is now in $\beta$.
    End.
```

Remarks:

(1) The uniqueness is guaranteed in a very straightforward way. The algorithm returns the smallest representation regarding the lexicographic order mentioned in the previous subsection.

(2) The algorithm is relatively fast. The complexity is as follows (counting integer operations only and disregarding the size of the integers): n divisions, $n(n+1)/2 \approx n^2/2$ multiplications and additions.

(3) Coefficient growth of the intermediate entries of the representation of β may occur, most of all the first entry, which suffers from $n-1$ relatively uncontrolled additions. Let N be an upper bound on the absolute values of the HNF matrix entries and of the entries of the vector. The worst case multiplicator of the first loop can be N (although this is very unlikely). So the coefficients of the vector are now bounded by N^2. Iterating this consideration, we see that the coefficient size is roughly bounded by N^{n-1}.

This consideration ignores the fact that the matrix is in HNF. Say the first multiplicator has a large absolute value (close to N) and the penultimate entry

[3] It is possible to take another convention here: find $q \in \mathbb{Z}$ such that $\mathrm{pr}_i(\beta - q\xi_i)$ has its minimal nonnegative value. This convention defines another possible HNF reduction.

$a_{n-1,n}$ of the last column of the HNF matrix has large absolute value (close to N) as well. Then the $(n-1)$-st entry of b is now close to N^2. But because the penultimate diagonal entry $a_{n-1,n-1}$ is larger than $a_{n-1,n}$, the second multiplicator must actually be smaller than N. This thought can be extended analogously to all entries of the vector: if the entry has significant growth because of large entries in the matching row of the HNF matrix, the diagonal entry is large as well and keeps the multiplicator small. So the entries of the vector are usually not larger than N^2.

In practice coefficient growth does not cause any trouble.

2.3.3 General basis reduction

The next three subsections describe a class of modulo functions which is a generalization of the HNF basis reduction.

Let $\mathfrak{a}$ be an integral ideal with $\mathbb{Z}$–basis as in equation (2.3.2). There is a much more general method of reducing an element modulo this ideal not requiring the HNF property of the basis matrix of $\mathfrak{a}$. This algorithm is based on the fact that the ideal basis can be considered as a vector space basis of the field $\mathcal{K}$.

Algorithm 2.3.2: General basis reduction of an algebraic number modulo an ideal

```
Input: A basis of an integral ideal $\xi_1, \dots, \xi_n$ and an integral algebraic number $\alpha$ represented as a vector.
Output: A representative $\beta$ of the class of $\alpha$ in $\mathcal{O}/\equiv^{\mathrm{M}}_{\mathfrak{a}}$.
Steps:
    1: Represent $\alpha$ as a linear combination of the $\xi_1 \dots \xi_n$ with rational coefficients $q_1, \dots, q_n$: $\alpha = \sum_{i=1}^{n} q_i \xi_i$.
    2: Assign $\beta = \alpha - \sum_{i=1}^{n} \lfloor q_i \rceil \xi_i$.
    3: End.
```

Remarks:

(1) The symbol $\lfloor q \rceil$ denotes the nearest integer to q preferring $q - \frac{1}{2}$ to $q + \frac{1}{2}$ in case q is exactly an integer plus a half. So $-\frac{1}{2} < q - \lfloor q \rceil \leq \frac{1}{2}$.

(2) The algorithm provides an element of the class of α because the element $\sum_{i=1}^{n} \lfloor q_i \rceil \xi_i$ is an element of the ideal $\mathfrak{a}$. This element is uniquely determined because $\xi_1, \dots, \xi_n$ is a basis of $\mathcal{K}$ so the q_i are uniquely determined.

(3) The complexity of this algorithm is determined by the complexity of finding the linear combination for α.

(4) If $\alpha \equiv^{\mathrm{M}}_{\mathfrak{a}} 0$ the algorithm returns $\beta = 0$.

(5) Consider any quality function χ satisfying

- $\chi(\mathsf{A} + \mathsf{B}) \leq \chi(\mathsf{A}) + \chi(\mathsf{B})$ for $\mathsf{A}, \mathsf{B} \in \mathcal{K}^n$ (triangle inequality) and
- $\chi(\lambda \mathsf{A}) = |\lambda| \chi(\mathsf{A})$ for $\mathsf{A} \in \mathcal{K}^n$, $\lambda \in \mathcal{K}$ (linearity with respect to multiplication).

Examples are the norm of the representation and the rooted $\mathrm{T}_1/\mathrm{T}_2$–norm. Then the quality of β (identifying the quality of an algebraic number as the quality of its representation here) is bounded

$$\chi(\beta) = \chi\left(\sum_{i=1}^{n} (q_i - \lfloor q_i \rceil)\, \xi_i\right) \leq \sum_{i=1}^{n} |q_i - \lfloor q_i \rceil|\, \chi(\xi_i) \leq \frac{1}{2} \sum_{i=1}^{n} \chi(\xi_i).$$

> The practical value of this algorithm is that we are able to choose an ideal basis with elements of a better quality than the HNF basis. The inequality guarantees a good quality of the reduced element.

So our problem splits into two other problems: how to find a good ideal basis and how to compute the coefficients for a linear combination. We begin with the latter.

2.3.4 Finding linear combinations

As $\xi_1, \ldots, \xi_n$ is a $\mathbb{Z}$–basis for the ideal $\mathfrak{a}$, it is also a field basis for $\mathcal{K}$ as a $\mathbb{Q}$–vector space. An element $\alpha \in \mathcal{K}$ is represented in the basis Ω as $\sum_{i=1}^{n} a_i \omega_i$ with $a_i \in \mathbb{Q}$. So what we have to do is nothing more than a basis transformation.

Because $\xi_1, \ldots, \xi_n$ are given in the basis Ω, we have a $\mathbb{Q}$–matrix $\mathbf{M}$ with

$$(\xi_1, \ldots, \xi_n) = (\omega_1, \ldots, \omega_n)\mathbf{M}.$$

So we have

$$\alpha = \Omega \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = (\xi_1, \ldots, \xi_n)\mathbf{M}^{-1} \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix},$$

which gives us the linear combination as the result of an inversion of a rational matrix and a matrix multiplication.

2.3.5 Finding a small ideal basis

Usually the HNF reduced basis for an ideal is represented by a reasonably small matrix. There are smaller bases[4], but in general it is cumbersome to minimize any size measure. We have to find a trade–off between computation costs and basis quality.

The LLL–algorithm is a good method to get a matrix which represents a better basis at reasonable computation costs. See [Coh95, p. 81] or [PZ93, sec. 3.3] for introductions.

The LLL–algorithm is polynomial, but usually much slower than HNF computation. It is especially useful if the better basis is not just used to reduce a single algebraic number. There are different versions of LLL to be taken into consideration. The Gram–Schmidt–coefficients might be computed either with rational arithmetic or approximated real arithmetic, for instance. The details will not be discussed here.

Henri Cohen suggests in [Coh96] another method called partial reduction, which is supposed to be more quickly than LLL and still have coefficients with pretty small absolute values, although nothing can be proven as with LLL. The experiments performed did not seem very promising, so this approach was not followed.

2.3.6 Reducing algebraic numbers with rationals

If the ideal to reduce with is a principal ideal with a rational generator, there is an obvious reduce algorithm which is much easier than the above algorithms.

[4] Let χ be a quality function. The size of a basis $\xi_1, \ldots, \xi_n$ is measured by the sum of the χ–values of the vectors representing ξ_i for $i \in \mathbb{N}_n$.

Algorithmic idea 2.3.3: Reducing algebraic numbers with rationals

If the algebraic number $\Omega \mathsf{A}$ (where $\Omega = (\omega_1, \ldots, \omega_n)$ is the basis of $\mathcal{O}$ and $\mathsf{A} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$ a rational vector) is to be reduced by $r \in \mathbb{Q}$ we can use the reduce function

$$\mathrm{mod}_r^{\mathrm{R}}(\Omega \mathsf{A}) = \Omega \begin{pmatrix} \mathrm{mod}_r^{\mathrm{R}}(a_1) \\ \vdots \\ \mathrm{mod}_r^{\mathrm{R}}(a_n) \end{pmatrix}.$$

Of the quality functions considered in subsection 2.3.1 we observe that

- the algorithm minimizes the quality function "vector norm of the representation";
- as the HNF–basis of an ideal generated by a rational number is a diagonal matrix, the result is the same as the result of the HNF–reduction algorithm. Therefore it gives the first number considering the lexicographic order, also.

Chapter 3

Equivalence and normal forms of matrices

Let $\mathcal{D}$ be an integral domain (a commutative unital ring without nontrivial zero divisors) and $\mathcal{K}$ its quotient field. This chapter deals with the important correspondance of finitely generated $\mathcal{K}$–modules and classes of matrices over $\mathcal{K}$. The key is to define an equivalence relation $\approx$ on $\mathcal{K}^{n\times m}$ such that $\mathcal{K}^{n\times m}/\approx$ is isomorphic to the $\mathcal{D}$–module of finitely generated $\mathcal{D}$–modules of rank not larger than n. The even more general approach is to extend the equivalence relation to matrices of different size, the set $\bigcup_{m\in\mathbb{N}} \mathcal{K}^{n\times m}$. This isomorphism is an important theoretic concept for the understanding of the structure of the module of finitely generated modules. Even more important though for the practical investigation is that modules can be represented by matrices.

The relation $\approx$ comes in three flavours: the

- module definition is aimed at the representation mapping,
- transformation definition is aimed at efficient algorithms,
- matrix multiplication definition supplies an additional theoretic background, first of all the use of the determinant.

The equivalence of these three definitions perfects the correspondance of modules and matrices. Unfortunately, the equivalence does not hold for the generality of integral domains — we will deal with the question of how much can be done.

3.1 Matrix equivalence definitions

Definition 3.1.1:

Let $\mathbf{M} = (A_1, \ldots, A_m) \in \mathcal{K}^{n\times m}$, where the A_i are the m columns of $\mathbf{M}$. Then the (finitely generated) $\mathcal{D}$–module represented by the matrix $\mathbf{M}$ is

$$\mathrm{Mod}\,(\mathbf{M}) \;\;=_{\mathrm{Def}}\;\; \mathcal{D}A_1 + \cdots + \mathcal{D}A_m \subset \mathcal{K}^n.$$

Two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ are called **module equivalent** iff their represented modules are equal:

$$\mathbf{M}_1 \approx_{\mathrm{mod}} \mathbf{M}_2 \;\;\Longleftrightarrow_{\mathrm{Def}}\;\; \mathrm{Mod}\,(\mathbf{M}_1) = \mathrm{Mod}\,(\mathbf{M}_2)\,.$$

Note that this definition includes matrices with a different number of columns.

Definition 3.1.2:

Two matrices are called **transformation equivalent**: $\mathbf{M}_1 \approx_{\text{trafo}} \mathbf{M}_2$ iff $\mathbf{M}_1$ can be transformed to $\mathbf{M}_2$ with a finite number of transformation steps. A single transformation step from $\mathbf{M}$ to $\mathbf{M}'$ is (assuming $\mathbf{M} \in \mathcal{K}^{n\times m}$) either

1. the permutation of two columns i and j:

$$\mathbf{M} = (A_1, \ldots, A_{i-1}, A_i, A_{i+1}, \ldots, A_{j-1}, A_j, A_{j+1}, \ldots, A_m)$$

and

$$\mathbf{M}' = (A_1, \ldots, A_{i-1}, A_j, A_{i+1}, \ldots, A_{j-1}, A_i, A_{j+1}, \ldots, A_m),$$

2. the transformation of two columns A_i and A_j $(i \neq j)$ involving four scalar coefficients $c_1, c_2, c_3, c_4 \in \mathcal{D}$ satisfying $c_1c_4 - c_2c_3 = 1$:

$$\mathbf{M} = (A_1, \ldots, A_{i-1}, A_i, A_{i+1}, \ldots, A_{j-1}, A_j, A_{j+1}, \ldots, A_m)$$

and

$$\mathbf{M}' = (A_1, \ldots, A_{i-1}, c_1A_i + c_2A_j, A_{i+1}, \ldots, A_{j-1},\\ c_3A_i + c_4A_j, A_{j+1}, \ldots, A_m),$$

3. the multiplication of a unit ϵ in the ring $\mathcal{D}$ to column A_i:

$$\mathbf{M} = (A_1, \ldots, A_{i-1}, A_i, A_{i+1}, \ldots, A_m)$$

and

$$\mathbf{M}' = (A_1, \ldots, A_{i-1}, \epsilon A_i, A_{i+1}, \ldots, A_m),$$

4. the insertion of a zero column[1] before position $i \in \mathbb{N}_m$ or as the last column:

$$\mathbf{M} = (A_1, \ldots, A_{i-1}, A_i, \ldots, A_m)$$

and

$$\mathbf{M}' = (A_1, \ldots, A_{i-1}, \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, A_i, \ldots, A_m),$$

or

5. the deletion of a zero column at position $i \in \mathbb{N}_{m+1}$:

$$\mathbf{M} = (A_1, \ldots, A_{i-1}, \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, A_{i+1}, \ldots, A_{m+1})$$

and

$$\mathbf{M}' = (A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_m)$$

with the convention that at least one column must remain.

[1] shorthand for a column with only zero entries

Remark:

For EUCLIDean rings the second transformation can be replaced by the simpler transformation:

the addition of the q–fold of a column A_i to a column A_j for $q \in \mathcal{D}, i \neq j$, where

$$\mathbf{M} = (\mathsf{A}_1, \ldots, \mathsf{A}_{i-1}, \mathsf{A}_i, \mathsf{A}_{i+1}, \ldots, \mathsf{A}_{j-1}, \mathsf{A}_j, \mathsf{A}_{j+1}, \ldots, \mathsf{A}_m)$$

and

$$\mathbf{M}' = (\mathsf{A}_1, \ldots, \mathsf{A}_{i-1}, \mathsf{A}_i, \mathsf{A}_{i+1}, \ldots, \mathsf{A}_{j-1}, \mathsf{A}_j + q\mathsf{A}_i, \mathsf{A}_{j+1}, \ldots, \mathsf{A}_m).$$

The EUCLIDean algorithm guarantees that this transformation can express the more general second transformation of definition 3.1.2. But this is not possible in case of non–EUCLIDean rings. The proof of theorem 4.5.6 requires the more expressive transformation, and it seems very likely that a proof of this theorem would not be possible without it.

Definition 3.1.3:

Two matrices over $\mathcal{K}$ are **matrix multiplication equivalent**, $\mathbf{M}_1 \approx_{\mathrm{mat}} \mathbf{M}_2$, iff there exist matrices $\mathbf{T}$ and $\mathbf{U}$ over $\mathcal{D}$ such that $\mathbf{M}_2 = \mathbf{M}_1\mathbf{T}$ and $\mathbf{M}_1 = \mathbf{M}_2\mathbf{U}$.

Remarks:

(1) This definition includes matrices of different sizes. Obviously $\mathbf{T}$ and $\mathbf{U}$ have to have the correct dimensions for the equalities to be true.

(2) Another notion of the matrix multiplication definition requires the unimodularity of $\mathbf{T}$ and does not require a matrix $\mathbf{U}$. But this notion excludes matrices of different sizes therefore this definition would not be as general.

(3) The definitions here are formulated as *column* equivalence. The analogue *row* equivalence can be defined either directly (with a few modifications of the column equivalence definitions) or based on the column equivalence definitions: Two matrices are module resp. transformation resp. matrix multiplication **row equivalent** iff their transposed matrices are module resp. transformation resp. matrix multiplication **column equivalent**.

(4) There is another notion of matrix equivalence. Again three different notions of equivalence can be defined. The module equivalence requires only that the represented modules are isomorphic but not equal. The transformation equivalence allows column *and* row transformations. The matrix multiplication demands the equalities: $\mathbf{M}_2 = \mathbf{T}_1\mathbf{M}_1\mathbf{T}_2$ and $\mathbf{M}_1 = \mathbf{U}_1\mathbf{M}_2\mathbf{U}_2$. The equivalence of those three equivalence relations for EUCLIDean domains leads to the SMITH normal form. It is not important for this thesis, so the details will not be given here.

3.2 Equivalence of the definitions

3.2.1 $\approx_{\mathrm{trafo}} \Longrightarrow \approx_{\mathrm{mod}}$

Lemma 3.2.1:

If two matrices over $\mathcal{K}$ are transformation equivalent, then they are module equivalent.

Proof. We have to show that an application of a transformation of any type does not change the generated module. This is trivial for the transformation of types 4 and 5 because the zero module is the neutral element of the operation "+" in the module of $\mathcal{D}$–modules. It is trivial for transformation of type 1 because "+" is commutative. It is trivial for transformation of type 3 because, for any $\mathcal{D}$–unit ϵ and any $\mathcal{D}$–module M, we have $\epsilon M = M$.

For transformations of type 2 (with the notation of definition 3.1.2), with the coefficients $c_1, c_2, c_3, c_4 \in \mathcal{D}$ and with $\mathsf{A}_i, \mathsf{A}_j \in \mathcal{K}^n$ for $i, j \in \mathbb{N}_m$ we have

$$\mathcal{D}(c_1\mathsf{A}_i + c_2\mathsf{A}_j) \subset \mathcal{D}c_1\mathsf{A}_i + \mathcal{D}c_2\mathsf{A}_j \subset \mathcal{D}\mathsf{A}_i + \mathcal{D}\mathsf{A}_j$$

and

$$\mathcal{D}(c_3\mathsf{A}_i + c_4\mathsf{A}_j) \subset \mathcal{D}c_3\mathsf{A}_i + \mathcal{D}c_4\mathsf{A}_j \subset \mathcal{D}\mathsf{A}_i + \mathcal{D}\mathsf{A}_j.$$

On the other hand, let $\mathsf{B}_i = c_1\mathsf{A}_i + c_2\mathsf{A}_j$ and $\mathsf{B}_j = c_3\mathsf{A}_i + c_4\mathsf{A}_j$. Since $c_1c_4 - c_2c_3 = 1$, we have $\mathsf{A}_i = c_4\mathsf{B}_i - c_2\mathsf{B}_j$ and $\mathsf{A}_j = c_1\mathsf{B}_j - c_3\mathsf{B}_i$. We conclude

$$\mathcal{D}\mathsf{A}_i = \mathcal{D}(c_4\mathsf{B}_i - c_2\mathsf{B}_j) \subset \mathcal{D}c_4\mathsf{B}_i + \mathcal{D}c_2\mathsf{B}_j \subset \mathcal{D}\mathsf{B}_i + \mathcal{D}\mathsf{B}_j$$

and

$$\mathcal{D}\mathsf{A}_j = \mathcal{D}(c_1\mathsf{B}_j - c_3\mathsf{B}_i) \subset \mathcal{D}c_1\mathsf{B}_j + \mathcal{D}c_3\mathsf{B}_i \subset \mathcal{D}\mathsf{B}_i + \mathcal{D}\mathsf{B}_j.$$

Therefore

$$\mathcal{D}\mathsf{B}_i + \mathcal{D}\mathsf{B}_j = \mathcal{D}\mathsf{A}_i + \mathcal{D}\mathsf{A}_j,$$

which completes the proof. □

3.2.2 $\approx_{\text{trafo}} \Longrightarrow \approx_{\text{mat}}$

Lemma 3.2.2:
If two matrices over $\mathcal{K}$ are transformation equivalent then they are matrix multiplication equivalent.

Proof. Let $\mathbf{M} \in \mathcal{K}^{n\times m}$. We relate every transformation type to an **elementary matrix** with the intention that the effect of the multiplication with the related matrix is identical to the transformation itself:

- The permutation of two columns i and j relates to an identity matrix of degree m except the diagonal elements on positions (i,i) and (j,j), which are zero, and the elements on positions (i,j) and (j,i), which are one:

$$\begin{array}{cccccccc}
 & & & i & & j & & \\
 & 1 & & & & & & 0 \\
 & & \ddots & & & & & \\
i & & & 0 & & 1 & & \\
 & & & & \ddots & & & \\
j & & & 1 & & 0 & & \\
 & & & & & & \ddots & \\
 & 0 & & & & & & 1
\end{array}.$$

  It is easy to see that multiplication with this matrix permutates A_i and A_j.

- The scaled transformation of two columns relates to the identity matrix of degree m except for the four positions $(i,i),(i,j),(j,i),(j,j)$, containing the scalar factors c_1,c_2,c_3,c_4:

$$\begin{array}{cc}
 & \begin{array}{ccccccc} & & i & & j & & \end{array} \\
\begin{array}{c} \\ \\ i \\ \\ j \\ \\ \\ \end{array} & \left(\begin{array}{ccccccc}
1 & & & & & & 0 \\
 & \ddots & & & & & \\
 & & c_1 & & c_2 & & \\
 & & & \ddots & & & \\
 & & c_3 & & c_4 & & \\
 & & & & & \ddots & \\
0 & & & & & & 1
\end{array}\right)
\end{array}.$$

Again it is easy to see that multiplication accomplishes the mentioned transformation.

- The multiplication of the i-th column with the unit ϵ relates to the identity matrix of degree m except that the i-th diagonal element is ϵ:

$$\begin{array}{cc}
 & \begin{array}{ccccc} & & i & & \end{array} \\
\begin{array}{c} \\ \\ i \\ \\ \\ \end{array} & \left(\begin{array}{ccccc}
1 & & & & 0 \\
 & \ddots & & & \\
 & & \epsilon & & \\
 & & & \ddots & \\
0 & & & & 1
\end{array}\right)
\end{array}.$$

The multiplication with this matrix multiplies the i-th column with ϵ.

- The insertion of a zero column at position i relates to an identity matrix with an inserted zero column at position i. It is a matrix in $\mathcal{D}^{m\times m+1}$:

$$\begin{array}{cc}
 & \begin{array}{ccccccc} & & i-1 & i & i+1 & & \end{array} \\
\begin{array}{c} \\ \\ i-1 \\ i \\ \\ \\ \end{array} & \left(\begin{array}{ccccccc}
1 & & & & & & 0 \\
 & \ddots & & & & & \\
 & & 1 & 0 & 0 & & \\
 & & 0 & 0 & 1 & & \\
 & & & & & \ddots & \\
0 & & & & & & 1
\end{array}\right)
\end{array}.$$

- The deletion of a zero column i relates to an identity matrix where the i-th column is removed. It is a matrix in $\mathcal{D}^{m\times m-1}$:

$$\begin{array}{cc}
 & \begin{array}{cccccc} & & i-1 & i & & \end{array} \\
\begin{array}{c} \\ \\ i-1 \\ i \\ i+1 \\ \\ \\ \end{array} & \left(\begin{array}{cccccc}
1 & & & & & 0 \\
 & \ddots & & & & \\
 & & 1 & 0 & & \\
 & & 0 & 0 & & \\
 & & 0 & 1 & & \\
 & & & & \ddots & \\
0 & & & & & 1
\end{array}\right)
\end{array}.$$

Because of the associativity of matrix multiplication, for any number of elementary matrices $\mathbf{T}_i$ we have

$$(\dots(\mathbf{M}_1\mathbf{T}_1)\cdot\dots\cdot\mathbf{T}_z) = \mathbf{M}_1(\mathbf{T}_1\cdot\dots\cdot\mathbf{T}_z).$$

It is easy to see that for any of the above transformation matrices there is another one which reverts its effect. The product of those inverse transformation matrices results in $\mathbf{U}$.

Note that the elementary matrix relating to the "delete zero column" transformation may not be applied in a general situation (like the other transformation matrices) but only if there is a zero column at the correct position. In this proof, however, we start with a valid transformation and this relates to a valid elementary matrix.

□

3.2.3 $\approx_{\text{mat}} \Longleftrightarrow \approx_{\text{mod}}$

Lemma 3.2.3:

Two matrices over $\mathcal{K}$ are matrix equivalent if and only if they are module equivalent.

Proof. $\approx_{\text{mat}} \Longrightarrow \approx_{\text{mod}}$

Let $\mathbf{M}_1 = (\mathsf{A}_1,\dots,\mathsf{A}_{m_1}) \in \mathcal{K}^{n\times m_1}$ and $\mathbf{M}_2 = (\mathsf{B}_1,\dots,\mathsf{B}_{m_2}) \in \mathcal{K}^{n\times m_2}$. By assumption, for any $i \in \mathbb{N}_{m_2}$ there are $t_{ij} \in \mathcal{D}$ such that $\mathsf{B}_i = \sum_{j=1}^{m_1} t_{ij}\mathsf{A}_j$, hence $\mathsf{B}_i \in \mathcal{D}\mathsf{A}_1 + \dots \mathcal{D}\mathsf{A}_{m_1}$ and

$$\mathcal{D}\mathsf{B}_1 + \cdots + \mathcal{D}\mathsf{B}_{m_2} \subset \mathcal{D}\mathsf{A}_1 + \cdots + \mathcal{D}\mathsf{A}_{m_1}.$$

The same conclusion can be drawn for $\mathsf{A}_i \in \mathcal{D}\mathsf{B}_1 + \dots \mathcal{D}\mathsf{B}_{m_2}$. Hence,

$$\mathcal{D}\mathsf{B}_1 + \cdots + \mathcal{D}\mathsf{B}_{m_2} = \mathcal{D}\mathsf{A}_1 + \cdots + \mathcal{D}\mathsf{A}_{m_1}.$$

$\approx_{\text{mod}} \Longrightarrow \approx_{\text{mat}}$

Let $\mathbf{M}_1 = (\mathsf{A}_1,\dots,\mathsf{A}_{m_1})$ and $\mathbf{M}_2 = (\mathsf{B}_1,\dots,\mathsf{B}_{m_2})$. By assumption,

$$\mathcal{D}\mathsf{A}_1 + \cdots + \mathcal{D}\mathsf{A}_{m_1} = \mathcal{D}\mathsf{B}_1 + \cdots + \mathcal{D}\mathsf{B}_{m_2}.$$

Therefore for any $i \in \mathbb{N}_{m_2}$,

$$\mathsf{B}_i \in \mathcal{D}\mathsf{A}_1 + \cdots + \mathcal{D}\mathsf{A}_{m_1};$$

hence, for any $j \in \mathbb{N}_{m_1}, i \in \mathbb{N}_{m_2}$ there exist $t_{ij} \in \mathcal{D}$ such that

$$\mathsf{B}_i = \sum_{j=1}^{m_1} t_{ij}\mathsf{A}_j, \text{ where } \quad i \in \mathbb{N}_{m_2}$$

and $\mathbf{M}_2 = \mathbf{M}_1\mathbf{T}$ with $\mathbf{T} = (t_{ij})_{i\in\mathbb{N}_{m_2}, j\in\mathbb{N}_{m_1}}$. On the other hand, we have for any $i \in \mathbb{N}_{m_1}$:

$$\mathsf{A}_i \in \mathcal{D}\mathsf{B}_1 + \cdots + \mathcal{D}\mathsf{B}_{m_2};$$

hence, there exist $u_{ij} \in \mathcal{D}$ for any $i \in \mathbb{N}_{m_1}, j \in \mathbb{N}_{m_2}$ such that

$$\mathsf{A}_i = \sum_{j=1}^{m_2} u_{ij}\mathsf{B}_j, \text{ where } \quad i \in \mathbb{N}_{m_1}.$$

With $\mathbf{U} = (u_{ij})_{i\in\mathbb{N}_{m_1}, j\in\mathbb{N}_{m_2}}$, we have $\mathbf{M}_1 = \mathbf{M}_2\mathbf{U}$.

□

3.2.4 $\approx_{\text{mat}} \Longrightarrow \approx_{\text{trafo}}$

This is by far the most difficult part of the equivalence proof which in general fails if the ring does not have special properties. An important aid for the equivalence proof and many other things is the following definition.

Definition 3.2.4 (Hermite normal form)**:**
Let $\mathcal{D}$ be an integral domain and $\mathcal{K}$ its quotient field. Let a reduce function according to definition 2.2.11 be fixed. Let $S \subset \mathcal{K}$ be a set of representatives for the classes of $\mathcal{K}^{\times}/\mathcal{D}^{*}$ where $\mathcal{K}^{\times}$ denotes the multiplicative group of $\mathcal{K}$ and $\mathcal{D}^{*}$ the group of multiplicative units of $\mathcal{D}$.

A matrix $\boldsymbol{M} = (a_{ij})_{i\in\mathbb{N}_n, j\in\mathbb{N}_m} \in \mathcal{K}^{n\times m}$ is in **Hermite normal form** iff there exists a strictly increasing map $\rho : \mathbb{N}_m \to \mathbb{N}_n$ with the properties

- $\forall i \in \mathbb{N}_n, j \in \mathbb{N}_m,\ i > \rho(j) \implies a_{ij} = 0$ (diagonal form)
- $\forall j, k \in \mathbb{N}_m,\ k > \rho(j) \implies a_{\rho(j)k}$ is reduced (see definition 2.2.11) modulo the ideal generated by $a_{\rho(j)j}$.
- $\forall j \in \mathbb{N}_m$, the (diagonal) element $a_{\rho(j)j} \in \mathcal{S}$.

This definition is a generalization of the well–known Hermite normal form of matrices over principal ideal rings, as for instance is given in [PZ93, p. 179]. The main prerequisite for the generalization is the introduction of reduce functions in section 2.2.

We can prove that every module equivalence class contains *at most* one matrix in Hermite normal form. (It is a special case of the proposition 4.4.4.)

The problem is to prove that every module equivalence class contains *at least* one matrix in Hermite normal form. If it is possible to show that for every matrix there exists a chain of elementary transformations which results in a matrix in Hermite normal form, we would have shown two things:

- Every module equivalence class contains exactly one matrix in Hermite normal form.
- Two matrices which are module equivalent are also transformation equivalent.

An important approach is the Gauss–Jordan algorithm. Most importantly, this algorithm only works with Euclidean rings. It can be modified to work for principal ideal domains, using the following lemma.

Lemma 3.2.5:
In a principal ideal domain the gcd of two elements always exists.

Proof. Let $\mathcal{D}$ be a principal ideal domain and $a, b \in \mathcal{D}$. Consider the ideal $a\mathcal{D} + b\mathcal{D}$. Every $\mathcal{D}$–ideal is principal, therefore $a\mathcal{D} + b\mathcal{D}$ has a principal generator $c \in \mathcal{D}$. $a\mathcal{D} \subset c\mathcal{D}$, hence $a \in c\mathcal{D}$ and $c|a$, likewise $c|b$. A $d \in \mathcal{D}$ dividing both a and d generates an ideal $d\mathcal{D} \supset a\mathcal{D} + b\mathcal{D} = c\mathcal{D}$ which implies $d|c$. □

If there is an algorithm which finds the principal generator for any given ideal then the proof of the lemma is constructive. There are Gauss–Jordan algorithm versions which use the gcd instead of remainder division as in [KB79], [CC82], or [Hop94]. In algebraic number rings, finding a principal generator of an ideal is generally not an easy task; for good methods see [Hes96, pp. 75–77].

[PZ93, p. 179] proves the existence of the Hermite normal form in principal ideal domains with a different constructive method.

The results of the next chapter will provide still another proof of theorem 4.5.6, which is based on the more general pseudomatrices. This proof is constructive and uses methods for pseudomatrices.

3.2.5 Summary

The proven statements about the three matrix equivalence definitions are summarized in the following picture:

$$\begin{tikzcd}
& \approx_{\text{trafo}} \arrow[dl, shift left, "\text{always}"] \arrow[dr, "\text{always}"] & \\
\approx_{\text{mat}} \arrow[ur, shift left, "\text{only for principal ideal domains}"] \arrow[rr, shift right, "\text{always}"'] & & \approx_{\text{mod}} \arrow[ll, shift right, "\text{always}"']
\end{tikzcd}$$

The question is still open if the equivalence of the matrix equivalence definitions holds for more general rings. One hypothesis is that this is the case for Dedekind domains but not for integral domains although there is neither proof nor disproof so far.

