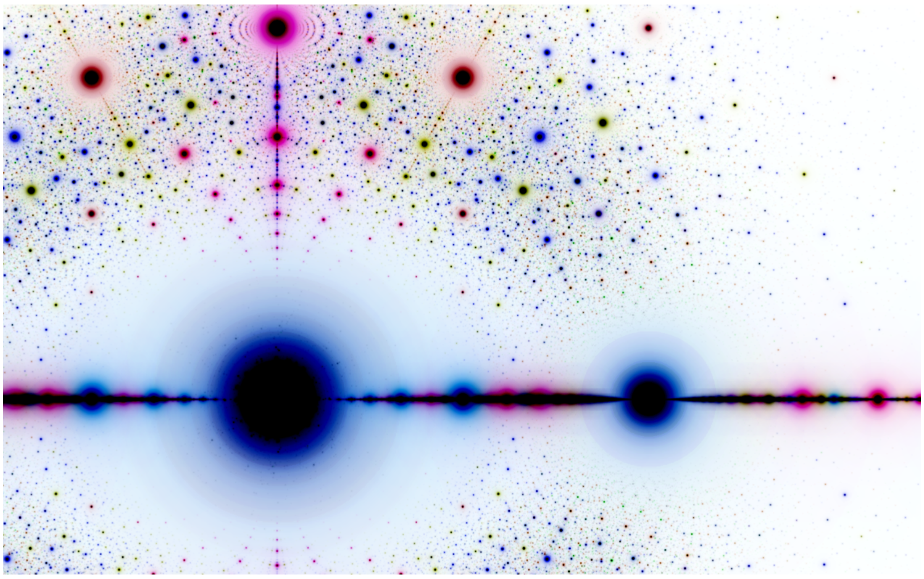
Radboud University

Master Thesis



Computing the power residue symbol

Koen de Boer

supervised by

dr. W. Bosma
dr. H.W. Lenstra Jr.

August 28, 2016

Foreword

Introduction

In this thesis, an algorithm is proposed to compute the power residue symbol $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ in arbitrary number rings containing a primitive m-th root of unity. The algorithm consists of three parts: principalization, reduction and evaluation, where the reduction part is optional. The evaluation part is a probabilistic algorithm of which the expected running time might be polynomially bounded by the input size, a presumption made plausible by prime density results from analytic number theory and timing experiments. The principalization part is also probabilistic, but it is not tested in this thesis.

The reduction algorithm is deterministic, but might not be a polynomial-time algorithm in its present form. Despite the fact that this reduction part is apparently not effective, it speeds up the overall process significantly in practice, which is the reason why it is incorporated in the main algorithm.

When I started writing this thesis, I only had the reduction algorithm; the two other parts, principalization and evaluation, were invented much later. This is the main reason why this thesis concentrates primarily on the reduction algorithm by covering subjects like lattices and lattice reduction. Results about the density of prime numbers and other topics from analytic number theory, on which the presumed effectiveness of the principalization and evaluation algorithm is based, are not as extensively treated as I would have liked to.

Since, in the beginning, I only had the reduction algorithm, I tried hard to prove that its running time is polynomially bounded. When I did not succeed, I attempted to pose some assumptions I thought to be plausible, in order to deduce from it that the reduction algorithm is effective. I did not succeed in making the assumptions plausible nor in deducing the effectiveness of the reduction algorithm. The short research about these assumptions is placed in the appendix (see section B.2).

Acknowledgements

I would like to thank my supervisor, dr. Wieb Bosma of the FNWI at the Radboud University, for his unstoppable faith and enthusiasm. The large amount of freedom and independence that characterizes his supervision might be overwhelming for others, but for me, it was exactly what I needed. His ability to see the big picture in difficult mathematical subjects has often saved me from losing myself in the details.

Besides my advisor, I would like to thank dr. H.W. Lenstra of the Mathematisch Instituut at the Universiteit Leiden, for being an outstanding teacher and for inspiring me with many very good ideas, including two-sided reduction and principalization. With his expertise and eye for detail, he has pointed out many areas of improvement in my thesis.

My sincere thanks also go to dr. J. Bouw of the Mathematisch Instituut at the Universiteit Leiden, for allowing me to read his PhD thesis before the publication, so that I could implement his algorithm that computes Hilbert symbols. This heavily motivated me to think about an effective algorithm to compute power residue symbols. I have good memories of my visit to Leiden, when we verified each other's computations.

I would also like to acknowledge dr. M. Kosters of the department of Mathematics at the University of California (Irvine), for sending me a summary of Bouw's algorithm that focusses on the algorithmic aspect, which has been very useful for me.

Furthermore, I would like to thank dr. D. Micciancio of the Computer Science & Engineering department at the University of California (San Diego), for giving a comprehensive answer to my question about q-ary lattices.

My thesis would be full of textual mistakes without the proofreaders: Elke de Boer, Els de Jong, Kris Roufs and Janneke de Wit. Many thanks for your effort.

Also, many thanks to Djordy Timmermans, who designed the beautiful cover of this thesis.

Finally, I must express my gratitude to my parents, to my brothers and sister, and to my partner Kris Roufs for supporting me spiritually throughout writing this thesis and my life in general. This accomplishment would not have been possible without them. Thank you.

Contents

CHAPTER 1

Number fields and completions

1.1 Introduction

The main subjects of this thesis are the power residue symbol and, to a lesser extent, the related Hilbert symbol. In order to obtain a clear understanding of these symbols, one has to be acquainted with algebraic number theory and its notions: number fields, ideals, orders, integral elements, completions, etcetera.

This chapter will give a quick, incomplete and subjective overview of the algebraic number theory topics needed. For professional and complete studies of number fields, I would like to recommend [Jan96] and [CF67]. Another goal of this chapter is introducing notation, to avoid misunderstandings in the remainder of this thesis.

We denote the integers by $\mathbb{Z}$, and the rational numbers by $\mathbb{Q}$. We denote rounding to the closest integer by $\lceil \cdot \rfloor$, and the group of invertible matrices with entries in $\mathbb{Z}$ by $\mathrm{GL}_n(\mathbb{Z})$.

1.2 Number fields

1.2.1 Finite degree field extensions

Definition 1.1 (Algebraic number field)**.** A number field is a finite degree field extension of the rational numbers $\mathbb{Q}$.

In this thesis, a number field is often denoted by the capital letter K (from the German word *Körper*) with degree $n = [K : \mathbb{Q}]$ over the rational numbers. Also, towers of finite extensions will occur. In that case, the field above K will be called L. The extension $L : K$ is called a relative extension, in contrast to $K : \mathbb{Q}$, to which is referred as an absolute extension.

In a computational context, a number field L is defined by an irreducible polynomial f over its ground field K. Via the isomorphism $L \simeq K[x]/f(x)$, any element of L can be uniquely represented by a vector $(k_1, \dots, k_n) \in K^n$, with $n = \deg f$.

Definition 1.2 (Galois extension)**.** Suppose $K \subseteq L$ are both number fields. The finite degree field extension $L : K$ is called a Galois extension if it is a normal extension; i.e., if for every irreducible polynomial $f(x) \in K[x]$ holds

$$f(x) \text{ has a root in } L \Longrightarrow f(x) \text{ splits in linear factors over } L.$$

Remark 1.3. Equivalently, a Galois extension $L : K$ is a splitting field of some polynomial $f(x) \in K[x]$, see [Lan05, V§3, i.p. Thm. 3.3]. Every Galois extension has a Galois group $G = \mathrm{Gal}(L : K)$ associated with it, which is a subgroup of the permutation group on the zeroes of the defining polynomial. ◀

Definition 1.4 (Abelian and cyclic extensions)**.** Suppose $K \subseteq L$ are both number fields. The extension $L : K$ is called an abelian extension if it is a Galois extension with an abelian Galois group. Similarly, an extension $L : K$ is called cyclic when the Galois group is cyclic.

1.2.2 Number rings

With a number field K one can associate a special subring of K, the ring of integers. Integers of K, also called integral elements, are recognizable by the form of their minimum polynomial over $\mathbb{Q}$ [SD01, §1, Thm. 1].

Definition 1.5 (Integral elements)**.** Let K be a number field. An element $\alpha \in K$ is called integral iff there exists a monic polynomial $f \in \mathbb{Z}[x]$ such that $f(\alpha) = 0$.

Definition 1.6 (Ring of integers)**.** The ring of integers of a number field K is now defined as the set of integral elements in K:

$$\mathcal{O}_K := \{\alpha \in K \mid \alpha \text{ is integral }\}.$$

Remark 1.7. It is not immediately clear that the above set is a ring. By examining equivalent notions of being integral one can indeed see that the set $\mathcal{O}_K$ is a ring containing $\mathbb{Z}$, see for example [Jan96, Thm. 2.3]. ◀

The following definition is a slight modification of the definition of an order[1]in [BS66, p. 88]. An alternative definition can be found in [Coh93, §4.6, Def. 4.6.1].

Definition 1.8 (Number ring)**.** Let K be a number field of degree $n = [K : \mathbb{Q}]$. Then, a ring $R \subseteq K$ is called a number ring, if:

(i) R is a free $\mathbb{Z}$-module with rank n;

(ii) $R \subseteq \mathcal{O}_K$.

Lemma 1.9. *The ring $\mathcal{O}_K$ is a number ring of K.*

A proof of this lemma can be found at [Cas86, § 10.3], for example.

[1] 'Order' is informally a synonym of 'number ring' here, although many authors treat these two notions differently.

Notation 1.10. For a number ring R of a number field K, we define the degree of R to be the degree $n = [K : \mathbb{Q}]$ of K.

Remark 1.11. In practice, a number ring is usually of the form $R = \mathbb{Z}[\theta_1, \dots, \theta_s]$, with $\theta_i \in \mathcal{O}_K$. In this case, part (ii) of Definition 1.8 is already fulfilled. Also, often one of the θ_i has the property that $[\mathbb{Q}(\theta_i) : \mathbb{Q}] = n$, implying that $\mathbb{Z}[\theta_i] \subseteq R$ is already a free, rank n $\mathbb{Z}$-module. Then, $R \subseteq \mathcal{O}_K$ is sandwiched between two free rank n modules, and is therefore [Lan05, Th. I.7.3] a free rank n $\mathbb{Z}$-module itself. ◀

In a computational context one often uses the property that number rings R always have a so-called integral basis.

Definition 1.12. Let R be a number ring in K, a number field of degree n. An integral basis of R is an n-tuple $(\theta_1, \dots, \theta_n) \in R^n$ such that every element $\alpha \in R$ can be written uniquely as

$$\alpha = \sum_{i=1}^{n} a_i \theta_i \text{ with } a_i \in \mathbb{Z}.$$

Example 1.13. The ring $\mathbb{Z}[i]$ is the ring of integers of $\mathbb{Q}(i)$, and is therefore a number ring. The integers $\mathbb{Z}[i]$ are called the Gaussian integers. Another example: The ring $\mathbb{Z}[\zeta_3]$ is the ring of integers of $\mathbb{Q}(\zeta_3)$, but it contains for example the ring $\mathbb{Z}[\sqrt{-3}]$, since $\zeta_3 = \frac{-1+\sqrt{-3}}{2}$. As $\sqrt{-3}$ has degree 2 over $\mathbb{Q}$, the ring $\mathbb{Z}[\sqrt{-3}]$ must have rank 2 over $\mathbb{Z}$, and therefore $\mathbb{Z}[\sqrt{-3}]$ is a number ring too (but not the ring of integers of $\mathbb{Q}(\zeta_3)$).

One can straightforwardly see that $(1, i)$ is an integral basis for $\mathbb{Z}[i]$, $(1, \zeta_3)$ is an integral basis for $\mathbb{Z}[\zeta_3]$ and $(1, \sqrt{-3})$ is an integral basis for $\mathbb{Z}[\sqrt{-3}]$. ◀

Remark 1.14. Note that for a number ring R with quotient field K, a $\mathbb{Z}$-basis $(\theta_1, \dots, \theta_n)$ of R is automatically a $\mathbb{Q}$-basis of K. So, every element $\alpha \in K$ can be written uniquely as $\sum_{i=1}^{n} q_i \theta_i$, with $q_i \in \mathbb{Q}$. ◀

Definition 1.15 (Multiplication matrix)**.** Suppose R is a number ring in a number field K of degree n, and R has given integral basis $(\theta_1, \dots, \theta_n)$. Then, given $\alpha \in K$, one can construct the multiplication matrix $M_\alpha \in M_{n \times n}(\mathbb{Q})$ of α. Write $\theta_i \cdot \alpha$ in the integral basis of R, for every $1 \leq i \leq n$:

$$\theta_i \cdot \alpha = \sum_{j=1}^{n} q_{ij} \theta_j.$$

One then defines the multiplication matrix as $M_\alpha := (q_{ij})_{i,j=1}^{n}$.

Remark 1.16. Seeing K as an n-dimensional $\mathbb{Q}$-vector space via the given integral basis $(\theta_1, \dots, \theta_n)$, above matrix M_α can be associated with the linear map induced by multiplication with α on the vector space K. Note that $M_\alpha \in M_{n \times n}(\mathbb{Z})$ when $\alpha \in R$. Also, observe that M_α heavily depends on the given integral basis of R. ◀

Definition 1.17 (Norm and trace)**.** Suppose K is a degree n number field and R is a number ring with given integral basis $B = (\theta_1, \dots, \theta_n)$. Then, for $\alpha \in K$, we have the following fundamental invariants, called the norm and the trace of α, respectively.

$$N(\alpha) := \det M_\alpha;$$

$$\mathrm{Tr}(\alpha) := \mathrm{Tr}\ M_\alpha.$$

Remark 1.18. The norm of an element $\alpha \in K$ does not depend on the given number ring in K nor the basis choice, since change of basis (even to another number ring) corresponds to (group-theoretic) conjugation of M_α with a transition matrix. This does not alter the value of the determinant, as it is a multiplicative homomorphism from $M_{n\times n}(\mathbb{Q})$ to $\mathbb{Q}$.

The trace, however, might depend on the chosen basis and given number ring. ◀

Definition 1.19 (Discriminant)**.** Suppose K is a degree n number field and R is a number ring with given integral basis $(\theta_1, \dots, \theta_n)$. Then we define the discriminant of R by

$$\Delta(R) := \det \operatorname{Tr}(\theta_i\theta_j)_{ij}.$$

I.e., the determinant of the matrix with as ij-th entry the value of the trace of $\theta_i\theta_j$ (which is in $\mathbb{Z}$).

Remark 1.20. The discriminant is independent of the chosen basis of R, but it does depend on the number ring. See for example [Cas86, §10.3, Lemma 3.2]. ◀

Notation 1.21. We will denote the discriminant of the ring of integers of K by $\Delta(K) := \Delta(\mathcal{O}_K)$.

Lemma 1.22. *For a number field K with ring of integers $\mathcal{O}_K$ and with a number ring R, we have the following identity:*

$$\Delta(R) = [\mathcal{O}_K : R]^2 \cdot \Delta(\mathcal{O}_K)$$

Here, $[\mathcal{O}_K : R]$ is the index of R in $\mathcal{O}_K$ as additive groups.

Proof. This lemma is a special case of [Neu99, Ch. 1, Prop. 2.12]. □

1.2.3 Ideal arithmetic

Unique factorization

Lemma 1.23. *The ring of integers $\mathcal{O}_K$ of a number field K is a Dedekind ring, i.e. it is Noetherian, integrally closed, and every nonzero prime ideal $\mathfrak{p}$ of $\mathcal{O}_K$ is a maximal ideal.*

Proof. See for example [Neu99, §I.3, Thm 3.1] □

Remark 1.24. In a Dedekind ring, every fractional ideal is invertible [AM69, Ch. 9, Thm. 9.8], and every ideal factors essentially uniquely as a product of prime ideals [AM69, Ch. 9, Cor. 9.4]. ◀

Definition 1.25. For a number field K, we denote by $\mathcal{I}_K$ the group of (nonzero) fractional ideals of $\mathcal{O}_K$.

Remark 1.26. The set $\mathcal{I}_K$ is indeed a group, under the following multiplication:

$$\mathfrak{a} \cdot \mathfrak{b} := \langle a \cdot b \mid a \in \mathfrak{a}, b \in \mathfrak{b} \rangle$$

i.e., $\mathfrak{a}\mathfrak{b}$ is the ideal generated by products of elements in $\mathfrak{a}$ and $\mathfrak{b}$. The group $\mathcal{I}_K$ has 'unit ideal' $\mathcal{O}_K = (1)$. This is the multiplication which is meant when one 'factorizes' an ideal. Note that the unique factorization property has as a direct consequence that the group $\mathcal{I}_K$ is a free $\mathbb{Z}$-module of countably infinite rank, with the prime ideals as its generators. ◀

Definition 1.27 (Valuation)**.** Given a prime ideal $\mathfrak{p}$ of $\mathcal{O}_K$, one can define the valuation $v_{\mathfrak{p}} : (\mathcal{I}_K, \cdot) \to (\mathbb{Z}, +)$, a group homomorphism. The $\mathfrak{p}$-valuation is defined on prime ideals[2] (the generators of $\mathcal{I}_K$) as follows:

$$v_{\mathfrak{p}}(\mathfrak{q}) = \begin{cases} 1 & \text{if } \mathfrak{p} = \mathfrak{q} \\ 0 & \text{if } \mathfrak{p} \neq \mathfrak{q} \end{cases}$$

Remark 1.28. The unique factorization property of fractional ideals in $\mathcal{O}_K$ can now be stated as follows. Every fractional ideal $\mathfrak{f} \in \mathcal{I}_K$ factors uniquely (up to order) as

$$\mathfrak{f} = \prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{f})}. \tag{1.1}$$

◀

Factorization of (p)

For a prime number p, the ideal (p) does not have to be a prime ideal in $\mathcal{O}_K$. In fact, in most cases it is not, and it factorizes as a product of prime ideals:

$$(p) = \prod_{i=1}^{g} \mathfrak{p}_i^{e_i}. \tag{1.2}$$

Via the inclusion $\mathbb{Z} \to \mathcal{O}_K$, we have $\mathbb{Z}/p\mathbb{Z} \to \mathcal{O}_K/\mathfrak{p}_i$ for every factor $\mathfrak{p}_i$ in (1.2). Since both $\mathbb{Z}/p\mathbb{Z}$ and $\mathcal{O}_K/\mathfrak{p}_i$ are fields, one can see this as a field extension. This leads to the following definition.

Definition 1.29. Let $\mathfrak{p}_i$ be a factor in the factorization of (p) in the ring of integers $\mathcal{O}_K$ of a number field K, as in (1.2). Then, we denote:

$$e_{K/\mathbb{Q}}(\mathfrak{p}_i) := e_i = v_{\mathfrak{p}_i}(p) \text{ and } f_{K/\mathbb{Q}}(\mathfrak{p}_i) := [\mathcal{O}_K/\mathfrak{p}_i : \mathbb{Z}/p\mathbb{Z}].$$

Remark 1.30. When there is no chance of confusion, one often drops the subscript $K/\mathbb{Q}$. Also, one calls $e(\mathfrak{p})$ the ramification index of $\mathfrak{p}$, and $f(\mathfrak{p})$ the residue class degree. A prime ideal that occurs in the factorization of a prime number (p), is called a prime (ideal) above p. So, in the case of (1.2), $\mathfrak{p}_i$ is a prime above p. ◀

Lemma 1.31. *Let $K/\mathbb{Q}$ be a Galois extension. Then, for all p, the factorization of (p) into prime ideals always has a particular form.*

$$(p) = \prod_{i=1}^{g} \mathfrak{p}_i^{e},$$

and $f_{K/\mathbb{Q}}(\mathfrak{p}_i) = f$, a fixed integer for all $1 \leq i \leq g$.

Proof. See for example [Neu99, Ch. 1, §9, Prop. 9.1]. □

Example 1.32. Note that above lemma does not mean that every prime number has the same factorization properties, as the following example shows. Consider

[2] By multiplicative continuation, $v_{\mathfrak{p}}$ defines a group homomorphism $\mathcal{I}_K \to \mathbb{Z}$.

the number field $K = \mathbb{Q}(\zeta_5)$, a Galois extension of $\mathbb{Q}$. It has ring of integers[3] $\mathbb{Z}[\zeta_5]$. We factorize (11) and (19) in $\mathbb{Z}[\zeta_5]$, with use of [Coh93, §4.8.2].

To obtain a factorization as in (1.2), one has to factorize $\Phi_5(x) = x^4 + x^3 + x^2 + x + 1$ (the 5-th cyclotomic polynomial) in $\mathbb{F}_{11}[x]$. We have

$$x^4 + x^3 + x^2 + x + 1 \equiv (x+2)(x+6)(x+7)(x+8) \bmod 11.$$

Therefore, $(11) = \prod_{i=1}^{4} \mathfrak{p}_i$, where $\mathfrak{p}_1 = (11, \zeta_5 + 2)$, $\mathfrak{p}_2 = (11, \zeta_5 + 6)$, $\mathfrak{p}_3 = (11, \zeta_5 + 7)$, and $\mathfrak{p}_4 = (11, \zeta_5 + 8)$. Note that all of these prime ideals $\mathfrak{p}_i$ have the same ramification index and residue class degree. In $\mathbb{F}_{19}[x]$, one obtains:

$$x^4 + x^3 + x^2 + x + 1 \equiv (x^2 + 5x + 1)(x^2 + 15x + 1) \bmod 19.$$

Therefore, $(19) = \mathfrak{q}_1\mathfrak{q}_2$, with $\mathfrak{q}_1 = (19, \zeta_5^2 + 5\zeta_5 + 1)$ and $\mathfrak{q}_2 = (19, \zeta_5^2 + 15\zeta_5 + 1)$. One sees that all prime ideals above the same prime number p have the same ramification index and residue class degree, as in Lemma 1.31. Prime ideals above different prime numbers, however, do not need to have common properties. ◀

Example 1.33. This example is about a non-Galois extension $\mathbb{Q}(\sqrt[3]{2})$ of $\mathbb{Q}$, having ring of integers $\mathbb{Z}[\rho]$ with $\rho = \sqrt[3]{2}$ (for a proof, see [AW03, Ex. 7.1.6, p. 153]). The factorization of prime numbers in $\mathbb{Z}[\rho]$ is not as in Lemma 1.31. The polynomial $x^3 - 2$ factors in the ring $\mathbb{F}_5[x]$ as

$$x^3 - 2 \equiv (x+2)(x^2 + 3x + 4) \bmod 5$$

and therefore $(5) = \mathfrak{p}_1\mathfrak{p}_2$, with $\mathfrak{p}_1 = (5, \rho + 2)$ and $\mathfrak{p}_2 = (5, \rho^2 + 3\rho + 4)$. The first prime ideal has residue class degree 1, whereas the second has residue class degree 2. ◀

The following definition is taken from [Coh93, Prop. 4.6.3].

Definition 1.34 (Norm of ideals)**.** The norm as in Definition 1.17 can be generalized to ideals of a number ring R. The norm of an ideal $\mathfrak{a}$ of R is defined as the cardinality of $R/\mathfrak{a}$,

$$N(\mathfrak{a}) := \#(R/\mathfrak{a}). \tag{1.3}$$

Remark 1.35. If $\mathfrak{a} = (\alpha)$ is a principal ideal (i.e., an ideal generated by one element), then the ideal norm coincides with the absolute value of the regular (element) norm, as in Definition 1.17. ◀

Example 1.36. Consider the quadratic number field $\mathbb{Q}(\sqrt{3})$; it has ring of integers $\mathbb{Z}[\sqrt{3}]$. The ideal $\mathfrak{a} = (2, \sqrt{3} + 1)$ is generated by two elements. Clearly we have $N(\mathfrak{a}) = 2$, since $a + b\sqrt{3} \equiv a - b$ modulo $\mathfrak{a}$, for $a, b \in \mathbb{Z}$. ◀

Remark 1.37. One can always effectively compute the norm of an ideal, since $\#(R/\mathfrak{a}) = |\det M_{\mathfrak{a}}|$, where $M_{\mathfrak{a}}$ is the basis matrix of $\mathfrak{a}$ in Hermite normal form, see subsection 2.3.1. ◀

Lemma 1.38 (Properties of the norm)**.** *The norm function of a number ring R has the following properties:*

(i) $N(\alpha\beta) = N(\alpha)N(\beta)$, for elements $\alpha, \beta \in R$;

[3] Every cyclotomic field $\mathbb{Q}(\zeta_m)$ has $\mathbb{Z}[\zeta_m]$ as its ring of integers [Jan96, Ch. 1, Thm. 10.4].

(ii) If $R = \mathcal{O}_K$ is the ring of integers, then $N(\mathfrak{a}\mathfrak{b}) = N(\mathfrak{a})N(\mathfrak{b})$, for any two ideals $\mathfrak{a}, \mathfrak{b}$ of $\mathcal{O}_K$;

(iii) If $R = \mathcal{O}_K$ is the ring of integers, and $\mathfrak{p}$ is a prime of R, we have $N(\mathfrak{p}) = p^f$, with f the residue class degree of $\mathfrak{p}$.

Proof. See [Jan96, p. 42–44, Prop. 8.1, 8.2, 8.4]. □

Greatest common divisor of ideals

The group $\mathcal{I}_K$ has, besides ideal multiplication, many other operations and one is of particular importance in this thesis.

Definition 1.39 (Greatest common divisor of ideals)**.** There is a greatest common divisor operation on $\mathcal{I}_K$, which is denoted by $+$. It is defined as follows:

$$\mathfrak{a} + \mathfrak{b} := \{a + b \mid a \in \mathfrak{a}, b \in \mathfrak{b}\}.$$

Remark 1.40. As expected, this operation is fully consistent with the unique ideal factorization; if one has $\mathfrak{a} = \prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{a})}$ and $\mathfrak{b} = \prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{b})}$, then

$$\mathfrak{a} + \mathfrak{b} = \prod_{\mathfrak{p}} \mathfrak{p}^{\min(v_{\mathfrak{p}}(\mathfrak{a}), v_{\mathfrak{p}}(\mathfrak{b}))},$$

just as in $\mathbb{Z}$. Note that the unique factorization of ideals is just a number field version of the fundamental theorem of arithmetic. ◀

1.2.4 Discriminant and singular primes

In practice, one obtains a number ring in the sense of Remark 1.11, without knowing if it is equal to the ring of integers. Finding the ring of integers of a given number field K is hard[4], and even the decision problem whether a given number ring R equals the ring of integers or not, is well-known to be hard in the worst case [Chi89]. On the other hand, finding the ring of integers in number fields with a defining polynomial having small coefficients and small degree is not that hard, in practice. Also, for 'larger' number fields, effective approximation algorithms are known [JB94].

In the main algorithms of this thesis (Algorithm 9 and Algorithm 10), one does not need the full ring of integers. In the tests and the timings of the algorithm, I only used cyclotomic fields $\mathbb{Q}(\zeta_m)$, where the ring of integers is known to be $\mathbb{Z}[\zeta_m]$. In the case when one *does not* know if the given ring R is the ring of integers, one has to take care of the so-called singular primes.

Singular prime ideals

The following definition is obtained from [Ste08, p. 13] in combination with [Ste08, Prop. 5.4], and requires localization (see [AM69, Ch. 3]). Denote by S a multiplicatively closed set, and by $S^{-1}R$ the ring of fractions of R at the set S (see [AM69, p. 37]). Sometimes, $S^{-1}R$ is called the localization of R at S, despite the fact that $S^{-1}R$ does not have to be a local ring at all.

[4] In the article [JB94, Th. 1.3] it is proven that finding the ring of integers of a number field K is equally as hard as finding the largest squarefree divisor of a number d of which the size equals the size of K.

Definition 1.41 (Singular prime ideals)**.** Let K be a number field with $\mathcal{O}_K$ as ring of integers, and let R be a number ring inside K. Let $\mathfrak{p}$ be a prime ideal of R and let $S = R\backslash\mathfrak{p}$. One calls $\mathfrak{p}$ a singular prime when the inclusion $R \subseteq \mathcal{O}_K$ induces a strict inclusion when localized at S, i.e.

$$S^{-1}R \subsetneq S^{-1}\mathcal{O}_K.$$

Definition 1.42 (Regular primes)**.** A prime ideal $\mathfrak{p}$ in R is called regular when it is not singular, i.e., when $S^{-1}R = S^{-1}\mathcal{O}_K$.

Remark 1.43. Since $\mathfrak{p}$ is a prime ideal inside R, $S^{-1}R$ is a local ring; it is denoted by $R_\mathfrak{p}$. On the other hand, $S^{-1}\mathcal{O}_K$ does not need to be local, since it is well possible that there are multiple prime ideals in $\mathcal{O}_K$ that do not touch the set S (see [AM69, Prop. 3.11(iv)]). ◀

Lemma 1.44. *Suppose R is a number field in K with regular prime $\mathfrak{p}$. Then $R_\mathfrak{p} = (\mathcal{O}_K)_{\mathfrak{p}'}$ for some unique prime ideal $\mathfrak{p}'$ of $\mathcal{O}_K$, with $\mathfrak{p}' \cap R = \mathfrak{p}$.*

Proof. The 'going-up theorem' [AM69, Thm. 5.10] shows the existence of such a prime $\mathfrak{p}'$, yielding $R_\mathfrak{p} \subseteq (\mathcal{O}_K)_{\mathfrak{p}'}$. As $R_\mathfrak{p} = S^{-1}R = S^{-1}\mathcal{O}_K$ is a Noetherian local ring of dimension 1 that is integrally closed[5], it is a valuation ring inside K [AM69, Prop. 9.2]. Valuation rings are maximal, and therefore $R_\mathfrak{p} = (\mathcal{O}_K)_{\mathfrak{p}'}$. □

Remark 1.45. In Lemma 1.44, the prime ideal $\mathfrak{p}' = \mathfrak{p}\mathcal{O}_K$ suffices, when $\mathfrak{p}$ is regular. This result can be obtained by localizing at every prime of $\mathcal{O}_K$; the localization of $\mathfrak{p}$ of R at any ideal $\mathfrak{q}' \neq \mathfrak{p}'$ of $\mathcal{O}_K$ vanishes. ◀

Lemma 1.46. *Suppose K is a number field and R be a number ring in K. Suppose $\mathfrak{p}$ is a regular prime ideal in R, satisfying $\mathfrak{p}' \cap R = \mathfrak{p}$ for a prime ideal $\mathfrak{p}'$ of $\mathcal{O}_K$ (see Lemma 1.44). Then the inclusion $R \subseteq \mathcal{O}_K$ induces a isomorphism*

$$R/\mathfrak{p} \xrightarrow{\sim} \mathcal{O}_K/\mathfrak{p}'. \tag{1.4}$$

Proof. A short application of a local-global principle will do the job, see for example [AM69, Prop. 3.9]. Remark that the inclusion induces a map $f : R/\mathfrak{p} \to \mathcal{O}_K/\mathfrak{p}'$. Seeing those two rings as $\mathcal{O}_K$-modules, it is enough to show that – after localization at any prime – the induced map of f is bijective.

For any prime ideal $\mathfrak{q}'$ other than $\mathfrak{p}'$, both $R/\mathfrak{p}$ and $\mathcal{O}_K/\mathfrak{p}'$ become the zero ring after localization with $\mathfrak{q}'$, and the 'localized' map f is trivially bijective. Localizing at $\mathfrak{p}'$ induces a bijection by the fact that $\mathfrak{p}$ is regular, and therefore $R_\mathfrak{p} = (\mathcal{O}_K)'_\mathfrak{p}$. □

Remark 1.47. Note that Lemma 1.46 implies that $N_{\mathcal{O}_K}(\mathfrak{p}') = N_R(\mathfrak{p})$, as in Definition 1.34. ◀

Example 1.48. Note that for the ring of integers $\mathcal{O}_K$, every prime $\mathfrak{p}'$ is regular. For an example of a singular prime, consider $\mathfrak{p} = (2, 1 + \sqrt{-3})$ in the ring $R = \mathbb{Z}[\sqrt{-3}]$, a number ring in the number field $K = \mathbb{Q}(\sqrt{-3})$. The ring of integers of K is $\mathcal{O}_K = \mathbb{Z}[\rho]$ with $\rho = \frac{1+\sqrt{-3}}{2}$. Take the prime ideal $\mathfrak{p}' = (2)$ inside $\mathcal{O}_K$, and note that the map $R \to \mathcal{O}_K/\mathfrak{p}'$ is not surjective, since the (reduction

[5] Those properties are preserved under localization by a multiplicatively closed set S [AM69, Prop. 7.3], [AM69, Prop. 5.12].

of the) element ρ is not in the image; every element of $\mathbb{Z}[\sqrt{-3}]$ is of the form $a + b\sqrt{-3} = (a-b) + 2b\rho$ with $a, b \in \mathbb{Z}$. This maps under $R \to \mathcal{O}_K/\mathfrak{p}'$ to $\overline{a-b}$ mod 2. So, $R/\mathfrak{p} \to \mathcal{O}_K/\mathfrak{p}'$ is not surjective, and thus, by Lemma 1.46, $\mathfrak{p}$ is singular. ◀

Lemma 1.49. *Let K be a number field with ring of integers $\mathcal{O}_K$ and let $R \subset \mathcal{O}_K$ be a number ring. Suppose $\mathfrak{p}$, a prime above p, is singular in R. Then $p \mid [\mathcal{O}_K : R]$, and therefore $p^2 \mid \Delta(R)$.*

Proof. Take the multiplicative closed set $T = \mathbb{Z}\backslash(p)$, and apply localization to the exact sequence

$$0 \to R \to \mathcal{O}_K \to \mathcal{O}_K/R \to 0, \tag{1.5}$$

yielding the exact sequence [AM69, Prop. 3.3]

$$0 \to T^{-1}R \xrightarrow{f} T^{-1}\mathcal{O}_K \to T^{-1}(\mathcal{O}_K/R) \to 0. \tag{1.6}$$

For $S = R\backslash\mathfrak{p}$, we have $T \subseteq S$, and therefore

$$S^{-1}(T^{-1}R) = S^{-1}R \text{ and } S^{-1}(T^{-1}\mathcal{O}_K) = S^{-1}\mathcal{O}_K.$$

Since $S^{-1}R \subsetneq S^{-1}\mathcal{O}_K$, we must have $T^{-1}R \subsetneq T^{-1}\mathcal{O}_K$ as well, meaning that f in (1.6) is not surjective, and in particular, $T^{-1}(\mathcal{O}_K/R)$ is non-trivial.

Now, seeing the rings in (1.5) as $\mathbb{Z}$-modules, and remarking that $\mathcal{O}_K/R$ is then a finite $\mathbb{Z}$-module, one obtains that $T^{-1}(\mathcal{O}_K/R)$ is a finite $\mathbb{Z}_p$-module. By the structure theorem for finitely generated modules of a principal ideal domain (see for example [Hun03, pp. Lm. IV.6.11]), one has:

$$\mathcal{O}_K/R \simeq \bigoplus_{i=1}^{r} \mathbb{Z}/q_i,$$

where q_i are powers of prime numbers. Since localizing at (p) makes all $\mathbb{Z}/q_i$ vanish when $p \nmid q_i$, one has

$$T^{-1}(\mathcal{O}_K/R) \simeq \bigoplus_{i=0}^{r'} \mathbb{Z}/(p^{k_i}).$$

Together with the fact that $T^{-1}(\mathcal{O}_K/R)$ is non-trivial, one necessarily has $p \mid \#(\mathcal{O}_K/R)$. The rest of the claim follows from Lemma 1.22. □

Lemma 1.50. *Suppose K is a number field with number ring R, that has a set of singular prime ideals S. Suppose $\mathfrak{a} \neq 0$ is an ideal in R with $\mathfrak{p} + \mathfrak{a} = R$ for all $\mathfrak{p} \in S$ (i.e., no singular prime divides $\mathfrak{a}$). Then $\mathfrak{a}$ can uniquely be decomposed as a product of regular prime ideals:*

$$\mathfrak{a} = \prod_{\mathfrak{p} \notin S} \mathfrak{p}^{n_\mathfrak{p}} \tag{1.7}$$

Proof. According to [AM69, Prop. 9.1], each nonzero ideal of R can uniquely be expressed as a product of primary ideals, whose radicals are all distinct.

$$\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{q}_i.$$

Suppose $\mathfrak{q}_i$ is a $\mathfrak{p}$-radical ideal. Since $\mathfrak{p}$ is an ideal above a regular prime p, the ring $R_\mathfrak{p}$ is a discrete valuation ring [CF67, p. 6, Prop. 1]. In such rings, every ideal is a power of $\mathfrak{p}R_\mathfrak{p}$. So, $(\mathfrak{q}_i)R_\mathfrak{p} = (\mathfrak{p}R_\mathfrak{p})^j$, for some $j > 0$. At all other localizations, both $\mathfrak{q}_i$ and $\mathfrak{p}$ vanish. Using the global-local property [AM69, Prop. 3.8], we have $\mathfrak{q}_i = \mathfrak{p}^j$, see also [AM69, Thm. 9.3]. Applying this reasoning to each primary ideal, we obtain a factorization of $\mathfrak{a}$ in prime ideals, which is unique by the same reasoning as in Dedekind rings, see [Neu99, Thm. 3.3, p. 18]. □

Remark 1.51. Note that Lemma 1.49 does not yield a procedure to find singular primes, other than factoring the discriminant $\Delta(R)$, but is very useful when one wants to avoid singular primes, which indeed is needed in Algorithm 8 and Algorithm 10 of this thesis. For an element $\alpha \in R$, one can calculate $d = \gcd(N(\alpha), \Delta(R))$. There are two cases.

(i) $d = 1$, which means that α does not have a singular prime in its factorization. Therefore, the ideal (α) has unique factorization into prime ideals, making it suitable for the 'naive' computation of the power residue symbol, see Definition 3.4.

(ii) $d \neq 1$, which means that one has likely a partial factorization of $\Delta(R)$. This is computationally profitable, since this brings us closer to finding the ring of integers, or proving that R is the ring of integers of K. Also, one can calculate $d' = \gcd(N(\alpha)^2, \Delta(R))$, and compute $c := d'/d$.

  (a) If $c = 1$, then none of the singular primes divide $N(\alpha)$, which means that α has also unique factorization into prime ideals.

  (b) If $c \neq 1$, it is possible that α does not have unique prime ideal factorization in R, making α unsuitable to calculate with, since the power residue symbol above α is then undefined (see Definition 3.15). Note that factorization of c gives possibly singular primes, which allows us to enlarge the ring R, meaning that it will become closer to $\mathcal{O}_K$.

◀

1.3 Local Fields and Completions

1.3.1 Introduction

In this thesis, one needs the definition of the Hilbert symbol. This symbol is defined over local fields, which arise naturally as completions of number fields. A short outline about completions, local fields, and their relation with number fields will be treated in this section. Also, some computational issues in local fields will be discussed. For a thorough treatment, I would like to recommend [Cas86], [Jan96] or [Wei98].

1.3.2 Absolute values

Just as one obtains $\mathbb{R}$ from $\mathbb{Q}$ by completion, one also can make a completion of a number field, in a similar way. As completion is a topological construct, one first needs a topology on the number field K – in this case, a metric topology. The following definition is obtained from [Chi07, p. 9, p. 64].

Definition 1.52 (Absolute value)**.** Suppose K is a number field. A function $|\cdot| : K \to [0, \infty)$ is called an absolute value if

(i) $|\alpha| = 0$ if and only if $\alpha = 0$;

(ii) $|\alpha\beta| = |\alpha||\beta|$ for all $\alpha, \beta \in K$;

(iii) There is a constant $C \in \mathbb{R}_{\geq 1}$ such that $|1 + \alpha| \leq C$ when $|\alpha| \leq 1$.

An absolute value as above induces a metric topology on K with neighbourhoods of the form $\{\alpha \in K \mid |\alpha| < \epsilon\}$ for $\epsilon \in \mathbb{R}_+$.

Definition 1.53 (Equivalent absolute values)**.** Two absolute values $|\cdot|_1, |\cdot|_2 : K \to [0, \infty)$ are called equivalent when $|\cdot|_1 = |\cdot|_2^c$ for some $c \in \mathbb{R}\backslash\{0\}$.

Remark 1.54. Equivalent absolute values induce the same topology on K. In this thesis, we exclude the trivial absolute value, that has value 1 everywhere. Note that for every absolute value $|\cdot|_1$ on K, there exists $c \in \mathbb{R}\backslash\{0\}$ such that $|\cdot|_2 := |\cdot|_1^c$ satisfies the triangle identity:

$$|\alpha + \beta|_2 \leq |\alpha|_2 + |\beta|_2.$$

◀

Definition 1.55 (Places)**.** A place of K is an equivalence class of absolute values, with equivalence as in Definition 1.52 denoted by $\sim$. We define the set of places of K as

$$V_K := \Big\{ |\cdot| \;\Big|\; |\cdot| \text{ is an absolute value on } K \Big\} / \sim .$$

Theorem 1.56 (Ostrowski)**.** *All places of a number field fall into one of the following categories:*

(i) The $\mathfrak{p}$-adic places. They contain an absolute value defined by $|\alpha|_{\mathfrak{p}} := N(\mathfrak{p})^{-v_{\mathfrak{p}}(\alpha)}$, with $v_{\mathfrak{p}}$ the $\mathfrak{p}$-adic valuation as in Definition 1.27, and the norm N of an ideal as in Definition 1.34. These are also called the non-Archimedean, finite or discrete places of K.

(ii) The infinite real places. They contain an absolute value defined by a real embedding $\sigma : K \to \mathbb{R}$, with: $|\alpha|_\sigma := |\sigma(\alpha)|_{\mathbb{R}}$, where $|\cdot|_{\mathbb{R}}$ is the standard real absolute value of $\mathbb{R}$.

(iii) The infinite complex places. They contain an absolute value defined by a pair of conjugate complex embeddings $\sigma, \bar{\sigma} : K \to \mathbb{C}$. The absolute value is then $|\alpha|_\sigma = |\alpha|_{\bar{\sigma}} = |\sigma(\alpha)|_{\mathbb{C}}^2$, with $|a + bi|_{\mathbb{C}}^2 = a^2 + b^2$, the standard metric on $\mathbb{C}$.

Proof. A proof can be found in [ZH80, Ch. 13]. □

Remark 1.57. If we speak about the places above, we will always associate the 'standard absolute value' with it. These are the absolute values as described in Theorem 1.56. ◀

1.3.3 $\mathfrak{p}$-adic completions

A number field is not topologically complete. In order to make the number field K complete with respect to an absolute value $|\cdot|$, one can construct the completion of K.

Definition 1.58 (Cauchy and null sequences)**.** A sequence $(\alpha_i)_{i=0}^{\infty}$ is called a Cauchy sequence with respect to $|\cdot|$ if we have: For all $\epsilon > 0$ there exists $N \in \mathbb{N}$ such that for all $n, m \geq N$ holds

$$|\alpha_n - \alpha_m| < \epsilon.$$

A sequence $(\alpha_i)_{i=0}^{\infty}$ is called a null sequence with respect to $|\cdot|$ if we have: For all $\epsilon > 0$ there exists $N \in \mathbb{N}$ such that for all $n \geq N$ holds

$$|\alpha_n| < \epsilon.$$

Definition 1.59 (Completion)**.** Suppose K is a number field and $|\cdot|$ an absolute value on K. We define the following abelian additive group (under row-wise addition)

$$\mathcal{C} := \{(\alpha_i)_{i=0}^{\infty} \mid (\alpha_i)_{i=0}^{\infty} \text{ is a Cauchy sequence w.r.t. } |\cdot|\}$$

and the following subgroup

$$\mathcal{N} := \{(\alpha_1)_{i=0}^{\infty} \mid \alpha_i)_{i=0}^{\infty} \text{ is a null-sequence w.r.t. } |\cdot|\}.$$

Then the completion $K_{|\cdot|}$ of K with respect to the absolute value $|\cdot|$ is defined as the following quotient group:

$$K_{|\cdot|} := \mathcal{C}/\mathcal{N}.$$

Lemma 1.60. *The group $K_{|\cdot|}$ is a complete field with multiplication defined row wise, and has K as a subfield.*

Proof. See for example [Jan96, Ch. 2, Thm. 2.1]. □

Remark 1.61. One denotes $K_\mathfrak{p}$ for the completion of a number field with respect to the $\mathfrak{p}$-adic metric. Also one uses the notation K_σ for the completion with respect to the absolute value defined by the embedding σ (inside the real- or the complex numbers). Note that $K_\sigma \simeq \mathbb{R}$ when σ is a real embedding and $K_\sigma \simeq \mathbb{C}$ if σ is a complex embedding. ◀

The above abstract construction might not appeal to one's mind intuitively, and is in fact very rarely used in a computational context. In the next section, we will explain how one copes with such fields in an algorithmic context. The following theorem, of which a generalization is stated in [Jan96, Ch. 2, Thm. 2.2], already gives an idea how a completion looks like.

Lemma 1.62 (Extension of $\mathbb{Q}_p$)**.** *Suppose K is a number field and $\mathfrak{p}$ is a prime ideal of $\mathcal{O}_K$ above a prime number p. Then, $K_\mathfrak{p}$ is a finite extension of $\mathbb{Q}_p$, the p-adic numbers.*

Remark 1.63. Lemma 1.62 also works the other way around; every finite extension of the p-adic field $\mathbb{Q}_p$ is isomorphic as a topological field to a completion of a number field [Koc97, p.55–56]. So, in some sense, there is no difference between completions of number fields and finite extensions of the p-adic rationals. ◀

1.3.4 $\mathfrak{p}$-adic local fields

Definition 1.64 ($\mathfrak{p}$-adic local fields)**.** A $\mathfrak{p}$-adic local field is a completion of a number field with respect to the $\mathfrak{p}$-adic absolute value.

Remark 1.65*.* A $\mathfrak{p}$-adic local field $K_{\mathfrak{p}}$ has a valuation ring

$$R := \{\alpha \in K_{\mathfrak{p}} \mid |\alpha|_{\mathfrak{p}} \leq 1\},$$

with unique maximal ideal

$$\mathfrak{m} := \{\alpha \in K_{\mathfrak{p}} \mid |\alpha|_{\mathfrak{p}} < 1\}.$$

Moreover, this maximal ideal $\mathfrak{m} = (\pi)$ is a principal ideal in R, and π is called a uniformizer of $\mathfrak{m}$. It can be obtained by taking an element $\alpha \in K$ with $\alpha \in \mathfrak{p}\backslash\mathfrak{p}^2$, and taking the image of α under the inclusion $K \hookrightarrow K_{\mathfrak{p}}$ [Jan96, Prop. 2.4]. ◀

Lemma 1.66. *Given a system of representatives S (with $0 \in S$) of $R/\mathfrak{m}$, every element $\alpha \in K_{\mathfrak{p}}$ has a unique (possibly infinite) expression as a power series*

$$\alpha = \pi^r \sum_{i=0}^{\infty} s_i \pi^i,$$

with $s_i \in S$, $s_0 \neq 0$ and $r \in \mathbb{Z}$.

Proof. See, for example [Jan96, Prop. II.2.8] or [Koc97, Prop. 1.70]. □

Notation 1.67. In the remainder of this thesis, F denotes a local field that is a finite extension of $\mathbb{Q}_p$. We will denote by $\mathcal{O}_F$ the ring of integers and by $\mathfrak{m}_F = (\pi_F)$ the unique maximal ideal. Also, we denote $\mathbb{F}_F = \mathcal{O}_F/\mathfrak{m}_F$. Sometimes, we will denote the $\mathfrak{m}_F$-valuation of an element in F by $v_F : F \to \mathbb{Z}$. Of course, the subscript F will be dropped when there is no confusion about the local field.

Definition 1.68. Let $E : F : \mathbb{Q}_p$ be a tower of finite extensions, then we define

$$f(E/F) = [\mathbb{F}_E : \mathbb{F}_F] = [\mathcal{O}_E/\mathfrak{m}_E : \mathcal{O}_F/\mathfrak{m}_F],$$
$$e(E/F) = v_E(\mathfrak{p}_F),$$

for the residue class degree and the ramification index, respectively.

Remark 1.69*.* For a completion $K_{\mathfrak{p}}$ of a number field K, we have $e(K_{\mathfrak{p}}/\mathbb{Q}_p) = e(\mathfrak{p})$ and $f(K_{\mathfrak{p}} : \mathbb{Q}_p) = f(\mathfrak{p})$ [Jan96, Thm. II.3.8], with $f(\mathfrak{p})$ and $e(\mathfrak{p})$ as in Definition 1.29. Therefore, the fact that those invariants have the same name will not lead to conflicts. ◀

Lemma 1.70. *Suppose K is a number field and $\mathfrak{p}$ a is prime in $\mathcal{O}_K$. Then the extension $K_{\mathfrak{p}} : \mathbb{Q}_p$ has degree $e(\mathfrak{p}) \cdot f(\mathfrak{p})$.*

$$\begin{tikzcd}
& K_{\mathfrak{p}} \arrow[d, dash, "e(\mathfrak{p})f(\mathfrak{p})"] \arrow[dl, dash, dashed] \\
K \arrow[d, dash, "n"] & \mathbb{Q}_p \arrow[dl, dash, dashed] \\
\mathbb{Q} &
\end{tikzcd}$$

Proof. See for example [Jan96, §II, Th.3.8]. □

Remark 1.71. Also for an 'arbitrary extension' $F : \mathbb{Q}_p$, i.e., if F is the completion of some unknown number field, the equality

$$[F : \mathbb{Q}_p] = e(F/\mathbb{Q}_p)f(F/\mathbb{Q}_p)$$

is still valid. ◀

Notation 1.72. For a tower of finite extensions $E : F : \mathbb{Q}_p$, the extension $E : F$ is called unramified when $f(E/F) = [E : F]$ and it is called totally ramified when $e(E/F) = [E : F]$. An extension is not necessarily either unramified or totally ramified, it can be some 'mixture' of these two.

Definition 1.73 (Ramified representation)**.** A finite extension $F : \mathbb{Q}_p$ is given in a ramified representation if one has a subfield $E \subseteq F$, such that $F : E$ is totally ramified and $E : \mathbb{Q}_p$ is unramified.

$$\begin{tikzcd}
F \arrow[d, dash, "e"] \\
E \arrow[d, dash, "f"] \\
\mathbb{Q}_p
\end{tikzcd}$$

Lemma 1.74 (Ramified Representation)**.** *Every finite extension $F : \mathbb{Q}_p$ has a ramified representation $F : E : \mathbb{Q}_p$.*

Proof. We follow [Wei98, Thm. 3-2-5] in combination with [Wei98, Thm. 3-2-10]. We can choose a generator of the extension $[\mathbb{F}_F : \mathbb{F}_{\mathbb{Q}_p}]$, and denote it $\bar{\gamma}$. Calculate its minimum polynomial over $\mathbb{F}_{\mathbb{Q}_p} = \mathbb{F}_p$, with linear algebra techniques. Then lift this polynomial to $\mathbb{Z}[X]$, and denote it $f(x)$. This will be the defining polynomial for $E : \mathbb{Q}_p$. With Newton approximation [Wei98, §3-1], one can find an element $\gamma \in F$ such that $f(\gamma) = 0$. So, $E = \mathbb{Q}_p(\gamma) \subseteq F$.

Now, take $\pi_F \in F$, a generator of the maximum ideal $\mathfrak{m}_F$. Seeing F as an E-vectorspace, one can obtain the minimum polynomial $g(z) \in E[z]$ of π_F over E, which is an Eisenstein polynomial [Wei98, Thm. 3-3-1]. □

Remark 1.75. The inclusion $K \hookrightarrow K_\mathfrak{p}$ is generally given by the power series expression of Lemma 1.66. Taking the representative set $S = \mathbb{Z} + \mathbb{Z}\gamma + \cdots + \mathbb{Z}\gamma^{f-1}$, and taking an element $\pi \in K$ that lies in $\mathfrak{p} \backslash \mathfrak{p}^2$, one can write, for every $\alpha \in K$,

$$\alpha = \pi^r \sum_{i=0}^{\infty} s_i \pi^i.$$

This can be done in the following way. First assume $r = 0$, otherwise divide α by an appropriate power of π. Then, find an element $s \in S$ such that $\alpha - s \in \mathfrak{p}$, and divide $\alpha - s$ by π, etcetera, until a suitable precision is reached. ◀

Definition 1.76 (Teichmüller map)**.** Any extension $F : \mathbb{Q}_p$ with residue class degree f, has a primitive $p^f - 1$-th root of unity [Jan96, Th. II.3.9], which is in fact a 'Newtonian' lift of a generator of the residue field $\mathbb{F}_F$. Writing μ_m for the m-th roots of unity, we have $\mu_{p^f-1} \subseteq F^*$. We denote by $\omega : \mathbb{F}_F^* \to \mu_{p^f-1} \subseteq F^*$ the Teichmüller map, which takes the Newton lift of the elements in the residue field.

Definition 1.77 (Tame and wild ramifications)**.** Let $F : \mathbb{Q}_p$ be a finite extension. We call $F : \mathbb{Q}_p$ tamely ramified (or tame) when $p \nmid e(F/\mathbb{Q}_p)$. On the other hand, we call $F : \mathbb{Q}_p$ wildly ramified when $p \mid e(F/\mathbb{Q}_p)$.

Lemma 1.78. *Suppose R is a number ring in a number field K, or a ring of integers of an extension $F : \mathbb{Q}_p$. Suppose $\zeta_m \in R$. Then, for any prime ideal $\mathfrak{p}$ of R with $\mathfrak{p} \nmid m$, we have an injection:*

$$\langle \zeta_m \rangle = \mu_m \hookrightarrow R/\mathfrak{p}$$

Proof. Suppose *ad absurdum* that $\zeta_m^j - \zeta_m^k \equiv 0$ modulo $\mathfrak{p}$ for some $j \neq k$. Multiplying with an appropriate power of ζ_m gives $1 - \zeta_m^i \equiv 0$ modulo $\mathfrak{p}$, for some $i \neq 0$. That is, $\mathfrak{p} \mid 1 - \zeta_m^i$.

Using $f(x) = x^{m-1} + \ldots + x + 1 = \prod_{i=1}^{m}(x - \zeta_m^i)$, we can conclude that $\mathfrak{p} \mid f(1) = m$, contradiction.

Therefore, the reduction map $\mu_m \to R/\mathfrak{p}$ is injective. □

Remark 1.79. In particular, $\bar{\mu}_m \subseteq (R/\mathfrak{p})^*$ is a multiplicative subgroup of $(R/\mathfrak{p})^*$, and therefore $m \mid \#(R/\mathfrak{p}) - 1 = N(\mathfrak{p}) - 1$.

◀

CHAPTER 2

Ideals and lattices

2.1 Introduction

The main component of the heuristic reduction Algorithm 9 in this thesis is Lenstra-Lenstra-Lovász lattice reduction, often called LLL-reduction. This polynomial time reduction algorithm [LLL82] gives a 'relatively good' solution to the shortest vector problem (SVP), which is an NP-hard problem [EB81], [Ajt98].

The heuristic algorithms 9 and 10 in this thesis use the greatest common divisor of ideals, which is calculated by applying the Hermite normal form to the basis matrices of those ideals [Coh93, p. 67].

In order to explain those crucial ingredients, I need some notation and definitions.

2.2 Lattices

Although there are many different ways to define lattices [CS99, p. 3, p. 42], I will use the following from [Coh93, p. 79–80], which is preferable because of its simplicity and conciseness.

Definition 2.1 (Bilinear form)**.** Let V be an F-vector space, with $F = \mathbb{Q}$ or $\mathbb{R}$. Then the map $b : V \times V \to F$ is called a positive-definite (symmetric) bilinear form if:

(a) $b(\cdot, v_0) : V \to F$ is a linear map, for fixed $v_0 \in V$;

(b) $b(v, w) = b(w, v)$;

(c) $b(v, v) > 0$ for all $v \in V \backslash \{0\}$ (positive definite).

Definition 2.2 (Lattice)**.** A lattice L is a free $\mathbb{Z}$-module of finite rank, together with a positive definite bilinear form on the $\mathbb{R}$-vector space $L \otimes_{\mathbb{Z}} \mathbb{R}$.

Notation 2.3. If we replace $\mathbb{R}$ by $\mathbb{Q}$ in Definition 2.2, then we call L a $\mathbb{Q}$-lattice.

Notation 2.4. For L a lattice with bilinear form b, we will denote

$$\|v\|_b = \sqrt{b(v,v)}$$

for the b-length of the vector $v \in L$. When one has a basis B of L, one can enrich L with the Euclidean norm. One then writes $\|v\|_2$ for the Euclidean vector length, which equals $\sqrt{v_1^2 + \ldots + v_n^2}$, where $(v_1, \ldots, v_n)$ is v written on the basis B of L.

Definition 2.5. An n-dimensional lattice L is called an integral lattice if a basis is given by the rows of a matrix $M \in M_{n\times n}(\mathbb{Z})$.

Definition 2.2 could be considered as quite abstract, since a lattice is often just represented by an integer-valued matrix together with the standard inner product as the bilinear form. In this thesis, it is not much different; the following example gives you an idea how lattices will be treated here.

Example 2.6. Let L be the lattice $\mathbb{Z}$-generated by the rows of the following matrix, together with the standard inner product on $L \otimes_{\mathbb{Z}} \mathbb{R}$:

$$M = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 4 & 2 & 1 & 1 \\ 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}.$$

The group L is a $\mathbb{Z}$-module, and it is free of rank 4, because it can be sandwiched between two free modules of rank 4:

$$L_5 \subseteq M \subseteq L_1,$$

where L_5 is the lattice generated by $5 \cdot I$, and L_1 the lattice generated by I. Here, I is the unit matrix with dimension 4. ◀

In fact, any lattice can be viewed as an integer-valued matrix together with an inner product [Coh93, p. 80]: since L is free $\mathbb{Z}$-module of finite rank, it has a finite $\mathbb{Z}$-basis, say: $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$. Then, one can write every element $x \in L$ as a $\mathbb{Z}$-linear combination of those basis elements:

$$x = \sum_{i=1}^{n} v_i \mathbf{b}_i \text{ with } v_i \in \mathbb{Z}.$$

Therefore, such an x can be represented by $v = (v_1, \ldots, v_n) \in \mathbb{Z}^n$. Note that this representation heavily relies on the choice of the basis. Taking $y \in L$, represented by $w = (w_1, \ldots, w_n)$, the bilinear form b on L satisfies:

$$b(x,y) = b\left(\sum_{i=1}^{n} v_i \mathbf{b}_i, \sum_{j=1}^{n} w_j \mathbf{b}_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} v_i w_j q_{ij} \text{ where } q_{ij} = b(\mathbf{b}_i, \mathbf{b}_j),$$

which equals

$$\begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix} Q \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \quad \text{with } Q = (q_{ij}).$$

The matrix Q gives rise to an important invariant of the lattice L.