Chapter 4

The theory of pseudomatrices

Let $\mathcal{D}$ be an integral domain (a commutative unital ring without nontrivial zero divisors) and $\mathcal{K}$ its quotient field. Finitely generated $\mathcal{D}$–modules in $\mathcal{K}^n$ (where $n \in \mathbb{N}$) can be represented by matrices over $\mathcal{D}$ with n rows. This was the subject of the last chapter. The equivalence of the three equivalence relations for matrices and the existence of the normal form can only be shown if $\mathcal{D}$ is a principal ideal domain.

This chapter generalizes the concept of $\mathcal{D}$–matrices to $\mathcal{D}$–pseudomatrices. The definitions of the three matrix equivalence relations for pseudomatrices and the normal form are an analogue of those for matrices. The advantage of the generalization is that equivalence of the three equivalence relations and the existence of the normal form can not only be shown for principal ideal domains, but for the more general Dedekind rings. Although the complete theoretical solution can only be obtained for Dedekind rings, definitions and propositions in the more general context of integral domains will be given, where possible.

The idea of pseudomatrices can be found in [O'M63, §81:3]. This theorem proves the existence of a finite sum of products of an ideal and an module as a replacement for integral bases which do not exist in general in the case of Dedekind rings. This sum was named pseudobasis in [Coh96] by Henri Cohen. Matrices represent bases of finitely generated modules — analogously pseudomatrices represent pseudobases.

4.1 The definition of pseudomatrices

Definition 4.1.1:

Let $\mathcal{D}$ be an integral domain and $\mathcal{K} = Q(\mathcal{D})$ its quotient field. Let $n, m \in \mathbb{N}$ and

$$\mathbf{A} = (\mathsf{A}_1, \ldots, \mathsf{A}_m) = \begin{pmatrix} a_{11} & \ldots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nm} \end{pmatrix}$$

be an $(n \times m)$–matrix over $\mathcal{K}$ with column vectors $\mathsf{A}_1, \ldots, \mathsf{A}_m$ in $\mathcal{K}^n$.

Let $\mathfrak{a}_1, \ldots, \mathfrak{a}_m$ be m fractional $\mathcal{D}$–ideals[1]. Then the scheme formed by

$$\mathfrak{M} = \left[\begin{array}{ccc} \mathfrak{a}_1 & \ldots & \mathfrak{a}_m \\ & \mathbf{A} & \end{array} \right] = \left[\begin{array}{ccc} \mathfrak{a}_1 & \ldots & \mathfrak{a}_m \\ \mathsf{A}_1 & \ldots & \mathsf{A}_m \end{array} \right] = \left[\begin{array}{c} \begin{array}{ccc} \mathfrak{a}_1 & \ldots & \mathfrak{a}_m \end{array} \\ \begin{pmatrix} a_{11} & \ldots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nm} \end{pmatrix} \end{array} \right]$$

is called a **pseudomatrix** over $\mathcal{D}$ with n rows and m columns.

$\mathfrak{a}_1, \ldots, \mathfrak{a}_m$ are called the **coefficient ideals** of $\mathfrak{M}$.

The notion of pseudomatrices was introduced in [Coh96, Th. 2.5]. Analogous to the definition of the operator Mod for matrices in definition 3.1.1 we define

Definition 4.1.2:

Let $\mathcal{D}$ be an integral domain and let

$$\mathfrak{M} = \left[\begin{array}{ccc} \mathfrak{a}_1 & \ldots & \mathfrak{a}_m \\ \mathsf{A}_1 & \ldots & \mathsf{A}_m \end{array} \right]$$

be a pseudomatrix over $\mathcal{D}$ with n rows. Then the $\mathcal{D}$–module

$$\mathrm{Mod}\,(\mathfrak{M}) = \sum_{i=1}^{m} \mathfrak{a}_i \mathsf{A}_i \subset \mathcal{K}^n$$

is called the **module generated by the pseudomatrix $\mathfrak{M}$**.

For every finitely generated $\mathcal{D}$–module in $\mathcal{K}^n$ there exists a pseudomatrix which represents it. Let $\mathsf{A}_1, \ldots, \mathsf{A}_m$ be the m generators of the module M. Then

$$\mathrm{Mod}\left(\left[\begin{array}{ccc} 1\mathcal{D} & \ldots & 1\mathcal{D} \\ \mathsf{A}_1 & \ldots & \mathsf{A}_m \end{array} \right] \right) = M.$$

($1\mathcal{D}$ is an abbreviation for the $\mathcal{D}$–ideal generated by 1.)

Definition 4.1.3:

Two pseudomatrices $\mathfrak{M}$ and $\mathfrak{N}$ over an integral domain $\mathcal{D}$ are called **module equivalent** (written $\mathfrak{M} \approx_{\mathrm{mod}} \mathfrak{N}$) iff $\mathrm{Mod}\,(\mathfrak{M}) = \mathrm{Mod}\,(\mathfrak{N})$.

It is easy to see that this an equivalence relation. A very important aim is to define and to construct *canonical representatives of the equivalence classes of the relation* $\approx_{\mathrm{mod}}$.

The set of matrices over $\mathcal{K}$ can be considered a subset of the set of pseudomatrices over $\mathcal{D}$ via the identification map $\mathbf{A} \mapsto \left[\begin{array}{ccc} 1\mathcal{D} & \ldots & 1\mathcal{D} \\ & \mathbf{A} & \end{array} \right]$, where $\mathbf{A}$ is a matrix over $\mathcal{K}$. This makes sense because the modules which are represented by $\mathbf{A}$ and $\mathfrak{M}$ are equal: $\mathrm{Mod}\,(\mathbf{A}) = \mathrm{Mod}\,(\mathfrak{M})$. Therefore the concept of the representation of $\mathcal{D}$–modules in $\mathcal{K}^n$ by pseudomatrices is a generalization of the concept of the representation by matrices.

[1] The zero ideal is not a fractional $\mathcal{D}$–ideal by convention of definition 1.0.1 .

Definition 4.1.4:

A pseudomatrix $\begin{bmatrix} \begin{array}{ccc} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \end{array} \\ \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix} \end{bmatrix}$ is called integral if for every $i \in \mathbb{N}_n$ and $j \in \mathbb{N}_m$ the ideal $a_{ij}\mathfrak{a}_j$ is an integral ideal.

Lemma 4.1.5:

A pseudomatrix $\mathfrak{M}$ is integral iff $\mathrm{Mod}\,(\mathfrak{M}) \subset \mathcal{D}^n$.

4.2 Transformations on pseudomatrices

Transformations on pseudomatrices are defined in analogy to transformations on matrices. Most importantly, an application of a transformation should not change the module generated by the pseudomatrix. This is proved in the subsequent proposition 4.2.3.

Definition 4.2.1:

Let $\mathfrak{M}$ and $\mathfrak{N}$ be two pseudomatrices of the same number n of rows. They are called **transformation equivalent** (written $\mathfrak{M} \approx_{\text{trafo}} \mathfrak{N}$) iff $\mathfrak{N}$ can be produced from $\mathfrak{M}$ by a finite number of applications of **elementary transformations** which are:

Swap:

Swaps the columns i and j of $\mathfrak{M}$:

$$\begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & A_i & \dots & A_j & \dots \end{bmatrix} \rightarrow \begin{bmatrix} \dots & \mathfrak{a}_j & \dots & \mathfrak{a}_i & \dots \\ \dots & A_j & \dots & A_i & \dots \end{bmatrix}.$$

Push factor:

This transformation involves only one column and its ideal. The nonzero factor $e \in \mathcal{K}$ is pushed from the ideal $\mathfrak{a}_i$ ($\mathfrak{a}_i$ is divided by e) to the column A_i (A_i is multiplied by e):

$$\begin{bmatrix} \dots & \mathfrak{a}_i & \dots \\ \dots & A_i & \dots \end{bmatrix} \rightarrow \begin{bmatrix} \dots & \frac{\mathfrak{a}_i}{e} & \dots \\ \dots & eA_i & \dots \end{bmatrix}.$$

Two scaled:

This transformation changes two columns without modifying the coefficient ideals:

$$\begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & A_i & \dots & A_j & \dots \end{bmatrix} \rightarrow \begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & B_i & \dots & B_j & \dots \end{bmatrix},$$

where $B_i, B_j \in \mathcal{K}^n$ and $c_1, c_2, c_3, c_4 \in \mathcal{K}$ satisfy

$$c_1, c_4 \in \mathcal{D}, \quad c_2\mathfrak{a}_i \subset \mathfrak{a}_j, \quad c_3\mathfrak{a}_j \subset \mathfrak{a}_i, \qquad \begin{vmatrix} c_1 & c_2 \\ c_3 & c_4 \end{vmatrix} = 1,$$

$$B_i = c_1A_i + c_2A_j, \quad B_j = c_3A_i + c_4A_j.$$

Collect:

This transformation collects the ideals of two columns of $\mathfrak{M}$:

$$\begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j & \dots \end{bmatrix} \to \begin{bmatrix} \dots & 1\mathcal{D} & \dots & \mathfrak{a}_i\mathfrak{a}_j & \dots \\ \dots & \mathsf{B}_i & \dots & \mathsf{B}_j & \dots \end{bmatrix},$$

where $\mathsf{B}_i, \mathsf{B}_j \in \mathcal{K}^n$ and $c_1, c_2, c_3, c_4 \in \mathcal{K}$ satisfy

$$c_1 \in \mathfrak{a}_i, \quad c_2 \in \mathfrak{a}_j, \quad c_3\mathfrak{a}_j \subset \mathcal{D}, \quad c_4\mathfrak{a}_i \subset \mathcal{D},$$
$$\begin{vmatrix} c_1 & c_2 \\ c_3 & c_4 \end{vmatrix} = 1, \quad \mathsf{B}_i = c_1\mathsf{A}_i + c_2\mathsf{A}_j, \quad \mathsf{B}_j = c_3\mathsf{A}_i + c_4\mathsf{A}_j.$$

Spread:

This is the inverse transformation of the Collect transformation. It may only be applied if the coefficient ideal of the i-th column is $\mathcal{D}$:

$$\begin{bmatrix} \dots & 1\mathcal{D} & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j & \dots \end{bmatrix} \to \begin{bmatrix} \dots & \mathfrak{b}_i & \dots & \mathfrak{b}_j & \dots \\ \dots & \mathsf{B}_i & \dots & \mathsf{B}_j & \dots \end{bmatrix},$$

where $\mathfrak{b}_i$ and $\mathfrak{b}_j$ are fractional $\mathcal{D}$–ideals, $\mathsf{B}_i, \mathsf{B}_j \in \mathcal{K}^n$, and $c_1, c_2, c_3, c_4 \in \mathcal{K}$ satisfy

$$\mathfrak{b}_i\mathfrak{b}_j = \mathfrak{a}_j, \quad c_1\mathfrak{b}_i \subset \mathcal{D}, \quad c_2 \in \mathfrak{b}_j, \quad c_3\mathfrak{b}_j \subset \mathcal{D}, \quad c_4 \in \mathfrak{b}_i,$$
$$\begin{vmatrix} c_1 & c_2 \\ c_3 & c_4 \end{vmatrix} = 1, \quad \mathsf{B}_i = c_1\mathsf{A}_i + c_2\mathsf{A}_j, \quad \mathsf{B}_j = c_3\mathsf{A}_i + c_4\mathsf{A}_j.$$

Insert zero column:

This transformation appends a zero column together with an arbitrary fractional ideal to $\mathfrak{M}$:

$$\begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \\ \mathsf{A}_1 & \dots & \mathsf{A}_m \end{bmatrix} \to \begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_m & \mathfrak{a} \\ \mathsf{A}_1 & \dots & \mathsf{A}_m & \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \end{bmatrix}$$

where $\mathfrak{a}$ is an arbitrary fractional $\mathcal{D}$–ideal.

Delete zero column:

This transformation deletes the last column of $\mathfrak{M}$, if it is a zero column, together with its (arbitrary) coefficient ideal:

$$\begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_m & \mathfrak{a} \\ \mathsf{A}_1 & \dots & \mathsf{A}_m & \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \end{bmatrix} \to \begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \\ \mathsf{A}_1 & \dots & \mathsf{A}_m \end{bmatrix}$$

for any fractional $\mathcal{D}$–ideal $\mathfrak{a}$. This transformation must not be applied if $\mathfrak{M}$ has only one column.

The actual definition of pseudomatrices forbids pseudomatrices with no columns which is just one possible convention. To deal with pseudomatrices whose matrix contains only zero entries (as counterpart of the zero module which is represented by it) must be artificial at some point — either in this definition of transformations or in all statements about pseudomatrices, where the pseudomatrix with no columns at all would have be dealt with as a special case.

Other forms of the definition of elementary transformations would have been possible. Here, the transformations are chosen for their simplicity, while still being expressive and reasonable, even for non–DEDEKIND rings.

Lemma 4.2.2:
Let $\mathcal{D}$ be an integral domain. For every transformation of pseudomatrices over $\mathcal{D}$, there exists an inverse transformation.

Proof. Obviously INSERT ZERO COLUMN and DELETE ZERO COLUMN are inverse to each other. The SWAP transformation is inverse to itself.

Let the TWO SCALED transformation be described as

$$\begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j & \dots \end{bmatrix} \to \begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{B}_i & \dots & \mathsf{B}_j & \dots \end{bmatrix},$$

where $\mathsf{B}_i, \mathsf{B}_j \in \mathcal{K}^n$ and $c_1, c_2, c_3, c_4 \in \mathcal{K}$ satisfy

$$c_1, c_4 \in \mathcal{D}, \quad c_2\mathfrak{a}_i \subset \mathfrak{a}_j, \quad c_3\mathfrak{a}_j \subset \mathfrak{a}_i, \qquad \begin{vmatrix} c_1 & c_2 \\ c_3 & c_4 \end{vmatrix} = 1,$$
$$\mathsf{B}_i = c_1\mathsf{A}_i + c_2\mathsf{A}_j, \quad \mathsf{B}_j = c_3\mathsf{A}_i + c_4\mathsf{A}_j.$$

The inverse transformation is a TWO SCALED transformation which is described as

$$\begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{B}_i & \dots & \mathsf{B}_j & \dots \end{bmatrix} \to \begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j & \dots \end{bmatrix},$$

where

$$\begin{matrix} d_1 = c_4, & d_2 = -c_2, \\ d_3 = -c_3, & d_4 = c_1, \end{matrix} \quad \begin{pmatrix} d_1 & d_2 \\ d_3 & d_4 \end{pmatrix} = \begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix}^{-1}, \quad \begin{matrix} \mathsf{A}_i = d_1\mathsf{B}_i + d_2\mathsf{B}_j, \\ \mathsf{A}_j = d_3\mathsf{B}_i + d_4\mathsf{B}_j. \end{matrix}$$

We can conclude

$$c_4 \in \mathcal{D} \implies d_1 \in \mathcal{D},$$
$$c_2\mathfrak{a}_i \subset \mathfrak{a}_j \implies d_2\mathfrak{a}_i \subset \mathfrak{a}_j,$$
$$c_3\mathfrak{a}_j \subset \mathfrak{a}_i \implies d_3\mathfrak{a}_j \subset \mathfrak{a}_i,$$
$$c_1 \in \mathcal{D} \implies d_4 \in \mathcal{D}.$$

The COLLECT transformation with the parameters c_1, c_2, c_3, c_4 is inverse to the SPREAD transformation with the parameters $c_4, -c_2, -c_3, c_1$. These parameters can be shown to satisfy the properties required by the SPREAD transformation analogously to the TWO SCALED transformation.

The SPREAD transformation is inverse to a COLLECT transformation in a similar way. □

Proposition 4.2.3:
Let $\mathfrak{M}$ and $\mathfrak{N}$ be two pseudomatrices over an integral domain $\mathcal{D}$ with the same number of rows. If they are transformation equivalent, then they are module equivalent:

$$\mathfrak{M} \approx_{\text{trafo}} \mathfrak{N} \implies \mathfrak{M} \approx_{\text{mod}} \mathfrak{N}.$$

Proof. We have to show that the transformations Swap, Push factor, Two scaled, Collect, Spread, Insert zero column, and Delete zero column of definition 4.2.1 do not change the represented module of a pseudomatrix $\mathfrak{M}$ with n rows.

It is trivial for the Swap and Push factor transformations.

Let A be the zero vector of length n, $\mathfrak{a}$ any fractional $\mathcal{D}$–ideal. Then $\mathfrak{a}\mathsf{A}$ is the zero module. Therefore the Insert zero column and Delete zero column do not change the represented module.

Let a Two scaled transformation be described by

$$\begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j & \dots \end{bmatrix} \to \begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{B}_i & \dots & \mathsf{B}_j & \dots \end{bmatrix},$$

where $\mathsf{B}_i, \mathsf{B}_j \in \mathcal{K}^n$ and $c_1, c_2, c_3, c_4 \in \mathcal{K}$ satisfy

$$c_1, c_4 \in \mathcal{D}, \quad c_2\mathfrak{a}_i \subset \mathfrak{a}_j, \quad c_3\mathfrak{a}_j \subset \mathfrak{a}_i, \qquad \begin{vmatrix} c_1 & c_2 \\ c_3 & c_4 \end{vmatrix} = 1,$$
$$\mathsf{B}_i = c_1\mathsf{A}_i + c_2\mathsf{A}_j, \quad \mathsf{B}_j = c_3\mathsf{A}_i + c_4\mathsf{A}_j.$$

We have to show that

$$\mathfrak{a}_i\mathsf{A}_i + \mathfrak{a}_j\mathsf{A}_j = \mathfrak{a}_i\mathsf{B}_i + \mathfrak{a}_j\mathsf{B}_j.$$

By lemma 4.2.2 it suffices to show that

$$\mathfrak{a}_i\mathsf{B}_i + \mathfrak{a}_j\mathsf{B}_j \subset \mathfrak{a}_i\mathsf{A}_i + \mathfrak{a}_j\mathsf{A}_j.$$

We can show

$$\mathfrak{a}_i\mathsf{B}_i = \mathfrak{a}_i(c_1\mathsf{A}_i + c_2\mathsf{A}_j) \subset \mathfrak{a}_ic_1\mathsf{A}_i + \mathfrak{a}_ic_2\mathsf{A}_j \subset \mathfrak{a}_i\mathsf{A}_i + \mathfrak{a}_j\mathsf{A}_j$$

and similarly $\mathfrak{a}_j\mathsf{B}_j \subset \mathfrak{a}_i\mathsf{A}_i + \mathfrak{a}_j\mathsf{A}_j$.

Let a Collect transformation be described by

$$\begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j & \dots \end{bmatrix} \to \begin{bmatrix} \dots & 1\mathcal{D} & \dots & \mathfrak{a}_i\mathfrak{a}_j & \dots \\ \dots & \mathsf{B}_i & \dots & \mathsf{B}_j & \dots \end{bmatrix},$$

where $\mathsf{B}_i, \mathsf{B}_j \in \mathcal{K}^n$ and $c_1, c_2, c_3, c_4 \in \mathcal{K}$ satisfy

$$c_1 \in \mathfrak{a}_i, \quad c_2 \in \mathfrak{a}_j, \quad c_3\mathfrak{a}_j \subset \mathcal{D}, \quad c_4\mathfrak{a}_i \subset \mathcal{D},$$
$$\begin{vmatrix} c_1 & c_2 \\ c_3 & c_4 \end{vmatrix} = 1, \quad \mathsf{B}_i = c_1\mathsf{A}_i + c_2\mathsf{A}_j, \quad \mathsf{B}_j = c_3\mathsf{A}_i + c_4\mathsf{A}_j.$$

We have to show that

$$\mathfrak{a}_i\mathsf{A}_i + \mathfrak{a}_j\mathsf{A}_j = \mathcal{D}\mathsf{B}_i + \mathfrak{a}_i\mathfrak{a}_j\mathsf{B}_j.$$

By lemma 4.2.2 it suffices to show that

$$\mathcal{D}\mathsf{B}_i + \mathfrak{a}_i\mathfrak{a}_j\mathsf{B}_j \subset \mathfrak{a}_i\mathsf{A}_i + \mathfrak{a}_j\mathsf{A}_j.$$

We have

$$\begin{aligned} &c_1 \in \mathfrak{a}_i \implies c_1\mathsf{A}_i \in \mathfrak{a}_i\mathsf{A}_i \\ &c_2 \in \mathfrak{a}_j \implies c_2\mathsf{A}_j \in \mathfrak{a}_j\mathsf{A}_j \\ &\implies \mathcal{D}\mathsf{B}_i = \mathcal{D}(c_1\mathsf{A}_i + c_2\mathsf{A}_j) \subset \mathfrak{a}_i\mathsf{A}_i + \mathfrak{a}_j\mathsf{A}_j \end{aligned}$$

and

$$\begin{aligned} &c_3\mathfrak{a}_j \in \mathcal{D} \implies c_3\mathfrak{a}_i\mathfrak{a}_j\mathsf{A}_i \in \mathfrak{a}_i\mathsf{A}_i \\ &c_4\mathfrak{a}_i \in \mathcal{D} \implies c_4\mathfrak{a}_i\mathfrak{a}_j\mathsf{A}_j \in \mathfrak{a}_j\mathsf{A}_j \\ &\implies \mathfrak{a}_i\mathfrak{a}_j\mathsf{B}_i = \mathfrak{a}_i\mathfrak{a}_j(c_3\mathsf{A}_i + c_4\mathsf{A}_j) \subset \mathfrak{a}_i\mathsf{A}_i + \mathfrak{a}_j\mathsf{A}_j. \end{aligned}$$

which proves the claim for the COLLECT transformation.

Likewise, it can be proved that the SPREAD transformation does not change the represented module. □

4.2.1 Existence of parameters for the transformations

Let $\mathcal{D}$ be a DEDEKIND ring. We ask the question whether there exist parameters for every elementary transformation type of definition 4.2.1 such that the transformation may be applied.

Obviously, INSERT ZERO COLUMN can always be applied and DELETE ZERO COLUMN only if the last column is indeed zero. It is easy to see that parameters are quite arbitrary for the transformations PUSH FACTOR, SWAP, TWO SCALED.

The SPREAD transformation cannot always be applied because one of the coefficient ideals of $\mathfrak{M}$ must be $\mathcal{D}$. From the proposition 4.2.4 below and the existence of the inverse transformation in lemma 4.2.2 it follows that if one coefficient ideal of $\mathfrak{M}$ is indeed $1\mathcal{D}$, parameters can be found to apply the SPREAD transformation. This leaves us with the task of showing the following proposition:

Proposition 4.2.4:

Let $\mathcal{D}$ be a DEDEKIND ring. For every two fractional $\mathcal{D}$–ideals $\mathfrak{a}_i$ and $\mathfrak{a}_j$, it is possible to find $c_1 \in \mathfrak{a}_i$, $c_2 \in \mathfrak{a}_j$, $c_3 \in \mathfrak{a}_j^{-1}$, and $c_4 \in \mathfrak{a}_i^{-1}$ such that

$$\begin{vmatrix} c_1 & c_2 \\ c_3 & c_4 \end{vmatrix} = 1.$$

These coefficients satisfy the requirements of a COLLECT transformation of definition 4.2.1.

If $\mathcal{D}$ is a maximal order of a number field, algorithm 1.8.5 gives a constructive proof.

If $\mathcal{D}$ is not an algebraic number ring, proposition 1.8.4 proves the existence.

4.2.2 Special transformations

There are other important transformations besides the elementary transformations.

General Two Columns:

This transformation changes two columns $\begin{bmatrix}\mathfrak{a}_i\\ \mathsf{A}_i\end{bmatrix}$ and $\begin{bmatrix}\mathfrak{a}_j\\ \mathsf{A}_j\end{bmatrix}$ of $\mathfrak{M}$ while the rest of the matrix stays constant:

$$\begin{bmatrix}\dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots\\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j & \dots\end{bmatrix} \to \begin{bmatrix}\dots & \mathfrak{b}_i & \dots & \mathfrak{b}_j & \dots\\ \dots & \mathsf{B}_i & \dots & \mathsf{B}_j & \dots\end{bmatrix},$$

where $\mathfrak{a}_i$ and $\mathfrak{a}_j$ must be invertible and where $\mathfrak{b}_i$ and $\mathfrak{b}_j$ are fractional $\mathcal{D}$–ideals, $\mathsf{B}_i, \mathsf{B}_j \in \mathcal{K}^n$, and $c_1, c_2, c_3, c_4 \in \mathcal{K}$ satisfy

$$\begin{aligned} &c_1\mathfrak{b}_i \subset \mathfrak{a}_i, \quad c_2\mathfrak{b}_i \subset \mathfrak{a}_j, \quad c_3\mathfrak{b}_j \subset \mathfrak{a}_i, \quad c_4\mathfrak{b}_j \subset \mathfrak{a}_j,\\ &e := \begin{vmatrix}c_1 & c_2\\ c_3 & c_4\end{vmatrix} = c_1c_4 - c_2c_3 \neq 0, \quad \mathfrak{a}_i\mathfrak{a}_j = e\mathfrak{b}_i\mathfrak{b}_j,\\ &\mathsf{B}_i = c_1\mathsf{A}_i + c_2\mathsf{A}_j, \quad \mathsf{B}_j = c_3\mathsf{A}_i + c_4\mathsf{A}_j.\end{aligned}$$

Unimodular:

Special case of General two columns where $e = 1$. It is otherwise identical.

The Swap, Push factor, Collect, Spread, Unimodular, and Two scaled transformations are all special cases of the General two columns transformation.

The effect of a General two columns transformation can also be obtained by an application of a Unimodular and a Push factor transformation.

The question is if the General two columns transformation can be based on the elementary transformations. It can be answered positively for Dedekind rings:

Proposition 4.2.5:

Let $\mathcal{D}$ be a Dedekind ring. Any one application of the Unimodular transformation can be reduced to one Collect, one Spread, and one Two scaled transformation application in this order.

Proof. Since $\mathcal{D}$ is a Dedekind ring every ideal is invertible. Let the Unimodular transformation be described by

$$\begin{pmatrix}\mathsf{B}_i\\ \mathsf{B}_j\end{pmatrix} = \begin{pmatrix}c_1 & c_2\\ c_3 & c_4\end{pmatrix}\begin{pmatrix}\mathsf{A}_i\\ \mathsf{A}_j\end{pmatrix}, \quad \begin{vmatrix}c_1 & c_2\\ c_3 & c_4\end{vmatrix} = 1, \quad \mathfrak{a}_i\mathfrak{a}_j = \mathfrak{b}_i\mathfrak{b}_j,$$

$$c_1 \in \mathfrak{a}_i\mathfrak{b}_i^{-1}, \quad c_2 \in \mathfrak{a}_j\mathfrak{b}_i^{-1}, \quad c_3 \in \mathfrak{a}_i\mathfrak{b}_j^{-1}, \quad c_4 \in \mathfrak{a}_j\mathfrak{b}_j^{-1}.$$

From proposition 4.2.4, it is clear that there are d_1, d_2, d_3, and d_4 such that

$$\begin{vmatrix}d_1 & d_2\\ d_3 & d_4\end{vmatrix} = 1, \quad d_1 \in \mathfrak{a}_i, \quad d_2 \in \mathfrak{a}_j, \quad d_3 \in \mathfrak{a}_j^{-1}, \quad d_4 \in \mathfrak{a}_i^{-1}.$$

This transforms

$$\begin{bmatrix}\dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots\\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j & \dots\end{bmatrix} \Longrightarrow \begin{bmatrix}\dots & \mathcal{D} & \dots & \mathfrak{a}_i\mathfrak{a}_j & \dots\\ \dots & \mathsf{C}_i & \dots & \mathsf{C}_j & \dots\end{bmatrix},$$

where $\begin{pmatrix}\mathsf{C}_i\\ \mathsf{C}_j\end{pmatrix} = \begin{pmatrix}d_1 & d_2\\ d_3 & d_4\end{pmatrix}\begin{pmatrix}\mathsf{A}_i\\ \mathsf{A}_j\end{pmatrix}.$

As an easy deduction of proposition 4.2.4 (because unimodular matrices can be inverted), it is clear that there exists a Spread transformation with

$$\begin{vmatrix} e_1 & e_2 \\ e_3 & e_4 \end{vmatrix} = 1, \quad e_1 \in \mathfrak{b}_i^{-1}, \quad e_2 \in \mathfrak{b}_j, \quad e_3 \in \mathfrak{a}_j^{-1}, \quad e_4 \in \mathfrak{a}_i$$

which transforms

$$\begin{bmatrix} \dots & \mathcal{D} & \dots & \mathfrak{a}_i\mathfrak{a}_j & \dots \\ \dots & \mathsf{C}_i & \dots & \mathsf{C}_j & \dots \end{bmatrix} \Longrightarrow \begin{bmatrix} \dots & \mathfrak{b}_i & \dots & \mathfrak{b}_j & \dots \\ \dots & \mathsf{D}_i & \dots & \mathsf{D}_j & \dots \end{bmatrix},$$

$$\text{where} \quad \begin{pmatrix} \mathsf{D}_i \\ \mathsf{D}_j \end{pmatrix} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} \begin{pmatrix} \mathsf{C}_i \\ \mathsf{C}_j \end{pmatrix}.$$

These two transformations applied together form a transformation

$$\begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j & \dots \end{bmatrix} \Longrightarrow \begin{bmatrix} \dots & \mathfrak{b}_i & \dots & \mathfrak{b}_j & \dots \\ \dots & \mathsf{D}_i & \dots & \mathsf{D}_j & \dots \end{bmatrix},$$

$$\text{where} \quad \begin{pmatrix} \mathsf{D}_i \\ \mathsf{D}_j \end{pmatrix} = \begin{pmatrix} f_1 & f_2 \\ f_3 & f_4 \end{pmatrix} \begin{pmatrix} \mathsf{A}_i \\ \mathsf{A}_j \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} f_1 & f_2 \\ f_3 & f_4 \end{pmatrix} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} \begin{pmatrix} d_1 & d_2 \\ d_3 & d_4 \end{pmatrix}.$$

For this transformation we can check

$$\begin{vmatrix} f_1 & f_2 \\ f_3 & f_4 \end{vmatrix} = 1, \quad \mathfrak{a}_i\mathfrak{a}_j = \mathfrak{b}_i\mathfrak{b}_j,$$

$$f_1 \in \mathfrak{a}_i\mathfrak{b}_i^{-1}, \quad f_2 \in \mathfrak{a}_j\mathfrak{b}_i^{-1}, \quad f_3 \in \mathfrak{a}_i\mathfrak{b}_j^{-1}, \quad f_4 \in \mathfrak{a}_j\mathfrak{b}_j^{-1}.$$

The coefficients

$$\begin{pmatrix} g_1 & g_2 \\ g_3 & g_4 \end{pmatrix} = \begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} \begin{pmatrix} f_4 & -f_2 \\ -f_3 & f_1 \end{pmatrix}$$

satisfy the requirements for the Matrix Two Scaled transformation since

$$\begin{aligned} g_1 &= c_1 f_4 - c_2 f_3 \in \mathfrak{a}_j\mathfrak{b}_i^{-1}\mathfrak{a}_i\mathfrak{b}_j^{-1} = 1\mathcal{D} \\ g_2 &= -c_1 f_2 + c_2 f_1 \in \mathfrak{a}_i\mathfrak{a}_j\mathfrak{b}_i^{-2} = \mathfrak{b}_i^{-1}\mathfrak{b}_j \\ g_3 &= c_3 f_4 - c_4 f_3 \in \mathfrak{a}_i\mathfrak{a}_j\mathfrak{b}_j^{-2} = \mathfrak{b}_j^{-1}\mathfrak{b}_i \\ g_4 &= -c_3 f_2 - c_4 f_1 \in \mathfrak{a}_j\mathfrak{b}_i^{-1}\mathfrak{a}_i\mathfrak{b}_j^{-1} = 1\mathcal{D}. \end{aligned}$$

□

Another transformation is the following:

Add q-times:

This transformation adds the q–fold of A_i to A_j:

$$\begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j & \dots \end{bmatrix} \to \begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j + q\mathsf{A}_i & \dots \end{bmatrix},$$

where $q \in \mathcal{K}$ satisfies $q\mathfrak{a}_j \subset \mathfrak{a}_i$.