Definition 2.7 (Determinant of a lattice)**.** Let L be a lattice, with a bilinear form b. Choose a basis $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ of L and set $Q = (b(\mathbf{b}_i, \mathbf{b}_j))$. Then, the determinant of the lattice is the following real invariant:

$$\Delta(L) = \sqrt{\det(Q)}.$$

The definition of $\Delta(L)$ is independent of base change; a change of basis from $(\mathbf{c}_1, \dots, \mathbf{c}_n)$ to $(\mathbf{b}_1, \dots, \mathbf{b}_n)$ coincides with multiplying the vectors $v \in \mathbb{Z}^n$ with a ‘transition matrix’ $M_{\mathbf{c}\to\mathbf{b}} \in \mathrm{GL}_n(\mathbb{Z})$. So, if v represents the element $x \in L$ with respect to the chosen basis $(\mathbf{c}_1, \dots, \mathbf{c}_n)$, then the vector $M_{\mathbf{c}\to\mathbf{b}}v$ represents the same element $x \in L$ but now with respect to the other basis $(\mathbf{b}_1, \dots, \mathbf{b}_n)$. Let Q be the same matrix as in the description above, then:

$$b(x, y) = (M_{\mathbf{c}\to\mathbf{b}}v)^T \, Q \, (M_{\mathbf{c}\to\mathbf{b}}w) = v^T M_{\mathbf{c}\to\mathbf{b}}^T Q M_{\mathbf{c}\to\mathbf{b}} w = v^T Q' w$$

with $Q' = M_{\mathbf{c}\to\mathbf{b}}^T Q M_{\mathbf{c}\to\mathbf{b}}$. Since $M_{\mathbf{c}\to\mathbf{b}}$ is in $\mathrm{GL}_n(\mathbb{Z})$, and those matrices have determinant ± 1, we conclude that $\det Q' = \det Q$. Also, Q is necessarily a positive definite matrix, by definition, and therefore $\det Q$ has to be positive, making $\sqrt{\det Q}$ a real number. So, in short, the determinant of a lattice is well defined. Note that the determinant of the lattice clearly depends on the chosen bilinear form on L.

Example 2.8. Using the same lattice as in Example 2.6, one sees that $B = (\mathbf{b}_1, \dots, \mathbf{b}_4) = ((4, 3, 2, 1), (4, 2, 1, 1), (5, 0, 0, 0), (0, 5, 0, 0))$ is a basis for L, because

$$5(\mathbf{b}_1 - \mathbf{b}_2) - \mathbf{b}_4 = (0, 0, 5, 0),$$

and

$$5\mathbf{b}_1 - 4\mathbf{b}_3 - 3\mathbf{b}_4 - 2(0, 0, 5, 0) = (0, 0, 0, 5).$$

Calculating inner products yields the following matrix $Q = (q_{ij})$:

$$Q = \begin{bmatrix} 30 & 25 & 20 & 15 \\ 25 & 22 & 20 & 10 \\ 20 & 20 & 25 & 0 \\ 15 & 10 & 0 & 25 \end{bmatrix}$$

with determinant $625 = 5^4$, so the lattice L has determinant $\det L = \sqrt{625} = 25$. ◀

The invariant $\Delta(L)$ also has a more intuitive geometrical interpretation; it is the volume of the parallelepiped solid inside $L \otimes \mathbb{R}$, spanned by a basis of L:

$$S := \{r_1\mathbf{b}_1 + \ldots + r_n\mathbf{b}_n \mid 0 \le r_i < 1\}.$$

This parallelepiped is often referred to as the covolume of the lattice and is denoted by V/L, where $V = L \otimes \mathbb{R}$. Now, we will end this section with a useful lemma, that helps us calculate the determinant of an integral lattice in an easier way.

Lemma 2.9. *For a full rank lattice L generated by an integral matrix M, together with the standard inner product, we have*

$$\Delta(L) = \#(\mathbb{Z}^n/L).$$

Proof. See [Cas97, p. 14, Lm. 1] or [PZ89, §3.2, Lm. 3.6]. □

2.3 Ideals as lattices

2.3.1 Basis matrix of a lattice

Suppose we have a number field K of degree n, with some number ring $R \subseteq K$. Since R is a $\mathbb{Q}$-lattice inside K, it has a $\mathbb{Z}$-basis, $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$. After the choice of this basis, we have

$$\mathbb{Z}\mathbf{b}_1 + \ldots + \mathbb{Z}\mathbf{b}_n = R,$$

as $\mathbb{Z}$-modules. So, in this particular way, R is identifiable with the lattice generated by I_n (the unit matrix in dimension n).

An ideal $\mathfrak{a}$ of R is a sublattice of the lattice of R, and can therefore be expressed in a basis of its own: $(\mathbf{c}_1, \ldots, \mathbf{c}_n)$. Since each of those basis elements $\mathbf{c}_i$ are in R, one can write them as a linear combination of the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$:

$$\mathbf{c}_i = \sum_{j=1}^{n} t_{ij}\mathbf{b}_j,$$

meaning:

$$\begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_n \end{bmatrix} = T \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{bmatrix} \text{ with } T = (t_{ij}) \text{ an integer valued matrix.}$$

So, in fact, the matrix T expresses the basis of $\mathfrak{a}$ on the basis of R.

Definition 2.10. A matrix $T_{\mathfrak{a}}$ that expresses a $\mathbb{Z}$-basis of the lattice $\mathfrak{a}$ in the chosen $\mathbb{Z}$-basis of the lattice R, is called a basis matrix of $\mathfrak{a}$.

In a computer algebra system like Magma or Sage, ideals are often represented by a basis matrix, after a fixed basis choice of R. The disadvantage of the above definition is that a basis matrix is not unique, since it clearly depends on the basis of $\mathfrak{a}$. Also, such a matrix T can have large entries, making it unpleasant to calculate with.

Lemma 2.11. *Let $\mathfrak{b}$ be an ideal of $\mathcal{O}_K$, and let $L_{\mathfrak{b}}$ be the ideal lattice generated by the basis matrix of $\mathfrak{b}$, with respect to some integral basis of $\mathcal{O}_K$. Then one has*

$$\det L_{\mathfrak{b}} = N(\mathfrak{b}).$$

Proof. Let $n = [K : \mathbb{Q}]$. The group $\mathbb{Z}^n/L_{\mathfrak{b}}$ is then canonically isomorphic to $\mathcal{O}_K/\mathfrak{b}$. According to Lemma 2.9, we have

$$\det L_{\mathfrak{b}} = \#(\mathbb{Z}^n/L_{\mathfrak{b}}) = \#(\mathcal{O}_K/\mathfrak{b}) = N(\mathfrak{b}).$$

□

2.3.2 The Hermite normal form

To ensure uniqueness of the basis matrix and to obtain good matrix properties, most computer algebra systems use the Hermite normal form, abbreviated the HNF. The following definition is adapted[1] from [Coh93, p. 67].

Definition 2.12 (Hermite normal form)**.** An $m \times n$ integer valued matrix $M = (m_{ij})$ is in Hermite normal form if there exists an $r \leq m$ and a strictly increasing map $f : \{1, \ldots, m-r\} \to \{1, \ldots, n\}$ satisfying the following properties.

(a) The last r rows of M are equal to zero;

(b) $m_{i,f(i)} > 0$ for $1 \leq i < m - r$;

(c) $m_{i,j} = 0$ when $j < f(i)$;

(d) $0 \leq m_{j,f(i)} < m_{i,f(i)}$ for $j < i$.

The above definition is quite formalistic and does not really appeal to one's imagination. In the case that M is a full rank $n \times n$ integer valued matrix, Definition 2.12 simplifies drastically.

Lemma 2.13 (Hermite normal form for full rank square matrices)**.** *A full rank integer valued $n \times n$ matrix $M = (m_{ij})$ is in Hermite normal form if*

(i) M is upper-triangular;

(ii) The diagonal entries of M are strictly positive;

(iii) For $i < j$ we have $0 \leq m_{ij} < m_{jj}$, i.e., every upper-diagonal entry is (strictly) smaller than the diagonal entry in its column.

Proof. Following Definition 2.12, we have $n = m$ in this case. Therefore $r = 0$ and $f = id$, since f is strictly increasing. Then the upper-triangle form of M follows from part (c) of Definition 2.12, positiveness of the diagonal entries from part (b) and property (iii) is a direct translation of part (d) of Definition 2.12. □

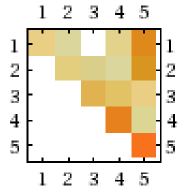


Figure 2.1: A matrix plot of a matrix in Hermite normal form. More red means a larger number (in absolute value).

Also full rank $m \times n$-matrices with $m \geq n$ in Hermite normal form have a shape that is easy to describe.

[1] In the literature, most authors differentiate between row-HNF and column-HNF. Cohen uses column-HNF, whereas I prefer row-HNF, so I altered the definition somewhat.

Lemma 2.14 (Hermite normal form for full rank matrices)**.** *A full-rank integer valued $m \times n$ matrix $M = (m_{ij})$ with $m \geq n$ is in HNF if*

(i) The last $m - n$ rows are zero;

(ii) The first n rows form a square matrix in HNF (as in Lemma 2.13).

Proof. The rank of M is equal to n, so M has at least n nonzero rows, so $r \leq m - n$, with r as in Definition 2.12. Because f is strictly increasing, it is in particular injective. Therefore, $m - r \leq n$, which is equivalent to $m - n \leq r$. So, we can conclude $r = m - n$, which proves (i). Note that this directly implies that $f = id$ and, thus, with the same arguments as in Lemma 2.13, the upper square submatrix of M is in Hermite normal form. □

Example 2.15. The following matrices are in HNF:

$$\begin{bmatrix} 3 & 1 & 0 & 2 \\ 0 & 3 & 2 & 1 \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad \begin{bmatrix} 6 & 4 & 2 & 1 & 3 \\ 0 & 0 & 4 & 3 & 2 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 5 & 20 & 13 & 2 \\ 0 & 23 & 3 & 0 \\ 0 & 0 & 15 & 3 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The left matrix clearly satisfies the requirements of Lemma 2.13, and the middle matrix satisfies Definition 2.12 with $r = 1$, $f(1) = 1$, $f(2) = 3$, $f(3) = 5$. Note that this matrix is not a full rank one. The right matrix is clearly in HNF as in Lemma 2.14. ◀

2.3.3 Computing the HNF

The proof of the following theorem about the Hermite normal form partially consists of an algorithm, which is adapted from [Coh93, p. 67]. The theorem is stated in its full form, whereas only a short outline of the HNF-algorithm is given, for sake of brevity.

Theorem 2.16. *Let M be an $m \times n$ integer-valued matrix, then there exists a unique $m \times n$ matrix H in Hermite normal form such that $H = UA$, with $U \in \mathrm{GL}_m(\mathbb{Z})$.*

Proof. Algorithm 1 ensures the existence of such an Hermite normal form; and since every row operation used is representable by a $\mathrm{GL}_m(\mathbb{Z})$-matrix, one can conclude that $H = UA$ for some $U \in \mathrm{GL}_m(\mathbb{Z})$, where U is just the product of these row operations.

Suppose $H = UA$ and $H' = U'A$, then $H' = U'U^{-1}UA = U'U^{-1}H$. Writing $U'' = U'U^{-1}$, one sees: For uniqueness it is sufficient to prove that $H' = UH$ for some $U \in \mathrm{GL}_m(\mathbb{Z})$ implies $H = H'$.

Both H' and H are in Hermite normal form; we denote the corresponding strictly increasing functions with f' respectively f and the amount of zero-rows at the bottom with r' respectively r. The row rank of the matrices H and H' are $m - r$, $m - r'$ respectively. Since a matrix $U \in \mathrm{GL}_m(\mathbb{Z})$ does not alter the row rank, we immediately conclude $r = r'$. So, f and f' have the same domain and codomain. The strictly increasing property of f and f' implies that U must be a upper triangular matrix – and with the fact that $U \in \mathrm{GL}_m(\mathbb{Z})$ and that the so-called ‘pivot entries’ $h_{k,f(k)}$ and $h'_{k,f'(k)}$ are strictly positive, we must have that the diagonal of U consists of ones. This immediately implies $f = f'$ and $h_{k,f(k)} = h'_{k,f'(k)}$. The property (d) of Definition 2.12 then implies $H = H'$. □

Algorithm 1 is obtained from [Coh93, p. 69], again with slight modifications. Although it is possible to keep track of the matrix U, this algorithm does not do that – for sake of brevity. Additionally, in practice, Algorithm 1 is nót the right recipe to compute Hermite normal forms effectively, because of coefficient explosion [HM90]. Instead, in real implementations, one uses tricky modifications of Algorithm 1, with modular arithmetic. This results in an algorithm, computing the HNF, that has a running time of $\mathrm{O}(mn^4 \log^2(M))$ for $m \times n$-matrices with entries bounded by M [MW01].

Algorithm 1: Computes row-Hermite normal form of a matrix A

```
HNF($A$);
Input  : An $m \times n$ integral matrix $A$
Output: A matrix $H$ such that $H = AU$ is in HNF, with $U \in \mathrm{GL_n}(\mathbb{Z})$
$c := 1$, $r := 1$ ;  // c and r are for column and row where we are calculating
while $c \leq n$ do
    while there exist a non-zero entry below $a_{r,c}$ do
        Choose smallest non-zero $a_{ic}$ with $i \geq r$ ; // smallest in abs. value
        and swap rows $A_i$ and $A_r$ ;
        for $i > r$ do
            $A_i := A_i - \left\lceil \frac{a_{ic}}{a_{rc}} \right\rceil \cdot A_r$ ; // Reduces every row below $r$ with row $A_r$
        end
    end
    if $a_{rc} < 0$ then
        $A_r := -A_r$ ;
    end
    if $a_{rc} \neq 0$ then
        for $i < r$ do
            $A_i := A_i - \left\lfloor \frac{a_{ic}}{a_{rc}} \right\rfloor \cdot A_r$ ; // Reduces every row above $r$ with row $A_r$
        end
        // Row $r$ and column $c$ of $A$ are reduced now.
        $c := c + 1$, $r := r + 1$ ;
    else
        $c := c + 1$ ;
    end
end
Move all zero rows of $A$ to the bottom ;
return $H := A$ ;
```

2.3.4 HNF and operations on ideals

The HNF has many applications [Coh93, p. 74], but in this thesis it is only used for computing the greatest common divisor of two ideals, see Definition 1.39.

Algorithm 2, adapted from [Coh93, p. 74, Sum of Modules], computes the ideal GCD of two ideals in a number field, using the HNF.

Algorithm 2: Computes the ideal $\mathfrak{a} + \mathfrak{b}$ from the ideals $\mathfrak{a}$ and $\mathfrak{b}$.

```
1 IdealGCD($\mathfrak{a}, \mathfrak{b}$);
  Input  : Two ideals $\mathfrak{a}, \mathfrak{b}$ in a number ring $R$, represented by their basis
           matrices
  Output: The ideal $\mathfrak{a} + \mathfrak{b}$, represented by its basis matrix
2 Concatenate the two basis matrices $M_\mathfrak{a}$ and $M_\mathfrak{b}$ vertically, and call this
    matrix $M$ ;
3 Calculate the HNF of this matrix, call the result $M$ ;
4 Extract $S$, the upper square matrix of this matrix $M$ ;
5 Return $S$ as basis matrix of the ideal $\mathfrak{a} + \mathfrak{b}$ ;
```

2.4 Lattice reduction: LLL

2.4.1 Introduction

The invention of the LLL-algorithm in 1982 by Lovász and the two Lenstra brothers was a breakthrough in the research about reduced bases [LLL82]. Although first intended for solving factorization in $\mathbb{Q}[X]$ and integer linear programming, the algorithm has a vast amount of applications now, mostly appearing in cryptography and number theory [NV10].

In this thesis, I will give a short description of the LLL-algorithm, together with some lemmata saying something about the running time and the quality of the output.

2.4.2 Reduced bases

From Definition 2.2, we know that every lattice L has a $\mathbb{Z}$-basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ in L; indeed, a lattice has infinitely many $\mathbb{Z}$-bases. From a computational viewpoint, some bases are better than others. Bases with relatively short and nearly orthogonal basis elements are a better choice for computing. In vector spaces it is clear that one can always find such a basis, by applying the Gram-Schmidt orthogonalization process. The following notation is based on that fact.

Notation 2.17. For a $\mathbb{Z}$-basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of L, define inductively the following orthogonalized basis by applying the Gram-Schmidt process.

$$\mathbf{b}_i^* := \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{ij} \mathbf{b}_j^* \text{ where } \mu_{ij} = \frac{b(\mathbf{b}_i, \mathbf{b}_j^*)}{b(\mathbf{b}_j^*, \mathbf{b}_j^*)} \text{ for } i = 1, \ldots, n.$$

Note that, in general, the basis $(\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ is not a $\mathbb{Z}$-basis of L, but it ís an $\mathbb{R}$-basis of $L \otimes \mathbb{R}$. This is mainly caused by the the fact that the Gram-Schmidt coefficients μ_{ij} are not integral, in general. The following definition uses above notation and is adapted from [NV10, p. 37, Def. 15].

Definition 2.18 (Size-reduced)**.** A basis $(\mathbf{b}_1, \dots, \mathbf{b}_n)$ of a lattice L is called size-reduced if

$$\left| \frac{b(\mathbf{b}_i, \mathbf{b}_j^*)}{b(\mathbf{b}_j^*, \mathbf{b}_j^*)} \right| = |\mu_{ij}| \leq \frac{1}{2} \text{ for } 1 \leq j < i \leq n.$$

Algorithm 3 is obtained from [NV10, p. 43] and computes a size-reduced basis from an arbitrary basis, in $O(n^5 \log(M)^2)$ time. Here n is the degree and M is the upper bound on the matrix entries.

Algorithm 3: Size-reduction algorithm

```
SizeReduce($B$);
Input  : A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ of a lattice $L$
Output: A size-reduced basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ of the lattice $L$
Compute Gram-Schmidt coefficients $\mu_{ij}$.
for $i := 2$ to $n$ do
    for $j := i - 1$ to $1$ do
        $\mathbf{b}_i := \mathbf{b}_i - \lceil \mu_{ij} \rfloor \mathbf{b}_j$
        for $k = 1$ to $j$ do
            $\mu_{ik} := \mu_{ik} - \lceil \mu_{ij} \rfloor \mu_{jk}$
        end
    end
end
```

Example 2.19. Let L be the lattice generated by $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ below, together with the standard inner product. The rows of the matrix form a $\mathbb{Z}$-basis of L, and this basis is not size-reduced. We have

$$\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 10 \\ 0 & 0 & 3 \end{bmatrix} \text{ and } \begin{bmatrix} \mathbf{b}_1^* \\ \mathbf{b}_2^* \\ \mathbf{b}_3^* \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ -\frac{16}{7} & -\frac{25}{7} & \frac{22}{7} \\ \frac{17}{130} & -\frac{1}{13} & \frac{1}{130} \end{bmatrix},$$

with $\mu_{2,1} = \frac{16}{7}$, $\mu_{3,1} = \frac{9}{14}$ and $\mu_{3,2} = \frac{22}{65}$. One sees that $\mu_{2,1}$ and $\mu_{3,1}$ violate the condition of being size-reduced, see Definition 2.18.

◀

The following definition from [LLL82] (in this form obtained from [NV10, p. 48]) gives a stronger version of 'being reduced' than being size-reduced only; the additional constraint is often called Lovász' condition. Recall (from Notation 2.4) that $\|v\|_b = \sqrt{b(v,v)}$, where b is the bilinear form on the lattice L.

Definition 2.20 (LLL-reduced)**.** Let δ be a real in $[\frac{1}{4}, 1]$. A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ is called δ-LLL-reduced if:

(Size reduced) B is size-reduced as in Definition 2.18.

(Lovász) For $1 < i \leq n$:

$$\delta \cdot \|\mathbf{b}_{i-1}^*\|_b^2 \leq \|\mathbf{b}_i^*\|_b^2 + \mu_{i,i-1}^2 \|\mathbf{b}_{i-1}^*\|_b^2$$

Algorithm 4 is obtained from [NV10, p. 48], and is a slightly simplified form of the 'real' LLL-algorithm. Although this simplified algorithm is much slower, LLL normally has time complexity $\mathrm{O}(n^5 m \cdot (\log_{1/\delta} M)^3)$ for an $n \times m$ matrix with entries bounded by M, see [NV10, p. 150, Thm. 3].

Algorithm 4: The simplified LLL-algorithm

```
LLL($B, \delta$);
Input  : A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$
Output: A $\delta$-LLL-reduced basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$
do
    Size-reduce the basis B;
    Find the smallest $i$ in $\{2, \dots, n\}$ which violates the Lovász condition,
      i.e.:
            $\delta \cdot \|\mathbf{b}_{i-1}^*\|_b^2 > \|\mathbf{b}_i^*\|_b^2 + \mu_{i,i-1}^2 \|\mathbf{b}_{i-1}^*\|_b^2$
    Swap $\mathbf{b}_i$ and $\mathbf{b}_{i-1}$;
while there exists an index i which does not satisfy the Lovász condition;
```

Definition 2.21 (Successive minima)**.** Let L be a lattice of dimension n. Define the set

$$\mathcal{B}_i := \left\{ V = \{v_1, \dots, v_i\} \subseteq L \;\middle|\; V \text{ is a linearly independent set of vectors} \right\},$$

and define, for $V = \{v_1, \dots, v_i\} \in \mathcal{B}_i$, $\|V\|_b = \max_{1 \le j \le i} \|v_j\|_b$. Then, the i-th successive minimum of L is defined as follows:

$$\lambda_i(L) := \min_{V \in \mathcal{B}_i} \|V\|_b.$$

Remark 2.22. Note that $\lambda_1(L)$ is just the b-length of the shortest vector in L. ◀

Theorem 2.23 (Minkowski's second theorem)**.** *Let L be a lattice of rank n. Then for any integer $1 \le r \le n$, one has*

$$\left(\prod_{i=1}^{r} \lambda_i(L) \right)^{1/r} \le \sqrt{\gamma_n} \cdot \Delta(L)^{1/n}$$

where γ_n is the nth Hermite's constant, which can be bounded by $1 + \frac{n}{4}$ [NV10, p. 33-35].

Proof. See for example [Mic, Thm. 12], [LG87, Thm. 2.9.1], [NV10, p. 35], [Cas97, Ch. VIII, p.202] or the historically correct [Min10, § 51]. □

The LLL-algorithm, of which a simplified form is shown in Algorithm 4, has the following bounds on its output [NV10, p. 48], where $\|\cdot\|_b$ is the b-induced norm.

Theorem 2.24. *Let $\delta \in (\frac{1}{4}, 1]$ and $\rho = 1/(\delta - \frac{1}{4})$, and let $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_n)$ be a δ-LLL-reduced basis of a lattice L. Then:*

(a) $\|\boldsymbol{b}_1\|_b \le \rho^{\frac{n-1}{4}} \cdot \Delta(L)^{\frac{1}{n}}$*;*

(b) For all $1 \le i \le n$ *we have* $\|\boldsymbol{b}_i\|_b \le \rho^{\frac{n-1}{2}} \lambda_i(L)$*;*

(c) $\prod_{i=1}^{n} \|\boldsymbol{b}_i\|_b \le \rho^{\frac{n(n-1)}{4}} \Delta(L)$*.*

Remark 2.25. In this thesis, δ always equals a half, which yields $\rho = 2$. For sake of readability, if we use the above result, we just take $\rho = 2$. One could be tempted to maximize δ, but since the running time of the algorithm depends on the choice of δ, a large δ is not always preferable. ◀

2.5 Element reduction modulo an ideal

In the heuristic algorithm, it is important that one can find 'small' representatives in the quotient ring $R/\mathfrak{b}$; the termination of the algorithm depends on it. For general number rings, finding a small representative for an element α mod $\mathfrak{b}$ is difficult, since this is equivalent to finding short vectors in translated ideal lattices, which is believed to be hard [ILC], [LPR10] and [SS11].

Fortunately, in the main algorithm of this paper, it is sufficient to find a 'relatively small' representative; and therefore, LLL applies here. We will first define and describe how to reduce modulo a basis matrix.

Definition 2.26. Let $\alpha \in R$, a number ring, and let $M_{\mathfrak{b}} = (\mathbf{m}_1, \ldots, \mathbf{m}_n)$ be a basis matrix of an ideal $\mathfrak{b}$ of R. The element α is $M_{\mathfrak{b}}$-small if

$$\alpha = \sum_{i=1}^{n} c_i \mathbf{m}_i \text{ for some } c_i \text{ with } |c_i| \le 1/2.$$

Alternatively, an element $\alpha \in R$ is $M_{\mathfrak{b}}$-small if it is in the following span: $\{c_1\mathbf{m}_1 + \ldots + c_n\mathbf{m}_n \mid |c_1|, \ldots, |c_n| \le 1/2\}$. Computing an $M_{\mathfrak{b}}$-small representative is not difficult and can be done by the following algorithm, adapted from [Coh00, Algorithm 1.4.13, p. 33].

Algorithm 5: Reduction modulo a basis matrix

```
ReductionModBasisMatrix($\alpha, M_{\mathfrak{b}}$);
Input  : An element $\alpha \in R$ and a basis matrix $M_{\mathfrak{b}}$ of an ideal $\mathfrak{b}$ of $R$.
Output: An element $\alpha' \in R$ such that $\alpha \equiv \alpha'$ mod $\mathfrak{b}$, and $\alpha'$ is $M_{\mathfrak{b}}$-small
Solve for $v$ in the linear system $vM_{\mathfrak{b}} = \alpha$;
Round every entry of $v$, i.e. $v_i := \lfloor v_i \rceil$ for every $i$;
Return $\alpha' := \alpha - vM_{\mathfrak{b}}$.
```

Remark that the size of a $M_{\mathfrak{b}}$-small element α heavily relies on the size of the basis elements $(\mathbf{m}_1, \ldots, \mathbf{m}_n)$. So, naturally, in order to find a relatively small representative of α modulo $\mathfrak{b}$, one first LLL-reduces the basis matrix $M_{\mathfrak{b}}$, and then one calculates an $M_{\mathfrak{b}}$-small representative of α, see Algorithm 6.

Lemma 2.27. *Suppose* α *is a small representative modulo an ideal* $\mathfrak{b}$*, then we have*

$$\|\alpha\|_b \le 2^{\frac{d-3}{2}} \sum_{i=1}^{d} \lambda_i(L_{\mathfrak{b}}),$$

where $L_{\mathfrak{b}}$ *denotes the lattice of the ideal* $\mathfrak{b}$*.*

Algorithm 6: Finding a relatively small representative modulo an ideal

```
SmallRepresentative($\alpha, \mathfrak{b}$);
Input  : An element $\alpha \in R$, an ideal $\mathfrak{b}$.
Output: An element $\alpha' \equiv \alpha \bmod \mathfrak{b}$, such that $\alpha'$ is ‘relatively small’
Find a basis matrix $M_{\mathfrak{b}}$;
LLL-reduce this matrix;
Return ReductionModBasisMatrix($\alpha, M_{\mathfrak{b}}$);
```

Proof. Using the triangle inequality yields

$$\|\alpha\|_b = \|\sum_{i=1}^{n} c_i \mathbf{m}_i\|_b \leq \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{m}_i\|_b \leq 2^{\frac{d-3}{2}} \sum_{i=1}^{d} \lambda_i(L_{\mathfrak{b}}).$$

□

Remark 2.28. The bound in Lemma 2.27 does not seem really useful, since the second successive minimum $\lambda_i(L)$ can be really large [NV10, p. 35], in arbitrary lattices L.

Note that – assuming that the $\lambda_i(L)$ do not vary too much for different i – one may heuristically deduce, by Theorem 2.23, that $\lambda_i(L) \approx \sqrt{\gamma_n}\Delta(L)^{1/n}$. Then, above bound simplifies to (also, see Lemma 2.11)

$$\|\alpha\|_b \leq 2^{\frac{d-3}{2}} \sum_{i=1}^{d} \lambda_i(L) \approx 2^{\frac{d-3}{2}} \cdot d \cdot \sqrt{\gamma_n} \cdot \Delta(L_{\mathfrak{b}})^{1/n} = 2^{\frac{d-3}{2}} \cdot d \cdot \sqrt{\gamma_n} \cdot N(\mathfrak{b})^{1/n},$$

which is, at least for ideals with superexponential norm in n, quite a good bound. When $\mathfrak{b} = (\beta)$ and $K : \mathbb{Q}$ is Galois, we have $N(\beta)^{1/n} = \sqrt[n]{\prod_{i=1}^{n} \sigma(\beta)} \leq \max_\sigma |\sigma(\beta)|_{\mathbb{C}}$. ◀

2.6 q-ary lattices

2.6.1 Introduction

In the starting phase of my research about the power residue symbol I attempted to compute the principal power residue symbol by reduction (i.e. Algorithm 9) solely. Proving that such a reduction algorithm terminates in polynomial time seemed to require extra assumptions about finding short vectors in q-ary lattices. These assumptions are formulated in the appendix, in Conjecture B.1, which is called the QSDL-conjecture.

Later in the process of writing my thesis, I found that applying reduction only is not enough (see Remark 5.11). To overcome this problem, I attempted a probabilistic way (Algorithm 10) to compute the principal power residue symbol, which seems to perform better than the reduction algorithm (see Figure 5.1, 5.2 and 5.3).

Having difficulties with proving that – assuming the QSDL-conjecture – the proposed reduction algorithm indeed terminates within polynomial time and lacking empirical evidence in the form of a trend in the timings (see Figure 5.2 and Figure 5.3), I decided to move the section about the QSDL-conjecture to the appendix (see section B.2). Also, it was suggested [Mic16] that, in theory, the chances that the QSDL-conjecture is true, are pretty slim.

So, briefly, I omit the QSDL-conjecture because it is not needed due to the evaluation algorithm, because I didn't find a way to deduce the termination of the reduction algorithm from it, and because some articles and authors suggest that the conjecture is not true (see Remark B.4).