This transformation is a special case of the Two scaled transformation. The question is whether the Two scaled transformation can be seen as a series of other transformations. If $\mathcal{D}$ were a Euclidean ring, it would be possible to determine a number of Add q–times transformations in a process similar to the Euclidean algorithm which is equivalent to a Two scaled transformation. Since we are interested in non–Euclidean rings, the Two scaled transformation is used as an elementary transformation and the Add q–times transformation is no longer important.

4.2.3 Special problems for non–Dedekind rings

In the General two columns transformation in subsection 4.2.2 the ideals $\mathfrak{a}_i$ and $\mathfrak{a}_j$ requested to be invertible? The answer is that the transformation should at least qualify the statements of lemma 4.2.2 and proposition 4.2.3, i.e. an application of it should be reversible and should not change the module generated by the pseudomatrix.

We assume the inverse transformation to be a General two columns transformation:

$$\begin{bmatrix} \dots & \mathfrak{b}_i & \dots & \mathfrak{b}_j & \dots \\ \dots & \mathsf{B}_i & \dots & \mathsf{B}_j & \dots \end{bmatrix} \rightarrow \begin{bmatrix} \dots & \mathfrak{a}_i & \dots & \mathfrak{a}_j & \dots \\ \dots & \mathsf{A}_i & \dots & \mathsf{A}_j & \dots \end{bmatrix},$$

where

$$d_1 := \frac{c_4}{e}, \quad d_2 := -\frac{c_2}{e}, \quad d_3 := -\frac{c_3}{e}, \quad d_4 := \frac{c_1}{e}$$
$$\mathsf{A}_i = d_1\mathsf{B}_i + d_2\mathsf{B}_j, \quad \mathsf{A}_j = d_3\mathsf{B}_i + d_4\mathsf{B}_j.$$

We have

$$d_1d_4 - d_2d_3 = \frac{1}{e} \neq 0, \quad \text{and} \quad \mathfrak{a}_i\mathfrak{a}_j\frac{1}{e} = \mathfrak{b}_i\mathfrak{b}_j.$$

From $c_4\mathfrak{b}_j \subset \mathfrak{a}_j$ (hence $c_4\mathfrak{a}_i\mathfrak{b}_j \subset \mathfrak{a}_i\mathfrak{a}_j$) and $e\mathfrak{b}_i\mathfrak{b}_j = \mathfrak{a}_i\mathfrak{a}_j$, we conclude $c_4\mathfrak{a}_i\mathfrak{b}_j \subset e\mathfrak{b}_i\mathfrak{b}_j$. But unless the ideal $\mathfrak{b}_j$ is invertible, we cannot conclude $d_4\mathfrak{a}_i \subset \mathfrak{b}_i$, which we would need for a valid General two columns transformation.

We only need the following simple property:

Lemma 4.2.6:
Let $\mathcal{D}$ be an integral domain and $\mathfrak{a}_1$ and $\mathfrak{a}_2$ two fractional ideals. Then $\mathfrak{a} := \mathfrak{a}_1\mathfrak{a}_2$ is invertible iff $\mathfrak{a}_1$ and $\mathfrak{a}_2$ are invertible.

Proof. If $\mathfrak{a}_1$ and $\mathfrak{a}_2$ are invertible, then $\mathfrak{a}' := \mathfrak{a}_2^{-1}\mathfrak{a}_1^{-1}$ is the inverse of $\mathfrak{a}$.

Let $\mathfrak{a}^{-1}$ be the inverse of $\mathfrak{a}$. Then $\mathfrak{a}' := \mathfrak{a}^{-1}\mathfrak{a}_2$ is the inverse of $\mathfrak{a}_1$ since

$$\mathfrak{a}'\mathfrak{a}_1 = \mathfrak{a}^{-1}\mathfrak{a}_2\mathfrak{a}_1 = 1\mathcal{D},$$

and likewise $\mathfrak{a}_2$ is invertible. □

Since we require the ideals $\mathfrak{a}_i$ and $\mathfrak{a}_j$ to be invertible, we have proved that the General two columns transformation is reversible and does not change the generated module.

4.3 Matrix multiplication equivalence definition

There is another important equivalence relation for pseudomatrices.

Definition 4.3.1:
Let $\mathcal{D}$ be an integral domain, let

$$\mathfrak{M} = \left[\begin{array}{ccc} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \\ & \mathbf{A} & \end{array} \right] \quad \text{and} \quad \mathfrak{N} = \left[\begin{array}{ccc} \mathfrak{b}_1 & \dots & \mathfrak{b}_l \\ & \mathbf{B} & \end{array} \right]$$

be two pseudomatrices with n rows. Then $\mathfrak{M}$ and $\mathfrak{N}$ are **matrix multiplication equivalent**, $\mathfrak{M} \approx_{\text{mat}} \mathfrak{N}$, iff there exists a matrix $\mathbf{T} = (t_{ij})_{i,j} \in \mathcal{K}^{m \times l}$ such that $\mathbf{B} = \mathbf{AT}$ and for every $i \in \mathbb{N}_m, j \in \mathbb{N}_l$ we have $t_{ij}\mathfrak{b}_j \subset \mathfrak{a}_i$, and there exists a matrix $\mathbf{U} = (u_{ij})_{i,j} \in \mathcal{K}^{l \times m}$ such that $\mathbf{A} = \mathbf{BU}$ and for every $i \in \mathbb{N}_l, j \in \mathbb{N}_m$ we have $u_{ij}\mathfrak{a}_j \subset \mathfrak{b}_i$.

Proposition 4.3.2:
Two pseudomatrices are module equivalent iff they are matrix equivalent.

Proof. $\Longrightarrow$
Assume

$$\left[\begin{array}{ccc} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \\ \mathsf{A}_1 & \dots & \mathsf{A}_m \end{array} \right] \approx_{\text{mod}} \left[\begin{array}{ccc} \mathfrak{b}_1 & \dots & \mathfrak{b}_l \\ \mathsf{B}_1 & \dots & \mathsf{B}_l \end{array} \right].$$

For every $j \in \mathbb{N}_l$ we have $\mathfrak{b}_j\mathsf{B}_j \subset \sum_{i=1}^m \mathfrak{a}_i\mathsf{A}_i$. Hence there exist $t_{ij} \in \mathcal{K}$ such that $t_{ij}\mathfrak{b}_j \subset \mathfrak{a}_i$ and $\mathsf{B}_j = \sum_{i=1}^m t_{ij}\mathsf{A}_i$. The matrix $\mathbf{T} = \left(t_{ij}\right)_{i\in\mathbb{N}_m, j\in\mathbb{N}_l}$ satisfies $\mathbf{B} = \mathbf{AT}$. On the other hand there exist $u_{ij} \in \mathcal{K}$ such that $u_{ij}\mathfrak{a}_j \subset \mathfrak{b}_i$ and for every $j \in \mathbb{N}_m$ we have $\mathsf{A}_j = \sum_{i=1}^l u_{ij}\mathsf{B}_i$. With $\mathbf{U} = \left(u_{ij}\right)_{i\in\mathbb{N}_l, j\in\mathbb{N}_m}$ we have $\mathbf{A} = \mathbf{BU}$ which completes one direction of the proof.

$\Longleftarrow$
Let the two pseudomatrices be matrix multiplication equivalent. Then we have $\mathbf{T} = (t_{ij})_{i\in\mathbb{N}_j} \in \mathcal{K}^{m\times l}$ such that $\mathbf{B} = \mathbf{AT}$ and for every $i \in \mathbb{N}_m, j \in \mathbb{N}_l$ we have $t_{ij}\mathfrak{b}_j \subset \mathfrak{a}_i$. It follows that

$$\mathfrak{b}_j\mathsf{B}_j = \sum_{i=1}^m t_{ij}\mathfrak{b}_j\mathsf{A}_i \in \sum_{i=1}^m \mathfrak{a}_i\mathsf{A}_i \quad \text{and} \quad \sum_{j=1}^l \mathfrak{b}_j\mathsf{B}_j \subset \sum_{i=1}^m \mathfrak{a}_i\mathsf{A}_i.$$

We have $\mathbf{U} = (u_{ij})_{i\in\mathbb{N}_l, j\in\mathbb{N}_m} \in \mathcal{K}^{l\times m}$ such that $\mathbf{A} = \mathbf{BU}$, and for every $i \in \mathbb{N}_l, j \in \mathbb{N}_m$ we have $u_{ij}\mathfrak{a}_j \subset \mathfrak{b}_i$. It follows that

$$\mathfrak{a}_j\mathsf{A}_j = \sum_{i=1}^l u_{ij}\mathfrak{a}_j\mathsf{B}_i \in \sum_{i=1}^l \mathfrak{b}_i\mathsf{B}_i \quad \text{and} \quad \sum_{j=1}^m \mathfrak{a}_j\mathsf{A}_j \subset \sum_{i=1}^l \mathfrak{b}_i\mathsf{B}_i,$$

which completes the proof.

□

4.4 Normal forms of pseudomatrices

We want to have a description of finitely generated modules by pseudomatrices. Since different pseudomatrices may represent the same module, we are interested in how to decide if two pseudomatrices are module equivalent. Transformations of pseudomatrices approach the algorithmic solution to this question.

In the last section, several questions were left open. If two pseudomatrices are module (or equivalently matrix multiplication) equivalent, are they also transformation equivalent? In other words, for any two pseudomatrices which represent the same finitely generated module, does there always exist a course of transformations which transforms one to the other? This question becomes easier if we fix one pseudomatrix in each class of pseudomatrices (regarding the module equivalence) — the normal form.

Once defined, the normal form must satisfy the following properties to be useful at all:

- Uniqueness: If two module equivalent pseudomatrices are in the normal form, then they are equal.
- Existence: For any given pseudomatrix, there exists a pseudomatrix in the normal form which is module equivalent to the given pseudomatrix.
- Constructability: For any given pseudomatrix, there exists a course of transformations which transforms it to a pseudomatrix in normal form.

If these properties hold, it follows immediately that two module equivalent pseudomatrices are also transformation equivalent.

4.4.1 The definition of the Cohen–Hermite normal form

The following definition is based on the paper [Coh96]. Henri Cohen calls it Hermite normal form there because it can be seen as a generalization of the Hermite normal form. However it includes conventions which are not the only ones possible. [BP91] use a different convention (although this is not explicitly stated there, it is a consequence of the given algorithm).

Definition 4.4.1:

A pseudomatrix over a Dedekind ring $\mathcal{D}$

$$\mathfrak{M} = \left[\begin{array}{c} \begin{matrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \end{matrix} \\ \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix} \end{array}\right]$$

is in Cohen–Hermite normal form (CHNF) regarding a fixed reduce function (as in definition 2.2.11) if there exists a strictly increasing map $\rho : \mathbb{N}_m \to \mathbb{N}_n$ (therefore $n \geq m \geq 1$) with the properties:

1. $\forall i \in \mathbb{N}_n, j \in \mathbb{N}_m,\ i > \rho(j) \implies a_{ij} = 0$ (triangular form);
2. $\forall i \in \mathbb{N}_m,\ a_{\rho(i)i} = 1$ (ones on the diagonal); and
3. $\forall i, j \in \mathbb{N}_m,\ j > i \implies a_{\rho(i)j}$ is reduced modulo the ideal $\frac{\mathfrak{a}_i}{\mathfrak{a}_j}$.

By abuse of language we will say “triangular form” if we want to point to the property $a_{lk} = 0$ for $k \in \mathbb{N}_m$ and $l \in \mathbb{N}_n \setminus \mathbb{N}_k$. We will call the entries $a_{\rho(k)k} = 1$ for $k \in \mathbb{N}_m$ “diagonal entries”.

The first two properties describe the form of the CHNF pseudomatrix as

$$\left[\begin{array}{c} \begin{array}{cccc} \mathfrak{a}_1 & \dots & \mathfrak{a}_{m-1} & \mathfrak{a}_m \end{array} \\ \begin{pmatrix} a_{11} & & a_{1,m-1} & a_{1m} \\ \vdots & & \vdots & \vdots \\ a_{\rho(1)-1,1} & & & \\ 1 & & & \\ & \ddots & \vdots & \vdots \\ & & 1 & a_{\rho(m-1),m} \\ & & \vdots & \vdots \\ & & 0 & a_{\rho(m)-1,m} \\ & & & 1 \\ & & & \vdots \\ 0 & & & 0 \end{pmatrix} \end{array}\right]. \tag{4.4.1}$$

The last property implies that a reduce function according to definition 2.2.11 must be fixed to define the CHNF. For different reduce functions, different pseudomatrices are in CHNF. From now on let a reduce function be fixed.

To prove the uniqueness of the CHNF, we need the following lemmas:

Lemma 4.4.2:
A pseudomatrix $\mathfrak{M} = \left[\begin{array}{ccc} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \\ \mathsf{A}_1 & \dots & \mathsf{A}_m \end{array}\right]$ in CHNF is a pseudobasis for $\mathrm{Mod}\,(\mathfrak{M})$. This means that there is no pseudomatrix $\mathfrak{N}$ with fewer columns than $\mathfrak{M}$ and $\mathrm{Mod}\,(\mathfrak{N}) = \mathrm{Mod}\,(\mathfrak{M})$.

Proof. This is immediate because of the triangular form of the CHNF, which implies that for no $i \in \mathbb{N}_m$ we have $\mathsf{A}_i \in \sum_{j \in \mathbb{N}_m \setminus \{i\}} \mathfrak{a}_j \mathsf{A}_j$. □

Lemma 4.4.3:
Let the i-**th row ideal** of the pseudomatrix

$$\left[\begin{array}{c} \begin{array}{ccc} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \end{array} \\ \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix} \end{array}\right]$$

be the ideal $\sum_{j=1}^{m} a_{ij}\mathfrak{a}_j$. Let

$$\begin{aligned} \mathrm{pr}_i : \mathcal{K}^n &\twoheadrightarrow \mathcal{K} \\ \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} &\mapsto a_i \end{aligned}$$

be the epimorphism to the i-th component of a vector.

Then the i-th row ideal of a pseudomatrix $\mathfrak{M}$ is equal to the ideal $\mathrm{pr}_i(\mathrm{Mod}\,(\mathfrak{M}))$.

Proof. Let $c \in \mathrm{pr}_i(\mathrm{Mod}\,(\mathfrak{M}))$. Then there exists an $\boldsymbol{A} \in \mathrm{Mod}\,(\mathfrak{M})$ with $c = \mathrm{pr}_i(\boldsymbol{A})$. $\boldsymbol{A}$ can be represented with $a_j \in \mathfrak{a}_j$ for $j \in \mathbb{N}_n$ as $\boldsymbol{A} = \sum_{j=1}^{n} a_j \mathsf{A}_j$. We conclude

$$c = \sum_{j=1}^{n} a_j a_{ij} \in \sum_{j=1}^{m} a_{ij} \mathfrak{a}_j.$$

Let $c \in \sum_{j=1}^{m} a_{ij} \mathfrak{a}_j$. Then there exist $a_j \in \mathfrak{a}_j$ for $j \in \mathbb{N}_n$ such that $c = \sum_{j=1}^{n} a_j a_{ij}$. Therefore

$$c = \sum_{j=1}^{n} a_j \mathrm{pr}_i(\mathsf{A}_j) = \mathrm{pr}_i(\sum_{j=1}^{n} a_j \mathsf{A}_j) \in \mathrm{pr}_i(\mathrm{Mod}\,(\mathfrak{M})).$$

□

Proposition 4.4.4 (Uniqueness of the CHNF):
If two module equivalent pseudomatrices are in CHNF they are equal.

Proof. Let $\mathfrak{M}$ and $\mathfrak{N}$ be two module equivalent pseudomatrices in CHNF.

By definition 4.1.3, $\mathfrak{M}$ and $\mathfrak{N}$ must have the same number of rows to be module equivalent. From lemma 4.4.2 it follows that they have to have the same number of columns. By lemma 4.4.3, for every row i, the i-th row ideal of $\mathfrak{M}$ and $\mathfrak{N}$ must be equal.

We use induction on the number m of columns of the pseudomatrix $\mathfrak{M}$.

Induction start. Let m=1 and

$$\mathfrak{M} = \left[\begin{array}{c} \mathfrak{a}_1 \\ \begin{pmatrix} a_{11} \\ \vdots \\ a_{\rho(1)-1,1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \end{array}\right], \mathfrak{N} = \left[\begin{array}{c} \mathfrak{b}_1 \\ \begin{pmatrix} b_{11} \\ \vdots \\ b_{n1} \end{pmatrix} \end{array}\right].$$

Since all row ideals of $\mathfrak{M}$ and $\mathfrak{N}$ are equal, it follows that for $\rho(1) + 1 \leq i \leq n$ we have $b_{i1} = 0$ and $b_{\rho(1)1} \neq 0$. $\mathfrak{N}$ is in CHNF, hence $b_{\rho(1)1} = 1$ and $\mathfrak{a}_1 = \mathfrak{b}_1$. Again by the equality of the row ideals, we conclude for $i \in \mathbb{N}_{\rho(1)}$ that $a_{i1} = b_{i1}\epsilon_i$, where the ϵ_i are units of $\mathcal{D}$. As $\mathfrak{M}$ and $\mathfrak{N}$ are module equivalent, all ϵ_i are equal in pairs. But $a_{\rho(1)1} = b_{\rho(1)1} = 1$, therefore $\epsilon_{\rho(1)} = 1$. We conclude that $a_{i1} = b_{i1}$ for all $i \in \mathbb{N}_n$, which finishes the induction start.

Induction step. Let the notations of $\mathfrak{M}$ be as in (4.4.1) and the notations of $\mathfrak{N}$ likewise with b_{ij} in place of a_{ij} and $\mathfrak{b}_{ij}$ in place of $\mathfrak{a}_{ij}$.

Let $\mathfrak{M}'$ resp. $\mathfrak{N}'$ be the pseudomatrices $\mathfrak{M}$ resp. $\mathfrak{N}$ after removing their last columns. By the definition of the CHNF, it is clear that $\mathfrak{M}'$ and $\mathfrak{N}'$ are also in CHNF.

The rows $\rho(m-1)+1, \ldots, n$ of the pseudomatrix $\mathfrak{M}'$ contain only zero entries. The last column of $\mathfrak{M}$ contains nonzero elements at most at positions $\rho(m-1)+1, \ldots, n$, of which at least the entry at position $\rho(m)$ equals one.

Therefore Mod$(\mathfrak{M}')$ contains all the elements of Mod$(\mathfrak{M})$ which have zero entries at positions $\rho(m-1)+1,\dots,n$. The same is true for Mod$(\mathfrak{N}')$ and Mod$(\mathfrak{N})$. Since Mod$(\mathfrak{M})$ = Mod$(\mathfrak{N})$, this gives

$$\text{Mod}\,(\mathfrak{M}') = \text{Mod}\,(\mathfrak{N}')\,.$$

By the induction assumption, we conclude that

$$\mathfrak{M}' = \mathfrak{N}'. \tag{4.4.2}$$

Since all row ideals of $\mathfrak{M}$ and $\mathfrak{N}$ are equal (lemma 4.4.3) and the diagonal entries of a CHNF satisfy $a_{\rho(m)m} = 1$, $b_{\rho(m)m} = 1$ (as in the argumentation in the induction start) we yield $\mathfrak{a}_m = \mathfrak{b}_m$.

It is left to show $a_{km} = b_{km}$ for $k \in \mathbb{N}_n$. By definition, for the entries at position $k > \rho(m)$ we simply have $a_{km} = b_{km} = 0$. The diagonal entry satisfies $a_{\rho(m)m} = b_{\rho(m)m} = 1$. For the following conclusions, let k iterate over the remaining entries.

Iteration of k from $\rho(m)-1$ to 1. Assume that $a_{k+1,m} = b_{k+1,m}$,…,$a_{\rho(m)-1,m} = b_{\rho(m)-1,m}$ has already been shown and also assume $a_{km} \neq b_{km}$. Let

$$\mathsf{C} = \mathsf{A}_m - \mathsf{B}_m = \begin{pmatrix} c_1 = a_{1m} - b_{1m} \\ \vdots \\ c_k = a_{km} - b_{km} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{4.4.3}$$

be the difference vector of the last columns A_m of $\mathfrak{M}$ and B_m of $\mathfrak{N}$. We have the formula

$$\mathfrak{a}_m\mathsf{C} = \mathfrak{a}_m(\mathsf{A}_m - \mathsf{B}_m) \subset \mathfrak{a}_m\mathsf{A}_m + \mathfrak{a}_m\mathsf{B}_m \subset \text{Mod}\,(\mathfrak{M})\,. \tag{4.4.4}$$

Consider two cases: the index k might or might not be in the image set $\rho(\mathbb{N}_m)$ of $\mathfrak{M}$. In other words, I check if there exists a column index r such that $a_{kr} = 1$ or not.

Case 1: $k \in \rho(\mathbb{N}_m)$

Let r be the column number with $\rho(r) = k$. We know that the matrices of both $\mathfrak{M}$ and $\mathfrak{N}$ have the entry one at position (k,r).

Let A_r be the r–th column of the matrix of $\mathfrak{M}$. A_r has a one at position k and zeros at the positions with a higher index than k. C has the entry $c_k = a_{km} - b_{km}$ at position k and zeros at the positions with higher index than k. Only the summand $\mathfrak{a}_r\mathsf{A}_r$ of the sum Mod$(\mathfrak{M}) = \sum_{i=1}^m \mathfrak{a}_i\mathsf{A}_i$ could give a contribution to the entry of C at position k. More precisely:

$$\mathfrak{a}_m c_k \subset \mathfrak{a}_r. \tag{4.4.5}$$

On the other hand, the definition of the CHNF requires both a_{km} and b_{km} to be reduced modulo the ideal $\frac{\mathfrak{a}_r}{\mathfrak{a}_m}$, which is the third property.

But by proposition 2.2.12 and formula (4.4.5), we obtain $a_{km} = b_{km}$ which is a contradiction to the above assumption.

Case 2: $k \notin \rho(N_m)$

Let l be the smallest index with $k < l \leq n$ and $l = \rho(r)$. Again consider the difference vector of formula (4.4.4). But how should any element of $\mathfrak{a}_m \mathsf{C}$ be expressed as a linear combination $\mathrm{Mod}(\mathfrak{M}) = \sum_{i=1} m\mathfrak{a}_i \mathsf{A}_i$? The columns $\mathsf{A}_1, \ldots, \mathsf{A}_{r-1}$ have zero at position k and can not contribute to a possible $c_k \neq 0$. The columns $\mathsf{A}_r, \ldots, \mathsf{A}_m$ are not zero at position k but they are also nonzero at positions $> k$. The triangular form of the matrix of $\mathfrak{M}$ guarantees that any nonzero element of $\sum_{i=r} m\mathfrak{a}_i \mathsf{A}_i$ has at least one nonzero entry among the positions $\rho(r), \ldots, \rho(m)$. It follows that this case can not occur under the assumption $c_k \neq 0$.

Both cases lead to a contradiction, therefore we have shown $a_{km} = b_{km}$ for any $k \in \mathbb{N}_n$. Together with equation (4.4.2) we obtain $\mathfrak{M} = \mathfrak{N}$, which finishes the induction step. □

Remark:

The question arises if there are other possible conventions for a normal form of a pseudomatrix except the one given in Definition 4.4.1.

The triangular form is the crucial factor to guarantee uniqueness, existence, and constructability. As argumented in many papers about the integer HNF, it has no practical value to define a normal form of integer matrices which is not a triangular form.

Consider the matrix equivalent to a given matrix where the sum of the vector norms of the columns is minimal. The construction is very hard, assumably a NP–complete problem. (The problem is related to the problem of finding a unique Steinitz form, see section 4.6.)

For triangular forms, the normal form convention includes two separate points: norming the diagonal entries and norming the non–diagonal entries.

In definition 4.4.1, the freedom to choose a reduce function implies the freedom for any convention of the non–diagonal elements. For algebraic number fields, at least 2 conventions are useful. One demands non–diagonal entries to have a representation with the least possible positive coefficients. The other one demands non–diagonal entries to have a representation with coefficients with the least absolute value. See section 2.3.

According to definition 4.4.1, the diagonal entries are one but could be any fractional algebraic number while the coefficient ideal of its column is divided by the same algebraic number. From the theoretical point of view this one is the natural choice, and it is satisfying from the algorithmic point of view since it is very easy to transform a normal form with ones on the diagonal to a normal form with any other convention.

Dr. Claus Fieker (in private conversation) pointed to the fact that, for certain applications using the normal form as a basis of a number field lattice, other conventions for the diagonal entries might be more useful. It is not be dealt with here since it is more a property of the algorithms using the normal form than a feature of the normal form itself.

4.5 The existence of the normal form

4.5.1 The Cohen algorithm

Let $\mathcal{D}$ be a Dedekind ring. Then the following algorithm describes the construction of a pseudomatrix in CHNF which is (transformation) equivalent to a given arbitrary pseudomatrix.

From the theoretical point of view this algorithm serves two purposes:

- The explicit computation of the CHNF in maximal orders of algebraic number fields over $\mathbb{Q}$. It is based on the explicit algorithms 1.8.8 in step 9 and algorithm 2.3.1 as an implementation of a reduce function defined in definition 2.2.11 in step 13. And of course on the implementations of addition, multiplication, and inversion of algebraic numbers and ideals in maximal orders over algebraic number fields.
- The existence proof in Dedekind rings based on the existence proof of proposition 1.8.7 in step 9 and of proposition 2.2.12 in step 13.

Algorithm 4.5.1: CHNF computation, Cohen

Input: Pseudomatrix $\mathfrak{M} = \begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \\ A_1 & \dots & A_m \end{bmatrix} = \left[\begin{matrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \\ \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix} \end{matrix} \right]$.

Output: Pseudomatrix $\mathfrak{M}' = \begin{bmatrix} \mathfrak{b}_1 & \dots & \mathfrak{b}_r \\ \mathrm{B}_1 & \dots & \mathrm{B}_r \end{bmatrix}$ in CHNF, the rank r, the map ρ, a transformation matrix $\mathbf{T}$ with $(A_1, \dots, A_m)\mathbf{T} = (0, \dots, 0, \mathrm{B}_1, \dots, \mathrm{B}_r)$.

Steps:

```
1: $i := m, \mathbf{T} := \mathrm{Id}_m \in \mathcal{K}^{m \times m}$
2: Loop $k := n, \dots, 1$
3:     Try to select a $j \in \mathbb{N}_i$ such that $a_{kj} \neq 0$. If not possible, goto next loop cycle in step 2.
4:     If $j \neq i$ swap columns $\begin{bmatrix} \mathfrak{a}_i \\ \mathrm{A}_i \end{bmatrix}$ with $\begin{bmatrix} \mathfrak{a}_j \\ \mathrm{A}_j \end{bmatrix}$ of $\mathfrak{M}$ and columns $i$ and $j$ ($\mathrm{T}_i$ and $\mathrm{T}_j$) of $\mathbf{T}$. This is a Swap transformation.
5:     Set $\mathrm{A}_i := \frac{1}{a_{ki}}\mathrm{A}_i$, $\mathfrak{a}_i := a_{ki}\mathfrak{a}_i$, $\mathrm{T}_i := \frac{1}{a_{ki}}\mathrm{T}_i$. This is an application of the Push factor transformation.
6:     Loop $j := 1, \dots, i-1$.
7:         If $a_{kj} = 0$ goto next loop cycle in step 6.
8:         $\mathfrak{d} := a_{kj}\mathfrak{a}_j + \mathfrak{a}_i$.
9:         Find $u \in \mathfrak{a}_j\mathfrak{d}^{-1}$ and $v \in \mathfrak{a}_i\mathfrak{d}^{-1}$ such that $a_{kj}u + v = 1$ (algorithm 1.8.8).
10:        Set $\mathrm{A}_j := \mathrm{A}_j - a_{kj}\mathrm{A}_i$, $\mathrm{A}_i := u\mathrm{A}_j + v\mathrm{A}_i$, $\mathrm{T}_j := \mathrm{T}_j - a_{kj}\mathrm{T}_i$, $\mathrm{T}_i := u\mathrm{T}_j + v\mathrm{T}_i$. Set $\mathfrak{a}_j := \mathfrak{a}_j\mathfrak{a}_i\mathfrak{d}^{-1}$, $\mathfrak{a}_i := \mathfrak{d}$. This is an application of the Unimodular transformation.
11:        Goto next loop cycle in step 6.
12:    Loop $j := i+1, \dots, m$.
13:        Reduce $a_{kj}$ modulo the ideal $\frac{\mathfrak{a}_i}{\mathfrak{a}_j}$ to $a$.
14:        If $a_{kj} \neq a$ then set $q = a - a_{kj}$, $\mathrm{A}_j := \mathrm{A}_j + q\mathrm{A}_i$, $\mathrm{T}_j := \mathrm{T}_j + q\mathrm{T}_i$. This is an application of the Add $q$-times transformation.
15:        Goto next loop cycle in step 12.
16:    Set $\hat{\rho}(i) := k$, $i := i - 1$.
17:    Goto next loop cycle in step 2.
```

```
18: Delete all zero columns of $\mathfrak{M}$.
19: The rank of the pseudomatrix is $r := n - i$.
20: Set $\rho(j) := \hat{\rho}(j + i)$ for all $j = 1, \dots, r$.
21: End.
```

This algorithm is a generalization (for modules of not necessarily full rank) on the basis of the algorithm described in [Coh96, Algorithm 2.6].

Proof. Since $\mathfrak{M}$ is subject to valid transformations, proposition 4.2.3 ensures that $\mathrm{Mod}\,(\mathfrak{M})$ is constant in the course of the algorithm.

Steps 4 and 5 guarantee the entry at position (i, k) of $\mathfrak{M}$ to be 1.

The loop in step 6 eliminates all entries of the k-th row at columns $1, \dots, i-1$, and from proposition 1.8.7, it follows that parameters can be found.

The loop in step 12 reduces the entries of the k-th row at columns $i+1, \dots, m$ to satisfy the third property in the definition of the CHNF, by proposition 2.2.12. □

Remarks:

(1) The freedom to choose a pivot entry in step 3 of the algorithm is the main factor to increase the efficiency of the algorithm with good heuristics. Any nonzero entry would do but the problem is that coefficient growth slows down further arithmetic computation. Therefore it is important to choose a pivot entry which can be expected to produce the least coefficient growth. A suggestion for a reasonable heuristics is:

- The pivot entry should be “small” since it directly affects the coefficient growth. For the notion of size, see 2.3.1, where the 1–norm of the coefficient vector representing the pivot entry is sufficient here. This should be the primary heuristics.
- The combined size of the entries of the pivot column, called column size here, also affects the coefficient growth. The sum of the 1–vector norms of the representations of the entries of the pivot column is a good measure. This should be the secondary heuristics.
- An interesting refinement is to accept a little larger pivot size (primary heuristics) for a considerable smaller column size (secondary heuristics).
- The secondary heuristics can be refined further: A little larger column size can be acceptable if the column has more zero entries.

(2) For integer matrices, a much more extensive approach, called **preview strategy**, is described in [Hop94, section 3.4]. At this stage of the development it is not a hopeful approach for pseudomatrices over Dedekind domains. Integer HNF computation is very fast, the only trouble is the coefficient explosion. So the gap between HNF computation “at a glance” and “to difficult to compute” is relatively large. CHNF computation for pseudomatrices over algebraic number fields is slow since the underlying arithmetics is for algebraic numbers, not integers! Therefore only normal forms of reasonably sized pseudomatrices will be computable at all. The actual gain by a clever heuristics is therefore much smaller for CHNF computations. Therefore only cheap heuristic computations are acceptable at all.

(3) The **best remainder strategy**, introduced in [HHR93], has marked a large step forward for integer HNF computations. The question arises if it is also applicable for pseudomatrices over DEDEKIND domains. The idea of the best remainder strategy is as follows.

The entries except one of a row must be cleared. This is not approached directly by using the gcd of the row. Instead, a pivot is chosen, all entries of the row are reduced (using a factor obtained with a remainder division) with this pivot. Only if the pivot is already the gcd of this row the row is finished, otherwise another pivot is chosen and the process is repeated. The correctness of the EUCLIDean algorithm assures that the process terminates.

This is the problem for DEDEKIND rings, where the correctness of the EUCLIDean algorithm cannot be guaranteed, provided that an analogy of the remainder division is used by selecting a certain measure for an algebraic number.

An immediate consequence from the existence of the CHNF and proposition 4.4.4 is

Corollary 4.5.2:

If two pseudomatrices over a DEDEKIND ring are module equivalent, then they are also transformation equivalent.

For each class of module equivalent pseudomatrices, there exists exactly one pseudomatrix in CHNF.

4.5.2 The BOSMA–POHST algorithm

This algorithm was introduced in the paper [BP91] and was the first algorithm to compute a normal form over a maximal order in an algebraic number field. It is formulated there in the context of relative extensions but can be seen as an algorithm to transform a given pseudomatrix into the CHNF.

This formulation has a peculiarity. The input is not a general pseudomatrix but one with only trivial coefficient ideals. Using the algorithm to compute a two element representation of an ideal (see [vS87, pp.40–41]), it is not difficult to transform a given pseudomatrix into one with trivial coefficient ideals.

Algorithm 4.5.3: CHNF computation, BOSMA–POHST

Input: Pseudomatrix $\mathfrak{M} = \begin{bmatrix} 1\mathcal{D} & \dots & 1\mathcal{D} \\ & \mathbf{A} & \end{bmatrix}$.

Output: Pseudomatrix $\mathfrak{M}'$ in CHNF.

Steps:

```
Initialize $\mathbf{D}^{(n)} := \mathbf{A}$.
Loop $t := n, \dots, 1$.
    $\mathbf{D}^{(t)}$ can be written as $\mathbf{D}^{(t)} = (\mathrm{D}_1, \dots, \mathrm{D}_m) = \begin{pmatrix} d_{11} & \cdots & d_{1m} \\ \vdots & & \vdots \\ d_{t1} & \cdots & d_{tm} \\ 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{pmatrix}$.
    Let $\mathfrak{a}_t$ be the ideal generated by $d_{t1}, \dots, d_{tm}$. If $\mathfrak{a}_t$ is the zero ideal, then
        $\mathbf{D}^{(t-1)} := \mathbf{D}^{(t)}$, $\mathrm{B}_t := 0$, and go to the next loop cycle.
```

```
5:  Apply algorithm 1.8.6 to find $e_1, e_2 \in \mathfrak{a}_t$ and $f_1, f_2 \in \mathfrak{a}_t^{-1}$ such that
        $e_1 f_1 + e_2 f_2 = 1.$
6:  Apply algorithm 1.8.9 to find $g_1, \dots, g_m \in \mathcal{D}$ such that $\sum_{i=1}^m d_{ti} g_i = e_1$
        and $h_1, \dots, h_m \in \mathcal{D}$ such that $\sum_{i=1}^m d_{ti} h_i = e_2$.
7:  Set $\mathsf{C}_1 := \begin{pmatrix} \sum_{i=1}^m g_i d_{1i} \\ \vdots \\ \sum_{i=1}^m g_i d_{t-1,i} \\ e_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ and $\mathsf{C}_2 := \begin{pmatrix} \sum_{i=1}^m h_i d_{1i} \\ \vdots \\ \sum_{i=1}^m h_i d_{t-1,i} \\ e_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$.
8:  Let $\mathsf{B}_t := f_1 \mathsf{C}_1 + f_2 \mathsf{C}_2$.
9:  Let $\mathbf{D}^{(t-1)} := (\mathsf{D}'_1, \dots, \mathsf{D}'_m)$, where $\mathsf{D}'_i := \mathsf{D}_i - d_{it} \mathsf{B}_t$ for $i \in \mathbb{N}_m$. Go to
        the next loop cycle.
10: Let $\mathfrak{M}' = \begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_n \\ \mathsf{B}_1 & \dots & \mathsf{B}_n \end{bmatrix}$. Delete all columns of $\mathfrak{M}'$ whose coefficient ideal
        is the zero ideal.
11: Reduce all the entries above the diagonal of the pseudomatrix $\mathfrak{M}'$, as described
        in the algorithm 4.5.1 in the loop from step 12.
12: End.
```

Proof. In step 5 we constructed e_i and f_i such that $e_1 f_1 + e_2 f_2 = 1$, therefore B_t has a one in position t. Hence step 9 guarantees that $\mathbf{D}^{(t-1)}$ has indeed the form which is proposed in step 3.

From the construction of the g_i and h_i, it becomes clear that we actually constructed

$$\mathsf{C}_1 = \mathbf{D}^{(t)} \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix} \quad \text{and} \quad \mathsf{C}_2 = \mathbf{D}^{(t)} \begin{pmatrix} h_1 \\ \vdots \\ h_m \end{pmatrix}.$$

and therefore we know $\mathsf{C}_1, \mathsf{C}_2 \in \operatorname{Mod}\left(\mathbf{D}^{(t)}\right)$. From $f_1, f_2 \in \mathfrak{a}_t^{-1}$, we conclude that $f_1 \mathfrak{a}_t \subset \mathcal{D}$ and $f_2 \mathfrak{a}_t \subset \mathcal{D}$. Hence

$$\mathfrak{a}_t \mathsf{B}_t = \mathfrak{a}_t (f_1 \mathsf{C}_1 + f_2 \mathsf{C}_2) \subset \operatorname{Mod}\left(\mathbf{D}^{(t)}\right). \tag{4.5.1}$$

A simple induction (on t beginning with n and decreasing) shows $\operatorname{Mod}\left(\mathbf{D}^{(t)}\right) \subset \operatorname{Mod}(\mathbf{A})$. The induction start is trivial because $\mathbf{A} = \mathbf{D}^{(n)}$. Assume $\operatorname{Mod}\left(\mathbf{D}^{(t)}\right) \subset \operatorname{Mod}(\mathbf{A})$. By formula (4.5.1) it follows that $\mathfrak{a}_t \mathsf{B}_t \subset \operatorname{Mod}(\mathbf{A})$. On the other hand, every column of $\mathbf{D}^{(t)}$ is an element of $\operatorname{Mod}(\mathbf{A})$; hence, by construction in step 9, all the columns of $\mathbf{D}^{(t-1)}$ are elements of $\operatorname{Mod}(\mathbf{A})$, which finishes the induction.

Let E be an arbitrary element of $\operatorname{Mod}\left(\mathbf{D}^{(t)}\right)$. Then there exist $k_i \in \mathcal{D}$ such that $\mathsf{E} = \sum_{i=1}^m k_i \mathsf{D}_i$. By the definition of the D'_i in step 9, it follows that

$$\mathsf{E} = \sum_{i=1}^m k_i (\mathsf{D}'_i + d_{it} \mathsf{B}_t) = \sum_{i=1}^m k_i \mathsf{D}'_i + \mathsf{B}_t \sum_{i=1}^m k_i d_{it} \in \mathfrak{a}_t \mathsf{B}_t + \operatorname{Mod}\left(\mathbf{D}^{(t-1)}\right)$$

since $\mathfrak{a}_t = \sum_{i=1}^{m} \mathcal{D} d_{it}$. With formula (4.5.1) and the easy consequence from algorithm step 9, $\operatorname{Mod}\left(\mathbf{D}^{(t-1)}\right) \subset \operatorname{Mod}\left(\mathbf{D}^{(t)}\right)$, and we conclude that

$$\operatorname{Mod}\left(\mathbf{D}^{(t)}\right) = \mathfrak{a}_t \mathtt{B}_t + \operatorname{Mod}\left(\mathbf{D}^{(t-1)}\right). \tag{4.5.2}$$

Equation (4.5.2) iteratively applied

$$\operatorname{Mod}(\mathbf{A}) = \mathfrak{a}_1 \mathtt{B}_1 + \cdots + \mathfrak{a}_n \mathtt{B}_n.$$

The triangular form of $(\mathtt{B}_1, \ldots, \mathtt{B}_n)$ is another consequence of formula (4.5.1). The ones on the diagonal are already mentioned. □

4.5.3 Discussion of the differences of both algorithms

The Cohen algorithm has the freedom to choose a pivot entry, contrary to the Bosma–Pohst algorithm. This might be an advantage, given a good pivoting strategy.

The chain of actions of the Bosma–Pohst algorithm can be seen as n (where n is the number of rows) complex pseudomatrix transformations. Therefore it can be assumed that the intermediate entry growth is less than in the Cohen algorithm. This view is supported by the practical investigation in section 6.4.

In the next chapter the concept of algorithms with reduction will be introduced. Here a drawback of the complex pseudomatrix transformations in the Bosma–Pohst algorithm becomes apparent. The Cohen algorithm uses the simpler elementary transformations and therefore enables the reduction after every transformation step. There is roughly one transformation for every matrix entry.

In sections 6.3 and 6.4 it is demonstrated how both algorithms behave practically.

4.5.4 Consequences of the existence of the normal form for matrices over principal ideal rings

The existence of the CHNF for pseudomatrices has implications on the existence of the HNF for matrices over principal ideal domains. It is not really necessary since the proof (e.g. in [PZ93, p. 179]) is possible without the theory of pseudomatrices. However, see the argumentation in subsection 3.2.4. The alternative proof is intended to clarify the relationship of matrices and pseudomatrices.

Lemma 4.5.4:

Let $\mathcal{D}$ be a principal ideal domain (which implies $\mathcal{D}$ is a Dedekind ring) and $\mathcal{K}$ its quotient field. For every matrix $\mathbf{M}$ over $\mathcal{K}$, there exists a matrix in Hermite normal form which is module equivalent to $\mathbf{M}$.

Proof. We consider the pseudomatrix $\mathfrak{M}$ consisting of $\mathbf{M}$ and the trivial ideal for every pseudomatrix column. There exists a course of pseudomatrix transformations which transform $\mathfrak{M}$ into a pseudomatrix $\mathfrak{N}$ in CHNF. Since $\mathcal{D}$ is a principal ideal domain, we can transform this $\mathfrak{N}$ into another pseudomatrix $\mathfrak{N}'$ with a triangular matrix and trivial ideals with some Push Factor transformations. Let $\mathbf{N}$ be the matrix of the pseudomatrix $\mathfrak{N}'$. Then $\mathbf{N}$ is clearly in the Hermite normal form by definition 3.2.4.

With the equation[2]

$$\operatorname{Mod}(\mathbf{M}) = \operatorname{Mod}(\mathfrak{M}) = \operatorname{Mod}(\mathfrak{N}) = \operatorname{Mod}(\mathfrak{N}') = \operatorname{Mod}(\mathbf{N}),$$

the proof is completed. □

But we can do more than that. Let $\mathcal{D}$ be a principal ideal domain and $\mathcal{K}$ its quotient field. Consider the pseudomatrix $\mathfrak{M} = \begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \\ \mathsf{A}_1 & \dots & \mathsf{A}_m \end{bmatrix}$, where $\mathfrak{a}_i$ is a fractional $\mathcal{D}$–ideal and $\mathsf{A}_i \in \mathcal{K}^n$ for $i \in \mathbb{N}_m$. For every ideal $\mathfrak{a}_i$ for $i \in \mathbb{N}_m$, we can find an $a_i \in \mathcal{K}$ which generates $\mathfrak{a}_i$. It makes sense to consider the matrix

$$\mathbf{M} = \left(a_1\mathsf{A}_1, \dots, a_m\mathsf{A}_m\right)$$

(which will be called a **corresponding matrix** to $\mathfrak{M}$ in the sequel) since we know $\operatorname{Mod}(\mathfrak{M}) = \operatorname{Mod}(\mathbf{M})$. There might be more than one possible generator for a given ideal. Therefore there are usually many matrices corresponding to $\mathfrak{M}$.

We know that all generators of a given principal ideal only differ by a factor which is a unit of $\mathcal{D}$. Therefore the columns of two matrices corresponding to $\mathfrak{M}$ also differ by a factor which is a unit in $\mathcal{D}$.

Consider the definition 3.1.2, the transformation number 3 allows the multiplication of a unit in $\mathcal{D}$. Therefore *any two matrices corresponding to the same pseudomatrix are transformation equivalent.*

Therefore we are able to prove the following fact.

Lemma 4.5.5:
Let $\mathcal{D}$ be a principal ideal domain. Two matrices corresponding to (pseudomatrix) transformation equivalent pseudomatrices are (matrix) transformation equivalent.

Proof. It is sufficient to prove the lemma for a pseudomatrix $\mathfrak{M}$ and another pseudomatrix $\mathfrak{N}$ produced from $\mathfrak{M}$ by a single elementary transformation of definition 4.2.1.

For the Swap transformation, there is a corresponding transformation in definition 3.1.2 (number 1).

For the Push Factor transformation, there is nothing to prove because $\mathfrak{M}$ and $\mathfrak{N}$ correspond to the identical matrix (provided that the same generators of the coefficient ideals are used.)

For the Two scaled transformation, there is a corresponding transformation in definition 3.1.2 (number 3). Assume the same notations as in definition 4.2.1. Let the ideals $\mathfrak{a}_i$ and $\mathfrak{a}_j$ resp. be generated by elements $a_i, a_j \in \mathcal{K}$ resp. For transformation 3 of definition 3.1.2 the parameters

$$\begin{aligned} d_1 &:= c_1 \in \mathcal{D}, \\ d_2 &:= \frac{c_2 a_i}{a_j} \in \mathcal{D}, \\ d_3 &:= \frac{c_3 a_j}{a_i} \in \mathcal{D}, \\ d_4 &:= c_4 \in \mathcal{D} \end{aligned}$$

[2] Compare the different notions of Mod for matrices in definition 3.1.1 and for pseudomatrices in definition 4.1.2

are used to transform the corresponding matrices.

For the Collect transformation, the corresponding matrix transformation is again number 3 of definition 3.1.2. Let the ideals $\mathfrak{a}_i$ and $\mathfrak{a}_j$ resp. be generated by elements $a_i, a_j \in \mathcal{K}$ resp. The parameters are

$$\begin{aligned} d_1 &:= \frac{c_1}{a_i} \in \mathcal{D}, \\ d_2 &:= \frac{c_2}{a_j} \in \mathcal{D}, \\ d_3 &:= c_3 a_j \in \mathcal{D}, \\ d_4 &:= c_4 a_i \in \mathcal{D}. \end{aligned}$$

The Spread transformation can be dealt with analogous to the Collect transformation.

For the Insert zero column transformation, there is a corresponding transformation in definition 3.1.2 (number 4).

For the Delete zero column transformation, there is a corresponding transformation in definition 3.1.2 (number 5). □

Now we have proved the following theorem

Theorem 4.5.6:

Let $\mathcal{D}$ be a principal ideal domain. For every matrix there exists a course of elementary transformations which transform the matrix to a matrix in Hermite normal form.

Proof. Let $\mathbf{M}$ be a matrix over $\mathcal{K}$, the quotient field of $\mathcal{D}$. Let a reduce function be fixed.

Let $\mathfrak{M}$ be the pseudomatrix which consists of $\mathbf{M}$ and trivial ideals. $\mathfrak{M}$ can be transformed to a pseudomatrix $\mathfrak{N}$ in CHNF considering the chosen reduce function. Let $\mathbf{N}$ be any matrix corresponding to $\mathfrak{N}$. Lemma 4.5.5 guarantees that $\mathbf{M}$ and $\mathbf{N}$ are transformation equivalent.

According to definition 3.2.4 of the HNF and definition 4.4.1 we know that $\mathbf{N}$ is in HNF except for the condition on the diagonal entries. This condition can be satisfied with at most one application for every column of the transformation 3 in definition 3.1.2. □

4.6 Steinitz forms

Triangular forms are not the only desirable forms in the class of equivalent pseudomatrices. Another aim is to find the smallest possible pseudomatrix with trivial coefficient ideals. This is equivalent to the task of finding a minimal $\mathcal{D}$–generating system of a given finitely generated $\mathcal{D}$–module.

Let $\mathfrak{M}$ be a pseudomatrix with n rows and rank n. Let

$$\mathfrak{M}' = \begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_n \\ \mathsf{A}_1 & \dots & \mathsf{A}_n \end{bmatrix}$$

be the CHNF of $\mathfrak{M}$. We know that in general it is not possible to find an equivalent pseudomatrix with n columns which has only trivial coefficient ideals. If this would be the case, every relative extension would have a relative integral basis. But this is not the case, as argumented in [BP91].

For the ideals $\mathfrak{a}_i$ with $i \in \mathbb{N}_n$, we can find the two–element presentations

$$\mathfrak{a}_i = \alpha_i \mathcal{D} + \beta_i \mathcal{D}.$$

Therefore $\mathfrak{M}'$ gives a $\mathcal{D}$–generating set for $\mathrm{Mod}\,(\mathfrak{M})$ of $2n$ elements:

$$\mathrm{Mod}\,(\mathfrak{M}) = \sum_{i=1}^{n} \big(\alpha_i \mathsf{A}_i \mathcal{D} + \beta_i \mathsf{A}_i \mathcal{D}\big).$$

But we can do better than that! It is possible to find a pseudomatrix $\mathfrak{M}''$ with n columns which has trivial coefficient ideals with the exception of one ideal. The ideal class of this ideal is an important invariant of $\mathrm{Mod}\,(\mathfrak{M})$ and is called the Steinitz class. Therefore pseudomatrices of the described form will be referred to as Steinitz forms, with the convention that at most the last coefficient ideal may be nontrivial.

From the Steinitz form

$$\mathfrak{M}'' = \left[\begin{array}{cccc} 1\mathcal{D} & \ldots & 1\mathcal{D} & \mathfrak{b} \\ \mathsf{B}_1 & \ldots & \mathsf{B}_{n-1} & \mathsf{B}_n \end{array}\right]$$

we can construct a minimal $\mathcal{D}$–generating system of $\mathrm{Mod}\,(\mathfrak{M})$. If $\mathfrak{b}$ is a principal ideal generated by b then

$$\mathrm{Mod}\,(\mathfrak{M}) = b\mathsf{B}_n\mathcal{D} + \sum_{i=1}^{n-1} \mathsf{B}_i \mathcal{D}$$

which is even a $\mathcal{D}$–basis. Otherwise, letting $\mathfrak{b} = \alpha\mathcal{D} + \beta\mathcal{D}$,

$$\mathrm{Mod}\,(\mathfrak{M}) = \alpha\mathsf{B}_n\mathcal{D} + \beta\mathsf{B}_n\mathcal{D} + \sum_{i=1}^{n-1} \mathsf{B}_i \mathcal{D}.$$

A difficult question is how to choose a unique Steinitz *normal* form from the the set of all Steinitz forms. Unfortunately, uniqueness can not be obtained in the relatively natural way like the CHNF. The following example shows that there are pseudomatrix classes where all Steinitz forms are not triangular forms.

Example 4.6.1. *Let ρ be a root of the integral polynomial $x^3 + 42x + 154$. Let $\mathcal{K}$ be the algebraic number field $\mathcal{K} = \mathbb{Q}[\rho]$. The maximal order $o_{\mathcal{K}}$ of $\mathcal{K}$ is generated by the powers of ρ. The class group is isomorphic to $\mathrm{C}_3 \times \mathrm{C}_3 \times \mathrm{C}_3$. Its generators are the prime ideals $\mathfrak{p}_1 = 2\mathcal{D} + \rho\mathcal{D}$, $\mathfrak{p}_2 = 3\mathcal{D} + (1+\rho)\mathcal{D}$, and $\mathfrak{p}_3 = 7\mathcal{D} + \rho\mathcal{D}$.*

The ideals $\mathfrak{a}_1 = \mathfrak{p}_1\mathfrak{p}_2$ and $\mathfrak{a}_2 = \mathfrak{p}_1\mathfrak{p}_3$ are also nonprincipal. Let a reduce function be fixed such that one is reduced modulo the ideal $\mathfrak{a}_1\mathfrak{a}_2^{-1}$, and let

$$\mathfrak{M} = \left[\begin{array}{cc} \mathfrak{a}_1 & \mathfrak{a}_2 \\ \multicolumn{2}{c}{\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}} \end{array}\right].$$

Then $\mathfrak{M}$ is in CHNF. Lemma 4.4.3 states that all module equivalent pseudomatrices have the same row ideals. The row ideals of $\mathfrak{M}$ are $\mathfrak{a}_1 + \mathfrak{a}_2 = \mathfrak{p}_1$ and $\mathfrak{a}_2$, both of which are not principal.

Assume

$$\mathfrak{N} = \left[\begin{array}{cc} 1\mathcal{D} & \mathfrak{b} \\ \multicolumn{2}{c}{\begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix}} \end{array}\right]$$

is a triangular pseudomatrix in a STEINITZ form which is module equivalent to $\mathfrak{M}$. By lemma 4.4.3, the second row ideal of $\mathfrak{M}$ and $\mathfrak{N}$ is equal, and therefore $\mathfrak{b} = \mathfrak{a}_2$. The equality of the first row ideal gives us $b\mathfrak{a}_2 + \mathcal{D} = \mathfrak{a}_1$. Since $\mathfrak{M}$ is integral, by lemma 4.1.5, $\mathfrak{N}$ is also integral. Therefore $b\mathfrak{a}_2 \subset \mathcal{D}$ and we conclude $b\mathfrak{a}_2 = \mathfrak{a}_1$. But this would imply $b\mathfrak{p}_3 = \mathfrak{p}_2$ which is a contradiction to the fact that $\mathfrak{p}_2$ and $\mathfrak{p}_3$ belong to different ideal classes.

Therefore, in this there is no triangular pseudomatrix in a STEINITZ form $\mathfrak{N}$.

The example implies that the triangular shape cannot be used for the definition of a STEINITZ normal form.

A possible choice of a STEINITZ normal form is the STEINITZ form with the least combined size (see subsection 2.3.1 for the notion of the quality of algebraic numbers for instance) of the entries. This notion would be impractical because it would probably be an exponentially hard problem to compute.

The following algorithm computes a STEINITZ form. It is important that $\mathcal{D}$ is a DEDEKIND ring because it is so for algorithm 1.8.5.

```
Algorithm 4.6.2: STEINITZ form computation
  Input: Pseudomatrix $\mathfrak{M}$.
  Output: Pseudomatrix in STEINITZ form: $\mathfrak{M}' = \begin{bmatrix} 1\mathcal{D} & \dots & 1\mathcal{D} & \mathfrak{b} \\ \mathsf{B}_1 & \dots & \mathsf{B}_{n-1} & \mathsf{B}_n \end{bmatrix}$.
  Steps:
    1: Transform $\mathfrak{M}$ into its CHNF $\mathfrak{M}' = \begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_n \\ \mathsf{A}_1 & \dots & \mathsf{A}_n \end{bmatrix}$.
    2: Loop $i = 1, \dots, n-1$.
    3:     If $\mathfrak{a}_i$ is a principal ideal, generated by $a_i \in \mathcal{K}$, apply a PUSH FACTOR trans-
                formation with the parameter $a_i$, and go to the next loop step.
    4:     Apply algorithm 1.8.5 to the ideals $\mathfrak{a}_i$ and $\mathfrak{a}_{i+1}$ which yields the parameters
                for a COLLECT transformation which is applied to $\mathfrak{M}'$.
    5: $\mathfrak{M}'$ holds the required STEINITZ form.
    6: End.
```

Remarks:

(1) It is possible to record a transformation matrix — it is based on the transformation matrix of the CHNF computation and the elementary matrix belonging to the COLLECT and PUSH FACTOR transformations.

(2) The step 3 does not require “If and only if”. This is important for algebraic number fields, where the decision if an ideal is principal might be very difficult to obtain. The standard implementation involves an enumeration of probably exponential complexity. It is possible to improve this with methods described in [Hes96] in cases where class group computations are feasible. But still it is expensive.

In the recent implementation there is a fast check on principality, where a few elements of the ideal are tested if they generate the ideal. This check has proved to be effective and cheap. A negative result does not guarantee that the ideal in question is not principal — but this is not required by this algorithm.

The advantage of using a principal ideal is obvious: the COLLECT transformation might cause a coefficient growth which can sum up considerably.

(3) It is not really necessary for i to go from 1 to $n-1$. Any other order would do as good, which opens the possibility of a heuristic decision. But the order from 1 to $n-1$ is very good since the first columns of a pseudomatrix in CHNF are very sparse. If COLLECT transformations are required this strategy diminishes coefficient growth and preserves part of the sparsity of the CHNF. Assuming a full rank pseudomatrix the resulting pseudomatrix contains at least $\frac{(n-1)(n-2)}{2}$ zero entries (compared to $\frac{n(n-1)}{2}$ zero entries of the CHNF).

4.7 The determinant of a pseudomatrix

Definition 4.7.1:
Let $\mathfrak{M} = \left[\begin{array}{ccc} \mathfrak{a}_1 & \dots & \mathfrak{a}_n \\ & \mathbf{A} & \end{array}\right]$ be a square pseudomatrix with n columns and rows over an integral domain $\mathcal{D}$. Then the **determinant** of $\mathfrak{M}$ is defined as the fractional $\mathcal{D}$–ideal (or the zero ideal)

$$\det \mathfrak{M} \quad =_{\mathrm{Def}} \quad \det \mathbf{A} \prod_{i=1}^{n} \mathfrak{a}_i .$$

Definition 4.7.2:
Let $\mathfrak{M} = \left[\begin{array}{ccc} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \\ & \mathbf{A} & \end{array}\right]$ be a pseudomatrix with n rows and m columns over an integral domain $\mathcal{D}$. Let $r \in \mathbb{N}$ with $r \leq m, r \leq n$. Let $\mathbf{A}'$ be an $r \times r$ submatrix of $\mathbf{A}$ and $i_1, \dots, i_r$ the indices of the columns of $\mathbf{A}$ which are columns of $\mathbf{A}'$. Then the pseudomatrix $\left[\begin{array}{ccc} \mathfrak{a}_{i_1} & \dots & \mathfrak{a}_{i_r} \\ & \mathbf{A}' & \end{array}\right]$ is called an r**–subpseudomatrix** of $\mathfrak{M}$. The determinant of this r–subpseudomatrix is called an r**–minor**.

Definition 4.7.3:
The r**–minor sum** of a (possibly not square) pseudomatrix $\mathfrak{M}$ is the sum of the determinants of all r–subpseudomatrices of $\mathfrak{M}$. If no r–subpseudomatrices exist (because $r > \min(m, n)$), then the r–minor sum is the zero ideal per convention.

Remark:
We cannot expect minor sums of nonsquare matrices to obey the multiplicativity law. Therefore they should not be viewed as a generalization of the determinant although minor sums have important applications similar to determinants. They allow coefficient reduction of matrices during successive matrix transformation. The following example demonstrates the main problem to define a determinant–like function for nonsquare matrices:

Example 4.7.4. *Let $\mathcal{D}$ be an integral domain, $\mathcal{K}$ its quotient field. Let $\mathcal{M}$ be the set of all nontrivial matrices (at least one row and one column) over $\mathcal{D}$. Consider*

$$\begin{pmatrix}1\\1\end{pmatrix}(1\;1) = \begin{pmatrix}1\;1\\1\;1\end{pmatrix} \quad \text{and} \quad (1\;1)\begin{pmatrix}1\\1\end{pmatrix} = (2).$$

So how should a multiplicative function det : $\mathcal{M} \to \mathcal{D}$ *be defined? In any case it cannot be consistent with* $\left|\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\right| = 0$ *and* $|2| = 2$.

Definition 4.7.5:

Let r be the maximum of all natural numbers such that the r–minor sum of $\mathfrak{M}$ is not the zero ideal. r is called the **rank** of $\mathfrak{M}$. The **rank minor sum** of $\mathfrak{M}$ is the r–minor sum of $\mathfrak{M}$.

Lemma 4.7.6:

The determinant of a square pseudomatrix does not change upon applications of the size preserving elementary transformations.

Proof. It suffices to show that the General transformation (see subsection 4.2.2) does not change the determinant.

The product of the coefficient ideals changes by $\frac{1}{e}$. The transformation of the matrix can be viewed as the multiplication of an elementary matrix as in subsection 3.2.2. The determinant of the elementary transformation matrix equals e. Therefore the determinant of the matrix of the pseudomatrix changes by e and the determinant of the whole pseudomatrix stays constant. □

Proposition 4.7.7:

The r–minor sum of a pseudomatrix does not change upon applications of the elementary transformations.

Proof. Let $\mathfrak{M}$ be a pseudomatrix before $\mathfrak{M}'$ resulting from an application of a transformation.

It is proved first that the insertion of a zero column does not change the r–minor sum. The set of r–subpseudomatrices of $\mathfrak{M}'$ contains the r–subpseudomatrices of $\mathfrak{M}$ and some pseudomatrices whose determinant is zero because they contain a zero column. This is true even if the set of r–subpseudomatrices of $\mathfrak{M}$ is empty and that of $\mathfrak{M}'$ is not. If both sets are empty, both r–minors are zero per convention.

If a zero column is deleted, the set of r–subpseudomatrices of $\mathfrak{M}$ contains the r–subpseudomatrices of $\mathfrak{M}'$ and some pseudomatrices whose determinant is zero because they contain a zero column. The equality is correct even if one or both sets of r–subpseudomatrices are empty.

It is left to show that an application of a transformation of the General type does not change the r–minor sum.

Let

$$\mathfrak{M} = \left[\begin{array}{c} \begin{matrix}\mathfrak{a}_1 & \cdots & \mathfrak{a}_m\end{matrix} \\ \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nm}\end{pmatrix}\end{array}\right].$$

Let Φ be the set of all injections from $\mathbb{N}_r$ to $\mathbb{N}_m$. Let Ψ be the set of all injections from $\mathbb{N}_r$ to $\mathbb{N}_n$. For $\phi \in \Phi$ and $\psi \in \Psi$, let

$$\mathfrak{M}_{\phi\psi} = \left[\begin{array}{c} \begin{matrix} \mathfrak{a}_\phi(1) & \dots & \mathfrak{a}_\phi(m) \end{matrix} \\ \begin{pmatrix} a_{\phi(1)\psi(1)} & \cdots & a_{\phi(1)\psi(m)} \\ \vdots & & \vdots \\ a_{\phi(n)\psi(1)} & \cdots & a_{\phi(n)\psi(m)} \end{pmatrix} \end{array}\right].$$

Then the r–minor sum of $\mathfrak{M}$ is

$$\sum_{\phi\in\Phi}\sum_{\psi\in\Psi} \det(\mathfrak{M}_{\phi\psi}).$$

Applying a General two columns transformation to $\mathfrak{M}$ results in the pseudomatrix $\mathfrak{M}'$. This transformation involves two columns A_i and A_j together with their coefficient ideals $\mathfrak{a}_i$ and $\mathfrak{a}_j$. The rest of the pseudomatrix $\mathfrak{M}$ is constant. The variables of this transformation are four elements $c_1, \dots, c_4 \in \mathcal{K}$.