2.6.2 q-ary lattices in the reduction algorithm

In this subsection – where we assume all lattices to live in $\mathbb{R}^n$ – a special form of lattices is treated: so-called q-ary lattices. We assume that lattices are given with some chosen basis, on which a bilinear map can be defined, see Remark 2.41.

Definition 2.29 (q-ary lattice)**.** Let $q \in \mathbb{Z}$ be a prime. An n-dimensional integral lattice L is called q-ary iff

$$q\mathbb{Z}^n \subseteq L$$

Notation 2.30. Let $q \in \mathbb{N}$. We denote L_q for the lattice $q\mathbb{Z}^n$.

Definition 2.31 (square-dense q-ary lattices)**.** Let $q \in \mathbb{Z}$ be a prime and let n be even. An n-dimensional q-ary integral lattice L is called square-dense if

(a) $L_q \subseteq L$ (q-ary);

(b) $\Delta(L) = \sqrt{\Delta(L_q)}$.

There occur q-ary square-dense lattices in the main algorithm of this thesis, and they are always like in the following notation.

Notation 2.32. Suppose R is a number ring of degree n, $\alpha, \beta \in R$ are coprime elements of R and q is a prime number, also coprime to α, β. Then, the kernel of the map $f_{\alpha,\beta} : R \times R \to R/q, (\gamma_1, \gamma_2) \mapsto \gamma_1\alpha - \gamma_2\beta \bmod q$ is denoted by $L^q_{\alpha,\beta}$.

Remark 2.33. The requirement that α and β are coprime is not necessary in this definition. However, this specific lattice will be used in Algorithm 9 to calculate the power residue symbol $\left(\frac{\alpha}{\beta}\right)_m$, in which one does require α and β to be coprime. In the following lemma the exactness at R/qR is proven with the coprimeness of α and β; in fact $R\alpha + R\beta + Rq = R$ is enough to prove this claim. ◀

Lemma 2.34. *The abelian group $L^q_{\alpha,\beta}$ is a q-ary square-dense lattice inside $\mathbb{R}^{2n}$. It fits inside the following exact sequence:*

$$0 \to L^q_{\alpha,\beta} \to R \times R \xrightarrow{f_{\alpha,\beta}} R/qR \to 0$$

Proof. Since the number field R has degree n over $\mathbb{Q}$, the lattice $R \times R$ is $2n$-dimensional, and so is $L^q_{\alpha,\beta}$. It is q-ary, because $q\mathbb{Z}^{2n} = qR \times qR \subseteq L^q_{\alpha,\beta}$, since q-multiples are mapped to zero under $f_{\alpha,\beta}$. Assuming that $L^q_{\alpha,\beta}$ is indeed well-posed in the exact sequence, one immediately sees that $(R \times R)/L^q_{\alpha,\beta} \simeq R/qR$, which has q^n elements. Therefore $L^q_{\alpha,\beta}$ has covolume q^n.

We still have to prove that the above sequence is exact. It is exact at R/qR, since $f_{\alpha,\beta}$ is clearly surjective, because α and β are coprime; one can find γ_1, γ_2 such that $\gamma_1\alpha + \gamma_2\beta = 1$. It is exact at $R \times R$ and $L^q_{\alpha,\beta}$ by definition. □

In the reduction Algorithm 9, the lattice $L^q_{\alpha,\beta}$, is given by a matrix, instead of given as a kernel of some map. The matrix is constructed with a fixed recipe, see Algorithm 7.

Algorithm 7: Constructs generating matrix of $L^q_{\alpha,\beta}$

```
GeneratingMatrix ($\alpha, \beta, q, B$);
Input  : Elements $\alpha, \beta \in R$ a number ring, a prime number $q$ and a
         basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ of the ring $R$.
Output: The matrix $N$ that generates the lattice $L^q_{\alpha,\beta}$ w.r.t. the basis $B$.
Construct the $n \times n$ multiplication matrices $M_\beta$ and $M_\alpha$ with respect to
  the given basis $B$ of $R$, see Definition 1.15 ;
Reduce every entry in $M_\beta$ and $M_\alpha$ modulo $q$ ;
Call these matrices $\bar{M}_\beta$ and $\bar{M}_\alpha$, respectively ;
Concatenate $\bar{M}_\beta$ and $\bar{M}_\alpha$ horizontally, yielding a $n \times 2n$ matrix ;
Concatenate the resulting matrix vertically above the matrix $qI_{2n}$ ;
Call the resulting matrix $N$ ;
return $N$ ;
```

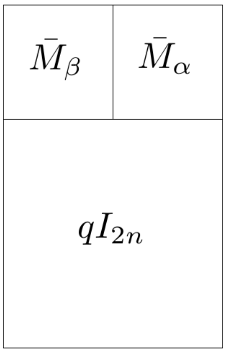


Figure 2.2: The construction of the matrix N, as in Algorithm 7

Lemma 2.35. *The rows of the matrix N as in Algorithm 7 generate the lattice $L^q_{\alpha,\beta}$.*

Proof. Let L_N be the lattice generated by the rows of N.

($L_N \subseteq L^q_{\alpha,\beta}$) It is enough to show that every row of N is inside $L^q_{\alpha,\beta}$. For the rows of qI_{2n} this is evident. The upper n rows of N consists of elements of the form $(\beta\cdot\mathbf{b}_i, \alpha\cdot\mathbf{b}_i)$, modulo q. Under $f_{\alpha,\beta}$ this sends to $\beta\cdot\mathbf{b}_i\cdot\alpha - \alpha\cdot\mathbf{b}_i\cdot\beta \equiv 0 \bmod q$. So, indeed this inclusion is right.

($L_N \supseteq L^q_{\alpha,\beta}$) Suppose $(\gamma_1, \gamma_2) \in L^q_{\alpha,\beta}$. Then, $\gamma_1\alpha - \gamma_2\beta \equiv 0 \bmod q$, i.e. $\gamma_1\alpha \equiv \gamma_2\beta \bmod q$. So, we have $\eta = \gamma_1\beta^{-1} \equiv \gamma_2\alpha^{-1}$ modulo q, where the inverses[2] are taken modulo q. Then we have

$$\beta\eta \equiv \gamma_1 \text{ and } \alpha\eta \equiv \gamma_2 \bmod q.$$

[2] Note that α, β are coprime to q.

Therefore, $(\gamma_1, \gamma_2) \equiv (\beta\eta, \alpha\eta)$ modulo q, so, $(\gamma_1, \gamma_2) \in L_N$.

□

2.6.3 Different inner products

The goal of finding a short vector inside the lattice $L^q_{\alpha,\beta}$, is obtaining $\gamma_1, \gamma_2 \in R$ such that

(1) γ_1 is 'small' (i.e. smaller than α and β);

(2) $\frac{\gamma_1\alpha - \gamma_2\beta}{q}$ is 'small' (idem).

One can use these elements to compute the power residue symbol $\left(\frac{\alpha}{\beta}\right)_m$ as in Algorithm 9. A natural choice for an norm on $L^q_{\alpha,\beta}$, is then a weighted norm:

Definition 2.36. The following norm on $L^q_{\alpha,\beta}$ is called the weighted norm (weighted by α, β and q).

$$\|(\gamma_1, \gamma_2)\|_{\alpha,\beta} := \sqrt{\left(\frac{\|\gamma_1\alpha + \gamma_2\beta\|_2}{q}\right)^2 + \|\gamma_1\|_2^2}.$$

Here $\|\cdot\|_2$ denotes the ℓ_2-norm on the vectors $\gamma_1\alpha + \gamma_2\beta$ and γ_1 with respect to a fixed basis B of R.

Definition 2.37. The following norm on $L^q_{\alpha,\beta}$ is called the unweighted norm.

$$\|(\gamma_1, \gamma_2)\|_2 = \sqrt{\|\gamma_1\|_2^2 + \|\gamma_2\|_2^2},$$

again with $\|\cdot\|_2$ the ℓ_2-norm.

Lemma 2.38. *For $L^q_{\alpha,\beta}$ together with the unweighted norm $\|\cdot\|_2$, we have:*

$$\Delta(L^q_{\alpha,\beta}) = q^n$$

Proof. This is just Lemma 2.34. □

Lemma 2.39. *For $L^q_{\alpha,\beta}$ together with the weighted norm $\|\cdot\|_{\alpha,\beta}$, we have:*

$$\Delta(L^q_{\alpha,\beta}) = N(\beta)$$

Proof. The norm $\|(\gamma_1, \gamma_2)\|_{\alpha,\beta} := \sqrt{\left(\frac{\|\gamma_1\alpha+\gamma_2\beta\|_2}{q}\right)^2 + \|\gamma_1\|_2^2}$ coincides with the following inner product matrix

$$Q = \frac{1}{q^2}\begin{bmatrix} M_\alpha^T M_\alpha + q^2 I_n & M_\beta^T M_\alpha \\ M_\alpha^T M_\beta & M_\beta^T M_\beta \end{bmatrix}.$$

Since

$$\begin{aligned}\|\gamma_1\alpha + \gamma_2\beta\|_2^2 &= (\gamma_1 M_\alpha + \gamma_2 M_\beta) \bullet (M_\beta^T \gamma_2^T + M_\alpha^T \gamma_1^T) \\ &= \begin{bmatrix}\gamma_1 & \gamma_2\end{bmatrix} \cdot \begin{bmatrix} M_\alpha \\ M_\beta \end{bmatrix} \begin{bmatrix} M_\alpha^T & M_\beta^T \end{bmatrix} \cdot \begin{bmatrix}\gamma_1^T \\ \gamma_2^T\end{bmatrix}\end{aligned}$$

$$= \begin{bmatrix} \gamma_1 & \gamma_2 \end{bmatrix} \cdot \begin{bmatrix} M_\alpha M_\alpha^T & M_\alpha M_\beta^T \\ M_\beta M_\alpha^T & M_\beta M_\beta^T \end{bmatrix} \cdot \begin{bmatrix} \gamma_1^T \\ \gamma_2^T \end{bmatrix}$$

we have

$$\left(\frac{\|\gamma_1\alpha + \gamma_2\beta\|_2}{q}\right)^2 + \|\gamma_1\|_2^2 = \begin{bmatrix} \gamma_1 & \gamma_2 \end{bmatrix} \cdot \frac{1}{q^2} \begin{bmatrix} M_\alpha M_\alpha^T + q^2 I & M_\alpha M_\beta^T \\ M_\beta M_\alpha^T & M_\beta M_\beta^T \end{bmatrix} \cdot \begin{bmatrix} \gamma_1^T \\ \gamma_2^T \end{bmatrix}.$$

The matrix determinant formula[3] $\det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(D) \cdot \det(A - BD^{-1}C)$ applies here:

$$\det(Q) = \frac{1}{q^{4n}} \det(M_\beta M_\beta^T) \det(M_\alpha M_\alpha^T + q^2 I_n - M_\alpha M_\beta^T \cdot (M_\beta^T)^{-1} M_\beta^{-1} \cdot M_\beta M_\alpha^T)$$

$$= \frac{1}{q^{4n}} \det(M_\beta^T M_\beta) \det(q^2 I_n) = \frac{1}{q^{2n}} \det(M_\beta M_\beta^T) = \frac{N(\beta)^2}{q^{2n}}$$

Denote L_0 for the lattice defined by the identity matrix, with the weighted norm. Then we have $\Delta(L_0) = \sqrt{\det Q} = \frac{N(\beta)}{q^n}$, by above calculations. Now, using the identity ([Cas97, p. 14, Lm. 1] [PZ89, §3.2, Lm. 3.6])

$$\Delta(L') = [L : L'] \cdot \Delta(L),$$

for sublattices $L' \subseteq L$, we obtain

$$\Delta(L^q_{\alpha,\beta}) = [L_0 : L^q_{\alpha,\beta}] \cdot \Delta(L_0) = q^n \cdot \frac{N(\beta)}{q^n} = N(\beta).$$

□

Remark 2.40. Lemma 2.39 shows us that we might expect that using LLL with the weighted norm does not give us much shorter vectors than ordinary two-sided reduction, as is discussed in subsection 4.3.2. I use the unweighted norm in the reduction algorithm. The disadvantage of using the unweighted norm is that one has no control over the size of $\gamma_1\alpha + \gamma_2\beta$, even if γ_1 and γ_2 are small. However, heuristically, one might assume – at least in cyclotomic fields $\mathbb{Q}(\zeta_m)$ with not 'too composite' m – that the product of two small elements in the Euclidean norm, will have a small Euclidean norm too. This assumption is not a very plausible one. ◀

Remark 2.41. One could also use the 'canonical norm' on K. Denoting $K_\mathbb{R} := K \otimes_\mathbb{Z} \mathbb{R}$, one defines the canonical bilinear form as follows.

$$b(x, y) := \sum_{\substack{\sigma : K_\mathbb{R} \to \mathbb{C} \\ \mathbb{R}\text{-algebra homomorphisms}}} \sigma(x)\overline{\sigma(y)}.$$

The main advantage of this form is that it is independent of choice of basis, and that it is submultiplicative, i.e. $\|xy\|_b \leq \|x\|_b \|y\|_b$, easily proven by comparing sums. On the other hand, in a computational context one often chooses a basis and defines a norm with respect to that basis. ◀

[3] This formula is obtained from [Mat].

Remark 2.42. What is the minimum length of a vector we can obtain in $L^q_{\alpha,\beta}$, only using easy, manual reductions? As each vector in $L^q_{\alpha,\beta}$ has entries of absolute value below $\frac{q-1}{2}$, we have an 'obvious' bound

$$\|\gamma_1\alpha+\gamma_2\beta\|_2 = \left\| \begin{bmatrix}\gamma_1 & \gamma_2\end{bmatrix} \begin{bmatrix} M_\alpha \\ M_\beta \end{bmatrix} \right\|_2 \leq \| \begin{bmatrix}\gamma_1 & \gamma_2\end{bmatrix} \|_2 \left\| \begin{bmatrix} M_\alpha \\ M_\beta \end{bmatrix} \right\|_2 \leq \frac{q\sqrt{2n}}{2} \left\| \begin{bmatrix} M_\alpha \\ M_\beta \end{bmatrix} \right\|_2 .$$

Where $\|M\|_2$ denotes the matrix 2-norm. So, we only have to calculate the matrix-norm of $M = \begin{bmatrix} M_\alpha \\ M_\beta \end{bmatrix}$, and since

$$\begin{bmatrix} M_\alpha \\ M_\beta \end{bmatrix} = \begin{bmatrix} M_\alpha & 0 \\ 0 & M_\beta \end{bmatrix} \cdot \begin{bmatrix} I_n \\ I_n \end{bmatrix},$$

we have, by the submultiplicative property of the matrix norm,

$$\left\| \begin{bmatrix} M_\alpha \\ M_\beta \end{bmatrix} \right\|_2 \leq \left\| \begin{bmatrix} M_\alpha & 0 \\ 0 & M_\beta \end{bmatrix} \right\|_2 \cdot \left\| \begin{bmatrix} I_n \\ I_n \end{bmatrix} \right\|_2 = 2 \cdot \max_\sigma(|\sigma(\alpha)|_{\mathbb{C}}, |\sigma(\beta)|_{\mathbb{C}}),$$

with σ the $\mathbb{R}$-algebra homomorphisms from $K_\mathbb{R} \to \mathbb{C}$ for the extension $K : \mathbb{Q}$. Above inequality holds because the eigenvalues of M_α and M_β respectively, are α respectively β, embedded in the complex plane by different $\mathbb{R}$-algebra homomorphisms $\sigma : K_\mathbb{R} \to \mathbb{C}$. And clearly, the singular values of the matrix $\begin{bmatrix} I_n \\ I_n \end{bmatrix}$ are $\sigma = 0, 2$. Since the matrix norm is equal to the square root of the maximum complex norm of the singular values, we can conclude that the above bound is sound.

All together, this implies that one can find – only using straightforward reductions – a vector of the following length:

$$\|\gamma_1\alpha + \gamma_2\beta\|_2 \leq \max_\sigma(|\sigma(\alpha)|_{\mathbb{C}}, |\sigma(\beta)|_{\mathbb{C}}) \cdot \sqrt{2n} \cdot q$$

Therefore we have:

$$\frac{\|\gamma_1\alpha + \gamma_2\beta\|_2}{q} + \|\gamma_1\|_2 \leq \sqrt{2n} \left(\max_\sigma (|\sigma(\alpha)|, |\sigma(\beta)|) + \frac{q}{2} \right)$$

◀

Remark 2.43. In practice, for example in my 'largest' number field $\mathbb{Q}(\zeta_{91})$ with composite number m, the largest value observed for

$$\frac{\log \| \frac{\gamma_1\alpha+\gamma_2\beta}{q} \|_2 \log \|\gamma_1\|_2}{\log \|\alpha\|_2 \log \|\beta\|_2}$$

(after applying a loop of Algorithm 9) equals 0.85. Such values mostly occur when α and β are already quite small – i.e. for large α and β, reduction is much better. ◀

CHAPTER 3

Power residue symbols and Hilbert symbols

3.1 Introduction

In this thesis, a supposedly effective algorithm computing the power residue symbol is proposed. In the next section I define the power residue symbol and derive some of its properties. Also, in a later section, I point out which of these properties are utilized in specific parts of the main algorithm.

One of these properties is reciprocity, which uses the Hilbert symbol. This symbol will be defined in this chapter too, and in the final section I will treat Bouw's algorithm, which computes Hilbert symbols in polynomial time.

3.2 Power residue symbols

3.2.1 Definition

Notation 3.1. In this section, K is a number field containing a primitive m-th root of unity $\zeta_m \in K$, and $\mathfrak{p}$ is a prime ideal of $\mathcal{O}_K$, coprime to m. Also, we denote $\mu_m = \langle \zeta_m \rangle$, for the group of m-th roots of unity in K.

The following definitions can be found in [Lem00, p. 111].

Definition 3.2 (Power residue symbols above prime ideals)**.** Let $K, m, \mathfrak{p}, \zeta_m$ and μ_m be as in Notation 3.1, and let $\alpha \in \mathcal{O}_K$ be coprime[1] with $\mathfrak{p}$. Then we define $\left(\frac{\alpha}{\mathfrak{p}}\right)_m \in \mu_m$ to be the unique m-th root of unity that satisfies

$$\left(\frac{\alpha}{\mathfrak{p}}\right)_m \equiv \alpha^{\frac{N(\mathfrak{p})-1}{m}} \bmod \mathfrak{p}.$$

Since m is coprime to $\mathfrak{p}$, we have that $(\mathcal{O}_K/\mathfrak{p})^*$ is a cyclic group of order $N(\mathfrak{p}) - 1$ that contains the subgroup $\bar{\mu}_m = \langle \zeta_m \bmod \mathfrak{p} \rangle$ of order m. This di-

[1] I.e. $\alpha \notin \mathfrak{p}$ in this case.

rectly implies that $\alpha^{\frac{N(\mathfrak{p})-1}{m}} \bmod \mathfrak{p} \in \bar{\mu}_m$, making the power residue symbol well defined.

Remark 3.3. As in fractions, we refer to the upper part and lower part of the power residue symbol as the numerator and denominator, respectively. In the case that $\mathfrak{p}$ is not coprime to n or coprime to α, the symbol is undefined. In some texts the power residue symbol is given the value $\left(\frac{\alpha}{\mathfrak{p}}\right)_m = 0$ when $\alpha \in \mathfrak{p}$. Not in this thesis. ◀

General power residue symbols – i.e., above any ideal – are just multiplicative continuations of Definition 3.2. Since the ring of integers of any number field is a Dedekind ring (see Lemma 1.23), every ideal can be decomposed uniquely in a product of prime ideals. I.e., for every (fractional) ideal $\mathfrak{b}$ of $\mathcal{O}_K$, we have (see equation (1.1) in Remark 1.28):

$$\mathfrak{b} = \prod_{\mathfrak{p}|\mathfrak{b}} \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{b})}.$$

Definition 3.4 (Power residue symbol)**.** Let K be as in Notation 3.1, let $\mathfrak{b}$ an ideal of $\mathcal{O}_K$ coprime to m and let $\alpha \in \mathcal{O}_K$ be an element coprime to $\mathfrak{b}$. We define

$$\left(\frac{\alpha}{\mathfrak{b}}\right)_m := \prod_{\mathfrak{p}|\mathfrak{b}} \left(\frac{\alpha}{\mathfrak{p}}\right)_m^{v_{\mathfrak{p}}(\mathfrak{b})}.$$

Remark 3.5. In this chapter, we will tacitly assume that α, the element in the numerator of the power residue symbol, is always coprime to $\mathfrak{b}$, the ideal in the denominator. ◀

Remark 3.6. The power residue symbol can also be defined by the Artin map of a certain Kummer-extension [CF67, p. 73], [Koc97, Ch. 2,§1.8]. Suppose $\alpha \in K$ with K a number field. Then $K(\sqrt[m]{\alpha}) : K$ is a so-called Kummer extension, and the Artin map $\left(\frac{\cdot}{K(\sqrt[m]{\alpha})/K}\right) : \mathcal{I}_K^{\mathfrak{m}} \to \mathrm{Gal}(K(\sqrt[m]{\alpha})/K)$ defines a group homomorphism from a certain subgroup of the group of fractional ideals (see Definition 1.25) to the Galois group of the extension $K(\sqrt[m]{\alpha}) : K$.

Since $\zeta_m \in K$, every element in this Galois group sends $\sqrt[m]{\alpha}$ to $\zeta_m^i \cdot \sqrt[m]{\alpha}$, for some $i \in \mathbb{Z}/m\mathbb{Z}$. The power residue symbol is then defined by the following identity.

$$\left(\frac{\mathfrak{b}}{K(\sqrt[m]{\alpha})/K}\right)(\sqrt[m]{\alpha}) = \left(\frac{\alpha}{\mathfrak{b}}\right)_m \cdot \sqrt[m]{\alpha}.$$

Definition 3.4 is chosen as main definition of the power residue symbol in this thesis, because it is more down-to-earth and it fits better in the computational context of this thesis. ◀

Notation 3.7 (Principal power residue symbol)**.** If $\mathfrak{b} = (\beta)$ in Definition 3.4 is a principal ideal, we just denote the symbol $\left(\frac{\alpha}{(\beta)}\right)_m$ by $\left(\frac{\alpha}{\beta}\right)_m$. In this thesis this particular symbol will be called the principal power residue symbol.

Notation 3.8. In some cases there might be confusion about the 'ground field' of the power residue symbol. In that case we write the field as subscript: $\left(\frac{\alpha}{\mathfrak{b}}\right)_{m,K}$.

Example 3.9. A hopefully instructive example for Notation 3.8. Suppose K is as in Notation 3.1, and $L : K$ is a finite degree Galois extension of K. If $\alpha, \beta \in K$, then also $\alpha, \beta \in L$. Then

$$\left(\frac{\alpha}{\beta}\right)_{m,K} = \prod_{\mathfrak{p} \mid \beta \mathcal{O}_K} \left(\frac{\alpha}{\mathfrak{p}}\right)_{m,K}^{v_{\mathfrak{p}}(\beta)};$$

but

$$\left(\frac{\alpha}{\beta}\right)_{m,L} = \prod_{\mathfrak{P} \mid \beta \mathcal{O}_L} \left(\frac{\alpha}{\mathfrak{P}}\right)_{m,L}^{v_{\mathfrak{P}}(\beta)}.$$

Generally, those symbols are not equal. ◀

Lemma 3.10. *For the power residue symbol in K above a prime ideal $\mathfrak{p}$ coprime to m, we have the following results.*

(a) $\left(\frac{\alpha}{\mathfrak{p}}\right)_m = \left(\frac{\alpha'}{\mathfrak{p}}\right)_m$ *if $\alpha \equiv \alpha'$ mod $\mathfrak{p}$, for all $\alpha, \alpha' \in \mathcal{O}_K$;*

(b) $\left(\frac{\alpha}{\mathfrak{p}}\right)_m = 1 \iff \alpha \equiv \eta^m$ *mod $\mathfrak{p}$, for some $\eta \in \mathcal{O}_K$.*

$$\begin{tikzcd} & K \arrow[d, dash] \arrow[dd, dash, dashed, bend left=60, "G"] \\ \zeta_m \in & k \arrow[d, dash] \\ & F \end{tikzcd}$$

Suppose we have a tower of fields $F \subseteq k \subseteq K$, with $\zeta_m \in k$, as in the picture left. Then we have m-th power residue symbols with respect to ground fields K and k. If $K : F$ is a Galois extension, its Galois group $G = \mathrm{Gal}(K : F)$ acts naturally on the power residue symbol. The next lemma shows how.

Lemma 3.11. *Let $F \subseteq k \subseteq K$ be a tower of fields with $\zeta_m \in k$ and let $K : F$ be a Galois extension with Galois group G. Let $\mathfrak{b}$ an ideal of $\mathcal{O}_K$ coprime to m. Then we have:*

(a) (Galois action) $\left(\frac{\sigma(\alpha)}{\sigma(\mathfrak{b})}\right)_m = \sigma\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ *for every $\sigma \in G$ and $\alpha \in \mathcal{O}_K$;*

(b) (Inertia free) For $\alpha \in \mathcal{O}_k$ and $\mathfrak{p}$ a prime ideal in $\mathcal{O}_k$ with inertia degree 1, then: $\left(\frac{\alpha}{\mathfrak{p}}\right)_{m,k} = \left(\frac{\alpha}{\mathfrak{P}}\right)_{m,K}$*, for any prime ideal $\mathfrak{P}$ in K above $\mathfrak{p}$;*

(c) (Norm) Suppose $K : k$ is abelian. For $\mathfrak{p}$ a prime ideal in $\mathcal{O}_k$, and $\alpha \in \mathcal{O}_K$ we have: $\left(\frac{\alpha}{\mathcal{O}_K \mathfrak{p}}\right)_{m,K} = \left(\frac{N_{K/k}(\alpha)}{\mathfrak{p}}\right)_{m,k}$.

Proof. The proofs of Lemma 3.10 and Lemma 3.11 can be found in [Lem00, p. 112-113]. □

3.2.2 Power residue symbols in number rings

In the case that one does not have the ring of integers, or does not even know whether or not the ring R is equal to the ring of integers in K, it is still possible to compute the power residue symbol in the sense of Definition 3.2 and Definition 3.4.

Lemma 3.12. *Suppose R is a number ring in K, and let $\mathfrak{p}$ be a regular prime ideal in R, coprime to m. Furthermore, let $\mathfrak{p}'$ be the prime ideal of $\mathcal{O}_K$ such that $\mathfrak{p}' \cap R = \mathfrak{p}$, as in Lemma 1.44. Then we have for every $\alpha \in K^*$*

$$\alpha^{\frac{N(\mathfrak{p})-1}{m}} \equiv \zeta_m^i \textit{ mod } \mathfrak{p} \iff \alpha^{\frac{N(\mathfrak{p}')-1}{m}} \equiv \zeta_m^i \textit{ mod } \mathfrak{p}'. \tag{3.1}$$

Proof. Since $\alpha \in K^*$ can be written as a fraction of two elements in R, we may assume that $\alpha \in R \backslash \{0\}$. Remark $N(\mathfrak{p}) = N(\mathfrak{p}') = p^f$, since the residue class degrees are equal by Lemma 1.46. Now, by the same Lemma 1.46, the inclusion $R \subseteq \mathcal{O}_K$ induces an isomorphism $R/\mathfrak{p} \to \mathcal{O}_K/\mathfrak{p}'$. Since the m-th roots of unity map injectively in $\mathcal{O}_K/\mathfrak{p}'$, by Lemma 1.78, one must have that (3.1) holds. □

Definition 3.13. Suppose $\alpha \in R$ and $\mathfrak{p}$ is a regular prime ideal of R, coprime to m. Then we define $\left(\frac{\alpha}{\mathfrak{p}}\right)_{m,R} \in \mu_m$ to be the unique m-th root of unity that satisfies

$$\left(\frac{\alpha}{\mathfrak{p}}\right)_{m,R} \equiv \alpha^{\frac{N(\mathfrak{p})-1}{m}} \bmod \mathfrak{p}.$$

Remark 3.14. Lemma 3.12 ensures that, when $\mathfrak{p}$ is a regular prime ideal of R, we have $\left(\frac{\alpha}{\mathfrak{p}}\right)_{m,R} = \left(\frac{\alpha}{\mathfrak{p}'}\right)_m$, where $\mathfrak{p}'$ is the prime ideal in $\mathcal{O}_K$ such that $\mathfrak{p}' \cap R = \mathfrak{p}$. Also note that, when $R = \mathcal{O}_K$, one has $\left(\frac{\alpha}{\mathfrak{p}}\right)_{m,R} = \left(\frac{\alpha}{\mathfrak{p}'}\right)_m$. ◀

Definition 3.15. Suppose $\alpha \in R$ and $\mathfrak{b}$ is an ideal in R coprime to all singular primes of R and coprime to m. Then we define

$$\left(\frac{\alpha}{\mathfrak{b}}\right)_{m,R} := \prod_{\mathfrak{p} | \mathfrak{b}} \left(\frac{\alpha}{\mathfrak{p}}\right)_{m,R}^{v_{\mathfrak{p}}(\mathfrak{b})}.$$

Remark 3.16. Definition 3.15 is justified by the fact that ideals of R that are coprime to all singular primes decompose uniquely into regular prime ideals, as in Lemma 1.50. ◀

Lemma 3.17. *Suppose R is a number ring and $\mathfrak{b}$ is an ideal coprime to all singular primes of R, and coprime to m. Then we have*

$$\left(\frac{\alpha}{\mathfrak{b}}\right)_{m,R} = \left(\frac{\alpha}{\mathcal{O}_K \mathfrak{b}}\right)_m. \tag{3.2}$$

Proof. Suppose that $\mathfrak{p}$ is a regular prime ideal of R, coprime to m. Then, by Remark 1.45, we can take $\mathfrak{p}' = \mathfrak{p}\mathcal{O}_K$ for the prime ideal $\mathfrak{p}'$ in Lemma 1.44. Since, by Lemma 3.12 and Remark 3.14, we have $\left(\frac{\alpha}{\mathfrak{p}}\right)_{m,R} = \left(\frac{\alpha}{\mathfrak{p}'}\right)_m$, we only have to prove the following claim. If $\mathfrak{b} = \prod_{\mathfrak{p}} \mathfrak{p}^{n_{\mathfrak{p}}}$ is the prime factorization of $\mathfrak{b}$ in R, as in Lemma 1.50, then we have the following prime ideal factorization of $\mathcal{O}_K \mathfrak{b}$ in $\mathcal{O}_K$:

$$\mathfrak{b}\mathcal{O}_K = \prod_{\mathfrak{p}} (\mathfrak{p}\mathcal{O}_K)^{n_{\mathfrak{p}}} \text{ (where } \mathfrak{p} \text{ ranges over the prime ideals in } R).$$

Since, by Remark 1.45, every ideal $(\mathfrak{p}\mathcal{O}_K)$ in $\mathcal{O}_K$ is a prime ideal, indeed, above factorization is the prime ideal factorization of $\mathfrak{b}\mathcal{O}_K$ in $\mathcal{O}_K$. This proves that identity (3.2) is valid. □

3.3 Hilbert symbols

In this section, F is finite extension of $\mathbb{Q}_p$ that contains a primitive m-th root of unity. We denote by $e = e(F/\mathbb{Q}_p)$ the ramification index and by $f = f(F/\mathbb{Q}_p)$ the residue class degree and by $\mathcal{O}$ the ring of integers of F with maximal ideal $\mathfrak{m}$. Let $v : F \to \mathbb{Z}$ be the valuation on F. Also, we denote the residue field by $\mathbb{F} = \mathcal{O}/\mathfrak{m}$.