$$\begin{pmatrix} \mathsf{B}_i \\ \mathsf{B}_j \end{pmatrix} = \begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} \begin{pmatrix} \mathsf{A}_i \\ \mathsf{A}_j \end{pmatrix}, \quad \begin{vmatrix} c_1 & c_2 \\ c_3 & c_4 \end{vmatrix} = e, \quad \mathfrak{a}_i\mathfrak{a}_j = e\mathfrak{b}_i\mathfrak{b}_j,$$
$$c_1 \in \mathfrak{a}_i\mathfrak{b}_i^{-1}, \quad c_2 \in \mathfrak{a}_j\mathfrak{b}_i^{-1}, \quad c_3 \in \mathfrak{a}_i\mathfrak{b}_j^{-1}, \quad c_4 \in \mathfrak{a}_j\mathfrak{b}_j^{-1}. \quad (4.7.1)$$

The question is how the summands in the ideal sum $\sum_{\phi\in\Phi}\sum_{\psi\in\Psi}\det(\mathfrak{M}_{\phi\psi})$ are affected by the General transformation of columns i and j.

Because the column transformation changes every entry of a column simultaneously, we can use the simple implication

$$\begin{aligned} &\forall\psi\in\Psi, \ \sum_{\phi\in\Phi}\det(\mathfrak{M}_{\phi\psi}) = \sum_{\phi\in\Phi}\det(\mathfrak{M}'_{\phi\psi}) \\ &\implies \sum_{\phi\in\Phi}\sum_{\psi\in\Psi}\det(\mathfrak{M}_{\phi\psi}) = \sum_{\phi\in\Phi}\sum_{\psi\in\Psi}\det(\mathfrak{M}'_{\phi\psi}) \end{aligned}$$

and show the equality for every $\psi \in \Psi$. In the sequel let $\psi \in \Psi$ be fixed.

Let $\phi \in \Phi$. If neither i nor j is in the image set of ϕ, then obviously $\det(\mathfrak{M}_{\phi\psi}) = \det(\mathfrak{M}'_{\phi\psi})$. If both i and j are in the image set of ϕ, then we can apply lemma 4.7.6 and we have again $\det(\mathfrak{M}_{\phi\psi}) = \det(\mathfrak{M}'_{\phi\psi})$.

This simple 1–1 correspondence is not true for those summands where exactly one of i and j is in the image set of ϕ. Without loss of generality we can assume $i \in \mathrm{Im}\phi$. There exists a $\phi' \in \Phi$ such that $\mathrm{Im}\phi' = \mathrm{Im}\phi \setminus \{i\} \cup \{j\}$. It is clear that if we can show $\det\mathfrak{M}_{\phi\psi} + \det\mathfrak{M}_{\phi'\psi} = \det\mathfrak{M}'_{\phi\psi} + \det\mathfrak{M}'_{\phi'\psi}$, we can deduce $\sum_{\phi\in\Phi}\det(\mathfrak{M}_{\phi\psi}) = \sum_{\phi\in\Phi}\det(\mathfrak{M}'_{\phi\psi})$.

Applying the definition of the determinant for pseudomatrices and dividing the ideals which are equal on both sides, this leaves us to show

$$\mathfrak{a}_i\det\Big(\dots\mathsf{A}_i\dots\Big) + \mathfrak{a}_j\det\Big(\dots\mathsf{A}_j\dots\Big) = \mathfrak{b}_i\det\Big(\dots\mathsf{B}_i\dots\Big) + \mathfrak{b}_j\det\Big(\dots\mathsf{B}_j\dots\Big)$$

where the four matrix patterns denote the same matrix except for one column at a certain position which is A_i, A_j, B_i, and B_j resp.

We have

$$\begin{aligned}
&\mathfrak{b}_i \det\Big(\dots \mathsf{B}_i \dots\Big) + \mathfrak{b}_j \det\Big(\dots \mathsf{B}_j \dots\Big)\\
&= \mathfrak{b}_i \det\Big(\dots c_1\mathsf{A}_i + c_2\mathsf{A}_j \dots\Big) + \mathfrak{b}_j \det\Big(\dots c_3\mathsf{A}_i + c_4\mathsf{A}_j \dots\Big).
\end{aligned}$$

Since the determinant of matrices is linear in every column, we have

$$\begin{aligned}
&\mathfrak{b}_i \det\Big(\dots \mathsf{B}_i \dots\Big) + \mathfrak{b}_j \det\Big(\dots \mathsf{B}_j \dots\Big)\\
&\quad = \mathfrak{b}_i \Bigg(c_1 \det\Big(\dots \mathsf{A}_i \dots\Big) + c_2 \det\Big(\dots \mathsf{A}_j \dots\Big)\Bigg)\\
&\qquad + \mathfrak{b}_j \Bigg(c_3 \det\Big(\dots \mathsf{A}_i \dots\Big) + c_4 \det\Big(\dots \mathsf{A}_j \dots\Big)\Bigg).
\end{aligned}$$

For a fractional $\mathcal{D}$–ideal over an integral domain $\mathcal{D}$, we know

$$\mathfrak{a}(a+b) \subset \mathfrak{a}a + \mathfrak{a}b, \text{ where } \quad a, b \in \mathcal{K};$$

therefore

$$\begin{aligned}
&\mathfrak{b}_i \det\Big(\dots \mathsf{B}_i \dots\Big) + \mathfrak{b}_j \det\Big(\dots \mathsf{B}_j \dots\Big)\\
&\quad \subset (\mathfrak{b}_i c_1 + \mathfrak{b}_j c_3) \det\Big(\dots \mathsf{A}_i \dots\Big) + (\mathfrak{b}_i c_2 + \mathfrak{b}_j c_4) \det\Big(\dots \mathsf{A}_j \dots\Big).
\end{aligned}$$

Because of the membership requirements on $c_1, \dots, c_4$ in formula (4.7.1), we conclude that

$$\begin{aligned}
&\mathfrak{b}_i \det\Big(\dots \mathsf{B}_i \dots\Big) + \mathfrak{b}_j \det\Big(\dots \mathsf{B}_j \dots\Big)\\
&\quad \subset \mathfrak{a}_i \det\Big(\dots \mathsf{A}_i \dots\Big) + \mathfrak{a}_j \det\Big(\dots \mathsf{A}_j \dots\Big).
\end{aligned}$$

Up to this point, we have only proved that the r–minor sum of $\mathfrak{M}'$ is contained in the r–minor sum of $\mathfrak{M}$. But there is an inverse transformation from $\mathfrak{M}'$ to $\mathfrak{M}$ (lemma 4.2.2) which yields the opposite containment. □

4.8 Reduction of pseudomatrices

Let $\mathcal{D}$ be a DEDEKIND ring and $\mathcal{K}$ its quotient field. Let $\mathfrak{M}$ be a pseudomatrix over $\mathcal{D}$. The normal form algorithm 4.5.1 or 4.5.3 transform $\mathfrak{M}$ ssuccessively into its normal form:

$$\mathfrak{M} = \mathfrak{M}_1 \longrightarrow \dots \longrightarrow \mathfrak{M}_k \longrightarrow \mathfrak{M}_{k+1} \longrightarrow \dots \longrightarrow \mathfrak{M}_z = \left[\begin{array}{c} \mathfrak{c}_1 \quad \dots \quad \mathfrak{c}_n \\ \begin{pmatrix} 1 & & * \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \end{array}\right].$$

In every step $k \in \mathbb{N}_{z-1}$ the property $\mathrm{Mod}\,(\mathfrak{M}_k) = \mathrm{Mod}\,(\mathfrak{M}_{k+1})$ is satisfied.

This section describes modifications to this algorithm resulting from the insertion of a reduction step $\xrightarrow{\mathrm{red}_{\mathfrak{R}}}$ after every step:

$$\mathfrak{M}_k \xrightarrow{\mathrm{red}_{\mathfrak{R}}} \mathfrak{M}'_k \longrightarrow \mathfrak{M}_{k+1}.$$

The purpose of this is as follows: The normal form algorithm directly approaches the triangular form of the pseudomatrix without care about the size (see subsection 2.3.1 for the meaning of size in the case of algebraic number rings) of the entries of the intermediate pseudomatrices $\mathfrak{M}_k$. If the pseudomatrix is large and difficult, the growth of the entries of $\mathfrak{M}_k$ causes the algorithm to fail from memory and calculation time problems. Reduction steps will avoid or delay the entry explosion.

4.8.1 The general reduction process

Let $\mathfrak{R}$ be a pseudomatrix which satisfies $\mathrm{Mod}\,(\mathfrak{R}) \subset \mathrm{Mod}\,(\mathfrak{M})$. Only pseudomatrices of a very simple form are useful here — we will consider only diagonal forms. $\mathrm{red}_{\mathfrak{R}}$ denotes a reduction (defined below in definition 4.8.1) of $\mathfrak{M}_k$ using the submodules of $\mathfrak{R}$ such that

$$\mathrm{Mod}\,(\mathfrak{M}_k) + \mathrm{Mod}\,(\mathfrak{R}) = \mathrm{Mod}\,(\mathfrak{M}'_k) + \mathrm{Mod}\,(\mathfrak{R})\,. \tag{4.8.1}$$

Actually, it seems more natrural to demand $\mathrm{Mod}\,(\mathfrak{M}_k) = \mathrm{Mod}\,(\mathfrak{M}'_k)$. But the above equation is more general and allows a reduction process, which has proved to be very effective.

Definition 4.8.1 (Reduction of pseudomatrices):
Let $\mathfrak{R}$ be a pseudomatrix with $\mathrm{Mod}\,(\mathfrak{R}) \subset \mathrm{Mod}\,(\mathfrak{M})$. A proper reduction $\mathrm{red}_{\mathfrak{R}}$ is any finite number of single reduction step applications. In a single reduction step, we choose a column $\left[\begin{matrix}\mathfrak{r}\\ \begin{pmatrix} r_1\\ \vdots\\ r_n\end{pmatrix}\end{matrix}\right]$ of $\mathfrak{R}$, a column $\left[\begin{matrix}\mathfrak{a}\\ \begin{pmatrix} a_1\\ \vdots\\ a_n\end{pmatrix}\end{matrix}\right]$ of $\mathfrak{M}$, and an element $q \in \frac{\mathfrak{r}}{\mathfrak{a}}$. We replace the column $\left[\begin{matrix}\mathfrak{a}\\ \begin{pmatrix} a_1\\ \vdots\\ a_n\end{pmatrix}\end{matrix}\right]$ of $\mathfrak{M}$ with the column $\left[\begin{matrix}\mathfrak{a}\\ \begin{pmatrix} a_1+qr_1\\ \vdots\\ a_n+qr_n\end{pmatrix}\end{matrix}\right]$.

This definition does not specify a method to reduce. It only gives a frame for the allowed operations.

Lemma 4.8.2:
For any reduction of the above kind, the equation (4.8.1) holds.

Proof. It will be shown that the equation holds for every single reduction step. Let the notations be as in the above definition. Let $\mathfrak{M}$ be the pseudomatrix before reduction, $\mathfrak{M}'$ after.

$$\mathrm{Mod}\left[\begin{matrix}\mathfrak{a}\\ \begin{pmatrix} a_1+qr_1\\ \vdots\\ a_n+qr_n\end{pmatrix}\end{matrix}\right] \subset \mathrm{Mod}\left[\begin{matrix}\mathfrak{a}\\ \begin{pmatrix} a_1\\ \vdots\\ a_n\end{pmatrix}\end{matrix}\right] + \mathrm{Mod}\left[\begin{matrix}\mathfrak{a}\\ \begin{pmatrix} qr_1\\ \vdots\\ qr_n\end{pmatrix}\end{matrix}\right]$$

$$= \mathfrak{a}\begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} + q\mathfrak{a}\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix} \subset \mathfrak{a}\begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} + \mathfrak{r}\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix} \subset \operatorname{Mod}(\mathfrak{M}) + \operatorname{Mod}(\mathfrak{R})\,.$$

On the other hand,

$$\operatorname{Mod}\begin{bmatrix} \mathfrak{a} \\ \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \end{bmatrix} = \operatorname{Mod}\begin{bmatrix} \mathfrak{a} \\ \begin{pmatrix} a_1 + qr_1 - qr_1 \\ \vdots \\ a_n + qr_n - qr_n \end{pmatrix} \end{bmatrix}$$

$$\subset \operatorname{Mod}\begin{bmatrix} \mathfrak{a} \\ \begin{pmatrix} a_1 + qr_1 \\ \vdots \\ a_n + qr_n \end{pmatrix} \end{bmatrix} + \operatorname{Mod}\begin{bmatrix} \mathfrak{a} \\ \begin{pmatrix} qr_1 \\ \vdots \\ qr_n \end{pmatrix} \end{bmatrix} \subset \operatorname{Mod}(\mathfrak{M}') + \operatorname{Mod}(\mathfrak{R})\,.$$

□

The general principles of reduction are

Two–phase method We choose $\mathfrak{R}$ such that $\operatorname{Mod}(\mathfrak{R}) \subset \operatorname{Mod}(\mathfrak{M})$. This implies $\operatorname{Mod}(\mathfrak{M}) = \operatorname{Mod}(\mathfrak{M}_z) + \operatorname{Mod}(\mathfrak{R})$. After we transformed $\mathfrak{M}$ into its normal form $\mathfrak{M}_z$ (with reduction modulo $\mathfrak{R}$), we know that the concatenation $\mathfrak{N} := \mathfrak{M}_z + \mathfrak{R}$ satisfies $\operatorname{Mod}(\mathfrak{N}) = \operatorname{Mod}(\mathfrak{M})$. Therefore we can apply the normal form algorithm to $\mathfrak{N}$, this time without reduction. The first normal form application with reduction is called **first phase** and the second normal form application without reduction **second phase**.

Strict equality method We choose an $\mathfrak{R}$ such that, for every step, $\operatorname{Mod}(\mathfrak{M}_k) = \operatorname{Mod}(\mathfrak{M}'_k)$ is guaranteed. If $\mathfrak{M}_z$ is in the required normal form, then we are finished because $\operatorname{Mod}(\mathfrak{M}) = \operatorname{Mod}(\mathfrak{M}_z)$.
In practice it is much too difficult to ensure $\operatorname{Mod}(\mathfrak{M}_k) = \operatorname{Mod}(\mathfrak{M}'_k)$ in every step.

Determinant reduction method We use

$$\mathfrak{R} = \begin{bmatrix} \mathfrak{d} \quad \dots \quad \mathfrak{d} \\ \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \end{bmatrix},$$

where $\mathfrak{d}$ is an integral multiple of the rank minor sum of $\mathfrak{M}$, provided $\mathfrak{M}$ has full rank (see definition 4.7.5). As well as in the two–phase method the resulted CHNF is not equivalent to $\mathfrak{M}$. But there is a method to construct the correct CHNF which is simpler than another CHNF computation without reduction. It is described in [Coh96, p.16].
The drawback of this method is that there usually are diagonal reducers whose generated module is much larger than the reducer obtained with the determinant.

4.8.2 Suitable reducers

A reducer $\mathfrak{R}$ should represent a large $\operatorname{Mod}(\mathfrak{R})$, on the one hand, so that there are many options to reduce the given pseudomatrix. On the other hand, $\mathfrak{R}$ should be of a very simple form, such that the algorithm which uses the reducer can be fast and efficient. This can be contradictory, so there has to be a good trade–off between both.

Obviously the pseudomatrix itself is a valid reducer of it, but this would be very impractical. Three different types of reducers will be introduced. Only reducers with a diagonal matrix will be used here since they allow for the reduction of a single entry separately.

For a diagonal reducer the two–phase method is very useful. The second phase is relatively cheap since $\mathfrak{R}$ is very sparse and $\mathfrak{M}_z$ is already in normal form. This is demonstrated practically in section 6.6. In the first phase we have a maximum of reduction possibilities compared with the strict equality method and the determiant reduction method.

The following is based on the two–phase method.

General diagonal reducer

$$\mathfrak{R} = \left[\begin{array}{c} \mathfrak{r}_1 \quad \dots \quad \mathfrak{r}_n \\ \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \end{array}\right],$$

where $\mathfrak{r}_i$ is a fractional $\mathcal{D}$–ideal for $i \in \mathbb{N}_m$. The underlying method is as follows: Let

$$\left[\begin{array}{c} \begin{array}{ccc} \mathfrak{a}_1 & \dots & \mathfrak{a}_m \end{array} \\ \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix} \end{array}\right]$$

be the pseudomatrix to reduce. $\xrightarrow{\mathrm{red}_{\mathfrak{R}}}$ does not change the ideals $\mathfrak{a}_i$, but every single entry a_{ij} of the matrix is reduced with the following procedure:

$$\left[\begin{array}{c} \begin{array}{ccc} \dots & \mathfrak{a}_j & \dots \end{array} \\ \begin{pmatrix} & \vdots & \\ \dots & a_{ij} & \dots \\ & \vdots & \end{pmatrix} \end{array}\right] \xrightarrow{\mathrm{red}_{\mathfrak{R}}} \left[\begin{array}{c} \begin{array}{ccc} \dots & \mathfrak{a}_j & \dots \end{array} \\ \begin{pmatrix} & \vdots & \\ \dots & a_{ij} + r_{ij} & \dots \\ & \vdots & \end{pmatrix} \end{array}\right] \quad \text{with} \quad r_{ij} \in \frac{\mathfrak{r}_i}{\mathfrak{a}_j},$$

which is obviously a valid reduction as in definition 4.8.1.

But how can we choose a good r_{ij} for which a *reduction* indeed takes place? In chapter 2 I specify the notion of reduce functions and give different algorithms. Now let a reduce function $\operatorname{mod}^{\mathrm{R}}$ (see definition 2.2.11) be fixed. Then

$$r_{ij} = \operatorname{mod}^{\mathrm{R}}_{\frac{\mathfrak{r}_i}{\mathfrak{a}_j}}(a_{ij}) - a_{ij}.$$

One–ideal reducer

$$\mathfrak{R} = \left[\begin{array}{c} \mathfrak{r} \quad \dots \quad \mathfrak{r} \\ \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \end{array}\right],$$

where $\mathfrak{r}$ is a fractional $\mathcal{D}$–ideal. This reducer is a special case of the general diagonal reducer. The algorithmic advantage of its simplicity is as follows:

A basic step in the reduction process

$$\left[\begin{array}{c} \dots \quad \mathfrak{a}_j \quad \dots \\ \begin{pmatrix} & \vdots & \\ \cdots & a_{ij} & \cdots \\ & \vdots & \end{pmatrix} \end{array}\right] \xrightarrow{\text{red}_{\mathfrak{R}}} \left[\begin{array}{c} \dots \qquad \mathfrak{a}_j \qquad \dots \\ \begin{pmatrix} & \vdots & \\ \cdots & a_{ij} + r_{ij} & \cdots \\ & \vdots & \end{pmatrix} \end{array}\right], \text{ where } \quad r_{ij} \in \frac{\mathfrak{r}}{\mathfrak{a}_j},$$

uses the ideal quotient $\frac{\mathfrak{r}}{\mathfrak{a}_j}$ to reduce the a_{ij}. Before we can do that, we have to compute the ideal quotient $\frac{\mathfrak{r}}{\mathfrak{a}_j}$ of $\mathfrak{r}$ and $\mathfrak{a}_j$. Ideal division is computationally relatively expensive.

For one column of $\mathfrak{M}$, we have to do only one ideal division instead of n. The algorithm 4.5.1 frequently changes only the entries of the matrix and not the ideals. A basic computational idea is not to divide the two ideals for every entry, but to store the ideal quotient until the ideal $\mathfrak{a}_j$ changes.

So in this case we have to store m ideals, which is moderate. Compare this to the general diagonal reducer case where we had to store mn ideals, which can be considered intolerable. The ideals are usually stored with a $\mathbb{Z}$–basis, which requires n^2 integers (and some more $\mathrm{O}(n)$ information which is unimportant) for a total of mn^3 integers. Compare this to a whole pseudomatrix which requires m ideals (n^2 integers each) and mn algebraic numbers (n integers each) which totals to $2mn^2$ integers. So storing the ideal quotients raises the memory complexity by one power!

If we have a general reducer

$$\mathfrak{R} = \left[\begin{array}{c} \mathfrak{r}_1 \quad \dots \quad \mathfrak{r}_n \\ \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \end{array}\right],$$

we can simply transform this to a one–ideal reducer

$$\mathfrak{R}' = \left[\begin{array}{c} \mathfrak{r} \quad \dots \quad \mathfrak{r} \\ \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \end{array}\right] \quad \text{where} \quad \mathfrak{r} := \operatorname{lcm}\{\mathfrak{r}_1, \dots, \mathfrak{r}_n\},$$

satisfying $\operatorname{Mod}(\mathfrak{R}') \subset \operatorname{Mod}(\mathfrak{R})$ because $\mathfrak{r} \subset \mathfrak{r}_i,\ \forall i \in \mathbb{N}_n$. As the price for the more efficient reduction method $\operatorname{Mod}(\mathfrak{R}')$ is possibly smaller than $\operatorname{Mod}(\mathfrak{R})$, this translates to less reduction power.

Rational reducer

$$\mathfrak{R} = \left[\begin{array}{ccc} r\mathcal{D} & \dots & r\mathcal{D} \\ \multicolumn{3}{c}{\begin{pmatrix} 1 & & \text{\Large 0} \\ & \ddots & \\ \text{\Large 0} & & 1 \end{pmatrix}} \end{array} \right],$$

where $r\mathcal{D}$ denotes the ideal generated by the rational number r.

The point in using this even simpler reducer is to have no ideal division at all. Instead of reducing an algebraic number with the ideal $\frac{r\mathcal{D}}{\mathfrak{a}}$, we reduce with the ideal

$$r\,\mathrm{den}(\mathfrak{a}) \subset \tfrac{r\mathcal{D}}{\mathfrak{a}},$$

where den() denotes the denominator of the ideal.

It is easy to see that this is correct because

$$\begin{array}{rl} & \mathfrak{a}\,\mathrm{den}(\mathfrak{a}) \quad \text{is an integral ideal} \\ \Longrightarrow & \mathfrak{a}\,\mathrm{den}(\mathfrak{a}) \subset 1\mathcal{D} \\ \Longrightarrow & \mathrm{den}(\mathfrak{a})\mathcal{D} \subset \tfrac{1}{\mathfrak{a}} \\ \Longrightarrow & r\,\mathrm{den}(\mathfrak{a})\mathcal{D} \subset \tfrac{r\mathcal{D}}{\mathfrak{a}}. \end{array}$$

So one point of this algorithm is that we save the ideal divisions. Another one is still more important: It is much easier to reduce an algebraic number with a rational number than to reduce with an ideal, which was described in subsection 2.3.6.

The method can be refined to increase the reduction power at the expense of some extra computation time:

Let e be the maximal natural factor of $\mathfrak{a}\,\mathrm{den}(\mathfrak{a})$ (see definition 1.3.5),

$$e = \max\bigl\{e \in \mathbb{N} \mid \mathfrak{a}\,\mathrm{den}(\mathfrak{a}) \subset e\mathcal{D}\bigr\}.$$

Then

$$\frac{r\,\mathrm{den}(\mathfrak{a})}{e}\mathcal{D} \subset \frac{r\mathcal{D}}{\mathfrak{a}},$$

which means that we may reduce even by the rational number $\frac{r\,\mathrm{den}(\mathfrak{a})}{e}$ instead of $r\,\mathrm{den}(\mathfrak{a})$.

If we have a one–ideal reducer

$$\mathfrak{R} = \left[\begin{array}{ccc} \mathfrak{r} & \dots & \mathfrak{r} \\ \multicolumn{3}{c}{\begin{pmatrix} 1 & & \text{\Large 0} \\ & \ddots & \\ \text{\Large 0} & & 1 \end{pmatrix}} \end{array} \right],$$

we have the rational reducer

$$\mathfrak{R}' = \left[\begin{array}{c} r\mathcal{D} \quad \dots \quad r\mathcal{D} \\ \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \end{array}\right],$$

where r is the minimum of the ideal $\mathfrak{r}$ (see definition 1.3.1). Again $\mathrm{Mod}\,(\mathfrak{R}')$ is possibly smaller than $\mathrm{Mod}\,(\mathfrak{M})$ which translates to less reduction power.

Experiments to compare the efficiency of the different reducers can be found in section 6.3.

4.8.3 Obtaining reducers

In important applications (like relative ideals), we can assume reducers to be known in advance. If this is not the case, we obtain a reducer with the method below. Unfortunately, the reducers obtained with this method are usually quite bad.

Proposition 4.8.3:

Let $\mathfrak{M}$ be an integral pseudomatrix with n rows and m columns. Let $\mathfrak{d}$ be the n–minor sum of $\mathfrak{M}$ (see definition 4.7.3). Then $\mathfrak{d}\mathcal{D}^n \subset \mathrm{Mod}\,(\mathfrak{M})$.

Proof. Since $\mathfrak{M}$ is integral, by lemma 4.1.5, $\mathrm{Mod}\,(\mathfrak{M}) \subset \mathcal{D}^n$. If $n > m$, the n–minor sum is zero, which produces a zero pseudomatrix as a reducer which is equivalent to no reduction at all. Thus, there is nothing to prove. The rank of $\mathfrak{M}$ must be n for $\mathfrak{d}$ not to be zero.

By proposition 4.7.7, the n–minor sum is not altered by applications of elementary transformations. Since we established the equivalence of module and transformation equivalence, we know that the n–minor sum of $\mathfrak{M}$ equals the n–minor sum of the CHNF $\mathfrak{M}'$ of $\mathfrak{M}$. Since $\mathfrak{M}$ was integral, so is its CHNF.

It remains to prove the theorem for square matrices in CHNF. Let

$$\mathfrak{M} = \left[\begin{array}{c} \mathfrak{a}_1 \quad \dots \quad \mathfrak{a}_n \\ \begin{pmatrix} 1 & & * \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \end{array}\right] \quad \text{and} \quad \mathfrak{d} = \prod_{i=1}^{n} \mathfrak{a}_i.$$

We have to show

$$\forall i \in \mathbb{N}_n, \quad \mathfrak{d}\mathsf{E}_i \subset \mathrm{Mod}\,(\mathfrak{M}),$$

where E_i denotes the i-th canonical vector in $\mathcal{D}^n$.

Since $\mathfrak{M}$ is integral and has at least one entry equal to one in every column, we have $\mathfrak{a}_i \subset \mathcal{D}$, for any $i \in \mathbb{N}_n$. Let $\mathfrak{d}_i := \prod_{j=1}^{i} \mathfrak{a}_j$. It is sufficient to prove

$$\forall i \in \mathbb{N}_n, \quad \mathfrak{d}_i\mathsf{E}_i \subset \mathrm{Mod}\,(\mathfrak{M}). \tag{4.8.2}$$

We prove that by induction over i, starting from 1.

For $i = 1$ we see that the first column of $\mathfrak{M}$ is identical to $\begin{bmatrix}\mathfrak{a}_1\\ \mathsf{E}_1\end{bmatrix}$, so there is nothing to prove.

Let 4.8.2 be proved for all $j \in \mathbb{N}_{i-1}$. Let the j-th entry of the i-th column of $\mathfrak{M}$ be denoted by c_j. Then the i-th column of $\mathfrak{M}$ can be written as:

$$\mathsf{A}_i = \mathsf{E}_i + \sum_{j=1}^{i-1} c_j \mathsf{E}_j.$$

Let $j \in \mathbb{N}_{i-1}$. Since $\mathfrak{M}$ is assumed to be integral, we have $\mathfrak{a}_i c_j \subset \mathcal{D}$. By the definition of $\mathfrak{d}_i$, we conclude

$$\begin{aligned}&\mathfrak{d}_i c_j \subset \mathfrak{d}_i \mathfrak{a}_i^{-1} \subset \mathfrak{d}_j \quad \text{and}\\ &\mathfrak{d}_i c_j \mathsf{E}_j \subset \mathfrak{d}_j \mathsf{E}_j \subset \operatorname{Mod}(\mathfrak{M}) \quad \text{by induction assumption.}\end{aligned}$$

By definition, $\mathfrak{a}_i \mathsf{A}_i \subset \operatorname{Mod}(\mathfrak{M})$. By definition of $\mathfrak{d}_i$, this gives $\mathfrak{d}_i \mathsf{A}_i \subset \operatorname{Mod}(\mathfrak{M})$.

Now

$$\mathfrak{d}_i \mathsf{E}_i = \mathfrak{d}_i \left(\mathsf{A}_i - \sum_{j=1}^{i-1} c_j \mathsf{E}_j \right) \subset \mathfrak{d}_i \mathsf{A}_i + \sum_{j=1}^{i-1} \mathfrak{d}_i c_j \mathsf{E}_j \subset \operatorname{Mod}(\mathfrak{M})$$

which finishes the induction step and the proof. □

Definition 4.8.4:

The **denominator** of a pseudomatrix $\mathfrak{M}$ (over an algebraic number field over $\mathbb{Q}$) is the minimal natural number d such that $d\mathfrak{M}$ is an integral pseudomatrix.

Corollary 4.8.5:

Let d be the denominator of a pseudomatrix $\mathfrak{M}$ with n rows and $\mathfrak{d}$ be the n–minor sum of $d\mathfrak{M}$. Then

$$\frac{\mathfrak{d}}{d}\mathcal{D}^n \subset \operatorname{Mod}(\mathfrak{M}).$$

In other words $\left[\begin{matrix} \frac{\mathfrak{d}}{d} & \cdots & \frac{\mathfrak{d}}{d} \\ \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \end{matrix}\right]$ is a proper one–ideal reducer for $\mathfrak{M}$.

The reducer obtained by this corollary is tested in section 6.4.

4.9 The module of pseudomatrices

Definition 4.9.1:

Let

$$\mathfrak{M} = \begin{bmatrix} \mathfrak{a}_1 & \cdots & \mathfrak{a}_m \\ \mathsf{A}_1 & \cdots & \mathsf{A}_m \end{bmatrix} \quad \text{and} \quad \mathfrak{N} = \begin{bmatrix} \mathfrak{b}_1 & \cdots & \mathfrak{b}_l \\ \mathsf{B}_1 & \cdots & \mathsf{B}_l \end{bmatrix}$$

be two pseudomatrices with the same number of rows. Addition is defined as the concatenation

$$\mathfrak{M} + \mathfrak{N} \;=_{\mathrm{Def}}\; \begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_m & \mathfrak{b}_1 & \dots & \mathfrak{b}_l \\ \mathsf{A}_1 & \dots & \mathsf{A}_m & \mathsf{B}_1 & \dots & \mathsf{B}_l \end{bmatrix}.$$

Let $a \in \mathcal{K}$. Scalar multiplication is defined as

$$a\mathfrak{M} \;=_{\mathrm{Def}}\; \begin{bmatrix} a\mathfrak{a}_1 & \dots & a\mathfrak{a}_m \\ \mathsf{A}_1 & \dots & \mathsf{A}_m \end{bmatrix}.$$

These definitions are consistent with addition and scalar multiplication of $\mathcal{D}$–modules:

$$\begin{aligned} &\mathrm{Mod}\,(\mathfrak{M}) + \mathrm{Mod}\,(\mathfrak{N}) = \mathrm{Mod}\,(\mathfrak{M} + \mathfrak{N}) \quad \text{and} \\ &a\mathrm{Mod}\,(\mathfrak{M}) = \mathrm{Mod}\,(a\mathfrak{M})\,. \end{aligned} \tag{4.9.1}$$

It is easily seen that the pseudomatrices with a fixed number of rows form a $\mathcal{D}$–module with these definitions.

4.10 Dual pseudomatrices and intersection of modules

Definition 4.10.1:
Let $\mathfrak{M} = \begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_n \\ & \mathbf{A} & \end{bmatrix}$ be a square pseudomatrix with n rows and rank n. Then the **dual pseudomatrix** is the pseudomatrix $\mathfrak{M}' = \begin{bmatrix} \mathfrak{a}_1^{-1} & \dots & \mathfrak{a}_n^{-1} \\ & \left(\mathbf{A}^{\mathrm{tr}}\right)^{-1} & \end{bmatrix}$.

For invertible square matrices $\mathbf{A}$ over fields we have $(\mathbf{A}^{\mathrm{tr}})^{-1} = (\mathbf{A}^{-1})^{\mathrm{tr}}$ and therefore $\mathfrak{M} = \mathfrak{M}''$.

Lemma 4.10.2:
Let $\mathfrak{M}$ be a pseudomatrix and $\mathsf{C} \in \mathcal{K}^n$. Then

$$\mathsf{C} \in \mathrm{Mod}\,(\mathfrak{M}) \iff \forall \mathsf{D} \in \mathrm{Mod}\,(\mathfrak{M}') : \mathsf{C}^{\mathrm{tr}}\mathsf{D} \in \mathcal{D}.$$

Proof. Let the notation of $\mathfrak{M}$ and $\mathfrak{M}'$ be as in definition 4.10.1.

Let $\mathsf{C} \in \mathcal{K}^n$. Since $\mathbf{A}$ is invertible, there are unique $a_i \in \mathcal{K}$, for $i \in \mathbb{N}_n$, such that

$$\mathsf{C} = \mathbf{A} \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}.$$

Then,

$$\begin{aligned}
& \mathsf{C} \in \operatorname{Mod}(\mathfrak{M}) \\
\Longleftrightarrow\ & \mathbf{A}\begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \in \operatorname{Mod}(\mathfrak{M}) \\
\Longleftrightarrow\ & \forall i \in \mathbb{N}_n, \quad a_i \in \mathfrak{a}_i \\
\Longleftrightarrow\ & \forall i \in \mathbb{N}_n, \forall b_i \in \mathfrak{a}_i^{-1} : \quad a_i b_i \in \mathcal{D} \\
\Longleftrightarrow\ & \forall b_i \in \mathfrak{a}_i^{-1}, \text{ where } \quad i \in \mathbb{N}_n, \quad \sum_{i=1}^{n} a_i b_i \in \mathcal{D} \\
\Longleftrightarrow\ & \forall b_i \in \mathfrak{a}_i^{-1}, \text{ where } \quad i \in \mathbb{N}_n, (a_1, \ldots, a_n)\mathbf{A}^{\mathrm{tr}}(\mathbf{A}^{\mathrm{tr}})^{-1}\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \in \mathcal{D} \\
\Longleftrightarrow\ & \forall b_i \in \mathfrak{a}_i^{-1}, \text{ where } \quad i \in \mathbb{N}_n, \left(\mathbf{A}\begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}\right)^{\mathrm{tr}}(\mathbf{A}^{\mathrm{tr}})^{-1}\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \in \mathcal{D} \\
\Longleftrightarrow\ & \forall \mathsf{B} \in \operatorname{Mod}(\mathfrak{M}'), \mathsf{C}^{\mathrm{tr}}\mathsf{B} \in \mathcal{D}
\end{aligned}$$

since

$$\operatorname{Mod}(\mathfrak{M}') = \left\{ (\mathbf{A}^{\mathrm{tr}})^{-1}\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \middle| \, b_i \in \mathfrak{a}_i^{-1}, \forall i \in \mathbb{N}_n \right\}.$$

□

Proposition 4.10.3:

Let $\mathfrak{M}$ and $\mathfrak{N}$ be two pseudomatrices with n rows and columns of rank n. Then

$$\operatorname{Mod}(\mathfrak{M}) \cap \operatorname{Mod}(\mathfrak{N}) = \operatorname{Mod}(\mathfrak{R}'),$$

where $\mathfrak{R}$ is the CHNF of the pseudomatrix $\mathfrak{M}' + \mathfrak{N}'$.

Proof. Let $\mathsf{C} \in \mathcal{K}$.

$$\begin{aligned}
& \mathsf{C} \in \operatorname{Mod}(\mathfrak{M}) \cap \operatorname{Mod}(\mathfrak{N}) \\
\Longleftrightarrow\ & \mathsf{C} \in \operatorname{Mod}(\mathfrak{M}) \quad \text{and} \quad \mathsf{C} \in \operatorname{Mod}(\mathfrak{N}) \\
\Longleftrightarrow\ & \forall \mathsf{B} \in \operatorname{Mod}(\mathfrak{M}'),\ \mathsf{C}^{\mathrm{tr}}\mathsf{B} \in \mathcal{D} \quad \text{and} \\
& \quad \forall \mathsf{E} \in \operatorname{Mod}(\mathfrak{N}'),\ \mathsf{C}^{\mathrm{tr}}\mathsf{E} \in \mathcal{D}, \text{ by lemma 4.10.2} \\
\Longleftrightarrow\ & \forall \mathsf{B} \in \operatorname{Mod}(\mathfrak{M}') + \operatorname{Mod}(\mathfrak{N}'),\ \mathsf{C}^{\mathrm{tr}}\mathsf{B} \in \mathcal{D} \\
\Longleftrightarrow\ & \forall \mathsf{B} \in \operatorname{Mod}(\mathfrak{M}' + \mathfrak{N}'),\ \mathsf{C}^{\mathrm{tr}}\mathsf{B} \in \mathcal{D}, \text{ by (4.9.1)} \\
\Longleftrightarrow\ & \mathsf{C} \in \operatorname{Mod}(\mathfrak{R}'), \text{ where } \mathfrak{R} \text{ is the normal form of } \mathfrak{M}' + \mathfrak{N}', \\
& \quad \text{by lemma 4.10.2.}
\end{aligned}$$

For the last conclusion we needed the fact that the rank of $\mathfrak{M}$ and $\mathfrak{N}$ is n which implies that the rank of $\mathfrak{M}' + \mathfrak{N}'$ also equals n. Then the normal form $\mathfrak{R}$ is square and has rank n, therefore lemma 4.10.2 may be applied. □

Chapter 5

Relative ideals

For an introduction to relative extensions in algebraic number fields see [BP91]. See [DP98] for applications which stress the importance of computations in relative extensions in connection with the computation of subfields in [Klü97].

The arithmetic of relative ideals is one important application of the normal form algorithm over algebraic number rings. In [Fri97] the round–two–algorithm is developed for the computation of a pseudobasis of the relative maximal order where the arithmetic of relative ideals plays an important role.

What is meant by relative ideals? A relative ideal is an ideal in an order of an algebraic number field as described in chapter 1. The terminology "relative" refers to a special presentation of this ideal using a nontrivial subfield. Relative ideal is a shorthand for ideal in relative representation.

5.1 Relative ideals in algebraic number fields

Let $\mathcal{K}$ be an algebraic number field over $\mathbb{Q}$ with finite degree $[\mathcal{K} : \mathbb{Q}] = m > 1$ and $\mathcal{L}$ be a finite algebraic extension of $\mathcal{K}$ with degree $[\mathcal{L} : \mathcal{K}] = n > 1$. Let $o_{\mathcal{K}}$ be the ring of integers of $\mathcal{K}$, which is a Dedekind domain.

Let $\mathcal{O}$ be an order of $\mathcal{L}$. Then $\mathcal{O}$ does not always have an $o_{\mathcal{K}}$–basis. But, at least $\mathcal{O}$ has a presentation

$$\mathcal{O} = \sum_{i=1}^{n} \mathfrak{c}_i\omega_i, \text{ where } \mathfrak{c}_i \text{ are fractional } o_{\mathcal{K}}\text{–ideals and } \omega_i \in \mathcal{L}, \tag{5.1.1}$$

which is a relative pseudobasis of $\mathcal{O}$. $\Omega = (\omega_1, \ldots, \omega_n)$ is also a basis for $\mathcal{L}$ as a $\mathcal{K}$–vector space.

An algebraic number $\alpha \in \mathcal{L}$ can be represented as

$$\alpha = \sum_{i=1}^{n} a_i\omega_i, \text{ where } \quad a_i \in \mathcal{K},$$

which is called the **relative representation of the algebraic number** α.

Definition 5.1.1:
An $\mathcal{O}$–ideal $\mathfrak{A}$ with at least one of the following presentations is called an **ideal in relative representation** or, shortly, a **relative ideal**.

pseudobasis presentation $\mathfrak{A} = \mathfrak{a}_1\alpha_1 + \cdots + \mathfrak{a}_n\alpha_n$, where $\mathfrak{a}_i$ is a fractional $o_{\mathcal{K}}$–ideal and $\alpha_i \in \mathcal{L}$ are algebraic numbers in relative representation for all $i \in \mathbb{N}_n$.

two–element presentation $\mathfrak{A} = \mathcal{O}\alpha_1 + \mathcal{O}\alpha_2$, where $\alpha_1, \alpha_2 \in \mathcal{L}$ are algebraic numbers in relative representation.

generalized two–element presentation $\mathfrak{A} = \mathcal{O}\mathfrak{a}_1\alpha_1 + \mathcal{O}\mathfrak{a}_2\alpha_2$, where $\mathfrak{a}_1$ and $\mathfrak{a}_2$ are fractional $o_{\mathcal{K}}$–ideals and $\alpha_1, \alpha_2 \in \mathcal{L}$ are algebraic numbers in relative representation.

5.2 Basic functions

Relative ideals can be seen as a generalization of the ideals as they are described in chapter 1, which are called **absolute ideals** in this chapter[1]. The presentations there are based on the presentation of integral algebraic numbers as vectors over $\mathbb{Z}$. This leads to the representation of integral ideals with matrices over $\mathbb{Z}$.

In this chapter we deal with algebraic numbers represented as vectors of algebraic numbers, which leads to the presentation of ideals as pseudomatrices of algebraic numbers and ideals. The main complication is that we cannot use matrices over $o_{\mathcal{K}}$ to represent relative ideals if $o_{\mathcal{K}}$ is not a principal ideal domain.

We are able to generalize the algorithms for absolute ideals, to relative ideals but quite a few problems have to solved.

5.2.1 The corresponding absolute order

Since $\mathcal{L}$ can be viewed as an algebraic number field over $\mathbb{Q}$, $\mathcal{O}$ has a $\mathbb{Z}$–basis

$$\mathcal{O} = \sum_{i=1}^{nm} \xi_i\mathbb{Z}, \text{ where } \quad \xi_i \in \mathcal{L} \quad \text{for} \quad i \in \mathbb{N}_{nm}. \tag{5.2.1}$$

Assumption 5.2.1. *Writing $\xi_i \in \mathcal{L}$ in formula* (5.2.1) *and $\omega_i \in \mathcal{L}$ in* (5.1.1) *actually means two different things: in the former, algebraic numbers are presented as vectors over $\mathbb{Q}$ and in the latter, as vectors with entries over $\mathcal{K}$, which are presented as vectors over $\mathbb{Q}$ themselves. In the sequel I will identify both presentations, assuming that the transformation between both is well–established.*

The computational connection between different representations of one algebraic number field is not trivial. First of all it depends on the way the algebraic number field is presented. See [Dab93] for the background of the ideas which have been implemented in KANT.

The basis $\Xi = (\xi_1, \ldots, \xi_{nm})$ can be obtained from the representation in formula (5.1.1). Let the $o_{\mathcal{K}}$–ideals $\mathfrak{c}_1, \ldots, \mathfrak{c}_n$ be given by $\mathbb{Z}$–bases

$$\mathfrak{c}_i = \sum_{j=1}^{m} c_{ij}\mathbb{Z}, \text{ where } \quad c_{ij} \in \mathcal{K} \quad \text{for} \quad i \in \mathbb{N}_n,\ j \in \mathbb{N}_m.$$

Then

$$\mathcal{O} = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} c_{ij}\mathbb{Z} \right) \omega_i = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}\omega_i\mathbb{Z}.$$

[1] Absolute ideals are relative ideals in the special case $\mathcal{K}=\mathbb{Q}$.

5.2.2 Transformation from relative to absolute ideals

Let the ideal $\mathfrak{A}$ be given by an $o_{\mathcal{K}}$–pseudobasis, $\mathfrak{A} = \mathfrak{a}_1\alpha_1 + \cdots + \mathfrak{a}_n\alpha_n$, where $\mathfrak{a}_i$ is a fractional $o_{\mathcal{K}}$–ideal and $\alpha_i \in \mathcal{L}$ for $i \in \mathbb{N}_n$. Let the $\mathfrak{a}_i$ be given by $\mathbb{Z}$–bases,

$$\mathfrak{a}_i = a_{i1}\mathbb{Z} + \cdots + a_{im}\mathbb{Z}, \text{ where } \quad a_{ij} \in \mathcal{K},\ i \in \mathbb{N}_n,\ j \in \mathbb{N}_m.$$

Then we have a $\mathbb{Z}$–generating system of $\mathfrak{A}$ as:

$$\mathfrak{A} = \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i a_{ij}\mathbb{Z}.$$

If $\mathfrak{A}$ is given in the generalized two–element presentation

$$\mathfrak{A} = \mathcal{O}\mathfrak{a}_1\alpha_1 + \mathcal{O}\mathfrak{a}_2\alpha_2,$$

where $\mathfrak{a}_1$ and $\mathfrak{a}_2$ are fractional $o_{\mathcal{K}}$–ideals and $\alpha_1, \alpha_2 \in \mathcal{L}$, then we have to use the pseudobasis of $\mathcal{O}$:

$$\mathcal{O} = \sum_{i=1}^{n} \mathfrak{c}_i\omega_i, \quad \mathfrak{c}_i \text{ fractional } o_{\mathcal{K}}\text{–ideals},\ \omega_i \in \mathcal{L}, \quad \text{for} \quad i \in \mathbb{N}_m.$$

Let the $o_{\mathcal{K}}$–ideals $\mathfrak{a}_1, \mathfrak{a}_2, \mathfrak{c}_1, \ldots, \mathfrak{c}_n$ be given by $\mathbb{Z}$–bases,

$$\mathfrak{a}_l = \sum_{j=1}^{m} a_{lj}\mathbb{Z}, \text{ where } \quad a_{lj} \in \mathcal{K} \quad \text{for} \quad l = 1, 2,\ j \in \mathbb{N}_m, \quad \text{and}$$

$$\mathfrak{c}_i = \sum_{j=1}^{m} c_{ij}\mathbb{Z}, \text{ where } \quad c_{ij} \in \mathcal{K} \quad \text{for} \quad i \in \mathbb{N}_n,\ j \in \mathbb{N}_m.$$

Then

$$\begin{aligned}\mathfrak{A} &= \left(\sum_{i=1}^{n}\omega_i\sum_{j=1}^{m}c_{ij}\mathbb{Z}\right)\left(\sum_{k=1}^{m}a_{1k}\mathbb{Z}\right)\alpha_1 + \left(\sum_{i=1}^{n}\omega_i\sum_{j=1}^{m}c_{ij}\mathbb{Z}\right)\left(\sum_{k=1}^{m}a_{2k}\mathbb{Z}\right)\alpha_2 \\ &= \sum_{l=1}^{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m} c_{ij}a_{lk}\alpha_l\omega_i\mathbb{Z}.\end{aligned}$$

which is a set of $2m^2n$ $\mathbb{Z}$–generators for $\mathfrak{A}$ which can be reduced to a $\mathbb{Z}$–basis with a HNF computation.

If the ideals $\mathfrak{a}_1$ and $\mathfrak{a}_2$ are trivial the above formula simplifies to

$$\mathfrak{A} = \sum_{l=1}^{2}\sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}\alpha_l\omega_i\mathbb{Z}$$

and only $2mn$ $\mathbb{Z}$–generators have to be considered.

5.2.3 Transformation from absolute ideals in relative ideals

If the ideal is given in two–element presentation, nothing is to be done using assumption 5.2.1. If the ideal is given as a $\mathbb{Z}$–basis, we have nm $\mathbb{Z}$–generators. These are also $o_{\mathcal{K}}$–generators. They can be reduced to a $o_{\mathcal{K}}$–pseudobasis with a CHNF computation.

5.3 Arithmetic in relative ideals

Let the relative ideals $\mathfrak{A}$ and $\mathfrak{B}$ be represented as

$$\mathfrak{A} = \mathfrak{a}_1\alpha_1 + \cdots + \mathfrak{a}_n\alpha_n \quad \text{and} \quad \mathfrak{B} = \mathfrak{b}_1\beta_1 + \cdots + \mathfrak{b}_n\beta_n.$$

Then

$$\mathfrak{A} + \mathfrak{B} = \mathfrak{a}_1\alpha_1 + \cdots + \mathfrak{a}_n\alpha_n + \mathfrak{b}_1\beta_1 + \cdots + \mathfrak{b}_n\beta_n.$$

This is a presentation with $2n$ summands. The normal form algorithm is able to find fractional $o_{\mathcal{K}}$–ideals $\mathfrak{c}_1, \ldots, c_n$ and $\gamma_1, \ldots, \gamma_n \in \mathcal{L}$ such that

$$\mathfrak{a}_1\alpha_1 + \cdots + \mathfrak{a}_n\alpha_n + \mathfrak{b}_1\beta_1 + \cdots + \mathfrak{b}_n\beta_n = \mathfrak{c}_1\gamma_1 + \cdots + \mathfrak{c}_n\gamma_n,$$

which gives us a pseudobasis of the relative ideal $\mathfrak{A} + \mathfrak{B}$.

With the same notations as above, we have for the product

$$\mathfrak{A}\mathfrak{B} = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathfrak{a}_i\mathfrak{b}_j\alpha_i\beta_j.$$

So we multiply each $\mathfrak{a}_i\mathfrak{b}_j$ and $\alpha_i\beta_j$, which gives us a presentation with n^2 summands. The normal form algorithm can reduce this to a pseudobasis.

We can also use the two other algorithms introduced in subsection 1.2.1 for relative ideals.

For the mixed multiplication, let $\mathfrak{B}$ be presented with an $o_{\mathcal{K}}$–basis as above and $\mathfrak{A} = \mathcal{O}\alpha_1 + \mathcal{O}\alpha_2$, where $\alpha_1, \alpha_2 \in \mathcal{L}$. $\mathfrak{B}$ being an ideal implies $\mathcal{O}\mathfrak{B} = \mathfrak{B}$. Therefore

$$\mathfrak{A}\mathfrak{B} = \mathcal{O}\alpha_1\mathfrak{B} + \mathcal{O}\alpha_2\mathfrak{B} = \alpha_1\mathfrak{B} + \alpha_2\mathfrak{B} = \sum_{k=1}^{2}\sum_{i=1}^{n} \mathfrak{b}_i(\alpha_k\beta_i).$$

which is to be reduced by the normal form algorithm.

One important prerequisite for the four generator multiplication method is the existence of the representation matrix for algebraic elements in relative extensions, see [Pau96]. Let $\mathfrak{A} = \mathcal{O}\alpha_1 + \mathcal{O}\alpha_2$ and $\mathfrak{B} = \mathcal{O}\beta_1 + \mathcal{O}\beta_2$, where $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \mathcal{L}$. Then

$$\mathfrak{A}\mathfrak{B} = \mathcal{O}\alpha_1\beta_1 + \mathcal{O}\alpha_2\beta_1 + \mathcal{O}\alpha_1\beta_2 + \mathcal{O}\alpha_2\beta_2.$$

An efficient algorithm for inversion and division of relative ideals in the relative maximal order is developed in [Fri97, pp. 93–98].

5.4 Computing the minimum of a relative ideal

Let $\mathfrak{A}$ be represented as $\mathfrak{A} = \mathfrak{a}_1\alpha_1 + \cdots + \mathfrak{a}_n\alpha_n$. We want to compute the $o_{\mathcal{K}}$–minimum ideal of definition 1.3.1 which will simply be called minimum in the following. The general method, in analogy to the algorithm 1.3.2, uses the fact that

$(\alpha_1, \dots, \alpha_n)$ is a $\mathcal{K}$–vector space basis for $\mathcal{L}$. Therefore there exist $b_i \in \mathcal{K}$, where $i \in \mathbb{N}_n$, such that

$$\sum_{i=1}^{n} b_i \alpha_i = 1.$$

Let $c \in \mathcal{K} \cap \mathfrak{A}$. Then $c = \sum_{i=1}^{n} c b_i \alpha_i$. Since $\mathfrak{a}_1\alpha_1 + \dots + \mathfrak{a}_n\alpha_n$ is indeed a pseudobasis this is equivalent to $\forall i \in \mathbb{N}_n : cb_i \in \mathfrak{a}_i$. We conclude that

$$\mathcal{K} \cap \mathfrak{A} = \bigcap_{i=1}^{n} \frac{\mathfrak{a}_i}{b_i}.$$

Since the α_i are given in the basis Ω, we are given a matrix M which satisfies $(\alpha_1, \dots, \alpha_n) = \Omega M$. Let the inverse of this matrix be $M^{-1} = \left(m_{ij}\right)_{i,j \in \mathbb{N}_n}$. If we can find $c_i \in \mathcal{K}$, where $i \in \mathbb{N}_n$, such that $\sum_{i=1}^{n} c_i \omega_i = 1$, then we have

$$\sum_{j=1}^{n} \alpha_j \sum_{i=1}^{n} c_i m_{ji} = 1,$$

which provides

$$b_j = \sum_{i=1}^{n} c_i m_{ji} \in \mathcal{K} \quad \text{with} \quad \sum_{i=1}^{n} b_i \alpha_i = 1.$$

If the basis Ω has the property $\omega_1 = 1$ and the matrix M is the matrix of a pseudomatrix in CHNF, the above formula for the minimum simplifies to:

$$\mathcal{K} \cap \mathfrak{A} = \mathfrak{a}_1.$$

So the only difficult part is to find a representation of the 1. How this can be done depends on the representation of the order $\mathcal{O}$. We can apply the normal form algorithm to a generating system of $\mathcal{O}$ which gives us another pseudobasis $\mathcal{O}$ with the first basis element equal to 1. We can use the basis transformation to get a representation of the 1 in the original representation.

5.5 Using the minimum for arithmetic

In analogy to section 1.4 we can improve the efficiency of ideal arithmetic by using the minimum ideals.

Let $\mathfrak{A}$ be a relative ideal and $\mathfrak{m}$ its $o_{\mathcal{K}}$–minimum ideal of definition 1.3.1. Then we have $\mathfrak{m}\mathcal{O} \subset \mathfrak{A}$. Therefore $\mathfrak{m}\mathcal{O}$ is a reducer of $\mathfrak{A}$ in the sense of definition 4.8.1.

If we add or multiply two relative ideals with the above methods, we get a generating pseudomatrix which is subjected to a normal form computation. The following lemma provides a good reducer for this normal form computation:

Lemma 5.5.1:

Let $\mathfrak{A}$ and $\mathfrak{B}$ be (fractional) ideals in the order $\mathcal{O}$. Then

$$\begin{aligned} &\min \mathfrak{A} + \mathfrak{B} \supset \min \mathfrak{A} + \min \mathfrak{B} \quad \text{and} \\ &\min \mathfrak{A}\mathfrak{B} \supset \min \mathfrak{A} \min \mathfrak{B}. \end{aligned} \tag{5.5.1}$$

5.6 Least common multiplier for relative ideals

For ideals $\mathfrak{A}$ and $\mathfrak{B}$ over a maximal order we have the property

$$\operatorname{lcm}(\mathfrak{A}, \mathfrak{B}) = \frac{\mathfrak{A}\mathfrak{B}}{\mathfrak{A} + \mathfrak{B}},$$

this can also be used to compute the lcm of relative ideals.

There is a more sophisticated method to compute the lcm using the algorithm developed in section 4.10. Let $\mathfrak{M}_1$ and $\mathfrak{M}_2$ be the pseudomatrices representing $\mathfrak{A}$ and $\mathfrak{B}$. Proposition 4.10.3 gives us a pseudomatrix $\mathfrak{N}$ which satisfies

$$\operatorname{Mod}(\mathfrak{N}) = \operatorname{Mod}(\mathfrak{M}_1) \cap \operatorname{Mod}(\mathfrak{M}_2).$$

Since $\operatorname{lcm}(\mathfrak{A}, \mathfrak{B}) = \mathfrak{A} \cap \mathfrak{B}$, we conclude that $\mathfrak{N}$ represents $\operatorname{lcm}(\mathfrak{A}, \mathfrak{B})$.

Chapter 6

Examples

6.1 General remarks

Not all the experiments of this chapter have been performed on the same computer. To compare computation times of different computers the test program “relidmark” is used. This idea is similar to “GAPstones”, which is used to compare the run time for GAP programs on different computers. The “relidmark” number is inversely proportional to the computation time. The test file contains creation of absolute and relative orders, computations of class groups, and multiplication of relative ideals.

The “GAPstones” number only measures the speed of integer arithmetics. Therefore it is independent on any improvements of the GAP system. In opposition, “relidmark” is dependent on the speed of various number theoretic algorithms in KANT.

The “relidmark” is normed to be equal to the “GAPstones” number on the HP series 700 computers which are mainly used for the development of KANT. The computers used for the experiments range from 74000 to 259000 relidmark.

6.2 Comparison of the different multiplication algorithms for absolute ideals

Let $p \in \mathbb{Z}[x]$ be an irreducible polynomial of degree $n \in \mathbb{N}$. A root ρ of this polynomial generates an integral domain $\mathcal{O} = \mathbb{Z}[\rho]$ which is a finitely generated $\mathbb{Z}$–module. Let $\mathcal{K}$ be the quotient field of $\mathcal{O}$. Then $\mathcal{O}$ is an order of $\mathcal{K}$, called an equation order. Let $o_{\mathcal{K}}$ be a maximal order of an algebraic number field $\mathcal{K}$. There exists a $\mathbb{Z}$–basis $\Omega = (\omega_1, \ldots, \omega_n)$ for $o_{\mathcal{K}}$.

This situation is called an *absolute extension* and is the starting point for most of the algebraic number theoretic computations in KANT. Algebraic numbers and ideals over $o_{\mathcal{K}}$ are represented in this basis Ω.

Aim

How does the time efficiency of different multiplication algorithms for absolute ideals depend on the degree of the algebraic number field and the size of the ideals?

The answer will give heuristics for the question of how to use the different multiplication algorithms to increase the overall time efficiency of the multiplication for absolute ideals.

Notations

Presentations of an ideal

The following abbreviations for ideal presentations, introduced in section 1.1 are used in the tables.

basis

The $\mathbb{Z}$–basis presentation, where the matrix representing the ideal is assumed to be in HNF.

2elt

The two–element presentation where one element is assumed to be rational, or

normal

normal presentation.

Transformations

Using one presentation of an ideal it is possible to compute another presentation. The following abbreviations for presentation transformations are used in the tables.

basis→2elt

produces a two–element presentation of an ideal given by a basis presentation,

2elt→basis

produces a basis presentation of an ideal given by a two–element presentation,

2elt→normal

produces a normal presentation of an ideal given by a two–element presentation, and

basis→normal

produces a normal presentation of an ideal given by a basis presentation.

Because a normal presentation is a special two–element presentation, the transformation *normal→2elt* is trivial and the transformation *normal→basis* can be considered identical to the transformation *2elt→basis* because there is no special algorithm.

Multiplication algorithms

Four different multiplication algorithms were discussed in subsection 1.2.1 which are tabulated as

basis mult

multiplication using the two $\mathbb{Z}$–bases of the ideals. From the algorithm it is clear that a permutation of the two factors does not make a difference in computation costs.

mixed mult

multiplication using one $\mathbb{Z}$–basis and one two–element presentation. From the algorithm, it is clear that a permutation of the two factors might make a difference in computation costs.

four mult

multiplication using the two–element presentations of both ideals. From the algorithm it is clear that a permutation of the two factors does not make a difference in computation costs.

normal mult

multiplication of two ideals in normal presentation. This multiplication might involve another transformation of the presentations whose computation costs are added to the actual multiplication costs (which are very low). Again it is clear that a permutation of the two factors does not make a difference in computation costs.

Design of the test

Six different number fields from degree 3 to 33 are used. They are given by their maximal orders. For every number field, several ideal test sets are used. Each ideal test set contains ideals of roughly the same computational difficulty. For every ideal test set the average of the computation times of randomly chosen ideals is used.

6.2.1 Example number fields

Number fields are tabulated by their degrees, only one number field for each degree is used.

The number field is defined by a root ρ of a polynomial over $\mathbb{Z}$. Because the different roots of an irreducible polynomial are algebraically equivalent, it is not important which of the different roots of the polynomial p the root ρ actually is. The integral basis (which is a basis of the maximal order $o_{\mathcal{K}}$ of $\mathcal{K}$) is given, where it does not include huge coefficients. It is expressed in the powers of ρ. For the complete example generation, given as KASH programs, see [Hop]. Note that the actual integral basis is not really relevant for the significance of the result since many randomly chosen ideals are used, which should eliminate any effects of peculiarities of the maximal order basis.

number field of degree 3 Let $\mathcal{K}$ be the number field $\mathbb{Q}[\rho]$, where ρ satisfies $\rho^3 + \rho^2 + 81\rho + 1 = 0$. It has discriminant -529444 and class number 104. The powers of ρ form a basis of the maximal order $o_{\mathcal{K}}$.

number field of degree 6 Let $\mathcal{K}$ be the number field $\mathbb{Q}[\rho]$, where ρ satisfies $\rho^6 + 3\rho^5 - 3115\rho^4 - 6235\rho^3 + 2271309\rho^2 + 13868999\rho - 219506499 = 0$. It has discriminant $11^4 \cdot 263513^4$ and class number 4.

The maximal order $o_{\mathcal{K}}$ has the basis

$$(1, \rho, \rho^2, \rho^3, \rho^4,$$
$$\omega = \tfrac{1004429297+909886177\rho+3560597177\rho^2+1574028943\rho^3+2829966955\rho^4+\rho^5}{3858758129}).$$

number field of degree 9 Let $\mathcal{K}$ be the number field $\mathbb{Q}[\rho]$, where ρ satisfies $\rho^9 - 30\rho^8 + 291\rho^7 - 835\rho^6 - 573\rho^5 - 2661\rho^4 + 5256\rho^3 + 3435\rho^2 + 90\rho - 10394$. It has discriminant $2^6 \cdot 3^{15} \cdot 5^3 \cdot 409^3$.

The maximal order $o_{\mathcal{K}}$ has a basis

$$(1, \rho, \rho^2, \rho^3, \rho^4, \rho^5, \rho^6,$$
$$\omega_8 = \frac{\rho^2 + \rho^3 + \rho^4 + \rho^5 + \rho^6 + \rho^7}{2}, \omega_9 =$$
$$\frac{\begin{pmatrix}192482556886936+63967537334938\rho+18683832251437\rho^2+119111891060334\rho^3+\\172319480755568\rho^4+19755909260520\rho^5+155411992372966\rho^6+62922522292402\rho^7+\rho^8\end{pmatrix}}{211316676965006}.$$

number field of degree 12 Let $\mathcal{K}$ be the number field $\mathbb{Q}[\rho]$ where ρ satisfies $\rho^{12} - 2\rho^{11} + 4\rho^{10} - 8\rho^9 + 13\rho^8 + 53\rho^7 + 120\rho^6 - 100\rho^5 + 168\rho^4 - 46\rho^3 - 12\rho^2 + 14\rho + 7 = 0$. It has discriminant $5 \cdot 7 \cdot 101 \cdot 137 \cdot 22119145456439547247253650048219957 31$. The powers of ρ form a basis of the maximal order $o_{\mathcal{K}}$.