Definition 3.18 (m-th norm residue symbol or Hilbert symbol)**.** The m-th norm residue symbol is defined as follows, using the Artin map $\psi_F : F^* \to \mathrm{Gal}(F^{ab}/F)$:

$$\left(\frac{x,y}{\mathfrak{m}}\right)_m := \frac{\psi_F(x)(\sqrt[m]{y})}{\sqrt[m]{y}} \quad \text{for } x, y \in F^*.$$

Notation 3.19 (Alternative notation)**.** When $F = K_\mathfrak{p}$ is explicitly given as a completion of a number field K with respect to a prime $\mathfrak{p}$, the Hilbert symbol is denoted $\left(\frac{\cdot,\cdot}{\mathfrak{p}}\right)_m$ (with $\mathfrak{p}$ replacing $\mathfrak{m}$).

Remark 3.20. Definition 3.18 has much in common with the definition of the power residue symbol as in Remark 3.6, and can be seen as a 'local' version of it. Since all ideals are principal in local fields, the ideal group is replaced by the field F. ◀

Lemma 3.21. *The Hilbert symbol of F is a map*

$$\left(\frac{\cdot,\cdot}{\mathfrak{m}}\right)_m : F^*/(F^*)^m \times F^*/(F^*)^m \to \mu_m$$

that has the following properties, for every $\alpha, \alpha', \beta, \beta' \in F^*$.

(a) (Multiplicatively bilinear) $\left(\frac{\alpha\alpha',\beta}{\mathfrak{m}}\right)_m = \left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_m \left(\frac{\alpha',\beta}{\mathfrak{m}}\right)_m$ and $\left(\frac{\alpha,\beta\beta'}{\mathfrak{m}}\right)_m = \left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_m \left(\frac{\alpha,\beta'}{\mathfrak{m}}\right)_m$;

(b) (Anti-symmetry) $\left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_m = \left(\frac{\beta,\alpha}{\mathfrak{m}}\right)_m^{-1}$;

(c) (Symbol properties) $\left(\frac{\alpha,-\alpha}{\mathfrak{m}}\right)_m = 1$ and $\left(\frac{\alpha,1-\alpha}{\mathfrak{m}}\right)_m = 1$ for $\alpha \in F^*\backslash\{1\}$;

(d) (Non-degenerate) If $\left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_m = 1$ for all $\beta \in F^*$, then $\alpha \in (F^*)^m$;

(e) (Norm residue) $\left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_m = 1 \iff \alpha \in N_F^{F(\sqrt[m]{\beta})}(F(\sqrt[n]{\beta})^*)$.

Writing $m = p^k r$, with $p \nmid r$ and p the characteristic of the residue field[2], one often distinguishes the symbols $\left(\frac{\cdot,\cdot}{\mathfrak{m}}\right)_r$ and $\left(\frac{\cdot,\cdot}{\mathfrak{m}}\right)_{p^k}$, referring to them as the *tame* Hilbert symbol and the *wild* Hilbert symbol, respectively, see Definition 1.77. According to Lemma 1.78, we have $r \mid p^f - 1$. The tame Hilbert symbol can be computed using a nice and short formula.

[2] Note that F is an extension of the p-adic field $\mathbb{Q}_p$

Lemma 3.22 (Tame Hilbert symbol)**.** *For $\alpha, \beta \in F$, write $a = v(\alpha)$ and $b = v(\beta)$. Then one has:*

$$\left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_r = \omega\left((-1)^{ab}\frac{\beta^a}{\alpha^b} \bmod \mathfrak{m}\right)^{\frac{p^f-1}{r}}. \tag{3.3}$$

where $\omega : \mathbb{F}_F \to \mu_{p^f-1}$ is the Teichmüller map as in Definition 1.76.

No similar easy-to-calculate formula is known for the wild symbols, although there has been much research into finding one [Sha50], [Vos79], [Iwa68], [AH28]. A proof of Lemma 3.22 can be found in [Neu99, p. 336]. The algorithm of Bouw, which we will treat shortly, calculates this wild symbol effectively [Bou16]. In essence, Bouw's algorithm computes a symbol isomorphic to the Hilbert symbol, and then 'calibrates' it, using the following formula, obtained from [Mil13, Ch. 1, Ex. 3.13].

Notation 3.23. Suppose $\alpha \in F$. We define α_* (for Lemma 3.24 only) by the following equation:

$$\alpha_* p^t = N_{F/\mathbb{Q}_p}(\alpha) \text{ with } \alpha_* \in \mathbb{Z}_p, t \in \mathbb{Z} \text{ and } v_p(\alpha_*) = 0.$$

In words, α_* is the p-free part of $N_{F/\mathbb{Q}_p}(\alpha)$.

Lemma 3.24 (Calibration formula for the wild symbol)**.** *Let m be maximal such that $\zeta_{p^m} \in F$, then we have, for each $1 \leq n \leq m$:*

$$\left(\frac{\alpha,\zeta_{p^m}}{\mathfrak{m}}\right)_{p^n} = \zeta_{p^m}^{\frac{\alpha_*^{-1}-1}{p^n}}$$

In practice, one computes the wild and tame Hilbert symbol separately, to combine them afterwards in the following manner.

Lemma 3.25. *Suppose $m = p^k r$. Write $1 = a_p \cdot p^k + a_r \cdot r$, with the Euclidean algorithm. Then we have*

$$\left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_m = \left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_r^{a_p}\left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_{p^k}^{a_r}.$$

Proof.

$$\left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_r^{a_p}\left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_{p^k}^{a_r} = \left(\frac{\psi_F(\alpha)(\beta^{\frac{p^k}{m}})}{\beta^{\frac{p^k}{m}}}\right)^{a_p}\left(\frac{\psi_F(\alpha)(\beta^{\frac{r}{m}})}{\beta^{\frac{r}{m}}}\right)^{a_r}$$

$$= \left(\frac{\psi_F(\alpha)(\beta^{\frac{a_r r + a_p p^k}{m}})}{\beta^{\frac{a_r r + a_p p^k}{m}}}\right) = \frac{\psi_F(\alpha)(\sqrt[m]{\beta})}{\sqrt[m]{\beta}} = \left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_m.$$

□

Lemma 3.26 (Product formula)**.** *Let K be a number field, containing μ_m, the m-th roots of unity. Then, for every $\alpha, \beta \in K$, one has*

$$\prod_{\mathfrak{p}}\left(\frac{\alpha,\beta}{\mathfrak{p}}\right)_m = 1. \tag{3.4}$$

Proof. See, for example, [Neu99, Thm. VI.8.1]. For the notation, see Notation 3.19. □

Remark 3.27. In formula (3.4), α and $\beta \in K$ are seen as elements inside $K_\mathfrak{p}$ via the inclusion $K \hookrightarrow K_\mathfrak{p}$. The formula is in fact a finite product, since in the tame case – that is, when $\mathfrak{p} \nmid m$ – we have the tame formula from Lemma 3.22. This tame formula vanishes when when $\mathfrak{p} \nmid \alpha$ and $\mathfrak{p} \nmid \beta$. So, one can rephrase Lemma 3.26 as follows.

$$\prod_{\mathfrak{p} \mid \alpha\beta m} \left(\frac{\alpha, \beta}{\mathfrak{p}}\right)_m = 1. \tag{3.5}$$

◀

Remark 3.28. When one has an implementation of Bouw's algorithm (as in [Bou16] or section 3.5), one can use Formula (3.5) as a 'sanity check'. This check can be performed as follows. Choose $\alpha, \beta \in K$, relatively small, and factor them into prime ideals. Now, compute the tame symbols with the tame formula (3.3) as in Lemma 3.22, and the wild Hilbert symbols (i.e., when $\mathfrak{p} \mid m$) with Bouw's algorithm, and take the following product:

$$\prod_{\mathfrak{p} \mid \alpha\beta \text{ and } \mathfrak{p} \nmid m} \left(\frac{\alpha, \beta}{\mathfrak{p}}\right)_m \cdot \prod_{\mathfrak{p} \mid m} \left(\frac{\alpha, \beta}{\mathfrak{p}}\right)_m .$$

This should equal one for every $\alpha, \beta \in K$.

Note that all Hilbert symbols in the product (3.5) must be expressed in the same mth root of unity. To avoid problems, I took – in my own implementation – a fixed $\zeta_m \in K$ beforehand, and expressed all Hilbert symbols as a power of this ζ_m:

$$\left(\frac{\alpha, \beta}{\mathfrak{p}}\right)_m = \zeta_m^{i_\mathfrak{p}}.$$

The product formula is then equivalent to: $\sum_{\mathfrak{p} \mid \alpha\beta m} i_\mathfrak{p} \equiv 0 \bmod m$. ◀

3.4 Exploitable properties of power residue symbols

According to Bouw's algorithm [Bou16], Hilbert symbols can be computed in polynomial time. This leads to the first exploitable property of the power residue symbol: reciprocity.

Property 3.29 (Reciprocity). *For $\alpha, \beta \in K$, we have:*

$$\left(\frac{\alpha}{\beta}\right)_m \left(\frac{\beta}{\alpha}\right)_m^{-1} = \prod_{\mathfrak{p} \mid m\infty} \left(\frac{\alpha, \beta}{\mathfrak{p}}\right)_m \tag{3.6}$$

Proof. A proof of the above reciprocity law can be found in [Neu99, p. VI.8.3] or [Koc97, Ch. 2, Thm. 2.16]. □

Notation 3.30. In this thesis, the right hand side of (3.6) will be denoted by $U(\alpha, \beta)$, the *Umkehrfaktor* (German for inversion factor).

Remark 3.31. An immediate corollary of Bouw's algorithm is that the Umkehrfaktor can be computed in polynomial time, meaning that the reciprocity law can be used eminently in an algorithm that computes the principal power residue symbol. The heuristic reduction Algorithm 9 in this thesis uses reciprocity extensively. ◀

Property 3.32 (Translation-invariance)**.** *For $\alpha \in K$ and $\mathfrak{b}$ an ideal of $\mathcal{O}_K$, we have:*

$$\left(\frac{\alpha}{\mathfrak{b}}\right)_m = \left(\frac{\alpha+\beta}{\mathfrak{b}}\right)_m \quad \textit{for every } \beta \in \mathfrak{b}. \tag{3.7}$$

Example 3.33. Note that only the above two properties are used when $m = 2$ and $K = \mathbb{Q}$. In that case, $\left(\frac{\cdot}{\cdot}\right)_m$ is just the Jacobi symbol $\left(\frac{\cdot}{\cdot}\right)$, and then there is an easy formula for $U(a, b)$ [IR90]. This Jacobi symbol can be calculated in polynomial time using the Euclidean algorithm in $\mathbb{Z}$.

For $a, b \in \mathbb{Z}$, the symbol $\left(\frac{a}{b}\right)$ is computed by reducing a mod b, yielding an a' with $|a'| < |b|/2$. After using reciprocity the computation of $\left(\frac{a}{b}\right)$ comes down to the calculation of $\left(\frac{b}{a'}\right)$. Then, reducing b mod a', we have b' with $|b'| < |a'|/2$. Repeating these operations leads to a rapid decrease of the size of the input of the Jacobi symbol – eventually yielding the computation of the symbol. ◀

Unfortunately, many number fields do not have a Euclidean algorithm[3]. Also, straightforwardly reducing the numerator by the denominator and using reciprocity is not feasible, because that depends on finding (relatively) short vectors in the translated lattice $\alpha + \mathfrak{b} \subseteq \mathbb{R} \otimes K$. This is believed to be hard[4], and variations of this problem (LWE, SIS[5]) are used for post-quantum cryptosystems [MR08, §1.2].

Using LLL lattice reduction for finding short vectors solves this problem only partially, still leaving an 'exponential gap' (see Remark B.3). The algorithm of Squirrel basically consists of exploiting the above two properties, and finding short vectors with LLL [Squ]. Squirrel overcomes the 'exponential gap' with expensive precomputations ([Squ, §V.3], in the form of large tables), which are the main cause of the fact that his algorithm does not run in polynomial time for varying m. Also in practice these tables are, even for small number fields as $\mathbb{Q}(\zeta_{11})$, too large to compute in reasonable time, see section 4.2.

Property 3.34 (Primes)**.** *For $\alpha \in K$, and $\mathfrak{p}$ a prime ideal of $\mathcal{O}_K$ (coprime to m), we have:*

$$\left(\frac{\alpha}{\mathfrak{p}}\right)_m \equiv \alpha^{\frac{N(\mathfrak{p})-1}{m}} \textit{ mod } \mathfrak{p}. \tag{3.8}$$

The above property, together with the fact that prime numbers can be (probabilistic) effectively factorized in number fields [Coh93, §4.8, §6.2], plays an important role in the reduction algorithm (Algorithm 9) of this thesis. The use

[3] Euclidean number fields seem to occur quite rarely, especially those with large degree – this is suggested by the effort that has to be made to find them [Len77]. Also, Euclidean number fields have a trivial class group. The heuristics of Lenstra-Cohen [CL84], suggest that there are many number fields with non-trivial class group.

[4] Finding short vectors in some well-chosen lattices is hard [Ajt98], but [LPR10] and [SS11] suggest that it is hard in ideal lattices too. However, these ideal lattices do not live in number fields, and hardness is proven for LWE, which is slightly more difficult than finding relatively short vectors.

[5] These are abbreviations of Learning With Errors and Short Integer Solution, respectively.

of prime numbers gives rise to q-ary lattices, in which finding relatively short vectors is hopefully easier than in arbitrary ideal lattices. Also, the reduction algorithm uses so-called *two-sided reduction* (see subsection 4.3.2), a technique proposed by [Len15].

The remaining properties are *not* used in reduction Algorithm 9, although I really tried to find a way to exploit them in that algorithm. However, these properties *are* used in the evaluation algorithm (Algorithm 10 on page 61), which is a probabilistic algorithm. The properties are obtained from [Koc97, Th. 2.13, p. 100], and in all claims it is assumed that the numerator and denominator do not have a divisor in common.

Property 3.35 (Multiplicativity). *For $\alpha, \beta \in K$ and $\mathfrak{b}$ an ideal of $\mathcal{O}_K$, we have:*

$$\left(\frac{\alpha\beta}{\mathfrak{b}}\right)_m = \left(\frac{\alpha}{\mathfrak{b}}\right)_m \cdot \left(\frac{\beta}{\mathfrak{b}}\right)_m \tag{3.9}$$

Property 3.36 (Multiplicativity (2)). *For $\alpha \in K$, and $\mathfrak{b}, \mathfrak{c}$ ideals of $\mathcal{O}_K$, we have:*

$$\left(\frac{\alpha}{\mathfrak{b}\mathfrak{c}}\right)_m = \left(\frac{\alpha}{\mathfrak{b}}\right)_m \cdot \left(\frac{\alpha}{\mathfrak{c}}\right)_m \tag{3.10}$$

Property 3.37 (m-th residue). *For $\alpha, \gamma \in K$, and $\mathfrak{b}$ an ideal of $\mathcal{O}_K$, we have:*

$$\left(\frac{\alpha\gamma^m}{\mathfrak{b}}\right)_m = \left(\frac{\alpha}{\mathfrak{b}}\right)_m \tag{3.11}$$

The three properties above are multiplicative in essence, implying that it requires other computational techniques than lattice reduction, which is mainly additive. The evaluation algorithm (Algorithm 10) exploits these multiplicative properties, in particular Property 3.37 and 3.34.

3.5 Bouw's algorithm

3.5.1 Introduction

Let F be a finite extension of $\mathbb{Q}_p$, containing a m-th primitive root of unity ζ_m. Then, the m-th norm residue symbol, also called the m-th Hilbert symbol, can be defined using the Artin map $\psi_F : F^* \to \mathrm{Gal}(F^{ab}/F)$:

$$\left(\frac{x, y}{\mathfrak{m}}\right)_m := \frac{\psi_F(x)(\sqrt[m]{y})}{\sqrt[m]{y}}.$$

For a treatment of local class field theory and the Artin map, see [Mil13]. In his article [Dab01], Daberkow gives an algorithm to compute this symbol. But this algorithm does not run in polynomial time, generally [Bou16, §10.1]. Bouw proposes an algorithm in his PhD thesis, computing the Hilbert symbols in polynomial time [Bou16]. The following summary of Bouw's algorithm is written with help of a very nice outline from Michiel Kosters [Kos14].

We denote by F a finite extension of $\mathbb{Q}_p$, by $\mathcal{O}$ the ring of integers of F with maximal ideal $\mathfrak{m}$. We denote the ramification index by $e = e(F/\mathbb{Q}_p)$ and the residue field degree by $f = f(F/\mathbb{Q}_p)$. Let $v : F \to \mathbb{Z}$ be the valuation on F. Also, we write $\mathbb{F} = \mathcal{O}/\mathfrak{m}$. We choose an uniformizer $\pi \in \mathcal{O}$, i.e. an element such that $(\pi) = \mathfrak{m}$. Also, we choose $\gamma \in \mathcal{O}$ such that every element in $\mathbb{F}$ can be uniquely written[6] as $\sum_{i=0}^{f-1} c_i\gamma^i \bmod \mathfrak{m}$, with $c_i \in \{0, \dots, p-1\}$.

We will use the notation $U_i = \{u \in F \mid v(u-1) \geq i\}$ for the higher units; these are elements of the form $1 + c \cdot \pi^i$, for some $i \in \mathbb{N}_{>0}$ and $c \in \mathcal{O}$. Such an element is called an i-th higher unit. Remark that there is a natural isomorphism $\mathbb{F} \xrightarrow{\sim} U_i/U_{i+1}$ for $i \geq 1$, sending $\bar{r}$ to $\overline{1 + r\pi^i}$ for $\bar{r} \in \mathbb{F}$. Also the Teichmüller map $\omega : \mathbb{F}^* \to \mu_{p^f-1} \subseteq F^*$ plays an important role in Bouw's algorithm.

3.5.2 Roots of unity and the weakly distinguished unit

Coprime root of unity

Since the symbol $\left(\frac{x,y}{\mathfrak{m}}\right)_m$ is expressed as a power of the primitive root of unity $\zeta_m \in F$, one first has to compute a primitive root ζ_m. We split $m = d \cdot p^r$, with $p \nmid d$. Since d and p are coprime, we must have that $\langle \zeta_d \rangle \to \mathbb{F} = \mathcal{O}/\mathfrak{m}$ (taking modulo $\mathfrak{m}$) is an injective map (see Lemma 1.78). Therefore $d \mid (p^f - 1)$ and thus the root ζ_d can easily be found by applying Newtonian Hensel-lifting (which is possible since $\Phi'_d(\zeta_d) \not\equiv 0$ modulo $\mathfrak{m}$).

More difficult is the ζ_{p^r}-part of the root of unity – it actually uses the same machinery as is used for computing the symbol $(x, y)_m$. We will return to this later.

p-th powering map

Suppose $x \in \mathfrak{m}$, i.e., $v(x) = i \geq 1$. Then, using the binomial expansion, one has

$$(1+x)^p - 1 = x^p + px^{p-1} + \ldots + px.$$

So, $v((1+x)^p - 1) = \min(p \cdot i, e + i)$. Denoting $d(i) = \min(pi, e+i)$, we have maps

$$m_i : U_i/U_{i+1} \to U_{d(i)}/U_{d(i)+1}, \bar{u} \mapsto \overline{u^p},$$

which are bijections when $pi \neq e + i$, since p-th powering (when $e + i > pi$) and fixed multiplication (when $e + i < pi$) in $\mathbb{F}$ are bijections. We write $u_0 = \frac{-\pi^e}{p}$.

Lemma 3.38. *Write m_i for the p-th powering map restricted to U_i. The following properties are equivalent:*

(i) $\zeta_p \in F$;

(ii) $(p-1)|e$ *and* $N_{\mathbb{F}/\mathbb{F}_p}(\overline{u_0}) = \overline{u_0}^{\frac{p^f-1}{p-1}} = 1$;

(iii) $(p-1)|e$ *and* $\ker m_{e/(p-1)}$ *has dimension one over* $\mathbb{F}_p$*, the field of* p *elements.*

Remark 3.39. The lemma above is proven in the proof of [Bou16, Prop. 7.2], and in fact gives an algorithm for checking whether ζ_p exists in F (by computing the norm, property (ii)). In theoretical sense, this is not needed, because we already assumed that F has ζ_m and therefore has ζ_{p^r} too. In an algorithmic sense, this

[6] Note that finding γ and π is equivalent to computing the ramified representation as in Lemma 1.74. In Bouw's article, it is therefore assumed that an extension $F : \mathbb{Q}_p$ is given in this ramified representation. In a computational context, a ramified representation is useful when one wants to calculate the Teichmüller lift.

is highly useful, since an unobservant mathematician could accidentally ask for a p-th power residue symbol inside a p-adic field where ζ_p does not exist!

The lemma also proves the existence of a so-called weakly distinguished unit (WDU) δ, provided that $\zeta_p \in F$. Note the following exact sequence:

$$1 \to \ker m_{e/(p-1)} \to U_{e/(p-1)} \xrightarrow{m_{e/(p-1)}} U_{pe/(p-1)} \xrightarrow{q} \operatorname{coker}(m_{e/(p-1)}) \to 1$$

Since the dimensions over $\mathbb{F}_p$ of $\ker m_{e/(p-1)}$, $U_{e/(p-1)}$ and $U_{pe/(p-1)}$ are 1, f and f, respectively, we must have $\dim \operatorname{coker}(m_{e/(p-1)}) = 1$ (still provided that $\zeta_p \in F$, of course). So, in particular, $\operatorname{coker}(m_{e/(p-1)})$ has a non-trivial element. An element in $U_{pe/(p-1)}$ that maps under q to a non-trivial element, is called a weakly distinguished unit. This special unit is important in Bouw's algorithm. ◀

Definition 3.40 (p-adic exponentiation)**.** Suppose $a = \sum_{i=0}^{\infty} a_i p^i \in \mathbb{Z}_p$ and $u \in U_j$, with $j \geq 1$. Then we define:

$$u^a := \prod_{i=0}^{\infty} u^{a_i p^i}.$$

Remark 3.41. The above definition is well-defined, since $u^{a_i p^i} \in U_{d^i(j)} \subseteq U_{i+j}$ (here, the crude inequality $d^i(j) \geq i+j$ is used). Therefore, the partial products $\prod_{i=0}^{N} u^{a_i p^i}$ form a Cauchy sequence with respect to the p-adic absolute value; and such sequences converge, by the completeness of F. ◀

Exponential representation

One of the main components of Bouw's algorithm is the so-called exponential representation of principal units (which are the units of height one). This exponential representation writes an element $u \in U_1$ as a product of (p-adic) powers of elements in a finite set T, which we will call the basis units. From now on, we will assume $\zeta_p \in F$.

Definition 3.42. We define the following sets.

(i) $I := \{i \mid 1 \leq i < pe/(p-1) \text{ and } p \nmid i\}$;

(ii) $T_i := \{1 - \omega(\gamma)^j \pi^i \mid 0 \leq j < f\} \subseteq U_i$, for $i \in I$;

(iii) $T := \{\delta\} \sqcup \bigsqcup_{i \in I} T_i$, where δ is a weakly distinguished unit. The elements in the set T are called the basis units;

(iv) $T_i^{(j)} := m^j(T_i) = \underbrace{m \circ \ldots \circ m}_{j \text{ times}}(T_i) = \{t^{p^j} \mid t \in T_i\}$ where m is the p-th powering map.

Notation 3.43. We denote $e/(p-1) = p^k \cdot r$, with $p \nmid r$. Note that $r \in I$ (as in Definition 3.42).

The following lemma shows the importance of the basis units:

Lemma 3.44. *With the notation of Definition 3.42 and Notation 3.43, we have:*

(i) For $i \neq r$ and for all $j \geq 0$, we have that the residues of the elements in $T_i^{(j)}$ generate the group $U_{d^j(i)}/U_{d^j(i)+1}$.

(ii) For all $0 \leq j \leq k$, we have that the residues of the elements in $T_r^{(j)}$ generate the group $U_{d^j(r)}/U_{d^j(r)+1}$.

(iii) For $j \geq k+1$, we have that the residues of the elements in $T_r^{(j)} \cup \{m^{j-(k+1)}(\delta)\}$ generate the group $U_{d^j(r)}/U_{d^j(r)+1}$.

Proof. See [Bou16, §7.2]. □

Lemma 3.45. *Let $d(i) := \min(i+e, pi)$, and let $I = \{1 \leq i < pe/(p-1) \mid p \nmid i\}$. Then, for every $n \in \mathbb{N}_{>0}$ there exist unique $m \geq 0$ and $i \in I$ such that*

$$n = d^m(i),$$

where the superscript m means repeated evaluation.

Proof. We have that d is an injective map; suppose (the other cases are easy) $i + e = d(i) = d(i') = pi'$. Then, $pi \geq i + e = pi' \leq i' + e$. This implies both $i \geq i'$ and $i \leq i'$, i.e. $i = i'$.

(Uniqueness) Suppose $d^m(i) = d^{m'}(i')$. Because d is injective, we have $d^{m-m'}(i) = i' \in I$. If $m - m' > 0$, we have that i' is in the image of d, which is impossible, since then either i' is a multiple of p, or $i' \geq pe/(p-1)$. But then $i' \notin I$, contradiction – therefore $m = m'$ and $i = i'$.

(Existence) Take an arbitrary $n \in \mathbb{N}_{>0}$. Take $k \in \mathbb{N}$ such that $n - ke - e < e/(p-1) \leq n - ke$. Write $t = n - ke$, then $d^k(t) = n$, since[7] $t \geq e/(p-1)$. Notice that $t < e/(p-1) + e = pe/(p-1)$; so we have two cases. One, $p \nmid t$, then $t \in I$, and we are done. Two, $t = p^\ell \cdot b$ with $p \nmid b$, then $t = d^\ell(b)$ and $b \in I$. So in either case, there exist $m \in \mathbb{N}$ and $i \in I$ such that $n = d^m(i)$.

□

The following is a short version of theorem [Bou16, §8.4, Th. 8.15].

Theorem 3.46. *Suppose $\zeta_p \in F$. Then the group homomorphism*

$$\phi : \mathbb{Z}_p^T \to U_1, \ (a_t)_{t \in T} \mapsto \prod_{t \in T} t^{a_t} \tag{3.12}$$

is a surjection, with kernel equal to $(b_t)_{t \in T}\mathbb{Z}_p$ for some $(b_t)_{t \in T} \in \mathbb{Z}_p^T$. This element $(b_t)_{t \in T}$ satisfies

$$\zeta_{p^\ell} \in F \iff \min\{v_p(b_t) \mid t \in T\} \geq \ell. \tag{3.13}$$

The proof of this theorem is insightful, and gives a procedure for finding the maximum ℓ such that $\zeta_{p^\ell} \in F$. Therefore, I will give here a full proof of above theorem, obtained from [Bou16, Th. 8.15] and [Kos14, Th. 2.2].

Proof. We split the proof in three parts:

[7] If $i \geq e/(p-1)$ we have $pi - i = (p-1)i \geq e$, and therefore $pi \geq i + e$. So, for $i \geq e/(p-1)$, we have $d(i) = i + e$.

(Convergence) The product at the right side of (3.12) is finite, in the sense that the set T is finite. So, in order to obtain convergence, one has to prove that the expression t^{a_t} is well-defined and converges, which is explained in Remark 3.41.

(Surjective) Given $u \in U_1$, the powers $(a_t)_{t\in T}$ can be built inductively.