number field of degree 18 Let $\mathcal{K}$ be the number field $\mathbb{Q}[\rho]$, where ρ satisfies $\rho^{18} + 103\rho^{17} + 5654\rho^{16} + 208051\rho^{15} + 5656080\rho^{14} + 118519143\rho^{13} + 1952386178\rho^{12} + 25254464067\rho^{11} + 253773392888\rho^{10} + 1934686349631\rho^9 + 10684964678644\rho^8 + 38972994689559\rho^7 + 110317002224976\rho^6 + 47679505774513\rho^5 + 6617690323691\rho^4 + 1913538554456\rho^3 + 1118923004758\rho^2 - 222202371528\rho + 9309104652 = 0$. The field has discriminant $-2^4 \cdot 3^{10} \cdot 7^{12} \cdot 53^6 \cdot 400237 \cdot 737836479150440729697149699395082586 43$.
The basis of the maximal order is not given explicitly here because it would fill several pages.
This number field is also presented as a relative extension. In subsection 6.3.1 it is labeled "3 over 6".

number field of degree 25 Let $\mathcal{K}$ be the number field $\mathbb{Q}[\rho]$, where ρ satisfies $\rho^{25} - 10\rho^{23} + 16\rho^{22} + 108\rho^{21} - 15\rho^{20} - 790\rho^{19} + 1072\rho^{18} + 5124\rho^{17} - 5608\rho^{16} - 7984\rho^{15} - 758\rho^{14} + 93156\rho^{13} + 37420\rho^{12} - 240436\rho^{11} - 101240\rho^{10} + 182046\rho^9 + 2012960\rho^8 + 16972\rho^7 - 2449224\rho^6 - 3922137\rho^5 + 1881886\rho^4 + 21697150\rho^3 + 18723708\rho^2 + 25162760\rho - 6466833 = 0$. The field has discriminant $-2^{20} \cdot 3 \cdot 11^{20} \cdot 17^5 \cdot 19 \cdot 547 \cdot 4230289192683304949018485547006455230717 3943$.
Again the basis of the maximal order will not be given.
This number field is also presented as a relative extension. In subsection 6.3.1 it is labeled "5 over 5".

number field of degree 33 Let $\mathcal{K}$ be the number field $\mathbb{Q}[\rho]$, where ρ satisfies $\rho^{33} - 84\rho^{29} + 171\rho^{27} + 1347\rho^{25} + 3\rho^{24} - 6300\rho^{23} + 6\rho^{22} + 37565\rho^{21} + 210\rho^{20} - 166482\rho^{19} - 918\rho^{18} + 305595\rho^{17} - 4896\rho^{16} - 188758\rho^{15} + 22650\rho^{14} + 7323\rho^{13} - 31791\rho^{12} + 939\rho^{11} + 13872\rho^{10} - 290\rho^9 + 687\rho^8 - 1455\rho^7 - 105\rho^6 + 696\rho^5 + 90\rho^4 + 12\rho^3 + 12\rho^2 + 8 = 0$. The basis of the maximal order and the discriminant will not be given.
This number field is also presented as a relative extension. In subsection 6.3.1 it is labeled "11 over 3".

6.2.2 Ideal test sets

The ideals to compare the run times are chosen randomly from ideal test sets. The ideal test sets contain ideals of a similar computational difficulty. They are produced in a generalized process as follows.

We start with the list of all prime ideals over the 11 smallest prime numbers which range from 2 to 37. This list of ideals is the first test set which is denoted as test set 0. To produce the test set 1 each ideal of the test set 0 is multiplied with one randomly chosen ideal of the test set 0. The test set 1 contains the same number of ideals as the test set 0, and each factors into 2 prime ideals.

In the same way, the test set 2 is produced. Thus, it contains ideals which factor into 4 prime ideals and so forth. This process yields test sets n, where $n \in \mathbb{Z}^{\geq 0}$, each containing only ideals which factor in 2^n prime ideals. A test set with even simpler ideals is called test set $0'$. It contains the smallest 12 ideals from test set 0. Usually, it contains prime ideals over 2, 3, 5, and 7.

The decomposition behaviour of the ideals in one test set is very similar. This is not a problem for our purposes.

6.2.3 Number of repetitions per test set

For small examples the computation time is too small to be measured confidently. Randomly chosen ideals from the same ideal test sets might differ in their computational difficulty. For both reasons the process of choosing 2 ideals from a given test set randomly and measuring the computation time is repeated. Any computation time entry in the table always refers to the average computation time of those repetitions.

The number of repetitions per test set is chosen such that the total running time of a test series does not exceed a few minutes since longer computations tend to fragment the memory, which slow down further computations. This may blur the results of the test.

6.2.4 Computation times

The table contains two sorts of computation times, times for actual multiplications (assuming that the necessary presentation of the ideals are already given) and times for presentation transformations.

A complex multiplication method may either use the available presentations or transform one or both presentations to apply other algorithms, whatever is faster. The preferable complex multiplication method for each of the ideal test sets is given in the computation time table by the *typeface of the multiplication computation times* as follows:

boldface This is the least average computation time. The method belonging to this value is to be preferred.

italic The method belonging to this value should not be used. Even if the necessary presentation for another method is not given, it is worth it to transform the presentation.

normal face The method belonging to this value should be used if the necessary presentations are available and the necessary presentations for a faster method are not.

Blank spaces mean that the particular method was not included in the test run.

All times are given in milliseconds. A HP9000 series 700 computer with relidmark 124000 (see subsection 6.1) was used for the experiment.

number field identified by its degree										
	number of the ideal test set									
		repetitions per ideal test set								
			time (in *ms*) for the multiplication type				time (in *ms*) for the transformation			
			basis	*mixed*	*four*	*normal*	*basis*→ *2elt*	*2elt*→ *basis*	*2elt*→ *normal*	*basis*→ *normal*
3	$0'$	1000	**0.99**	1.16	1.56	1.00	1.16	0.55	0.75	1.66
3	0	1000	1.22	1.04	1.65	**0.49**	0.80	1.94	13.26	1.80
3	1	500	1.36	**1.28**	2.16	*4.46*	1.20	3.02	13.20	2.78
3	2	200	2.90	**2.55**	4.35	*21.60*	1.50	3.10	11.95	4.15
3	3	100	5.6	**4.0**	*7.7*	*55.8*	4.5	2.4	12.7	13.6
3	4	100	10.9	**7.7**	*16.8*	*80.9*	5.8	3.7	20.4	31.4
3	5	100	18.4	**14.1**	*28.4*	*157.1*	9.5	4.0	40.9	104.2
3	6	50	35.6	**26.6**	*54.8*	*345.6*	19.0	6.2	85.4	231.4
3	7	50	81.2	**60.2**	*125.0*	*1073.8*	32.0	9.6	316.0	508.6
3	8	50	224.6	**165.2**	*347.2*	*1705.0*	71.6	19.2	1050.6	1686.2
3	9	20	741.5	511.0	*1077.0*	**8.0**	387.5	56.5	3010.0	2793.0
3	10	10	*2450*	1714	*3485*	**19**	640	182	1310	2234
3	11	10	*9236*	5945	*12648*	**58**	5287	684	24437	25307
6	$0'$	1000	*7.35*	**4.59**	5.94	*11.43*	2.51	1.61	4.56	8.63
6	0	500	8.18	**5.00**	6.96	*9.92*	3.26	6.22	17.54	9.22
6	1	200	*11.15*	**6.75**	11.20	*49.10*	3.05	8.55	17.75	9.70
6	2	100	*33*	**15.3**	*25.9*	*88.3*	4.9	9.2	18.6	13.6
6	3	100	*60.9*	**25.2**	*44.7*	*101.9*	8.9	11.1	24.6	24.4
6	4	100	*121.2*	**43.8**	*83.8*	*206.4*	18.0	15.7	40.5	49.4
6	5	50	*217.0*	**76.0**	*148.6*	*270.6*	31.0	21.0	61.2	90.2
6	6	20	*420.0*	**144.0**	*283.0*	*414.5*	73.5	36.0	600.0	190.0
6	7	10	*983*	**331**	*660*	*645*	131	65	204	378
6	8	10	*2546*	882	*1755*	**9**	432	142	2469	1302
6	9	10	*8107*	2727	*5408*	**22**	859	411	10319	1829
6	10	10	*26005*	*9037*	*17472*	**43**	3582	1352	2112	5266
9	$0'$	200	23.90	**15.05**	16.35	*100.60*	11.50	3.30	14.90	29.80
9	0	200	*27.15*	**18.45**	24.75	*225.35*	7.30	13.55	33.05	26.75
9	1	100	35.6	**22.6**	32.1	*909.5*	14.3	13.4	33.6	36.1
9	2	50	*53.6*	**27.6**	37.6	*1226.8*	18.8	10.0	28.2	54.6
9	3	50	*160.4*	**52.0**	*83.6*	*1341.8*	60.2	16.4	34.0	86.0
9	4	20	*254.5*	**76.0**	*126.5*	*2150.0*	126.0	22.5	248.0	408.0
9	5	20	*486.5*	**130.5**	*232.0*	*232.5*	211.5	35.5	92.5	1217.0
9	6	10	929	233	*428*	**223**	1044	56	151	651
9	7	10	*2029*	493	*913*	**182**	923	112	592	1576
9	8	10	*5151*	1225	*2298*	**11**	2378	248	726	308315
12	$0'$	100	*48.9*	12.3	*16.5*	**6.6**	9.9	5.3	2.9	20.9
12	0	100	*53.0*	**12.3**	19.4	*60.0*	8.1	7.3	11.0	19.8
12	1	100	*71.5*	**20.3**	*42.8*	*514.4*	10.8	18.4	22.9	26.4
12	2	50	*239.6*	**52.8**	*111.8*	*315.4*	17.0	20.4	28.0	38.0
12	3	20	*476.5*	**95.5**	*191.0*	*740.0*	51.0	32.5	63.0	91.0
12	4	20	*1078.5*	**197.5**	*373.5*	*7268.5*	126.5	60.5	145.5	283.0
12	5	10	*2149*	**375**	*715*	*16711*	368	100	258	660
12	6	10	*4046*	**695**	*1325*	*6712*	621	173	557	1181
12	7	10	*8696*	**1542**	*2821*	*4724*	1161	337	1070	2710
12	8	5	*24518*	**4184**	*7606*	*18946*	3038	780	3694	4530

number field identified by its degree										
	number of the ideal test set									
		repetitions per ideal test set								
			time (in *ms*) for the multiplication type				time (in *ms*) for the transformation			
							basis→	*2elt*→	*2elt*→	*basis*→
			basis	*mixed*	*four*	*normal*	*2elt*	*basis*	*normal*	*normal*
18	0′	100	*162.0*	*154.2*	**82.6**	*351.9*	27.9	16.4	48.7	76.0
18	0	50	*194.8*	*157.6*	**87.2**	*260.0*	28.0	18.8	42.0	81.8
18	1	20	*282.5*	*171.5*	**132.0**	*1341.5*	36.5	26.0	70.0	96.5
18	2	20	*582.5*	**219.0**	223.0	*1779.5*	58.0	34.5	70.0	128.5
18	3	20	*1655.5*	**382.0**	*472.0*	*3495.5*	125.0	73.0	132.5	265.0
18	4	10	*2557*	**530**	*670*	*5629*	322	136	156	430
18	5	5	*4748*	**836**	*1170*	*70086*	730	214	268	1482
18	6	5	*7794*	**1216**	*1838*	1760	4078	322	602	10216
18	7	5	*19438*	2742	*4500*	**40**	5548	734	1306	2642
18	8	5	*47632*	6364	*10732*	**68**	6782	1366	4432	5482
25	0′	100	*499.9*	*520.6*	**147.7**	*301.3*	116.2	32.6	35.5	184.7
25	0	50	*595.6*	*526.4*	**164.8**	*372.0*	67.2	36.2	37.2	158.8
25	1	20	*837.0*	*560.5*	**245.0**	*1302.5*	134.5	61.5	103.0	266.0
25	2	20	*2006.0*	*672.5*	**486.0**	*9155.5*	125.0	72.5	164.0	343.5
25	3	10	*4854*	**996**	1032	*40245*	262	105	229	478
25	4	5	*8064*	**1382**	1524	*106222*	1014	250	638	2072
25	5	10	*14716*	**2236**	2597		4409	524		
25	6	5	*27442*	**3512**	*4698*		5540	800		
33	0′	50	*2191.6*	*1549.2*	**459.6**	*3586*	260	85.4	498.2	1887.4
33	0	20	*2293.5*	*1560*	**348.5**	*1716*	305	74	160	1987
33	1	10	*3072*	*2005*	**1242**	*134462*	283	251	1549	2081
33	2	5	*6854*	*2378*	**1838**	*353198*	366	300	2050	2218
33	3	10	*16426*	3281	**2968**		980	432		
33	4	10	*27442*	5673	**4366**		6612	981		
33	5	5	*50616*	**7658**	7714		4146	1476		

Discussion

The following heuristic rules for ideal multiplication strategies can be drawn from the table.

- For number fields of degree up to about 15 the mixed presentation method is usually best. If the necessary presentations are not available, it is worth it to compute a presentation, usually. The four multiplication method should be avoided.
- In number fields from about degree 15 on, the four multiplication method is best for smaller ideals and the mixed multiplication for larger ideals. If the necessary presentations are not available it is worth computing the presentations since the *basis*→*2elt* and *2elt*→*basis* are relatively fast.
- For small number fields and small ideals the transformations are expensive and the differences between the *basis, mixed, and four* multiplication are small. The multiplication should be done with the available transformations.
- Normal multiplication should not be used as a general method for multiplication. It is occasionally very fast but usually much slower than the other methods.

- The very fast computation times for the test ideal sets of around 8 have the following explanation. The ideal test sets contain only ideals whose minimum is a power product of prime numbers up to 37. A product of 256 prime ideals (referring to test set number 8) is very likely to have all those prime numbers dividing its minimum. If ideal $\mathfrak{a}$ of this test set is given in normal presentation, it is actually P–normal, where P is the set of the prime numbers dividing the minimum of $\mathfrak{a}$.
  The normal multiplication has the two difficult problems of finding normal presentations and making the normal presentation of two ideals compatible. After this, the actual multiplication is very cheap. Normal presentations of ideals in ideal test set 8 are very likely to be compatible already. Therefore the small computation time is more or less an artifact of the way the ideal test sets are produced.
- The small time for the actual multiplication is leveled off by the large time to produce the normal presentation.
- There is one exception to the previous statement: the order of degree 6, ideal test set 10. In this case the normal multiplication is the fastest no matter in which presentation the ideals are given. Under similar circumstances a single normal multiplication is usually quite fast compared with the other multiplication methods. But for some ideals it is very hard to find a normal presentation resulting in an enormous computation time which raises the average time considerably. (This happened in the case of the order of degree 6, ideal test set 9, the time for the transformation *2elt→normal*.)
- In some rows of the table the time for the transformation *basis→normal* is larger than the time for the transformations *basis→2elt* and *2elt → normal* combined. Obviously, the transformation *basis→normal* can be replaced by the transformations *basis→2elt* and *2elt→normal*. The implemented algorithms to compute *2elt* and *normal* presentations include random choices. In this case the effect of lucky/unlucky choices on the computation time is very strong since it is expensive to check if the elements in question indeed generate the ideal/are in normal presentation. The trouble is that it can not be determined *in advance* which is faster: *basis→normal* or *basis→2elt* and *2elt→normal* combined.

6.3 Comparison of different normal form algorithms with reduction using relative ideal arithmetic

Aim

To compare the computation times for the different normal form algorithms implemented in KANT on pseudomatrices with good reducers known in advance (like in relative ideal arithmetic).

Design

The overwhelming part of the computation costs of relative ideal arithmetic are consumed by the normal form computation. Thus, the time for the relative ideal multiplication instead of the normal form computation time is used, for convenience.

The actual modules fed into the normal form algorithm result from the basis multiplication algorithm (marked as test operation *) and of the basis addition (marked as test operation $+$), described in section 5.3.

Let n be the relative degree of the relative extension. The pseudomatrix whose CHNF is to be computed, for

- addition, has dimension $n \times 2n$ and is very sparse, $n(n-1)$ entries are zero, since it is the concatenation of two matrices already in CHNF;
- multiplication, has dimension $n \times n^2$ and is very dense since its columns are representations of products of algebraic numbers. Depending on the size of entries of the multiplication table for the order, its entries can be large even if the ideals are relatively small.

Compared methods

The following methods were compared and tabulated using the given abbreviations.

C1 Cohen algorithm (algorithm 4.5.1) with a one–ideal reducer (see subsection 4.8.2).

B1 Bosma–Pohst algorithm (algorithm 4.5.3) with a one–ideal reducer (see subsection 4.8.2).

C Cohen algorithm (algorithm 4.5.1) without reduction

B Bosma–Pohst algorithm without reduction

Cg Cohen algorithm (algorithm 4.5.1) with a general diagonal reducer (see subsection 4.8.2).

Cr Cohen algorithm (algorithm 4.5.1) with a rational reducer (see subsection 4.8.2).

6.3.1 Example relative number fields

The general situation is as follows:

Let $\mathcal{K} = \mathbb{Q}[\rho]$ be an algebraic number field, where ρ satisfies a certain integral polynomial equation with rational coefficients of degree n. Let $\Omega = (\omega_1, \ldots, \omega_m)$ be a $\mathbb{Z}$–basis of the maximal order $o_{\mathcal{K}}$ of the number field $\mathcal{L}$.

Let $\mathcal{L} = \mathcal{K}[\sigma]$ be an algebraic number field extension of $\mathcal{K}$, where σ satisfies a certain polynomial equation of degree m with coefficients in $\mathcal{K}$. The maximal order $o_{\mathcal{L}}$ of $\mathcal{L}$ does not always have a $\mathcal{K}$–basis, but at least a $\mathcal{K}$–pseudobasis $o_{\mathcal{L}} = \mathfrak{c}_1\xi_1 + \cdots + \mathfrak{c}_m\xi_m$, where $\mathfrak{c}_i$ is a fractional $\mathcal{D}$–ideal and $\xi \in \mathcal{L}$ for $i \in \mathbb{N}_m$. The ξ_i can be expressed in the powers of σ. $o_{\mathcal{L}}$ also has a $\mathbb{Z}$–basis of degree nm. This is the corresponding absolute extension.

In the examples, the relative orders used are tabulated by the degrees n and m as m over n, for each degree combination, only one number field tower is used.

relative number field of degree 3 over 3 Let $\mathcal{K}$ be the number field $\mathbb{Q}[\rho]$, where ρ satisfies $\rho^3 - 10\rho^2 - 3\rho - 2 = 0$. It has discriminant -8180 and class number 2. The powers of ρ form a basis of the maximal order $o_{\mathcal{K}}$.
Let $\mathcal{L}$ be the number field $\mathcal{K}[\sigma]$, where σ satisfies $\sigma^3 - 3 = 0$. It has the relative discriminant $243o_{\mathcal{K}}$. The powers of σ form a basis of the maximal order $o_{\mathcal{L}}$.
The absolute representation of this number field can be found in subsection 6.2.1 as the number field of degree 9.

relative number field of degree 3 over 6 Let $\mathcal{K}$ be the number field $\mathbb{Q}[\rho]$, where ρ satisfies $\rho^6 + 5\rho^5 - 6\rho^4 - 53\rho^3 + 3\rho^2 + 206\rho + 244 = 0$. It has discriminant -182099043 and class number 18. A basis of $o_{\mathcal{K}}$ is $(1, \rho, \rho^2, \rho^3, \omega_5 = \frac{\rho^4+\rho}{2}, \omega_6 = \frac{\rho^5+596\rho^4+140\rho^3+487\rho^2+120\rho+1256}{2740})$.
Let $\mathcal{L}$ be the number field $\mathcal{K}[\sigma]$, where σ satisfies $\sigma^3 + (18+\rho)\sigma^2 + (13 + 6\rho + 34\rho^2 + \rho^3)\sigma + 51 + 25\rho + \rho^2 = 0$. It has the relative discriminant $(-460552395 + 207692333\rho - 184201393\rho^2 - 58798978\rho^3 - 461579366\omega_5 + 1056396040\omega_6)o_{\mathcal{K}}$. The powers of σ form a basis of the maximal order $o_{\mathcal{L}}$.

relative number field of degree 6 over 3 Let $\mathcal{K}$ be as in the number field denoted with 3 over 3. Let $\mathcal{L}$ be the number field $\mathcal{K}[\sigma]$, where σ satisfies $\sigma^6 + (1+\rho)\sigma^5 + (-1 + 2\rho - 4\rho^2)\sigma^4 + (5 - 5\rho - 11\rho^2)\sigma^3 + (1 + 3\rho^2)\sigma^2 + (1 + \rho - 3\rho^2)\sigma + (-\rho - \rho^2) = 0$. It has the relative discriminant $(319743143192792 + 1351078531895460\rho - 134055393145420\rho^2)o_{\mathcal{K}}$.
The maximal order $o_{\mathcal{L}}$ has no basis in this case but a pseudobasis:
$o_{\mathcal{L}} = o_{\mathcal{K}} + \sigma o_{\mathcal{K}} + \sigma^2 o_{\mathcal{K}} + \sigma^3 o_{\mathcal{K}} + \sigma^4 o_{\mathcal{K}} + (\sigma + \sigma^2 + \sigma^4 + \sigma^5)(o_{\mathcal{K}} + \frac{\rho+\rho^2}{2} o_{\mathcal{K}})$.
The absolute representation of this number field can be found in subsection 6.2.1 as the number field of degree 18.

relative number field of degree 5 over 5 Let $\mathcal{K}$ be the number field $\mathbb{Q}[\rho]$, where ρ satisfies $\rho^5 + \rho^4 - 4\rho^3 - 14\rho^2 + 3\rho + 1 = 0$. It has discriminant -3982352 and class number 5. A basis of $o_{\mathcal{K}}$ is $(1, \rho, \rho^2, \omega_4 = \frac{\rho^3-\rho^2-\rho-1}{2}, \omega_5 = \frac{\rho^4+1}{2})$.
Let $\mathcal{L}$ be the number field $\mathcal{K}[\sigma]$, where σ satisfies $x^5 - 2\sigma^3 + (2 + \rho - \omega_5)\sigma^2 + (13 + 6\rho + \rho^2)\sigma + (25 + \rho - 2\rho^2 - \omega_5) = 0$. It has the relative discriminant $(859894333 + 788062556\rho - 3775373724\rho^2 + 405937120\omega_4 + 1052770984\omega_5])o_{\mathcal{K}}$.
The powers of σ form a basis of the maximal order $o_{\mathcal{L}}$.
The absolute representation of this number field can be found in subsection 6.2.1 as the number field of degree 25.

relative number field of degree 11 over 3 Let $\mathcal{K}$ be the number field $\mathbb{Q}[\rho]$, where ρ satisfies $\rho^3 + 42\rho + 154 = 0$. It has discriminant -936684 and class number 27. The powers of ρ form a basis of the maximal order $o_{\mathcal{K}}$.
Let $\mathcal{L}$ be the number field $\mathcal{K}[\sigma]$, where σ satisfies $\sigma^{11} + (-\rho + \rho^2)\sigma^7 + (1 - 2\rho^2)\sigma^5 + \sigma^3 + (1+\rho)\sigma^2 + 2 = 0$. It has the relative discriminant $(-502660717878142543753299537122 0 - 2214443634236656662286177134952\rho - 1814411876700860414909860841 20\rho^2)o_{\mathcal{K}}$.
The maximal order $o_{\mathcal{L}}$ has no basis in this case but only a pseudobasis:
$o_{\mathcal{L}} = o_{\mathcal{K}} + \sigma o_{\mathcal{K}} + \sigma^2 o_{\mathcal{K}} + \sigma^3 o_{\mathcal{K}} + \sigma^4 o_{\mathcal{K}} + \sigma^5 o_{\mathcal{K}} + \sigma^6 o_{\mathcal{K}} + \sigma^7 o_{\mathcal{K}} + \sigma^8 o_{\mathcal{K}} + \sigma^9 o_{\mathcal{K}} + (\sigma + \sigma^2 + \sigma^4 + \sigma^{10})(o_{\mathcal{K}} + \frac{\rho^2}{2} o_{\mathcal{K}})$.
The absolute representation of this number field can be found in subsection 6.2.1 as the number field of degree 33.

6.3.2 Ideal test sets

The ideal test sets are generated in analogy to the sets for absolute ideals in subsection 6.2.2.

Again 0 refers to prime ideals over small prime numbers (viewed as absolute ideals). 1 refers to products of 2 prime ideals, 2 to products of 4 prime ideals and so on.

As described in 6.2.3 two ideals are chosen randomly from the ideal test set a certain number of times (entered in a separate column). The average of the computation times of all repetitions is used for the particular method.

6.3.3 Relative times

To improve the clarity of the table it does not contain the computation times for each of the methods directly but as factors in relation to the fastest method. The fastest method can be identified as the column with the relative time 1. The actual time (in milliseconds) of the fastest method is entered in a separate column.

Note that in these test different computers were used. Thus, the computation time of the fastest method must be seen in relation to the speed of the computer (column marked relidmark, see subsection 6.1).

If no factor is entered in the table the particular method was excluded from the test run, for two possible reasons:

- Methods likely to be very slow were excluded, such that either a complete test run was possible at all, or such more repetitions were possible to increase the reliability of the result.
- Method “Cg” was occasionally excluded because of its high correlation to the “C1” method.

Note that on several occasions two test runs have been performed for the same ideal test set. The first run included all methods, with only a small repetition count possible. The second run excluded the slower methods to obtain a higher repetition number.

base field degree	relative degree	ideal test set	test operation	time of the fastest method for this example (in ms)	relidmark of the computer used, *1000	repetition count	time in relation to the fastest method					
							C1	B1	C	B	Cg	Cr
3	3	0	+	7.7	74	400	6.03	11.93	4.59	5.31	4.85	1
3	3	1	+	18.0	74	400	2.97	5.82	2.36	2.85	2.62	1
3	3	2	+	49.1	74	400	1.60	2.01	1	1.18	1.71	1.08
3	3	3	+	129.0	74	400	1.68	2.41	1	1.11	2.19	1.4
3	3	4	+	270.3	74	400	1.67	3.18	1	2.21	2.38	1.54
3	3	5	+	589.6	74	400	1.72	16.80	1	10.38	2.57	1.53
3	3	6	+	393.8	234	211	2	15.71	1	9.21	2.89	1.73
3	3	0	*	121	74	400	1.78	1.84	1.68	1	2.04	2.01
3	3	1	*	276.2	74	400	1.35	1.41	1.64	1	1.67	1.57
3	3	2	*	839.4	74	400	1	1.47	1.48	1.34	1.43	1.06
3	3	3	*	522.5	234	400	1.07	4.54	1.79	3.46	1.52	1
3	3	4	*	943.1	234	267	1.19	9.53	2.08	6.8	1.73	1
3	3	5	*	1722	234	129	1.15	12.11	2.39	8.44	1.69	1
3	3	6	*	4232	234	66	1.11	6.82	2.91	8.6	1.63	1

base field degree	relative degree	ideal test set	test operation	time of the fastest method for this example (in ms)	relidmark of the computer used, *1000	repetition count	time in relation to the fastest method					
							C1	B1	C	B	Cg	Cr
6	3	0	+	18.8	100	400	10.42	16.68	5.99	10.95	6.8	1
6	3	1	+	62.9	100	400	4.39	6.77	2.41	9.36	3	1
6	3	2	+	445	100	400	1.69	2.5	1	6.52	1.71	1.01
6	3	3	+	765.4	230	113	1.46	3.4	1	10.84	2.29	1.25
6	3	0	*	2152	100	203	1	2.06	4.19	9.15	1.29	1.54
6	3	1	*	4786	104	87	1	2.6	6.54	10.43	1.5	1.49
6	3	2	*	7662	104	21	1	4.37	12.67	13.01	1.41	1.34
6	3	3	*	11638	230	100	1	7.55	8.34	20.3	1.42	1.06
3	6	0	+	14.4	110	400	6.52	25.08	6.81	11.8	6.47	1
3	6	1	+	100.8	110	400	1.64	3.54	1.13	2.48	1.76	1
3	6	2	+	342.9	110	400	1.73	2.68	1	2.36	2.18	1.06
3	6	0	*	1258	110	25	1	1.65	469	2.63	1.67	1.24
3	6	1	*	2511	221	18	1.2	1.12	225	2.08	2.04	1
3	6	2	*	4287	230	10	1.13	1.85	545	26.8	2.71	1
3	6	2	*	5044	100	96	1.24	3.2			2.51	1
5	5	0	+	13.6	124	400	7.55	27.8	8.49	16.0	6.42	1
5	5	1	+	68.8	124	400	2.69	8.31	2.54	6.17	2.43	1
5	5	2	+	369.9	124	246	1.39	3.47	1.04	25.8	1.63	1
5	5	3	+	582	221	24	1.39	35.5	1	54.7	2.46	1.23
5	5	0	*	1854	124	5	1	8.44	844	55.3	1.51	1.63
5	5	0	*	4056	124	273	1	3.82			1.39	1.14
5	5	1	*	4560	230	7	1	11.8	967	370	2.33	1.24
5	5	1	*	8118	230	171	1.11	4.67				1
5	5	2	*	7695	230	2	1	44.6	2868	880	2.81	1.21
5	5	2	*	6783	230	20	1.02	33.8				1
5	5	3	*	10488	230	6	1	97.0				1.18
3	11	0	+	32.5	200	400	5.25	28.0	5.8	15.75	5.42	1
3	11	1	+	823.4	200	170	2.06	3.51	1.16	2.8	2.7	1
3	11	2	+	539.6	200	23	2.05	49.4	1	6.03	3.24	1.2
3	11	0	*	14856	200	13	1.27	1.24		7.39	2.15	1
3	11	1	*	22091	200	17	1	4.38			2.35	1.12
3	11	2	*	70649	234	51	1	15.8			2.60	1.14
3	11	3	*	35730	230	1	1	1742			3.57	3.07

Discussion

- Method C1 wins for multiplication of large ideals and large orders. It is relatively fast in all test so it might be used as a general method for CHNF computation. The worst factors for this method appear for the addition of small ideals. This is due to the obvious fact that reduction is not worthwile if the pseudomatrix is sparse and has entries of small absolute value.

- Method B1 is always slower than C1 (with the exception of one test run, but with an insignificant difference). Therefore the BOSMA–POHST algorithm is not particularly suited for reduction, although, for large examples, it is better than the BOSMA–POHST algorithm without reduction.
- Method Cr wins for addition of small ideals for all orders. This is partly due to the fact that one is detected as a reducer and can be used efficiently. It also wins for multiplications of small ideals and small orders, in particular if the base field is large and the relative degree is small. These results can be explained by the fact that Cr includes a cheap reduction, which might not be as effective as the reduction in C1.
- Method C wins the addition of larger ideals. Reduction seems not be effective if the pseudomatrix has not many columns but the reducer is very large. This might be due to the fact that the COHEN algorithm can use the sparsity of the example pseudomatrices efficiently.
- Method B wins multiplication of small ideals for small orders. It is particularly effective if the base field is simple.
- Method Cg has a high correlation to C1 and slower in most cases, which is not surprising from the algorithm.

All entries of one row of the table is taken from one KANT–session. If the examples are large, the whole session required up to several hours. If this is the case, strange things might happen (see the example order degrees 5 over 5, ideal test set 1, operation *). The first run included all methods and only 7 repetitions were possible. In the second run methods C, B, and Cg were excluded to obtain 171 repetitions. The interesting point is that method C1 is relatively slower than in the first run, and method Cr is faster. The explanation could be as follows.

Large computations tend to fragment the memory. If much memory is consumed and the memory is fragmented, the allocation of new memory is much more difficult. This promotes implementations which do not allocate memory as frequently. This seems to be the case here: method C1 is more affected by the fragmented memory than method Cr.