Suppose there is already found $(a_t)_{t\in T} \in \mathbb{Z}_p^T$ such that

$$\mathrm{v}\,(\phi((a_t)_{t\in T} - u) \geq n-1.$$

Now, write $n = d^m(i)$ with $m \in \mathbb{N}$ and $i \in I$. Then, according to Lemma 3.44, either the residues of $T_i^{(m)}$ generate U_n/U_{n+1}, or (when $i = r$) the residues of $T_i^{(m)} \cup \{\delta^{j-(k+1)}\}$ generate U_n/U_{n+1}. So, assigning appropriate powers of elements from $T_i^{(m)}$ (and maybe δ) to $(a_t)_{t\in T}$, resulting in $(a'_t)_{t\in T}$, yields

$$\mathrm{v}\,(\phi((a'_t)_{t\in T} - u) \geq n. \tag{3.14}$$

(Kernel) Writing $e/(p-1) = p^k \cdot r$, as before, we have found that the p-th powering map is an isomorphism $U_i/U_{i+1} \to U_{d(i)}/U_{d(i)+1}$, except in the case where $i = e/(p-1)$. So, the set $T_r^{(k+1)}$ has an element $w^{p^{k+1}}$ that – modulo $U_{pe/(p-1)}$ – can be expressed as a product of the other elements in $T_r^{(k+1)}$. This means in particular that we can find an expression for $w^{p^{k+1}}$ without the use of the element w:

$$w^{p^{k+1}} = \prod_{t\in T\setminus\{w\}} t^{b_t}$$

The element $(b_t)_{t\in T}$ with $b_w = -p^{k+1}$ is in the kernel of the map ϕ, and even generates the kernel. Now, equation (3.13) is easy to explain; if $\min\{v_p(b_t) \mid t \in T\} \geq \ell$, then we are able to construct the following element:

$$r = w^{p^{k+1-\ell}} \cdot \prod_{t\in T\setminus\{w\}} t^{b_t\cdot p^{-\ell}},$$

which equals one when raised to the power p^ℓ. And vice versa, if $\zeta_{p^\ell} \in U_1$, it must be equal to one when raised to the power p^ℓ; therefore, the exponential representation of $\zeta_{p^\ell}^{p^\ell}$ must be in the kernel of ϕ.

□

The goal of above theorem is to prove that there is a fast way to write any element $x \in F^*$ in the following form:

$$x = \omega(u) \cdot (-\pi)^{v(x)} \cdot \delta^{a_\delta} \cdot \prod_{t\in T\setminus\{d\}} t^{a_t} \tag{3.15}$$

with $u \in \mathbb{F}^*$ and $a_t \in \mathbb{Z}_p$ and δ the distinguished unit. Because the power of δ in this representation is so important, and is strongly related to the Hilbert symbol, we give it a special notation:

Notation 3.47. Suppose one can write $x \in F^*$ as in Equation (3.15). Then we denote by $d(x, \pi)$ the power a_δ of δ modulo p^n, where n is maximal such that $\zeta_{p^n} \in F$.

Remark 3.48. Note that $d(x, \pi) \in \mathbb{Z}/p^n\mathbb{Z}$. This is due to the non-uniqueness of the representation of x; the power of δ can vary by p^n-multiples, without varying x itself. Also, one can see that in the notation of $d(x, \pi)$ also the uniformizer π plays a role. Changing π results in changing the representation of x, and therefore changes the symbol $d(x, \pi)$. ◀

Corollary 3.49. *The set $U_{pe/(p-1)+ke}$ consists entirely of p^k-th powers.*

Remark 3.50. In an algorithmic context, this is highly useful; if one wants to compute the p^k-th Hilbert symbol, one only has to compute with precision $pe/(p-1) + ke$. ◀

3.5.3 Find the Hilbert symbol from exponential representation

Notation 3.51. In this subsection, $m = p^n$, with n maximal such that $\zeta_{p^n} \in F$.

Lemma 3.52. *We have $\left(\frac{\pi, x}{\mathfrak{m}}\right)_m = \left(\frac{\pi, \delta}{\mathfrak{m}}\right)_m^{d(x,\pi)}$.*

Proof. We have $1 = \left(\frac{\omega(\gamma)^j\pi^i, 1-\omega(\gamma)^j\pi^i}{\mathfrak{m}}\right)_m$, by the symbol properties of the Hilbert symbol from Lemma 3.21. This implies

$$1 = \left(\frac{\omega(\gamma)^j\pi^i, 1-\omega(\gamma)^j\pi^i}{\mathfrak{m}}\right)_m = \left(\frac{\omega(\gamma)^j, 1-\omega(\gamma)^j\pi^i}{\mathfrak{m}}\right)_m \left(\frac{\pi^i, 1-\omega(\gamma)^j\pi^i}{\mathfrak{m}}\right)_m$$

$$= \left(\frac{\pi^i, 1-\omega(\gamma)^j\pi^i}{\mathfrak{m}}\right)_m = \left(\frac{\pi, 1-\omega(\gamma)^j\pi^i}{\mathfrak{m}}\right)_m^i. \tag{3.16}$$

The second equation by the multiplicative property of the Hilbert symbol, and the third equation follows from the fact that $\omega(\gamma)$ is a p^m-th power, since it is a $(p^f - 1)$-th root of unity.

Therefore, from (3.16), it follows that, when $p \nmid i$

$$\left(\frac{\pi, 1-\omega(\gamma)^j\pi^i}{\mathfrak{m}}\right)_m = 1 \tag{3.17}$$

Now, writing $x = \omega(u) \cdot (-\pi)^{v(x)} \cdot \delta^{a_\delta} \cdot \prod_{t \in T\setminus\{d\}} t^{a_t}$, we have:

$$\left(\frac{\pi, x}{\mathfrak{m}}\right)_m = \left(\frac{\pi, \omega(u) \cdot (-\pi)^{v(x)} \cdot \delta^{a_\delta} \cdot \prod_{t \in T\setminus\{d\}} t^{a_t}}{\mathfrak{m}}\right)_m = \left(\frac{\pi, \delta}{\mathfrak{m}}\right)_m^{a_\delta},$$

since $\omega(u)$ is a p^n-th power, and $\left(\frac{\pi, -\pi}{\mathfrak{m}}\right)_m = 1$, by the symbol properties. The elements of the form t^{a_t} cancel, because t is of the form $1-\omega(\gamma)^j\pi^i$ in combination with equation (3.17).

□

Lemma 3.53. *Write $x = \omega(x)\pi^i u$, and set $\pi' = u\pi$. Then:*

$$\left(\frac{x, y}{\mathfrak{m}}\right)_m = \left(\frac{\pi, \delta}{\mathfrak{m}}\right)_m^{d(y,\pi)(i-1)+cd(y,\pi')}, \tag{3.18}$$

Proof. We have

$$\left(\frac{x, y}{\mathfrak{m}}\right)_m = \left(\frac{\omega(x)\pi^i u, y}{\mathfrak{m}}\right)_m = \left(\frac{\pi, y}{\mathfrak{m}}\right)_m^{i-1} \left(\frac{\pi', y}{\mathfrak{m}}\right)_m$$
$$= \left(\frac{\pi, \delta}{\mathfrak{m}}\right)_m^{(i-1)d(y,\pi)} \cdot \left(\frac{\pi', \delta}{\mathfrak{m}}\right)_m^{d(y,\pi')}.$$

So, we have to compute the c, such that $\left(\frac{\pi',\delta}{\mathfrak{m}}\right)_m = \left(\frac{\pi,\delta}{\mathfrak{m}}\right)_m^c$. We distinguish three cases:

(i) The case $m = 2$. We have $\left(\frac{\pi,\delta}{\mathfrak{m}}\right)_m = \left(\frac{\pi',\delta}{\mathfrak{m}}\right)_m = -1$. Suppose *ad absurdum* that $\left(\frac{\pi,\delta}{\mathfrak{m}}\right)_m = 1$. Then, because $\left(\frac{\pi,x}{\mathfrak{m}}\right)_m = \left(\frac{\pi,\delta}{\mathfrak{m}}\right)_m^{d(x,\pi)}$, we have $\left(\frac{\pi,x}{\mathfrak{m}}\right)_m = 1$ for all $x \in F^*$. By the non-degenerate property of the Hilbert symbol in Lemma 3.21, we can conclude $\pi \in (F^*)^2$. But an uniformizer of an extension of $\mathbb{Q}_2$ is never a square (since it is a prime element). Contradiction. So, set $c = 1$ in this case.

(ii) When $m \neq 2$ and $p \nmid c_0 = d(\pi, \pi')$, calculate $c_1 = d(\pi', \pi)$. Setting $c = -c_1/c_0 \in \mathbb{Z}/p^n\mathbb{Z}$, we have $\left(\frac{\pi',\delta}{\mathfrak{m}}\right)_m = \left(\frac{\pi,\delta}{\mathfrak{m}}\right)_m^c$, since

$$\left(\frac{\pi', \delta}{\mathfrak{m}}\right)_m^{-c_0} = \left(\frac{\pi', \delta}{\mathfrak{m}}\right)_m^{-d(\pi,\pi')} = \left(\frac{\pi', \pi}{\mathfrak{m}}\right)_m^{-1}$$
$$= \left(\frac{\pi, \pi'}{\mathfrak{m}}\right)_m = \left(\frac{\pi, \delta}{\mathfrak{m}}\right)_m^{d(\pi',\pi)} = \left(\frac{\pi, \delta}{\mathfrak{m}}\right)_m^{c_1}. \tag{3.19}$$

(iii) In the harder case when $m \neq 2$ and $p \mid c_0 = d(\pi, \pi')$, we set $\pi'' = -\delta\pi'$. We have

$$\left(\frac{\pi', \delta}{\mathfrak{m}}\right)_m = \left(\frac{\pi', -\pi'\delta}{\mathfrak{m}}\right)_m = \left(\frac{\pi', \pi''}{\mathfrak{m}}\right)_m = \left(\frac{\pi', \delta}{\mathfrak{m}}\right)_m^{d(\pi'',\pi')},$$

and therefore, $d(\pi'', \pi') = 1$. Set $c_2 = d(\pi'', \pi) = d(-1, \pi) + d(\delta, \pi) + d(\pi', \pi) = d(-1, \pi) + 1 + c_1$. In the case $p \neq 2$, $d(-1, \pi) = 0$, and when $p = 2$, one has $d(-1, \pi) = p^{n-1}d(\zeta_{p^n}, \pi)$, which is already calculated in the exponential representation of ζ_{p^n}. Since $p \mid d(-1, \pi)$ and[8] $p \mid c_1$, we have $p \nmid c_2 = d(\pi'', \pi)$, so that we can divide by c_2.

Now define $c_3 = d(\pi', \pi'')$ and $c_4 = d(\pi, \pi'')$. Then:

$$\left(\frac{\pi', \delta}{\mathfrak{m}}\right)_m = \left(\frac{\pi', \delta}{\mathfrak{m}}\right)_m \left(\frac{\pi', -\pi'}{\mathfrak{m}}\right)_m = \left(\frac{\pi', \pi''}{\mathfrak{m}}\right)_m$$

[8] Since $p \mid c_0$, we must have $p \mid c_1$ too. Otherwise one can conclude from equation (3.19), that π' is a p-th power, which is impossible because it is a prime element.

$$= \left(\frac{\pi'', \pi'}{\mathfrak{m}}\right)_m^{-1} = \left(\frac{\pi'', \delta}{\mathfrak{m}}\right)_m^{-d(\pi', \pi'')} = \left(\frac{\pi'', \delta}{\mathfrak{m}}\right)_m^{-c_3}$$

and

$$\left(\frac{\pi, \delta}{\mathfrak{m}}\right)_m^{c_2} = \left(\frac{\pi, \pi''}{\mathfrak{m}}\right)_m = \left(\frac{\pi'', \pi}{\mathfrak{m}}\right)_m^{-1} = \left(\frac{\pi'', \delta}{\mathfrak{m}}\right)_m^{-c_4}. \tag{3.20}$$

therefore, setting[9] $c = c_2 c_3 / c_4$, we have:

$$\left(\frac{\pi', \delta}{\mathfrak{m}}\right)_m = \left(\frac{\pi'', \delta}{\mathfrak{m}}\right)_m^{-c_3} = \left(\left(\frac{\pi'', \delta}{\mathfrak{m}}\right)_m^{-c_4}\right)^{c_3/c_4} = \left(\frac{\pi, \delta}{\mathfrak{m}}\right)_m^{c}.$$

□

Remark 3.54. Note that, with the above procedure, one can write an arbitrary Hilbert symbol as a power of $\left(\frac{\pi,\delta}{\mathfrak{m}}\right)_m$. In order to know the exact Hilbert symbol, one has to use the calibration formula as in Lemma 3.24. ◀

Lemma 3.55. *We still assume $m = p^n$. Pick an $t \in T \cup \{\pi\}$ such that $t_* \not\equiv 1$ modulo p^{n+1} (from Notation 3.23 and Lemma 3.24). And calculate – with the exponential representation – $r \in \mathbb{Z}/m\mathbb{Z}$ such that*

$$\left(\frac{t, \zeta_m}{\mathfrak{m}}\right)_m = \left(\frac{\pi, \delta}{\mathfrak{m}}\right)_m^{r}.$$

Taking the inverse r' of r modulo p^n, we have:

$$\left(\frac{\pi, \delta}{\mathfrak{m}}\right)_m = \zeta_m^{r' \cdot \frac{t_* - 1}{m}}$$

Proof. The existence of an element $t \in T \cup \{\pi\}$ such that $t_* \not\equiv 1$ modulo p^{n+1} follows from the calibration formula in Lemma 3.24, in combination with the fact that ζ_m cannot be an p-th power. Therefore, there exists an element in $u \in F^*$ such that $\left(\frac{u,\zeta_m}{\mathfrak{m}}\right)_m$ is a primitive m-th root, which implies – with the calibration formula – that $u_* \not\equiv 1$ modulo p^{n+1}. Since the elements in $T \cup \{\pi\}$ generate F^*, there must exist some element t in it such that $t_* \not\equiv 1$ modulo p^{n+1}.

From the calibration formula follows the claim. □

[9] Since c_2 is not divisible by p, we must have (by equation (3.20)) that c_4 is not divisible by p too. Therefore we can divide by c_4 in $\mathbb{Z}/p^n\mathbb{Z}$.

CHAPTER 4

Heuristic algorithm for the power residue symbol

4.1 Introduction

In his article [Squ], Squirrel gives an algorithm that reduces the computation of the power residue symbol $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ in arbitrary number fields to the computation of the power residue symbol in $\mathbb{Q}(\zeta_m)$, the m-th cyclotomic field. This algorithm is a generalization of the ideas in a paper of Lenstra [Len95], in which is proven that there is an efficient algorithm for computing quadratic residue symbols in algebraic number fields.

Squirrel also proposes an algorithm that computes the m-th power residue symbol, but it is only a polynomial-time algorithm if m is not allowed to vary. Also, this algorithm is not feasible in most practical situations, because it requires expensive precomputations. These precomputations include enumerating (prime) ideals $\mathfrak{b}$ with norm smaller than some bound that is exponential in m, and naively calculating power residue symbols of the form $\left(\frac{a}{\mathfrak{b}}\right)_m$ for such ideals, for every residue class a mod $\mathfrak{b}$, see [Squ, § V.3].

In the present thesis, I exhibit an algorithm for computing $\left(\frac{\alpha}{\beta}\right)_m$ with $\alpha, \beta \in R$, a number ring containing ζ_m, that does not rely on heavy precomputations, and that has shown to be remarkably fast during experiments. Unfortunately, I am not able to prove that it runs in polynomial time. The algorithm is able to compute power residue symbols of the form $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ for ideals $\mathfrak{b}$ in R, too, but that part is not tested in this thesis.

4.2 Squirrel's algorithm

4.2.1 General power residue symbol

The algorithm of Squirrel computes $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$, with $\alpha \in R$ and $\mathfrak{b}$ an ideal in R. In his approach, he reduces this symbol to computing the principal power residue symbol, by finding a bounded ideal $\mathfrak{c}$ of R such that $(\beta) = \mathfrak{b}\mathfrak{c}$ [Squ, § V.2,

p. 60]. With use of the LLL-algorithm, his algorithm finds $\beta \in \mathfrak{b}$ with $(\beta) = \mathfrak{bc}$, such that $|N(\beta)| < \rho^{\frac{n(n-1)}{4}}$ and $N(\mathfrak{c}) \leq \rho^{\frac{n(n-1)}{4}} \frac{\sqrt{|\Delta(R)|}}{2^s}$. Here $\rho > 1$ is the LLL-constant as in Theorem 2.24, and $s \in \mathbb{N}$ is the number of pairs of complex embeddings of the number ring R.

Remark 4.1. Note that this step results in an element β and an ideal $\mathfrak{c}$ that may have exponential size in $\log n$, even if $\mathfrak{b}$ is small. The norm of $\mathfrak{c}$ can even be superexponential in $\log n$, due to the discriminant part of the bound; $\Delta(R)$ might be around n^n with $n = [K : \mathbb{Q}]$, where K is the number field that is the quotient field of R. ◀

Using the identity $\left(\frac{\alpha}{\beta}\right)_m = \left(\frac{\alpha}{\mathfrak{bc}}\right)_m = \left(\frac{\alpha}{\mathfrak{b}}\right)_m \left(\frac{\alpha}{\mathfrak{c}}\right)_m$, only $\left(\frac{\alpha}{\beta}\right)_m$ and $\left(\frac{\alpha}{\mathfrak{c}}\right)_m$ need to be computed, in order to find $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$. Squirrels algorithm uses (very large) pre-calculated tables [Squ, §V.3] to obtain the value of $\left(\frac{\alpha}{\mathfrak{c}}\right)_m$. Denoting

$$\mathcal{B} = \left\{ \mathfrak{c} \text{ ideal of } R \;\middle|\; N(\mathfrak{c}) \leq \rho^{\frac{n(n-1)}{4}} \frac{\sqrt{|\Delta(R)|}}{2^s} \right\},$$

Squirrel assumes that one has already computed a table of power residue symbols for the following inputs.

$$\left(\frac{\alpha}{\mathfrak{c}}\right)_m \quad \text{for all } \mathfrak{c} \in \mathcal{B} \text{ and all } \alpha \bmod \mathfrak{c}.$$

These are computed by factoring the denominator and using Definition 3.4, which makes the computation of the tables – both practically and theoretically – unfeasible.

Remark 4.2. In this thesis, a probably more effective way is proposed to reduce a general power residue symbol to a principal power residue symbol, see Algorithm 8. ◀

4.2.2 Principal power residue symbol

In order to compute principal power residue symbols, Squirrel uses a reduction step. This reduction step is based on an idea of Hurwitz, adapted by Lenstra to obtain a Euclidean-like algorithm [Len80]. Given $\alpha, \beta \in R$, the reduction step finds $\gamma, \beta' \in R$ and $j \in \mathbb{Z}$ such that

$$j\beta = \gamma\alpha + \beta' \text{ with } N(\beta') < N(\alpha)/2. \tag{4.1}$$

This is very similar to the ordinary Euclidean condition, except for the $j \in \mathbb{Z}$ and the reduction of the norm by a half. Squirrel obtains the result as in (4.1) with use of LLL [Squ, § V.2, p. 61]. The main disadvantage of this method is that the integer j is not bounded polynomially by n, the degree of the number ring R:

$$|j| \leq 4\rho^{\frac{(n+1)(n-1)}{4}} \frac{\sqrt{|\Delta(R)|}}{2^s}.$$

Squirrel again uses tables to overcome this problem [Squ, §V.3]. Denote the set

$$\mathcal{C} = \left\{ j \;\middle|\; j \in \mathbb{N}, 0 < j \leq 4\rho^{\frac{(n+1)(n-1)}{4}} \frac{\sqrt{|\Delta(R)|}}{2^s} \right\}.$$

Then, Squirrel assumes that symbols $\left(\frac{\alpha}{jR}\right)_m$ have already been computed for all $j \in \mathcal{C}$ and all representatives α modulo jR. Since there are about j^n representatives modulo jR, this table has an approximate size of

$$\sum_{0<j\leq 4\rho^{\frac{(n+1)(n-1)}{4}}\frac{\sqrt{|\Delta(R)|}}{2^s}} j^n > \left(4\rho^{\frac{(n+1)(n-1)}{4}}\frac{\sqrt{|\Delta(R)|}}{2^s}\right)^n. \tag{4.2}$$

For an 'easy' number field like $\mathbb{Q}(\zeta_{11})$, taking $\rho = 2$ as the standard LLL-constant, the right side of (4.2) is equal to $(4 \cdot 2^{11\cdot 9/4} \cdot \sqrt{11^{10}}/2^5)^{10} \approx 3.5 \cdot 10^{117}$, which is much more than an estimation of the number of atoms in the observable universe ($\approx 10^{81}$).

Remark 4.3. The short calculation above stopped me from thinking about trying to implement this part of Squirrels algorithm. It is obvious that Squirrel invented this part only for theoretical reasons, in order to prove that there is a polynomial time algorithm for the computation of power residue symbols, for fixed cyclotomic order m. ◀

4.3 Preliminaries

4.3.1 Notation

We denote by K a number field containing ζ_m, and by R a number ring in K. It is assumed that α, β are coprime elements in R and that they are coprime to m too. That is, α, β and m do not share any (ideal) divisors. Let $n = [K : \mathbb{Q}]$ be the degree of K over $\mathbb{Q}$. Since R is a number ring, it has a $\mathbb{Z}$-basis of integral elements: $(\gamma_1, \dots, \gamma_n)$. It is assumed in the main algorithms that $\zeta_m \in R$ is already computed and expressed in the integral $\mathbb{Z}$-basis of R.

The prototype of the description above is $K = \mathbb{Q}(\zeta_m)$ and $R = \mathbb{Z}[\zeta_m] = \mathcal{O}_K$ with $\mathbb{Z}$-basis $(1, \zeta_m, \zeta_m^2, \dots, \zeta_m^{\phi(m)-1})$.

Remark 4.4. Although I did not test it, there is no reason why the main algorithm would not work for non-Galois extensions. ◀

4.3.2 Two-sided reduction

Two-sided reduction is a reduction technique proposed by [Len15]. It gives a partial solution for computing the principal power residue symbol. In order to calculate $\left(\frac{\alpha}{\beta}\right)_m$, one would like to find 'quite small' $\gamma_1, \gamma_2 \in \mathcal{O}_K$ such that

$$\gamma_1\alpha \equiv \gamma_2 \bmod \beta.$$

The lattice $\Lambda_{\alpha,\beta} = \{(\gamma_1, \gamma_2) \mid \gamma_1\alpha \equiv \gamma_2 \bmod \beta\} \subseteq \mathcal{O}_K \times \mathcal{O}_K$ with standard Euclidean metric has determinant $N(\beta)$, by similar arguments as in Lemma 2.34. Then, reduction with LLL, gives[1]

$$\sqrt{\|\gamma_1\|_2^2 + \|\gamma_2\|_2^2} \leq \rho^{\frac{n-1}{4}} \cdot N(\beta)^{\frac{1}{2n}},$$

[1] Note that γ_1 and γ_2 are here $\mathbb{Z}$-vectors, since they are 'written' in a chosen integral basis of $\mathcal{O}_K$.

which is a fairly good reduction. But, one wants to relate $N(\beta)$ and $\|\beta\|_2$, the Euclidean norm of the representation of β in the integral basis of $\mathcal{O}_K$.

To avoid the technical machinery needed for this, I made a slight modification of two-sided reduction, so called q-ary two-sided reduction, already introduced in this form in Notation 2.32. This is about the following modular relation, for a prime number q:

$$\gamma_1\alpha + \gamma_2\beta \equiv 0 \bmod q.$$

Since I believe that in the q-ary lattice $\Lambda^q_{\alpha,\beta}$, as in Notation 2.32, it is easier to find small vectors, I hope that using this modification leads to shorter results in practice. The main idea is to set $q \approx \max(\|\alpha\|_2, \|\beta\|_2)$ – the main algorithm takes q around $\|\beta\|_2$. Then, after this reduction, one proceeds recursively with the hopefully much smaller elements $\frac{\gamma_1\alpha+\gamma_2\beta}{q}$ and γ_1, in order to compute $\left(\frac{\alpha}{\beta}\right)_m$.

Remark 4.5. I did not practically compare ordinary two-sided reduction to q-ary two sided reduction extensively. However, in the construction phase of my implementation, I found that the q-ary two-sided reduction gave – at least in small cyclotomic fields – better reduction results than the ordinary two-sided reduction. ◀

Remark 4.6. A similar 'two-sided reduction' technique (but not q-ary) is used in [Gro03], to compute the tame kernel, i.e. the kernel of the map

$$(t_v)_{v<\infty} : K_2(F) \to \bigoplus_{v<\infty} k_v^*,$$

an important invariant in algebraic K-theory. Here F is a number field, v is a finite valuation and k_v is the residue field with respect to the valuation v. ◀

4.3.3 Near-prime ideals

In Algorithm 8 and Algorithm 10 the notion of B-near primeness is used. A near prime ideal has a norm that is the product of one single large prime and several other very small primes. More formally:

Definition 4.7 (B-near prime number). An integer $N \in \mathbb{N}$ is said to be a B-near prime number if N factorizes as follows:

$$N = p \cdot \prod_{i=1}^{k} p_i \text{ with } p_i \leq B \text{ for all } 1 \leq i \leq k.$$

Definition 4.8 (B-near prime ideal). An ideal $\mathfrak{a}$ of R is called a B-near prime ideal when the norm $N(\mathfrak{a})$ is a B-near prime number as in Definition 4.7.

Remark 4.9. The definition of B-near prime ideal can be considered as quite awkward, since one might expect that the set of B-near prime ideals contains the set of prime ideals. This is false, since prime ideals that have residue class degree larger than one are not considered as a 1-near prime ideal; such prime ideals do not have prime norm. The set of B-near prime ideals, however, does contain the completely split prime ideals. ◀

Remark 4.10. If B is sufficiently small, say of polynomial size in the degree $n = [K : \mathbb{Q}]$, then B-near prime ideals $\mathfrak{a}$ are effectively factorizable, since one can find the prime factorization of the norm. Write $N(\mathfrak{a}) = p \cdot \prod_{i=1}^{k} p_i^{m_i}$ with $p_i \leq B$ and $p_i \neq p_j \neq p$ for $i \neq j$.

(a) Set $\mathfrak{p}_p := (\alpha, p)$;

(b) Factor $(\alpha, p_i^{m_i}) = \prod_{j=1}^{k_i} \mathfrak{p}_{p_i,j}^{t_j}$;

(c) Now, $(\alpha) = \mathfrak{p}_p \prod_{i=1}^{k} \prod_{j=1}^{k_i} \mathfrak{p}_{p_i,j}^{t_j}$.

In the special (and often occurring) case when $m_i = 1$ for all i, one has $(\alpha) = \mathfrak{p}_p \prod_{i=1}^{k} \mathfrak{p}_{p_i}$, where $\mathfrak{p}_{p_i} = (\alpha, p_i)$.

Step (b) can be sped up by first computing (α, p_i), and then factorizing it, which gives the factors $\mathfrak{p}_{p_i,j}$ already. The calculation of the complete factorization of $(\alpha, p_i^{m_i})$ is then much easier. ◀

Remark 4.11. For B of polynomial size in the degree $n = [K : \mathbb{Q}]$, recognizing B-near prime ideals can clearly be done by a fast, polynomial time algorithm. For an ideal $\mathfrak{a}$, calculate the norm $N = N(\mathfrak{a})$, then apply trial division up to B to the number N, i.e. $N = r \cdot \prod_{i=1}^{k} p_i$ with $p_i \leq B$. Then, use a fast primality proving algorithm to check whether r is prime or not. If r is prime, return '$\mathfrak{a}$ is a B-near prime ideal'. Otherwise return '$\mathfrak{a}$ is not a B-near prime ideal'. Since primality proving can be done in polynomial time [AKS02], this procedure gives a polynomial time algorithm for recognizing B-near prime ideals.

Note that one might also want to use elliptic curve factorization to factor the norm N partially, instead of trial division, for speeding up this factorization process. I expect that this only pays off when the degree of the number field is really large. ◀

4.4 Description of the main algorithm

4.4.1 Outline

The main algorithm of this thesis has essentially three parts.

(i) **Principalization**
Reduce the calculation of $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ to the computation of $\left(\frac{\alpha}{\beta}\right)_m$ and $\left(\frac{\alpha}{\mathfrak{p}}\right)_m$ for some $\beta \in \mathfrak{b}$ and some prime ideal $\mathfrak{p}$. This process is called principalization, and is done by Algorithm 8.

(ii) **Reduction**
Reduce a power residue symbol $\left(\frac{\alpha}{\beta}\right)_m$, for large input α and β, to many power residue symbols $\left(\frac{\alpha_i}{\beta_i}\right)_m$ with small α_i and β_i, using two-sided reduction. This part is called reduction, and is done by Algorithm 9. Despite the fact that this part of the algorithm is not needed and is even suspected to have superpolynomial running time, it speeds up the overall process significantly, at least for number fields with small degree (say, below 100).

(iii) **Evaluation**
Compute a power residue symbol $\left(\frac{\alpha}{\beta}\right)_m$ with relatively small α and β, by finding a near-prime element in the residue class of $\gamma^m\alpha \bmod \beta$ for some $\gamma \in R$. This is called the evaluation part of the overall algorithm, which one can see in Algorithm 10.

Remark 4.12. The word 'principalization' also occurs in class field theory, where it is used in the principal ideal theorem [CG05, p. 169]. The meaning of the word 'principalization' in this thesis is not related to this theorem. ◀

4.4.2 Principalization

The principalization algorithm (Algorithm 8), consists of sampling 'random', relatively small elements $\beta \in \mathfrak{b}$, and hoping that the ideal $\mathfrak{c} = (\beta)/\mathfrak{b}$ is a B-near prime ideal. Such B-near prime ideals are easily factorizable, and one calculates $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ by computing $\left(\frac{\alpha}{\beta}\right)_m \cdot \left(\frac{\alpha}{\mathfrak{c}}\right)_m^{-1}$, where in the computation of $\left(\frac{\alpha}{\mathfrak{c}}\right)_m$, the factorization of $\mathfrak{c}$ is used.