Since the comparison of algorithms, not implementations, are intended this effect blurs the result. Thus, the strength of this effect shows the limitations of the approach to compare algorithms with actual implementations.

6.4 Comparison of different normal form algorithms without reducers known in advance

Aim

To compare the different normal form algorithm implemented in KANT for pseudomatrices, where reducers are not known, as opposed to the previous section. The results will form a basis for heuristics for the question of which of the algorithms is best to apply in different situations.

Design

Compared methods

The following methods were compared and tabulated using the abbreviations given:

C Cohen algorithm (algorithm 4.5.1) without reduction,

B Bosma–Pohst algorithm (algorithm 4.5.3) without a reduction,

C1 Cohen algorithm (algorithm 4.5.1) with a one ideal reducer (see subsection 4.8.2) obtained with the method of corollary 4.8.5,

B1 Bosma–Pohst algorithm (algorithm 4.5.3) with a one ideal reducer (see subsection 4.8.2) obtained with the method of corollary 4.8.5,

Cr Cohen algorithm (algorithm 4.5.1) with a rational reducer (see subsection 4.8.2) obtained with the method of corollary 4.8.5,

Br Bosma–Pohst algorithm (algorithm 4.5.3) with a rational reducer (see subsection 4.8.2) obtained with the method of corollary 4.8.5.

r For information, the last column of the table contains the time necessary for the preparation of the reducer obtained with the method of corollary 4.8.5.

6.4.1 Randomly generated pseudomatrices

Starting from algebraic number fields in absolute presentation of degree 2, 3, 5, and 10, relative extensions with a relative degree of 2, 3, 5, and 10 are defined over each of the four base fields to produce an array of 16 relative number fields.

The base fields have the property that

- the generating polynomial has coefficients with small absolute value;
- the order generated by the polynomial is the maximal order of the number field;
- the class group is nontrivial.

Further details of the fields will not be given here, they will be described in [Hop].

The relative extensions are produced with random polynomials which

- are relatively sparse, about half of the coefficients are zero;
- have coefficients of the base field with a representation of integers with relatively small absolute values.

The relative degree corresponds to the number of rows of the final example pseudomatrix which is recorded in the table.

For each relative order an ideal $\mathfrak{A}$ is chosen such that $\mathfrak{A} \cap \mathbb{Z}$ is either $2\mathbb{Z}$ or $64777\mathbb{Z}$. In the table this is indicated as “ideals over 2” or “ideals over 64777”. Using the pseudobasis

$$\mathfrak{M} = \begin{bmatrix} \mathfrak{a}_1 & \dots & \mathfrak{a}_n \\ A_1 & \dots & A_n \end{bmatrix}$$

for each of the ideals, $2n$ random elements $a_i \in \mathfrak{a}_i$, for $i \in \mathbb{N}_n$, are chosen randomly to produce $2n$ elements of $\operatorname{Mod}(\mathfrak{M})$. The pseudomatrix $\mathfrak{N}$ formed by these elements together with trivial ideals is the example pseudomatrix. $\mathfrak{N}$ is very likely to be equivalent to $\mathfrak{M}$. This pseudomatrix has obviously a very good reducer, the minimum of the ideal $\mathfrak{A}$, but it will not be used in this test.

The table contains relative times, described in subsection 6.3.3.

base field degree	pseudomatrix dimension	ideal over	time of the fastest method for this example (in ms)	relidmark of the computer used, *1000	time in relation to the fastest method						
					C	B	C1	B1	Cr	Br	r
2	2×4	2	8	190	2.38	1	10.5	10	10.5	9.5	8
2	2×4	64777	16	210	1	1.31	6.69	7.44	5.88	7.25	4.25
3	2×4	2	27	190	2.56	1	4.81	4.78	5.04	4.48	4.04
3	2×4	64777	11	210	3.73	1	9.45	9.82	11.09	9.55	7.82
5	2×4	2	27	190	3.96	1	8.67	9.22	10.37	8.89	8
5	2×4	64777	32	210	9.06	1	8.56	8.69	10.19	8.06	6.5
10	2×4	2	590	190	15.61	1	10.31	10.36	11.29	10.34	10.03
10	2×4	64777	6822	187	2.17	1	1.69	1.35	1.68	1.32	1.08
2	3×6	2	22	190	2.27	1	6.59	5.36	5.68	5.09	4
2	3×6	64777	22	210	3.05	1	9.14	7.05	7.5	6.64	4.59
3	3×6	2	162	190	1.13	1	1.3	1.15	1.25	1.11	0.93
3	3×6	64777	28	210	9.21	1	5.96	6.29	7.14	6.25	4.64
5	3×6	2	148	190	18.13	1	3.22	3.24	3.54	3.11	2.64
5	3×6	64777	124	210	3.89	1	4.03	4.23	4.32	3.84	2.61
10	3×6	2	12430	190	18.57	49.19	1.06	1	1.16	1.02	0.94
10	3×6	64777	13810	187	4.79	1.85	1.42	3.62	1	2.15	0.59
2	5×10	2	43	190	7.56	1	4.98	4.95	4.72	4.77	3.77
2	5×10	64777	43	210	4	1	7.49	6.91	6.51	6.3	3.88
3	5×10	2	413	190	322.31	1.05	1.04	1.06	1.16	1	0.68
3	5×10	64777	420	210	25.24	1	1.67	2.38	1.64	1.79	0.64
5	5×10	2	890	190	2794.46	1.89	1	1.13	1.06	1.07	0.8
5	5×10	64777	1220	237		20125.44	1	8.7	5.11	8.6	0.65
10	5×10	2	18610	190		6.85	1	1	1.25	1.13	0.92
10	5×10	64777	42170	237		2.66	1	2.57	2.69	1.7	0.55
2	10×20	2	1180	190		1.14	1	1.07	1.01	1.72	0.53
2	10×20	64777	980	210		1	1.17	1.77	27.93	2.4	0.59
3	10×20	2	1970	190		105.59	1	1.27	1.2	1.94	0.59
3	10×20	64777	2990	160		37.15	1	2.57	118	3.68	0.56
5	10×20	2	5840	190		20.95	1.01	1.27	1	1.29	0.8
5	10×20	64777	13680	160		12.07	1	4.43	171.1	4.02	0.68
10	10×20	2	50160	190			1	1.21	1176.52	4.19	0.8
10	10×20	64777	86460	210			1	36.5		18.74	0.35

Discussion

A considerable part (half to almost all, with the exception of one case) of the computation time of the methods which use a reducer is consumed by the computation of the reducer.

Two methods share the smallest computation times: the Bosma–Pohst algorithm without reduction and the Cohen algorithm with reduction using an ideal as a reducer.

If the dimension of the pseudoamtrix is small the Bosma–Pohst algorithm without reduction is fastest. If the dimension of the pseudoamtrix is large the Cohen algorithm using reduction is fastest.

For medium dimensions (5×10) the size of the base field is important. For smaller base fields the Bosma–Pohst algorithm without reduction is faster, for larger base fields the Cohen algorithm with reduction is faster.

6.5 On the importance of absolute ideal multiplications in the relative normal form computations

Aim

This test demonstrates the importance of the efficiency of the multiplication of absolute ideals for the efficiency of the CHNF.

Design

The results of the profiling options of the GNU–C compiler are used. The profile contains the accumulated time for all absolute ideal multiplications and the total time for a CHNF–application.

The multiplication of relative ideals (given as pseudomatrices) involves the creation of a large pseudomatrix. The CHNF computation of this pseudomatrix forms the actual example pseudomatrix for this test.

Use some relative ideals in relative orders already used in the test in section 6.3.1.

Example 1

Use the relative number field of subsection 6.3.1 identified by the degrees 11 over 3.

$\mathfrak{a}_1 = 14911 o_{\mathcal{K}} + ((-5604 - 2114\rho - 13622\rho^2) + (-5882 + 4444\rho + 5883\rho^2)\sigma + (-13010 - 9472\rho - 12973\rho^2)\sigma^2 + (-6978 + 6600\rho - 11612\rho^2)\sigma^3 + (4038 + 7534\rho + 12453\rho^2)\sigma^4 + (7552 + 11432\rho - 9802\rho^2)\sigma^5 + (2078 + 8988\rho + 11884\rho^2)\sigma^6 + (13834 + 6258\rho - 2750\rho^2)\sigma^7 + (7218 - 1806\rho + 13108\rho^2)\sigma^8 + (-12502 - 4584\rho - 322\rho^2)\sigma^9 + (-6330 - 10558\rho - 321\rho^2)\xi)/2 o_{\mathcal{K}}$

$\mathfrak{a}_2 = 101269 o_{\mathcal{K}} + ((10419 - 27252\rho - 25219\rho^2) + (4246 + 33055\rho + 21907\rho^2)\sigma + (-48621 - 14073\rho - 5214\rho^2)\sigma^2 + (35102 + 20495\rho + 9289\rho^2)\sigma^3 + (-15012 + 14703\rho + 36679\rho^2)\sigma^4 + (36843 + 12524\rho + 21543\rho^2)\sigma^5 + (45036 + 37594\rho - 2870\rho^2)\sigma^6 + (2711 + 19750\rho - 40548\rho^2)\sigma^7 + (47920 - 35796\rho + 34136\rho^2)\sigma^8 + (-27599 + 32510\rho - 15622\rho^2)\sigma^9 + (16609 + 42785\rho + 1365\rho^2)\xi)/2 o_{\mathcal{K}}$

The total relative ideal multiplication uses 38.27s. 6707 absolute ideal multiplications used 27.16s. This is a share of 71%.

Example 2

Use the relative number field of subsection 6.3.1 identified by the degrees 5 over 5.

$\mathfrak{a}_1 = 5865 o_{\mathcal{K}} + ((-353 - 2463\rho + 854\rho^2 + 1236\rho^3 + 2555\omega) + (-1813 - 2250\rho + 2381\rho^2 - 2722\rho^3 - 1222\omega)\sigma + (303 + 2181\rho + 745\rho^2 + 2560\rho^3 - 2135\omega)\sigma^2 + (2496 - 1598\rho - 2679\rho^2 + 833\rho^3 + 1997\omega)\sigma^3 + (2438 + 2904\rho - 1249\rho^2 + 2553\rho^3 - 2611\omega)\sigma^4) o_{\mathcal{K}}$

$\mathfrak{a}_2 = 1615 o_{\mathcal{K}} + ((434 - 497\rho - 387\rho^2 + 430\rho^3 - 255\omega) + (155 - 125\rho + 442\rho^2 + 450\rho^3 - 638\omega)\sigma + (-429 + 328\rho - 133\rho^2 - 365\rho^3 + 122\omega)\sigma^2 + (93 + 710\rho + 728\rho^2 - 425\rho^3 + 245\omega)\sigma^3 + (-384 + 118\rho - 377\rho^2 + 726\rho^3 - 474\omega)\sigma^4) o_{\mathcal{K}}$

The total relative ideal multiplication uses 11.87s. 645 absolute ideal multiplications used 9.25s. This is a share of 78%.

Discussion

These results clearly stress the importance of the absolute ideal multiplication algorithms. Since an important aim of this work is to improve the efficiency of the normal form algorithm, the results motivate the efforts for the absolute ideal arithmetic.

6.6 Comparison of the two phases in the reduced normal form algorithm

Aim

On page 75 the two–phases–method was introduced. This test compares the running time of each of the two phases. It was mentioned that the second phase is not too expensive. This test serves as an argument to this opinion.

Design of the test

The relative number fields of subsection 6.3.1, the ideal test set, described in subsection 6.3.2, and the random pseudomatrices of subsection 6.4.1.

A pseudomatrix generated by this method features a good reducer: the minimum of the relative ideal (see section 5.4).

An experimental version of KASH was used which has the option of only executing the first phase of a normal form computation.

All computation times (in milliseconds) are the result of a single execution of the standard CHNF–algorithm on a HP900 series 700 computer with 124000 relidmark, see subsection 6.1.

base field degree					
	relative degree				
		relative ideal test set			
			time of the first phase	total time	percentage
3	3	0	70	90	78
3	3	1	80	90	89
3	3	2	190	210	90
3	3	3	310	360	86
3	3	4	440	540	81
3	3	5	650	770	84
3	3	6	1180	1420	83
3	3	7	1750	2190	80
3	3	8	5230	6480	81
6	3	0	250	300	83
6	3	1	240	290	83
6	3	2	420	560	75
5	5	0	510	570	89
5	5	1	740	880	84
5	5	2	1310	1620	81
3	6	0	210	250	84
3	6	1	260	310	84
3	6	2	410	490	84
3	11	0	320	460	70
3	11	1	7400	10310	72
3	11	2	3080	3610	85

6.7 Comparison of the different multiplication algorithms for relative ideals

Aim

This test is very similar to the comparisons of the different absolute ideal multiplication algorithms in subsection 6.2, with relative in place of absolute.

The answer will give heuristics for the question how to use the different multiplication algorithms to increase the overall time efficiency of the multiplication for relative ideals.

Design of the test

The design of this test is identical to the comparison of the absolute ideal multiplication in section 6.2 with the following exceptions:

- Normal presentations and normal multiplications are not used. These algorithms are not yet implemented in KANT.
- The relative orders of subsection 6.3.1 are used. The ideal test sets are produced with the method described in subsection 6.3.2.

base field degree	relative degree	number of the ideal test set	repetitions per ideal test set	speed of the computer in 1000*relidmark	time (in *ms*) for				
					multiplication of type			transformation	
					basis	*mixed*	*four*	*basis→2elt*	*2elt→basis*
3	3	$0'$	100	78	*145.8*	**81.7**	94.6	18.9	68.8
3	3	0	100	78	*189.6*	**114**	163.4	22.5	80.4
3	3	1	100	78	*416.9*	**260.4**	365.8	29.4	179.8
3	3	2	50	78	*731.6*	**469.6**	612.8	32.2	269.8
3	3	3	50	78	*1420*	**980.2**	1307.6	46.6	403.2
3	3	4	20	78	*2430.5*	**1647**	*2261*	64	572.5
3	3	5	20	78	*3969*	**2796.5**	*4112.5*	106.5	1116
3	3	6	10	78	*9204*	**6440**	*9538*	182	2436
3	3	7	10	78	*14332*	**10857**	*16969*	300	4354
3	3	8	10	78	*43961*	**30480**	*53723*	613	11926
6	3	$0'$	50	186	*433.8*	**199.2**	219.6	48.6	308.2
6	3	0	50	186	*497.8*	**280.8**	349.4	58.2	361.2
6	3	1	50	186	*887.6*	**522**	657.8	68.8	558.4
6	3	2	50	186	*2051*	**1261**	1661.4	91.8	846.8
6	3	3	50	190	*4443*	**3010.4**	4429.8	117.6	2277.4
6	3	4	20	190	*10779*	**7328**	10882	292	4398.5
6	3	5	20	190	*23529*	**15824**	*24038*	723.5	7236
5	5	$0'$	50	186	*2249.4*	**824.6**	1219.8	700.4	733.4
5	5	0	50	186	*3322.8*	**2028.4**	2856.6	1148.8	1475
5	5	1	20	186	*5259.5*	**3009**	3629	466.5	1840
5	5	2	20	186	*7096.5*	**2888**	4348.5	217	1618
5	5	3	20	190	*16878*	**6666.5**	10080.5	321.5	4549
5	5	4	20	190	*33915.5*	**12932**	*23740.5*	398.5	8808
3	6	$0'$	100	186	*786.1*	**209.9**	308.4	69.7	156
3	6	0	100	186	*1762.6*	**444.7**	629.2	136.1	279.2
3	6	1	20	186	*2186.5*	**649**	926.5	105	439.5
3	6	2	20	186	*3119.5*	**1002**	1265.5	119	570
3	6	3	10	190	*8231*	**2614**	3695	223	1439
3	6	4	10	190	*12819*	**4292**	*6120*	206	1790
3	6	5	10	190	*27112*	**8923**	*14338*	372	3696
3	6	6	10	190	*62876*	**20206**	*29037*	989	8313
3	11	$0'$	50	186	*8403.6*	**1449.8**	2104	895.2	1157.8
3	11	0	20	186	*11053*	**1838.5**	2744.5	1073	1097.5
3	11	1	20	186	*24323.5*	**6008**	6557	1785	2958.5
3	11	2	10	186	*33079*	*13080*	**8237**	1960	4090
3	11	3	10	190	*85828*	**19537**	21860	3766	9327

Discussion

The table shows that the mixed presentation method is quite superior. But subsequent tests drew another picture:

- There are huge run time differences for the mixed multiplication algorithm for all of the tested relative orders except for the order 3 over 3. For the order 3 over 6 they differ by a factor of 200, for the order 6 over 3 by a factor 4000, and for the order 5 over 5 by a factor of 5000. Some ideal multiplications in order 11 over 3 could not be finished which guarantees a factor of at least 1000000.
- The run time differences are the result of ill–behaved two–element presentations. Ideals with well–behaved two element presentations can efficiently be multiplied with any ideal given in basis presentation.
- The randomly chosen ideals in the above table seemed to have missed the extremely ill–behaved two–element presentations in most cases. This appears as a good average performance of the algorithm.
- Ill–behaved two–element presentations do not occur for absolute ideals. Therefore they appear to be a property of relative presentations of algebraic numbers.
- Ill–behaved two–element presentations also trouble the four generators multiplication algorithm.

6.8 Comparison of the efficiency of relative ideal multiplication to absolute ideal multiplication

Aim

Which is faster, the relative or the absolute multiplication? How does it depend on the size of the algebraic number field and the ideals? Is it therefore worth dealing with relative extensions for the purpose of ideal multiplications?

Design

The 16 relative number fields introduced in subsection 6.4.1 are used and tabulated as m over n, where m is the relative degree and n the degree of the base field.

For each relative order five ideals $\mathfrak{A}$ are chosen such that $\mathfrak{A} \cap \mathbb{Z}$ is $2\mathbb{Z}$, $14\mathbb{Z}$, $174\mathbb{Z}$, $1296\mathbb{Z}$, or $88642\mathbb{Z}$. The ideal is tabulated in the column "ideals over" with the corresponding natural number.

Using the method of subsection 5.2.2, for each relative ideal a corresponding absolute ideal is obtained.

The test compares the running times ("times for mult") of the multiplication of two ideals in relative representation ("rel") and in absolute representation ("abs"). The tabulated times are measured in milliseconds on the same computer, with 190000 relidmark. Transformation times from absolute to relative representations and vice–versa are not considered. The last column contains the quotient of the time for the relative multiplication by the time for the absolute multiplication.

For both relative and absolute multiplication only one algorithm is used (the one which is the default for KANT computations). It is a combined algorithm using the heuristics obtained from the tests in sections 6.2 and 6.7.

For the smaller examples, the table contains the average time of up to 10 multiplications of randomly chosen ideals.

base field degree					
	relative extension degree				
		ideals over			
			times for mult		quotient
			rel	abs	rel/ abs
2	2	2	14	1.0	13.5
2	2	21	26	1.5	17.1
2	2	455	35	1.8	18.8
2	2	60214	47	4.0	11.7
2	2	1125443	43	4.8	8.83
2	3	2	22	3.2	7.12
2	3	21	65	4.4	14.8
2	3	455	114	6	19.0
2	3	60214	174	20	8.52
2	3	1125443	106	26	3.97
2	5	2	143	15	9.2
2	5	21	281	21	13.2
2	5	455	470	36	12.9
2	5	60214	708	96	7.31
2	5	1125443	558	147	3.78
2	10	2	1183	170	6.93
2	10	21	4142	488	8.48
2	10	455	7653	720	10.6
2	10	60214	10133	1556	6.50
2	10	1125443	8533	2510	3.39
3	2	2	19	3.7	5.16
3	2	21	38	5	7.62
3	2	455	56	7.5	7.58
3	2	60214	99	23	4.29
3	2	1125443	115	30	3.77
3	3	2	57	13	4.38
3	3	21	81	15	5.2
3	3	455	121	30	4.06
3	3	60214	347	86	4
3	3	1125443	412	108	3.79
3	5	2	292	63	4.58
3	5	21	521	81	6.42
3	5	455	860	162	5.29
3	5	60214	1783	544	3.27
3	5	1125443	1770	667	2.65
3	10	2	2770	380	7.28
3	10	21	5560	650	8.55
3	10	455	10560	1350	7.82
3	10	60214	12770	4740	2.69
3	10	1125443	9530	4440	2.14
5	2	2	53	20	2.66
5	2	21	126	30	4.22
5	2	455	176	43	4.07
5	2	60214	270	116	2.31
5	2	1125443	390	173	2.25
5	3	2	330	80	4.12
5	3	21	350	150	2.33
5	3	455	370	200	1.85
5	3	60214	800	530	1.50
5	3	1125443	1520	730	2.08
5	5	2	220	450	0.488
5	5	21	1510	540	2.79
5	5	453	2510	760	3.30
5	5	60214	3070	2770	1.10
5	5	1125443	3750	3250	1.15
5	10	2	8440	4340	1.94
5	10	21	16880	5410	3.12
5	10	453	14500	8870	1.63
5	10	60214	32230	37170	0.867
5	10	1125443	13830	25940	0.533
10	2	2	560	116	4.8
10	2	21	796	210	3.79
10	2	455	1443	300	4.81
10	2	60214	2343	1070	2.19
10	2	1125443	3413	1836	1.85
10	3	3	1010	1230	0.821
10	3	23	1080	1860	0.580
10	3	453	4510	1830	2.46
10	3	60213	6280	5420	1.15
10	3	1125443	10120	9570	1.05
10	5	3	5790	6600	0.877
10	5	21	20590	7030	2.92
10	5	455	25440	10460	2.43
10	5	60213	39530	30220	1.30
10	5	1125443	32780	66280	0.494
10	10	3	38360	91740	0.418
10	10	23	46560	688240	0.0676
10	10	453	65680	148030	0.443
10	10	60213	301960	295830	1.02

Discussion

For larger number fields the relative multiplication, for smaller number fields the absolute multiplication is faster.

6.9 Comparison KASH and gp of modular and nonmodular relative normal form computations

Aim

There are pseudomatrix normal form algorithms implemented in gp, see [Pari]. By comparing running times, it is possible to evaluate if the efforts to improve the efficiency of normal form algorithms have proved to be effective.

Design

The pseudomatrices of subsection 6.4.1 are used. Again the degree of the base field, the dimension of the pseudomatrix, and the minimal natural number of the relative ideal which is represented by the pseudomatrix are tabulated.

The column marked “non-mod” refers to the normal form algorithm without reduction. The column “det-mod” refers to the combined computation time for obtaining a reducer with the gcd of a few minors and computing the normal form using this reducer. The column “or-mod” refers to normal form computation with a very good reducer which is the minimum of the above relative ideal.

All computation times are in milliseconds on a computer with 149000 relidmark.

The following abbreviations are used in the table.

non-mod Using a normal form algorithm without a reducer.

det-mod Using a reducer obtained with minor computations.

mod Using a reducer assumed to be known in advance which is the minimum of the relative ideal represented by the pseudomatrix. This is not applicable to gp since a reducer which is an integral multiple of the rank minor gcd is required for the reduction algorithm.

∞ The computation does not terminate in an acceptable time which is at least 100 times of the time needed by another time the test of a similar difficulty.

overflow A typical message in gp is: “The stack overflows. Doubling the stack size.” If this message occurs with an initial stacksize of 32MB, which is as much as the test computer as usually available for a gp process, “overflow” is tabulated.

error Other error messages, which indicate inconsistencies in the representations of ideals or algebraic numbers.

base field degree	pseudomatrix dimension	test ideals over	computation times in *ms*				
			KASH			gp	
			non-mod	det-mod	mod	non-mod	det-mod
2	2×4	2	10	120	20	10	50
2	2×4	64777	40	150	30	20	40
3	2×4	2	20	180	0	140	error
3	2×4	64777	10	170	20	90	70
5	2×4	2	30	330	20	550	510
5	2×4	64777	50	360	80	390	260
10	2×4	2	750	7990	210	14320	overflow
10	2×4	64777	6960	13160	10	11900	9470
2	3×6	2	40	210	0	80	110
2	3×6	64777	30	260	70	50	110
3	3×6	2	240	330	0	430	error
3	3×6	64777	40	240	50	210	200
5	3×6	2	200	650	90	1790	790
5	3×6	64777	170	640	230	940	690
10	3×6	2	1586160	17730	10	100920	22340
10	3×6	64777	22920	22010	2230	27260	18590
2	5×10	2	60	300	40	320	error
2	5×10	64777	70	420	110	220	270
3	5×10	2	640	610	50	5440	880
3	5×10	64777	520	950	220	1000	790
5	5×10	2	2270	1450	240	8390	2270
5	5×10	64777	∞	1600	360	14860	4510
10	5×10	2	270440	21890	90	970400	35590
10	5×10	64777	176250	49080	7240	295670	50010
2	10×20	2	1800	1810	490	23590	1530
2	10×20	64777	1310	1370	650	11590	1270
3	10×20	2	767640	2880	220	80890	error
3	10×20	64777	76640	2460	800	32320	3870
5	10×20	2	200150	5200	1130	951390	error
5	10×20	64777	117700	6730	3900	130140	15650
10	10×20	2	∞	83520	1290	∞	470480
10	10×20	64777	∞	126380	70070	∞	210690

Discussion

non–modular method While the implementation in KANT is faster on average and in more cases there are large runtime differences in both directions. Large computations suffer from several irregularities:

- Fragmented memory slows down the computation extremely if a larger part of the available memory is used. This is influenced mainly by the structural decisions for the memory managemant of the system.
- The behavior of basic algorithms (integer HNF computation, long integer arithmetic) for large numbers becomes much more important for large examples. The basic algorithms can be implemented in several

variants none of which is perfect for all examples. The variant chosen has a great effect of the runtime for the normal form algorithm.

These irregularities affect gp and KANT to a different extent. Therefore the concept of comparing implementations in different systems with the aim of comparing algorithms is quite restricted.

modular method with determinant The computation times are quite similar.

non–modular method While not implemented in gp the times for this method are very small. This stresses the importance to obtain reducers in advance to the normal form computation which is possible in many applications.

Index

Bibliography

[Ber96] Daniel J. Bernstein, *Fast ideal arithmetic via lazy localization*, Algorithmic Number Theory (Talence, France) (Henri Cohen, ed.), Lecture Notes in Computer Science, no. 1122, Second International Symposium of Number Theory, Springer, May 1996, pp. 29–36.

[Bla63] W. A. Blankinship, *A new version of the Euclidean algorithm*, Amer. Math. Mon. (1963), no. 70, 742–745.

[BP91] Wieb Bosma and Michael E. Pohst, *Computations with finitely generated modules over Dedekind domains*, Proceedings ISSAC'91 (Stephen M. Watt, ed.), 1991, pp. 151–156.

[CC82] T.-W. J. Chou and G. E. Collins, *Algorithms for the solution of systems of linear Diophantine equations*, SIAM J. Comput. (1982), no. 11, 687–708.

[Coh95] Henri Cohen, *A course in computational algebraic number theory*, 2 ed., Graduate Texts in Mathematics, vol. 138, Springer–Verlag, Berlin Heidelberg, 1995.

[Coh96] Henri Cohen, *Hermite and Smith normal form algorithms over Dedekind domains*, Mathematical Computing **216** (1996), no. 65, 1681–1699.

[Dab93] Mario Daberkow, *Bestimmung relativer Ganzheitsbasen in relativquadratischen Zahlkörpern*, Diplomarbeit (Diploma thesis), Heinrich–Heine–Universität Düsseldorf, 1993.

[DP98] Mario Daberkow and Michael. E. Pohst, *On the computation of Hilbert class fields*, Journal of Number Theory **69** (1998), 213–230.

[Fri97] Carsten Friedrichs, *Berechnung relativer Ganzheitsbasen mit dem Round–2–Algorithmus*, Diplomarbeit (Diploma thesis), Technische Universität Berlin, 1997.

[Fru76] M. A. Frumkin, *An application of modular arithmetic to the construction of algorithms for solving systems of linear equations*, Soviet Math. Dokl. (1976), no. 17, 1165–1168.

[GMS94] Michel Goosens, Frank Mittelbach, and Alexander Samarin, *The LaTeX companion*, 2 ed., Addison–Wesley, 1994.

[Hes96] Florian Hess, *Zur Klassengruppenberechnung in algebraischen Zahlkörpern*, Diplomarbeit (Diploma thesis), Technische Universität Berlin, 1996.

[HHR93] George Havas, Derek F. Holt, and Sarah Rees, *Recognizing badly presented $\mathbb{Z}$–modules*, Linear Algebra and its Applications **192** (1993), 137–163.

[HM94] George Havas and Bohdan S. Majewski, *Hermite normal form* *computation for integer matrices*, Congressus Numerantium (1994), no. 105, 87–96.

[Hop] Andreas Hoppe, *mathematical web page*, `http://www.math.tu-berlin.de/~hoppe/mathematik.html`.

[Hop94] Andreas Hoppe, *Normalformen ganzzahliger Matrizen — effiziente Implementierung in GAP*, Diplomarbeit (Diploma thesis), Humboldt–Universität Berlin, 1994.

[Kant] Mario Daberkow, Claus Fieker, Jürgen Klüners, Katherine Roegner, Michael E. Pohst, and Klaus Wildanger, *Kant V4*, Journal of Symbolic Computing **24** (1997), no. 3, 267–283, the webpage `http://www.math.tu-berlin.de/~kant/` includes the latest version of KASH with full documentation and introduction manual.

[KB79] R. Kannan and A. Bachem, *Polynomial algorithms for computing Smith and Hermite normal forms of an integer matrix*, SIAM J. Comput. (1979), no. 8, 499–507.

[Klü97] Jürgen Klüners, *Über die Berechnung von Automorphismen und Teilkörpern algebraischer Zahlkörper*, Dissertation (thesis), Technische Universität Berlin, 1997.

[Knu86] Donald E. Knuth, *The TEX book, computers and typesetting*, vol. A, Addison–Wesley, 1986.

[Kop92] Helmut Kopka, *LATEX, Eine Einführung*, 4 ed., Addison–Wesley, 1992.

[Lan94] Serge Lang, *Algebra*, 3 ed., Addidson–Wesley, Reading, Massachusetts, 1994.

[Ned94] Nederlandstalige TEXGebruikersgroep, *Ntg document classes*, `http://www.ntg.nl/`, 1994.

[O'M63] O. T. O'Meara, *Introduction to quadratic forms*, Die Grundlagen der Mathematischen Wissenschaften in Einzeldarstellungen, vol. 117, Springer–Verlag, Berlin, Göttingen, Heidelberg, 1963.

[Pari] Christian Batut, Henri Cohen, Francesco Diaz y Diaz, and Michel Olivier, *g pari calculator*, Université Bordeaux, the webpage `http://pari.home.ml.org/` includes the latest version of `gp` with full documentation and manual., 1998, Version 2.0.2.

[Pau96] Sebastian Pauli, *Zur Berechnung von Strahlklassengruppen*, Diplomarbeit (Diploma thesis), Technische Universität Berlin, 1996.

[PB74] I. S. Pace and S. Barnett, *Efficient algorithms for linear system calculations; part I — Smith form and common divisors of polynomial matrices*, J. of System Science (1974), no. 5, 403–411.

[PZ93] Michael E. Pohst and Hans Zassenhaus, *Algorithmic algebraic number theory*, Cambridge Unversity Press, 1993, 3rd printing.

[Tho] Kresten Krab Thorup, *Auc TEX development*, Mathematics and Computer Science, University of Aalborg, DK 9000 Aalborg, `http://www.iesd.auc.dk/~amanda/auctex/`.

[vS87] Johannes Graf von Schmettow, *Über die Berechnung von Klassengruppen algebraischer Zahlkörper*, Diplomarbeit (Diploma thesis), Mathematisches Institut der Universität Düsseldorf, 1987.

List of Algorithms