Remark 4.13. In line 6 of Algorithm 8, one needs a bound C on the vector $(c_1, \ldots, c_n)$. According to [Coh00, Rm. (2), p. 24], $C = 3$ is more than sufficient for essentially all purposes in his book. I expect that it won't be different in this particular algorithm. ◀

Remark 4.14. The idea of principalization was put forward by dr. H.W. Lenstra, after my presentation about this thesis [Len16]. This is the main reason why this specific aspect of the overall algorithm is not tested; I didn't have the opportunity. The other parts, however, are tested extensively. ◀

Remark 4.15. In this thesis there is no algorithm given that describes how to inverse ideals, which is needed in line 10 of Algorithm 8. An efficient algorithm is given in [Coh93, §4.8.4]. ◀

4.4.3 Reduction

The recipe of Algorithm 9 gives a heuristic algorithm for reducing the principal power residue symbol with large-coefficient input to many instances of the principal power residue symbol with input having small coefficients. The crucial element in the algorithm is two-sided reduction in q-ary lattices, as in subsection 4.3.2.

The algorithm reduces the computation of $\left(\frac{\alpha}{\beta}\right)_m$ to the computation of $\left(\frac{\beta}{\gamma_1}\right)_m$ and $\left(\frac{\beta}{\eta}\right)_m$, for (hopefully) small γ_1 and η. Those last two symbols are then reduced again, and so on, inducing a tree-like structure, see Figure 4.1. At the leaves of this tree, the function `PrincipalPowResSym` (Algorithm 10) is invoked.

Remark 4.16. By Definition 3.4 and Definition 3.15, the symbol $\left(\frac{\alpha}{\beta}\right)_m$ is not well-defined when β has a singular prime in its factorization, or when β and m have a ideal divisor in common. In Algorithm 9 such instances might occur. However, note that this does not directly lead to severe miscalculations, since

Algorithm 8: Principalization: reducing the general power residue symbol to the principal power residue symbol

```
PowerResidueSymbol($\alpha, \mathfrak{b}$);
Input  : An element $\alpha$ in $R$ and an ideal $\mathfrak{b}$ in $R$
Output: The power residue symbol $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$
// Formally, the input also contains an integral basis of $R$, $\zeta_m$
   represented in that basis, and $m$.
Set $n := [K : \mathbb{Q}]$, where $K$ is the quotient field of $R$ ;
Set $B := n^3$ for the bound for near-primeness;
Compute an LLL-reduced basis $(\beta_1, \ldots, \beta_n)$ of $\mathfrak{b}$ ;
do
    // Pick a random but small element from $\mathfrak{b}$
    Pick a random vector $(c_1, \ldots, c_n) \in \mathbb{Z}^n$, with $|c_i| \leq C$ for all $i$ ;
    Set $\beta := \sum_{i=1}^{n} c_i \beta_i$;
    Calculate $N := N(\beta)/N(\mathfrak{b})$ ;
while $N$ is not a $B$-near prime number;
// $N$ is of the form $p \cdot \prod_{i=1}^{r'} p_i$ for 'small' $p_i$ now
Calculate the ideal $\mathfrak{c} := (\beta)/\mathfrak{b}$ ;
Factorize $\mathfrak{c} := \mathfrak{p} \cdot \prod_{i=1}^{r} \mathfrak{p}_i$, using the factorization of $N$ as in Remark 4.10;
Compute $\left(\frac{\alpha}{\mathfrak{c}}\right)_m$ using above factorization ;
Compute $\left(\frac{\alpha}{\beta}\right)_m$ = PowResRed($\alpha, \beta$) using Algorithm 9 ;
Return $\left(\frac{\alpha}{\beta}\right)_m \left(\frac{\alpha}{\mathfrak{c}}\right)_m^{-1}$ ;
```

applying reciprocity may vanish the problem; one could simply define

$$\left(\frac{\alpha}{\beta}\right)_m := U(\alpha, \beta) \left(\frac{\beta}{\alpha}\right)_m ,$$

which is a sound definition when α doesn't have a singular prime in its factorization and is coprime to m. Even when α does not satisfy these requirements, one could still consider the fact that α and β are coprime, and replace α by suitable translations $\alpha + \kappa\beta$, for some $\kappa \in R$.

The essence of this remark is that during Algorithm 9, it is not required that all inputs of symbols are coprime to m and coprime to all singular primes, if the numerator and the denominator are coprime at least. ◀

Remark 4.17. When the Galois group of the extension $K : \mathbb{Q}$ is cyclic, one can choose the prime q in such a way that q remains prime, i.e., q does not factor into smaller ideals. This might slightly increase the speed of the algorithm (because one doesn't have to factor q), even considering the fact that q remaining prime is an extra condition on q. In the case when $K = \mathbb{Q}(\zeta_{p^k})$ for p a prime number, the Galois group Gal$(K : \mathbb{Q})$ is cyclic, and q stays prime if and only if $\bar{q} \in (\mathbb{Z}/p^k\mathbb{Z})^*$ is a generator of this group [Chi07, Ch.1, Thm. 1.8]. This trick also works in cyclotomic fields $\mathbb{Q}(\zeta_n)$ when $(\mathbb{Z}/n\mathbb{Z})^*$ is cyclic. ◀

Notation 4.18. In Algorithm 9 the notation $\ll$ and $\gg$ is used. In my implementation, I used the following:

$$a \ll b \iff a < b;$$

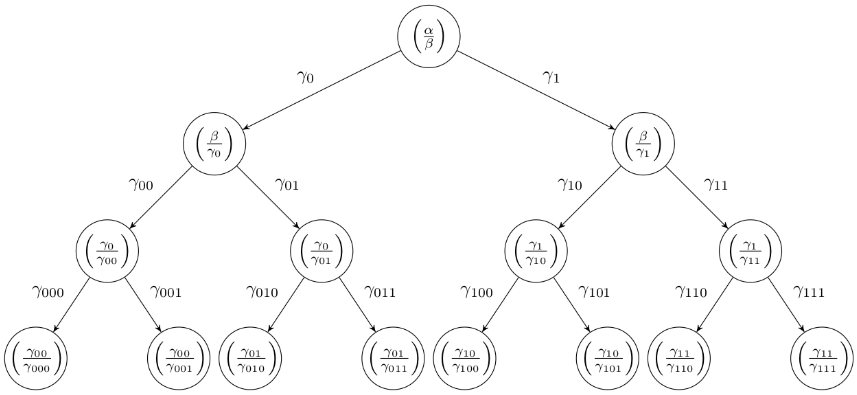

$\left(\frac{\alpha}{\beta}\right)$
γ_0
γ_1
$\left(\frac{\beta}{\gamma_0}\right)$
$\left(\frac{\beta}{\gamma_1}\right)$
γ_{00}
γ_{01}
γ_{10}
γ_{11}
$\left(\frac{\gamma_0}{\gamma_{00}}\right)$
$\left(\frac{\gamma_0}{\gamma_{01}}\right)$
$\left(\frac{\gamma_1}{\gamma_{10}}\right)$
$\left(\frac{\gamma_1}{\gamma_{11}}\right)$
γ_{000}
γ_{001}
γ_{010}
γ_{011}
γ_{100}
γ_{101}
γ_{110}
γ_{111}
$\left(\frac{\gamma_{00}}{\gamma_{000}}\right)$
$\left(\frac{\gamma_{00}}{\gamma_{001}}\right)$
$\left(\frac{\gamma_{01}}{\gamma_{010}}\right)$
$\left(\frac{\gamma_{01}}{\gamma_{011}}\right)$
$\left(\frac{\gamma_{10}}{\gamma_{100}}\right)$
$\left(\frac{\gamma_{10}}{\gamma_{101}}\right)$
$\left(\frac{\gamma_{11}}{\gamma_{110}}\right)$
$\left(\frac{\gamma_{11}}{\gamma_{111}}\right)$

Figure 4.1: Tree-like structure of Algorithm 9

$$a \gg b \Longleftrightarrow a > b.$$

An alternative interpretation might be

$$a \ll b \Longleftrightarrow a < \sqrt{b};$$

$$a \gg b \Longleftrightarrow \sqrt{a} > b.$$

Notation 4.19. Algorithm 9 uses a size function $s : \mathcal{O}_K \to \mathbb{N}$. It is defined by $s(\alpha) = n \cdot \|\alpha\|_\infty$, with $n = [K : \mathbb{Q}]$ the degree of the number field and $\|\alpha\|_\infty$ the maximum-norm of α written in some beforehand chosen basis of R. One can also use the Euclidean norm $\|\alpha\|_2$.

Remark 4.20. In line 19 of the reduction Algorithm 9, a matrix N will be LLL-reduced. The line states 'w.r.t. the chosen metric', which essentially means that, beforehand, the programmer has to choose between the weighted metric $\|\cdot\|_{\alpha,\beta}$ as in Definition 2.36 or the unweighted metric $\|\cdot\|_2$ as in Definition 2.37. In my implementation, I chose the unweighted metric. ◀

Remark 4.21. In line 3 of Algorithm 9, β is considered small if all of its coefficients are smaller than 10^4. This upper bound is quite practically chosen, since Algorithm 10 is quite fast with such small input. Of course, one can freely alter this upper bound; this might even be faster, see Remark 4.26. ◀

Remark 4.22. In line 23 in Algorithm 9 a quite qualitative notion 'short rows' is used. In my own implementation, I only use the upper three rows of the reduced matrix; if none of those three rows work, the program exits this loop (and jumps to line 17). ◀

4.4.4 Evaluation

The idea of Algorithm 10 is to repeatedly multiply α_0 by the m-th power of random γ, until $\gamma^m \alpha_0 \bmod \beta$ has a small representative $\hat{\alpha} \in R$ that generates a 'near prime ideal'. By 'near prime ideal' is meant an ideal that is the product

Algorithm 9: Reduction: heuristically reducing the principal power residue symbol

```
PowResRed($\alpha, \beta$);
Input  : Elements $\alpha, \beta \in R$.
Output: The power residue symbol $\left(\frac{\alpha}{\beta}\right)_m$
// Formally, the input also contains an integral basis of $R$, $\zeta_m$
   represented in that basis, $m$, and the size function $s$.
Calculate $s_\alpha = s(\alpha)$ and $s_\beta = s(\beta)$.
if $s_\beta < 10^4 \cdot n$ then
    Calculate PrincipalPowResSym($\alpha, \beta$) as in Algorithm 10, and return ;
end
if $s_\alpha \ll s_\beta$ then
    Calculate the Umkehrfaktor $U(\alpha, \beta)$ and ;
    recursively call PowResRed($\beta, \alpha$) ;
    Return $U(\alpha, \beta) \cdot$ PowResRed($\beta, \alpha$). ;
end
if $s_\alpha \gg s_\beta$ then
    Calculate the basis matrix $M_\beta$ of the ideal $(\beta)$ ;
    LLL-reduce the matrix $M_\beta$ ;
    Use Algorithm 5 and get a representative of $\alpha$ mod $\beta$ with reasonably small
      coefficients ;
end
// Now, $\alpha$ and $\beta$ are about the same size.
do
    Pick a (new) prime number $q \approx s(\beta)$ ;
    Construct the $3n \times 2n$-matrix $N$ with the first $n$ rows given by ($\gamma_i \cdot \alpha$ mod
      $q$, $\gamma_i \cdot \beta$ mod $q$) for $0 \leq i \leq n-1$ and the last $2n$ rows given by the
      diagonal matrix $q \cdot I_{2n}$ ;
    LLL-reduce the matrix $N$ w.r.t. the chosen metric ;
    do
        Extract the shortest row $r$ of $N$, and find the two elements $\delta_1$ and $\delta_2$
          given by the first $n$ entries, and the last $n$ entries of this row,
          respectively ;
        Remove row $r$ from the matrix $N$ ;
    while $\delta_2$ and $\beta$ have a common divisor and $N$ has still short rows;
while $\delta_2$ and $\beta$ have a common divisor;
Calculate $\eta = \frac{\delta_2\alpha - \delta_1\beta}{q} \in R$ ;
Recursively call PowResRed($\delta_2, \beta$) and PowResRed($\eta, \beta$). Also, calculate $U(q, \beta)$
  and compute $\left(\frac{\beta}{q}\right)_m$ by factoring the prime $q$ ;
Return $\left(\frac{\beta}{q}\right)_m \cdot U(q, \beta) \cdot \frac{\text{PowResRed}(\eta,\beta)}{\text{PowResRed}(\delta_2,\beta)}$ ;
```

of one single prime ideal having (large) prime norm with several other prime ideals with a tiny norm, see Definition 4.8.

After finding such an element $\hat{\alpha}$ generating a near-prime ideal, that is equivalent to $\gamma_0^m \alpha_0$ modulo β for some $\gamma_0 \in R$, one can use reciprocity – one only has to compute $\left(\frac{\beta}{\hat{\alpha}}\right)_m$ now. But since one knows the factorization[2] of $\hat{\alpha}$, one

[2] One can factorize $(\hat{\alpha})$ because $(\hat{\alpha})$ is a B-near prime for some B, see Remark 4.10.

can write $(\hat{\alpha}) = \mathfrak{p}_p \cdot \prod_{i=1}^{k} \mathfrak{p}_i$ and computes

$$\left(\frac{\beta}{\hat{\alpha}}\right)_m = \left(\frac{\beta}{\mathfrak{p}_p}\right)_m \cdot \prod_{i=1}^{k} \left(\frac{\beta}{\mathfrak{p}_i}\right)_m .$$

Remark 4.23. In line 11 of Algorithm 10, the bound $B = n^3$ is used, with $n = [K : \mathbb{Q}]$, where K is the number field of the ring R. This particular choice is loosely based on the Extended Riemann Hypothesis (see Remark 4.35), and may be increased to some other bound polynomial in n. Also, one might want to set $B = \min(10^6, p(n))$, so that the bound is not too small for fields of small degree. ◀

Remark 4.24. In line 12 of Algorithm 10, one wants to avoid $\hat{\alpha}$ that have a norm having a common divisor with m, because one cannot compute the power residue symbol above a prime ideal that divides m. In the case when R is not the full ring of integers, or when one does not know whether R is the ring of integers, one might want to replace 'N has divisors in common with m' by 'N has divisors in common with $\Delta(R)$'. In the case that one finds an N that has a common divisor with $\Delta(R)$, one might use that to one's own advantage, as in Remark 1.51. ◀

Remark 4.25. The algorithm is a Las Vegas probabilistic algorithm since the running time is a random variable [MR95, §1.2]. One can turn this algorithm into a Monte Carlo algorithm by halting the algorithm when a specific amount of multiplications with m-th powers of random γ is reached; the algorithm has failed in that case. The advantage of this method is that one has a clearer view of the running time. ◀

4.5 The correctness of the algorithm

4.5.1 Principalization correctness

Starting from the first line, we check the correctness of Algorithm 8, line-by-line. Lines 1 to 4 are initializations. Lines 6-7 let us obtain a relatively small $\beta \in \mathfrak{b}$. In line 8 one calculates the number $N = N(\beta)/N(\mathfrak{b}) = N((\beta)/\mathfrak{b}) = N(\mathfrak{c})$ by the multiplicative property of the norm (see Lemma 1.38). Although this multiplicative property might not be true in non-integrally closed R, it is true for ideals that do not contain singular primes in its factorization.

Since $N = N(\mathfrak{c})$, one sees that lines 10-12 are consistent with line 8. Lines 13 and 14 use the multiplicative property of the power residue symbol (see Property 3.36). Since $\mathfrak{c}\mathfrak{b} = (\beta)$, we have

$$\left(\frac{\alpha}{\beta}\right)_m = \left(\frac{\alpha}{\mathfrak{b}}\right)_m \left(\frac{\alpha}{\mathfrak{c}}\right)_m ,$$

proving line 14 to return the correct answer.

4.5.2 Reduction correctness

We check the correctness of Algorithm 9 line-by-line. Line 1 to 5 are obviously correct. The lines 6 to 10 use reciprocity: $\left(\frac{\alpha}{\beta}\right)_m = U(\alpha, \beta) \left(\frac{\beta}{\alpha}\right)_m$, and are

Algorithm 10: Evaluation; heuristically computing the principal power residue symbol

```
1  PrincipalPowResSym($\alpha_0, \beta$);
   Input  : Elements $\alpha_0, \beta \in R$.
   Output: The power residue symbol $\left(\frac{\alpha_0}{\beta}\right)_m$
   // Formally, the input also contains an integral basis of $R$, $\zeta_m$
      represented in that basis, and $m$.
2  Set $n$ as the degree of the number field of $R$ ;
3  do
4      do
5          Take a random $\bar{\gamma} \in R/\beta$ ;
6          Set $\alpha := \alpha_0 \cdot \bar{\gamma}^m$ modulo $\beta$, with modular exponentiation ;
7      while $\alpha$ is not invertible modulo $\beta$;
8      Find $\bar{\alpha}$, a small representative of $\alpha$ modulo $\beta$, as in Algorithm 6 on
        page 28 ;
9      Lift $\bar{\alpha}$ coordinate-wise to $R$, call it $\hat{\alpha}$ ;
10     Calculate its norm, $N := N(\hat{\alpha})$ ;
11     Factorize $N = \left(\prod_{i=1}^{k} p_i\right) \cdot r$ using trial division with bound $B = n^3$ ;
       // I.e. $p_i \leq B$ for all $i$, and $p_i$ are primes
12 while $r$ is not prime or $N$ has divisors in common with $m$;
   // $r$ is prime, and $\hat{\alpha}$ is invertible mod $\beta$
13 Set $\mathfrak{p}_r = (\alpha, r)$ ;
14 Factorize the ideal $(\alpha) = \mathfrak{p}_r \cdot \prod_{i=1}^{s} \mathfrak{p}_i$, using the factorization of $N$, as in
    Remark 4.10 ;
15 Calculate the Umkehrsymbol $U(\hat{\alpha}, \beta)$ ;
16 Return $\prod_{i=1}^{s} \left(\frac{\beta}{\mathfrak{p}_i}\right)_m \cdot \left(\frac{\beta}{\mathfrak{p}_r}\right)_m \cdot U(\hat{\alpha}, \beta)$ ;
```

therefore correct. Lines 11 to 14 only reduce α modulo β, and since the symbol $\left(\frac{\alpha}{\beta}\right)_m$ is invariant under translations of α by β, this is also a correct step.

Lines 17 to 19 create a lattice in $\mathbb{Z}^{2n}$ given by the matrix N. Elements ϵ_1, ϵ_2, given by the first n respectively the last n entries of a row of N, always satisfy $\epsilon_1 \cdot \beta - \epsilon_2 \cdot \alpha \equiv 0$ modulo q, see also Lemma 2.34 and 2.35. Therefore, after LLL-reduction, the elements δ_1, δ_2 that are formed by the shortest row, also satisfy

$$\delta_1 \cdot \beta - \delta_2 \cdot \alpha \equiv 0 \pmod{q}. \tag{4.3}$$

Lines 20 to 24 ensure us that δ_2 and β do not have a divisor in common and that the shortest rows are taken from the matrix N. By Equation (4.3), $\delta_2 \cdot \alpha - \delta_1 \cdot \beta$ is divisible by q, making η in line 25 well-defined. Line 26 only makes recursive calls. The correctness of line 27 is proven as follows:

$$\left(\frac{q}{\beta}\right)_m \left(\frac{\eta}{\beta}\right)_m = \left(\frac{q\eta}{\beta}\right)_m = \left(\frac{\delta_2\alpha - \delta_1\beta}{\beta}\right)_m = \left(\frac{\delta_2\alpha}{\beta}\right)_m = \left(\frac{\delta_2}{\beta}\right)_m \left(\frac{\alpha}{\beta}\right)_m .$$

Now, using $\left(\frac{q}{\beta}\right)_m = U(q,\beta)\left(\frac{\beta}{q}\right)_m$, one has:

$$\left(\frac{\alpha}{\beta}\right)_m = U(q,\beta)\left(\frac{\beta}{q}\right)_m \left(\frac{\eta}{\beta}\right)_m \left(\frac{\delta_2}{\beta}\right)_m^{-1}.$$

which verifies line 27. Note that δ_2 and β do not have a factor in common, by line 23 and 24. Therefore, there is no dividing or multiplying by zero (since, then η is also coprime to β).

4.5.3 Evaluation correctness

Starting with the last line of Algorithm 10, line 16, we have

$$\begin{aligned}\prod_{i=1}^{s}\left(\frac{\beta}{\mathfrak{p}_i}\right)_m \cdot \left(\frac{\beta}{\mathfrak{p}_r}\right)_m \cdot U(\hat{\alpha},\beta) &= \left(\frac{\beta}{\mathfrak{p}_r \cdot \prod_{i=1}^{s}\mathfrak{p}_i}\right)_m \cdot U(\hat{\alpha},\beta)\\ &= \left(\frac{\beta}{\hat{\alpha}}\right)_m \cdot U(\hat{\alpha},\beta) = \left(\frac{\hat{\alpha}}{\beta}\right)_m,\end{aligned}$$

by reciprocity and the fact that $\mathfrak{p}_r \cdot \prod_{i=1}^{s}\mathfrak{p}_i$ is the prime ideal factorization of $(\hat{\alpha})$.

But, since $\hat{\alpha} \equiv \bar{\alpha} \equiv \alpha \equiv \gamma_0^m \alpha_0$ modulo β, for some $\gamma_0 \in R/\beta$, we have

$$\left(\frac{\hat{\alpha}}{\beta}\right)_m = \left(\frac{\gamma_0^m \alpha_0}{\beta}\right)_m = \left(\frac{\alpha_0}{\beta}\right)_m.$$

by Property 3.37 (residuosity) and Property 3.32 (translation-invariance). So, indeed, Algorithm 10 is sound.

4.6 Analysis

4.6.1 Introduction

The analysis below is far from mathematically rigorous, due to the heuristic character of the algorithms. However, the analysis should give you a rough indication that the overall algorithm[3] will work in many cases. Also, this incomplete analysis might give an idea how to prove rigorously that the overall algorithm of this thesis is indeed an effective probabilistic algorithm, in which I did not succeed.

4.6.2 Reduction analysis

In order to keep the leaves of the reduction tree in Figure 4.1 polynomially bounded in the size of α and β, the logarithm of the size after one reduction must be reduced by a factor $c < 1$. Explicitly, if the reduction of the symbol $\left(\frac{\alpha}{\beta}\right)_m$ by Algorithm 9 yields η and δ_2, one would like to have

$$\log s(\eta) \leq c \cdot \min(\log s(\alpha), \log s(\beta)) \text{ and } \log s(\delta_2) \leq c \cdot \min(\log s(\alpha), \log s(\beta)).$$

[3] With the overall algorithm is meant the combination of Algorithm 8, Algorithm 9, and Algorithm 10.

The lattice $L^q_{\alpha,\beta}$ has, with the Euclidean norm, discriminant q^n, by Lemma 2.38. The element δ_2 is (often) obtained by the shortest row of the LLL-reduced matrix of the lattice $L^q_{\alpha,\beta}$ (see Notation 2.32, Algorithm 7 and line 18 of Algorithm 9). Therefore, using bounds on the output of the LLL-algorithm (see Theorem 2.24), we have

$$\|\delta_2\|_2 \leq 2^{\frac{n-1}{2}}\sqrt{q} \approx 2^{\frac{n-1}{2}}\sqrt{\|\beta\|_2}.$$

Note that q is chosen around $\|\beta\|_2$. So,

$$\log\|\delta_2\|_2 \lesssim \frac{n-1}{2}\log(2) + \frac{1}{2}\log\|\beta\|_2.$$

For $(c - \frac{1}{2})\log\|\beta\|_2 > \frac{n-1}{2}\log 2$, i.e. $\|\beta\|_2 > 2^{\frac{n-1}{2c-1}}$, we can use the above bound to prove $\log\|\delta_2\|_2 \lesssim c \cdot \|\beta\|_2$.

So, when the coefficients of β (and α) have size larger than $n = [K : \mathbb{Q}]$, this reduction 'works'. In practice, the reduction works too when the input is smaller.

The analysis of $\eta = \frac{\delta_2\alpha - \delta_1\beta}{q}$ is harder, since there is multiplication involved. Assume heuristically that $\|\kappa\lambda\|_2 \leq C\|\kappa\|_2\|\lambda\|_2$ and α and β are roughly of the same size. Since q is defined to be around $\|\beta\|_2$, we have:

$$\begin{aligned}\|\eta\|_2 &\leq 2C\max(\|\delta_1\|_2, \|\delta_2\|_2)\|\beta\|_2/q \approx 2C\max(\|\delta_1\|_2, \|\delta_2\|_2)\\ &\leq 2^{\frac{n+1}{2}}C\sqrt{q} \approx 2^{\frac{n+1}{2}}C\sqrt{\|\beta\|_2}.\end{aligned}$$

So, if C is not too large, the conclusion will be: Heuristically, the reduction algorithm works at least when the sizes of the coefficients of the elements are larger than n, the degree of the extension.

Remark 4.26. Using this reasoning, it is smart to stop when $s_\beta < n \cdot n$, in line 3 of Algorithm 9, instead of $s_\beta < 10^4 \cdot n$. The main reason why I didn't do this, is because I wanted to test whether it is possible to prove that this algorithm can reduce to elements with a fixed coefficient size, within polynomial time. According to Figure 5.1, this doesn't seem to be true.

In future research, one can test whether the reduction algorithm seems to terminate in polynomial time when above stopping criterion is used. ◀

4.6.3 Evaluation analysis

The 'loop' part of Algorithm 10, i.e. lines 3–12, is the most difficult part to analyse, since it is not clear when this loop terminates. The main question is: how often is $N(\hat{\alpha})$ of the form $p \cdot \prod_{i=1}^k p_i$ with p a large prime and $p_i \leq B$, i.e., when is $N(\hat{\alpha})$ a B-near prime number as in Definition 4.7?

Heuristically, one may assume that $N(\hat{\alpha}) \approx N(\beta) \approx C^n$, with $C = \max_i b_i$, where $b_i \in \mathbb{Z}$ are the coefficients of β written in the integral basis of the number ring, as in Definition 1.12.

Remark 4.27. If $N(\hat{\alpha})$ is randomly distributed among the numbers around C^n, one might hope, by the prime number theorem [MV07, Ch. 6] [Apo98, Ch. 13], that $N(\hat{\alpha})$ has approximately probability $\frac{1}{n\log C}$ to be a prime number, and even a slightly larger probability to be a B-near prime number. So the expected number of tries in 'the loop' (lines 3-12) before one finds a near-prime is approximately $n\log C$, and maybe slightly smaller. The above suggests that the running time of this part of the evaluation algorithm is a random variable having geometrical distribution with parameter $p \approx \frac{1}{\log N(\beta)}$. If this is true, the evaluation algorithm has expected polynomial running time. ◀

Although I didn't succeed in proving that $N(\hat{\alpha})$ is indeed randomly distributed among the numbers around C^n, I do have some arguments why $N(\hat{\alpha})$ might be B-near prime 'quite frequently'. The prime ideal theorem of Landau, [Ove15, Prop. 9.16] [Lan03] says something about the distribution of norms of prime ideals.

Theorem 4.28. *If K is a number field, then*

$$\sum_{N(\mathfrak{p}) \leq x} 1 \sim \frac{x}{\log x} \text{ as } x \to \infty, \tag{4.4}$$

where $\mathfrak{p}$ ranges over all prime ideals of $\mathcal{O}_K$.

Remark 4.29. We know that there are also prime ideals which have norm p^f with $f > 1$ and p prime, but note that such prime ideals contribute only $O(\sqrt{x})$ to the left part of (4.4), in the same reasoning as in [Ove15, p. 284]; since for such prime ideals $\mathfrak{p} = (p, \alpha)$ holds $x \geq N(\mathfrak{p}) \geq p^2$. Therefore only about $O(n\sqrt{x})$ of them might exist, with $n = [K : \mathbb{Q}]$. Therefore, a vast majority of ideals $\mathfrak{a}$ of $\mathcal{O}_K$ that pop up as a prime ideal have prime norm. ◀

In the particular context of Algorithm 10, one only samples principal ideals, whereas the above theorem is about non-principal ideals, too. The following result follows if one applies [Nar04, §7.2, Prop. 7.17, p. 347] to the set

$$A := \{\mathfrak{p} \text{ prime ideal of } \mathcal{O}_K \mid \mathfrak{p} \text{ is a principal ideal } \}.$$

This is a regular set of prime ideals (see for example [Neu99, §13, Thm. 13.2]) with Dirichlet density $\frac{1}{h_K}$, where $h_K = \#\mathrm{Cl}(K)$ the class number of K.

Theorem 4.30. *If K is a number field, then*

$$\sum_{\substack{\mathfrak{p}=(\pi) \\ N(\mathfrak{p}) \leq x}} 1 \sim \frac{x}{h_K \log x} \text{ as } x \to \infty, \tag{4.5}$$

where $\mathfrak{p}$ ranges over all principal prime ideals of $\mathcal{O}_K$.

Remark 4.31. Note that the algorithm searches for B-near (principal) prime ideals, instead of principal prime ideals only. One expects that the class number does not influence the density of those B-near prime ideals that much, because also prime ideals in other ideal classes can be sampled. So, one expects some asymptotic relation of the following form.

$$\sum_{\substack{\mathfrak{a}=(\alpha) \text{ B-near prime} \\ N(\mathfrak{a}) \leq x}} 1 \sim c_{B,K} \cdot \frac{x}{\log x} \text{ as } x \to \infty, \tag{4.6}$$

where $\frac{1}{h_K} \leq c_{B,K}$ and where $c_{B,K}$ is the Dirichlet density of the B-near principal prime ideals. ◀

Notation 4.32. Denote

$$\rho_K := \frac{2^{r_1+r_2}\pi^{r_2}R_K}{w_K\sqrt{|\Delta(K)|}},$$

where r_1 is the number of real embeddings of K, r_2 is the number of pairs of complex embeddings of K, R_K is the regulator of K (see for example [Neu99, p. 42-43]), w_K is the number of roots of unity in K, and $\Delta(K)$ is the discriminant of $\mathcal{O}_K$.

The following theorem counts the number of ideals in a particular ideal class C that have norm bounded by x, and is obtained from [Ove15, §9.5, Prop. 9.17].

Theorem 4.33. *For K a number field, and C an ideal class in $Cl(K)$, we have*

$$\sum_{\substack{N(\mathfrak{a})\leq x\\ \mathfrak{a}\in C}} 1 \sim \rho_K x$$

Combining equation (4.6) from Remark 4.31 and the equation of Theorem 4.33 with the trivial ideal class, one might expect that the probability that a random sampled principal ideal[4] is a B-near principal prime ideal equals

$$\mathbb{P}[\mathfrak{a} \text{ is a } B\text{-near prime ideal}] = \frac{c_{B,K}x/\log(x)}{\rho_K x} = \frac{c_{B,K}}{\rho_K}\cdot\frac{1}{\log x}. \tag{4.7}$$

If one wants to prove that evaluation Algorithm 10 has expected polynomial running time via this reasoning, one has to prove that $\frac{\rho_K}{c_{B,K}}$ is polynomially bounded in the degree n.

Remark 4.34 (Applying Brauer-Siegel). One can apply the Brauer-Siegel theorem [Lan94, Ch. XIII §4] to the sequence $K_m = \mathbb{Q}(\zeta_m)$ of cyclotomic fields [Was12, pp. Lm. 4.18, Lm. 4.19] to obtain the asymptotic relation

$$\log(h_m R_m) = \frac{1}{2}\log\Delta_m + o(\log\Delta_m),$$

where $h_m = h_{K_m}$, $R_m = R_{K_m}$ and $\Delta_m = \Delta(K_m)$. Note that $r_1 = 0$ and $r_2 = \phi(m)/2$ in this case. This means

$$\log\rho_{K_m} = \frac{1}{2}\phi(m)\log(2\pi) + \log R_m - \log w_{K_m} - \frac{1}{2}\log\Delta_m =$$
$$= \frac{1}{2}\phi(m)\log(2\pi) - \log h_m - \log w_{K_m} + o(\log\Delta_m) \text{ as } m\to\infty.$$

Using the fact [Was12, Thm. 4.20, Lm. 4.18] that $\log h_m^+ = \frac{1}{4}\phi(m)\log m + o(\phi(m)\log m)$ and $\log\Delta_m = \phi(m)\log m + o(\phi(m)\log m)$, we can conclude

$$\log\rho_{K_m} = -\frac{1}{4}\phi(m)\log m + o(\phi(m)\log m).$$

Here we use that the factors $\log w_{K_m} = \log m$ and $\phi(m)\log(2\pi)$ are eventually negligible compared to $\phi(m)\log m$. Note that we use the asymptotic behaviour of h_m^+ here, instead of that of h_m. Since $h_m = h_m^+ h_m^- \geq h_m^+$, this does not cause problems. ◀

[4] Randomly sampled with norm bounded by x.

Above heuristic reasoning suggests that for cyclotomic fields of increasing degree, ρ_{K_m} will eventually be very small. So, if the constant $\rho_{K_m}/c_{B,K_m}$ (see equation (4.7)) is polynomially bounded in $\phi(m)$ for $B = p(\phi(m))$, Algorithm 10 might be a expected polynomial time algorithm.

Note that Figure 5.1 suggest that, for $B = n^3$, we have $\rho_{K_m}/c_{B,K_m} \sim n^{2.34}$. Here $n = \phi(m) = [\mathbb{Q}(\zeta_m) : \mathbb{Q}]$.

Remark 4.35. According to [Bac90], assuming the Extended Riemann Hypothesis, the class group is generated by ideals with norm not exceeding $12\log(\Delta(K))^2$. Applying this in our context, it means that the class group of $\mathbb{Q}(\zeta_m)$ is generated by ideals with norm not exceeding $\sim 12{\cdot}\phi(m)^2\log(m)^2$. Taking $B = \phi(m)^3 = n^3$ in Algorithm 10 is therefore not a bad idea.

One then hopes heuristically that 'all classes are touched' in the sampling of Algorithm 10. ◀

4.6.4 Principalization analysis

The analysis of the principalization Algorithm 8 also heavily relies on conclusions from results in the field of analytic number theory. The $\beta \in \mathfrak{b}$ is obtained by some $\mathbb{Z}$-linear combination of basis elements from a LLL-reduced basis. From Theorem 2.24, one may assume that

$$\|\beta\|_2 = \|\sum_{i=1}^{n} c_i\beta_i\|_2 \le C\sum_{i=1}^{n}\|\beta_i\|_2 \le C\prod_{i=1}^{n}\|\beta_i\|_2 \le C\cdot 2^{\frac{n(n-1)}{2}}N(\mathfrak{b}). \tag{4.8}$$

So, β has coefficients around $C\cdot 2^{\frac{n(n-1)}{2}}N(\mathfrak{b})/\sqrt{n}$, therefore the norm of β is around $\frac{C^n}{n^{n/2}}\cdot 2^{\frac{n(n-1)}{2}}N(\mathfrak{b})^n$. Here is C the constant as in Remark 4.13. So,

$$N(\mathfrak{c}) = N(\beta)/N(\mathfrak{b}) \lesssim \frac{C^n}{n^{n/2}}\cdot 2^{\frac{n(n-1)}{2}}N(\mathfrak{b})^{n-1}.$$

Remark 4.36. Assuming that $N(\mathfrak{c})$ is a random distributed number around $N(\mathfrak{b})^n$, one might expect that the principalization algorithm has probabilistic running time that has the geometric distribution with parameter $p \approx \frac{1}{n\cdot N(\mathfrak{b})}$; if this is true, it has as a consequence that this algorithm is a polynomial expected time algorithm. However, it is not really plausible that $N(\mathfrak{c})$ is random distributed. ◀

The following theorem counts the number of ideals that have norm bounded by x, and is obtained from [Ove15, §9.5, Prop. 9.17].

Theorem 4.37. *For K a number field, $h_k = \#Cl(K)$ the class number and ρ_K as in Notation 4.32, we have*

$$\sum_{N(\mathfrak{a})\le x} 1 \sim h_K\rho_K x.$$

where $\mathfrak{a}$ ranges over all ideals of $\mathcal{O}_K$ with norm not exceeding x.

The above theorem, in combination with the theorem of Landau, as in Theorem 4.28, implies that the probability that a sampled ideal $\mathfrak{c}$ of norm around x is prime, equals

$$\frac{x/\log x}{h_K\rho_K x} = \frac{1}{h_K\rho_K\log x}.$$

Writing $\delta_{B,K}$ for the Dirichlet density of the B-near prime ideals[5], one has the following probability that a sampled ideal $\mathfrak{c}$ of norm around x is B-near prime.

$$\mathbb{P}[\mathfrak{a} \text{ is a } B\text{-near prime ideal}] = \frac{\delta_{B,K} x/\log(x)}{h_K \rho_K x} = \frac{\delta_{B,K}}{h_K \rho_K} \cdot \frac{1}{\log x}. \tag{4.9}$$

The difference with equation (4.7) in the analysis of the evaluation algorithm, is that above equation (4.9) has an extra factor $\frac{1}{h_K}$. Therefore I think this algorithm is harder to analyse, and that it might be not a expected polynomial time algorithm at all. I doubt that it is true that $\frac{h_K \rho_K}{\delta_{B,K}}$ is polynomially bounded in $n = \phi(m)$.

Unfortunately, I didn't have the time to make tests and timings for Algorithm 8.

4.7 Possible improvements

Remark 4.38. (Overall improvement) When $K : \mathbb{Q}$ is Galois, one can use the Galois action on ideals to factorize them partially: computing the greatest common divisor $\text{Gcd}(\mathfrak{b}, \sigma(\mathfrak{b}))$ for every $\sigma \in \text{Gal}(K/\mathbb{Q})$ often yields a partial factorization. This partial factorization is quite useful, since one then obtains ideals whose dividing prime ideals have the same splitting behaviour in $K : \mathbb{Q}$, see [Neu99, §I.9] or [Ste04, Ch. 14]. For example, an ideal $\mathfrak{b}_0$ that is coprime to every Galois-conjugate of itself, consists only of completely split primes. In such ideals, one can always obtain an integral representative of α modulo $\mathfrak{b}_0$, i.e. $\alpha \equiv \ell$ modulo $\mathfrak{b}_0$, with $\ell \in \mathbb{Z}$.

Note that it might be faster to factorize ideals in this way, and apply principalization again. However, I did not test this. ◀

Remark 4.39. (Improvement for reduction Algorithm 9) In line 27 of Algorithm 9, one has to factorize the prime number q in the number ring R. Although there are many effective probabilistic algorithms that can factorize prime numbers in number rings, there is no deterministic polynomial time algorithm known [GP01, §4.2].

According to [Len16], it is not needed to factorize q in order to compute $\left(\frac{\beta}{q}\right)_m$. It is enough if we know the splitting behaviour[6] $q = \prod_{i=1}^{r} \mathfrak{q}_i$ of q in $\mathcal{O}_K$ and the residue class degrees $f(\mathfrak{q}_i/q)$.

In the Galois case the splitting behaviour is partially known, see Lemma 1.31, since $f_q = f(\mathfrak{q}_i/q)$ is the same for all $0 \leq i \leq r$, so $rf = n$. With the Artin map, one can reveal the number f. For example, in $K = \mathbb{Q}(\zeta_m)$, $f_q = \text{Ord}(q \bmod m)$, the order of $\bar{q}$ in $(\mathbb{Z}/m\mathbb{Z})^*$.

When $K : \mathbb{Q}$ is Galois and one knows $f = f_q$, one can thus compute

$$c_j := \frac{1}{f} \log_q N(\text{Gcd}(\beta^{\frac{q^f - 1}{m}} - \zeta_m^j, q))$$

for $0 \leq j \leq m - 1$. This yields the following identity:

$$\left(\frac{\beta}{q}\right)_m = \zeta_m^{\sum_{j=0}^{m-1} j c_j}.$$

[5] This is *not* the same constant as $c_{B,K}$, since that is the density of the *principal* B-near prime ideals.

[6] We assume that q is unramified, since, often, q is quite large. In the ramified case something very similar can be done.

For the non-Galois case one might have different $f(\mathfrak{q}_i/q)$, making the situation slightly more complicated, but surely not impossible or infeasible. I didn't work out the non-Galois case.

Although it is not clear whether or not this improvement increases the running time of the reduction algorithm, this adaptation avoids the use of probabilistic polynomial factoring algorithms over finite fields, making the reduction algorithm fully deterministic. ◀

Remark 4.40. (Improvement for reduction Algorithm 9) In line 27 of Algorithm 9, one computes $U(q, \beta)$. At least in the case when $m = \ell$ is an odd prime, and $\beta \in \mathbb{Z}[\zeta_\ell]$, one can use Eisenstein reciprocity to compute $U(q, \beta)$, instead of Bouw's algorithm. In that case, compute an appropriate $c \in \mathbb{Z}/\ell\mathbb{Z}$ such that $\zeta_\ell^c \beta \equiv 1 \bmod (1 - \zeta_\ell)^2$ and $\zeta_\ell^c \beta$ is coprime to q. Then:

$$\left(\frac{\beta}{q}\right)_m = \left(\frac{\zeta_\ell^c \beta}{q}\right)_m \left(\frac{\zeta_\ell^c}{q}\right)_m^{-1} = \left(\frac{q}{\zeta_\ell^c \beta}\right)_m \left(\frac{\zeta_\ell^c}{q}\right)_m^{-1} = \left(\frac{q}{\beta}\right)_m \left(\frac{\zeta_\ell^c}{q}\right)_m^{-1},$$

see [Lem00, Thm. 11.6, Thm. 11.9]. Since we have $\left(\frac{\zeta_\ell^c}{q}\right)_m = \zeta_\ell^{c\frac{q^\ell - 1}{\ell}}$, we have the following formula [Lem00, Thm. 11.9(iii)] in this case:

$$\left(\frac{q}{\beta}\right)_m = \zeta_\ell^{c\frac{q^\ell - 1}{\ell}} \left(\frac{\beta}{q}\right)_m .$$

◀

Remark 4.41. (Improvement for evaluation Algorithm 10) If one knows that $\sigma(\beta)$ is coprime to β, for some Galois automorphism σ, then one deduces that the factorization of β into prime ideals consists entirely of completely split primes [Jan96, p. 54]. In that case, one can find for every α modulo β, some integer $k \in \mathbb{N}$ such that $\alpha \equiv k$ modulo β. In that case, an adaptation of Algorithm 10 can be used, that defines $N = k$ instead of $N = N(\hat{\alpha})$. Then, no assumptions about the distribution of primes in the norm-set are required. However, in step 13–16, one has to factorize the prime numbers of k into prime ideals, which might be the bottleneck in this adaptation. ◀

Remark 4.42. (Overall improvement) Let $m = \prod_{i=1}^r p_i^{n_i}$ be the prime factorization of m. One can reduce the computation of the power residue symbol $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ to the computation of the power residue symbols $\left(\frac{\alpha}{\mathfrak{b}}\right)_{p_i^{n_i}, K_i}$ where $K_i = \mathbb{Q}(\zeta_{p_i^{n_i}}, \alpha)$.

Write $m = rp^k$, with $p \nmid r$. By the consistency property of the Artin map [Chi07, p. 167], on has

$$\left(\frac{\alpha}{\mathfrak{b}}\right)_{m,\mathbb{Q}(\zeta_m)}^r = \left(\frac{\rho_{\mathbb{Q}(\zeta_m,\, \sqrt[m]{\alpha})/\mathbb{Q}(\zeta_m)}(\mathfrak{b})[\sqrt[m]{\alpha}]}{\sqrt[m]{\alpha}}\right)^r = \frac{\rho_{\mathbb{Q}(\zeta_m,\, \sqrt[m]{\alpha})/\mathbb{Q}(\zeta_m)}(\mathfrak{b})[\sqrt[p^k]{\alpha}]}{\sqrt[p^k]{\alpha}}$$

$$= \frac{\rho_{\mathbb{Q}(\zeta_{p^k},\, \sqrt[p^k]{\alpha})/\mathbb{Q}(\zeta_{p^k},\alpha)}(N_{\mathbb{Q}(\zeta_m)/\mathbb{Q}(\zeta_{p^k},\alpha)}(\mathfrak{b}))[\sqrt[p^k]{\alpha}]}{\sqrt[p^k]{\alpha}} = \left(\frac{\alpha}{\mathfrak{b}}\right)_{p^k,\mathbb{Q}(\zeta_{p^k},\alpha)} .$$

According to Squirrel [Squ, Ch. 4], one can reduce the computation of such symbols in K_i to the calculation of power residue symbols in $\mathbb{Q}(\zeta_{p_i^{n_i}})$.

Briefly, power residue symbols in $\mathbb{Q}(\zeta_m)$ can be obtained by calculating power residue symbols in each of the subfields $\mathbb{Q}(\zeta_{p_i^{n_i}})$, where $p_i^{n_i}$ are the prime powers occurring in m. ◀

CHAPTER 5

Computational Results

5.1 Introduction

In this chapter, the implementations of the reduction Algorithm 9 and the evaluation Algorithm 10 are tested and timed. Unfortunately, I didn't have the time to test the principalization Algorithm 8.

The algorithms are tested on cyclotomic fields $K = \mathbb{Q}(\zeta_m)$ solely, having ring of integers $\mathcal{O}_K = \mathbb{Z}[\zeta_m]$ with $\mathbb{Z}$-basis $(1, \zeta_m, \zeta_m^2, \ldots, \zeta_m^{\phi(m)-1})$. Every element of K is represented with respect to this fixed basis. If an element $\alpha \in \mathcal{O}_K$ is said to have coefficients of size 4, for example, then one means the coefficients of α with respect to the above integral basis of $\mathbb{Z}[\zeta_m]$.

5.2 Method

5.2.1 Reduction method

For this thesis, I implemented the reduction Algorithm 9 and the algorithm of Bouw that computes the Umkehrfaktor, using the computer algebra system Magma [BCP97]. Initially, I tried to get the running time of this algorithm as a whole. Unfortunately, I encountered problems with completions in Magma; the function[1] `Completion` runs really slow when the number field has a degree above 30 and seems to be the bottleneck. Completions are required in this algorithm, because Bouw's computation of the Umkehrfaktor works with complete fields. To overcome this problem, I do *not* calculate the Umkehrfaktors, but I only note when the Umkehrfaktor is called.

When the reduced symbols in the reduction algorithm are small enough, the evaluation algorithm is called – see line 3 to 5 of Algorithm 9. Instead of actually calculating these, I only kept track of the number of calls to the evaluation

[1] `http://magma.maths.usyd.edu.au/magma/handbook/text/352#3310`

algorithm that computes the small power residue symbols. To clarify this all, I will give an example.

Example 5.1. In the timings, I have calculated the power residue symbol of $\alpha, \beta \in \mathbb{Q}(\zeta_{15})$ where α and β have entries of about 20 digits in the $\mathbb{Z}$-basis $(1, \zeta_{15}, \ldots, \zeta_{15}^7)$. So, for example, α is of the form:

$$\alpha = 38294810592849103948 + 28501937593840392810\zeta + \ldots$$
$$+91058493017584639 28\zeta^7.$$

In a real computation, one of course computes all Umkehrfaktors and small power residue symbols. In my timings, however, I omitted these and I only count how many times those functions are called. So, after such a timing, the result might be: "The computation of the power residue symbol of `alpha` and `beta` needed 45 calls of the Umkehrfaktor and needed 129 'small' calls of the evaluation algorithm. The running time was 0.24s". ◀

So there are three variables in each running-time measurement: the number of Umkehrfaktor calls, the number of 'small' power residue symbol calls, and the running time in seconds. To give an indication of the relation of these variables with the input size, I made two log-log plots, which can be seen in Figure 5.2 and Figure 5.3. The first of these two has on the y-axis the number of Umkehrfaktor calls and 'small' calls, whereas the second has the running time on the y-axis. Both have the input size on the x-axis, of course.

Remark 5.2. The implemented algorithm uses the 'unweighted' norm as in Definition 2.37. I have not yet implemented a version with the weighted norm (as in Definition 2.36); so there might be an improvement if that norm is used. ◀

5.2.2 Evaluation method

I also have implemented the evaluation Algorithm 10. I made several timings in cyclotomic fields $\mathbb{Q}(\zeta_m)$ with various m and with elements that have coefficient sizes $4, 8$ and 16. The results can be found in Table A.3 in the appendix and in the chart (Figure 5.1).

Remark 5.3. In these timings, I omitted the time that was needed to calculate the Umkehrsymbol, which is only called one single time per calculation. I did not calculate these Umkehrsymbols at all, because of the problems I encountered with the `Completion` function of Magma. Also, I have separated the time needed for lines 3-12 (the 'loop'), and the time needed for lines 13-16. This because of the possible unpredictable behaviour of the loop, since I do not know whether this loop indeed terminates within reasonable time, see the analysis of the evaluation algorithm (subsection 4.6.3). ◀

5.3 Results

The results of the timings of the evaluation algorithm can be seen in Table A.3 in the appendix and in the chart in Figure 5.1 of this section. For the reduction algorithm, the results are stated in Table A.2 and the charts in Figure 5.2 and Figure 5.3. Although there is a short explanation below each chart, Table A.1 in the appendix elaborates on the meaning of the variables occurring in the tables and charts in more detail.

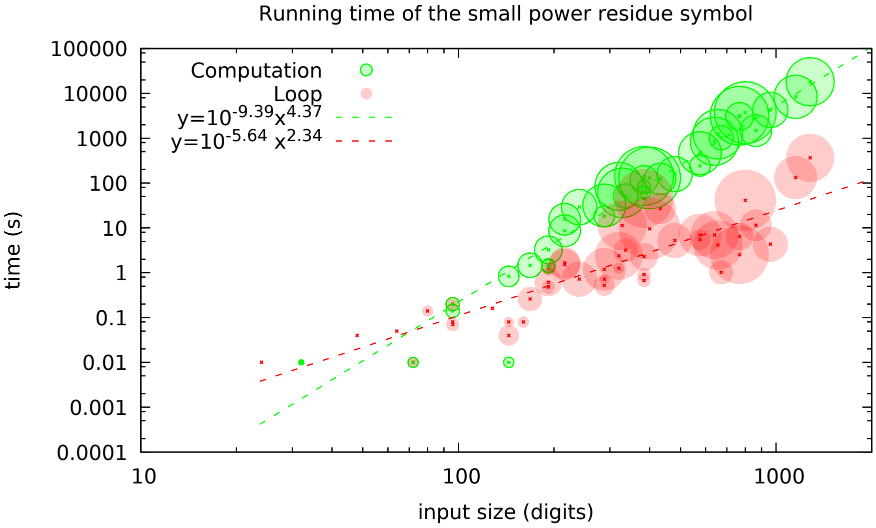


Figure 5.1: A log-log plot of the evaluation algorithm, with on the x-axis the size of the input. The red circles describe the running time for the loop (i.e. line 3-12) of Algorithm 10, and the green circles describe the time needed for lines 13-16 in Algorithm 10. The size of the circles is proportional to the degree of the extension $[K : \mathbb{Q}] = \phi(m)$.

5.4 Conclusion

5.4.1 Evaluation

Figure 5.1 suggests that evaluation Algorithm 10 might be a probabilistic algorithm with expected polynomial time.

Remark 5.4. Remark that lines 13-16 (green circles) require much more time than line 3-12 (the loop, red circles). This is a consequence of calculating the ideal $\mathfrak{p}_p = (\alpha, p)$; computing the greatest common divisor of two ideals is done by applying the Hermite normal form, which runs in $O(kn^4 \log^2(M))$ time for $n \times k$ matrices with coefficients bounded by M [MW01], see also subsection 2.3.3. However, in this specific case, $p \approx N(\alpha) \approx C^n$, where $C = \|\alpha\|_\infty$, the absolute value of the maximum coefficient of α with respect to the chosen integral basis $(1, \zeta, \ldots, \zeta_m^{\phi(m)-1})$ of $\mathbb{Z}[\zeta_m]$. Therefore, $M \approx C^n$, which means that one expects that computing the ideal $\mathfrak{p}_p = (\alpha, p)$ might require $O(n^7 \log(C))$ time (by setting $n = 2k$). The timings give a slightly better order (≈ 4.4), which might be explained by fast multiplication methods.

So, the probabilistic part (lines 3-12) does not seem to be the bottleneck of the evaluation algorithm, which surprises me positively. ◀

Remark 5.5. In the tests of the evaluation Algorithm 10, the class numbers of the fields $\mathbb{Q}(\zeta_m)$ that were tested vary heavily. For example (see Table A.3) the field $\mathbb{Q}(\zeta_{73})$ has class number around $12 \cdot 10^6$ [OEIS]. One might conclude that the size of the class group doesn't have any influence on the running time of the algorithm.

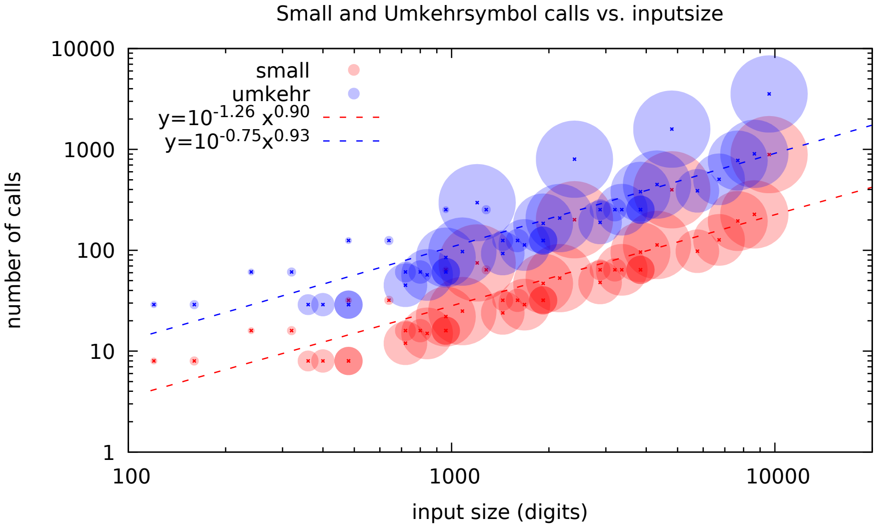


Figure 5.2: A log-log plot of the reduction algorithm, with on the x-axis the size of the input. The blue circles represent the number of calls of the Umkehrfaktor function, the red circles represent the number of calls of the power residue symbols with 'small input' (the evaluation algorithm). The size of the circles is proportional to the degree of the extension $[K : \mathbb{Q}] = \phi(m)$.

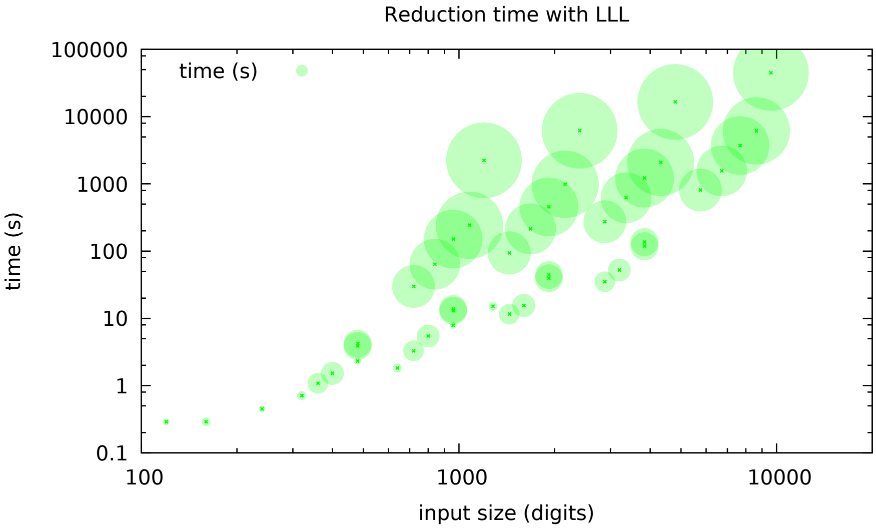


Figure 5.3: A log-log plot of the reduction algorithm, with on the x-axis the size of the input. The green circles represent the running time of the reduction. The size of the circles is proportional to the degree of the extension $[K : \mathbb{Q}] = \phi(m)$.

◀

5.4.2 Reduction

As the charts in Figure 5.2 and 5.3 suggest, the reduction Algorithm 9 is probably, in the present form, not a polynomial-time algorithm. Although the algorithm seems to have order one in the coefficient size, there is no indication that its running time is polynomial in the degree, which one can see by examining the sizes of the circles in Figure 5.2 and 5.3.

Remark 5.6. The running time of the reduction algorithm seems to depend only linearly on the coefficient size. As the evaluation Algorithm 10 has order ≈ 4.4 in the coefficient size, it is likely that it is more effective to run a suitable combination of the reduction and the evaluation algorithm, at least when the coefficient sizes are fairly large. ◀

Remark 5.7. In Figure 5.2 and 5.3 one can see four 'trails' of circles. They are formed by the different coefficient sizes; $40, 60$ and 80 and 160 digits. ◀

Remark 5.8. The observation that increasing the degree influences the running time much more than increasing the coefficient size might partially be caused by the LLL-algorithm, whose running time is order 6 in the degree, but only order 3 in the entry size [NV10, p. 150, Thm. 3], see subsection 2.4.2. ◀

Remark 5.9. The number of Umkehrfaktor calls is consistently approximately three times as big as the number of small calls. This is not a coincidence. The recursive steps of the heuristic Algorithm 9 form a binary tree, as in Figure 5.4. In every ramification of this binary tree, the Umkehr symbol is called three times: to reverse $\left(\frac{p}{\beta}\right)_m$, $\left(\frac{\delta_1}{\beta}\right)_m$ and $\left(\frac{\eta}{\beta}\right)_m$, see Algorithm 9, line 27. At the end of the branches (the leaves), there is one single call for a small power residue symbol computation (the evaluation algorithm). Since the number of ramifications is almost equal to the number of leaves in a binary tree, it is evident that there are three times as many Umkehrcalls as small calls. ◀

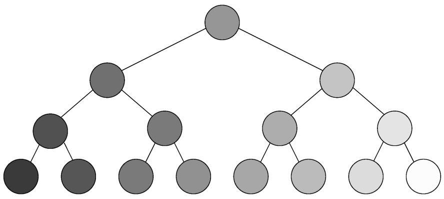


Figure 5.4: A binary tree

Remark 5.10. Remark that the Umkehrcalls in Figure 5.2 are counted as if they are all equally hard to compute. They are not; in the beginning of a computation the Umkehrcalls will have larger input, implying a harder computation. At the end of the computation of a power residue symbol, the input of the Umkehrcalls will be small. Those computations are not as hard, of course. For sake of simplicity and conciseness, I have made no distinction between those Umkehrcalls. ◀

5.5 Discussion

Remark 5.11. In the starting phase of my research about the power residue symbol, I attempted to compute the power residue symbol deterministically, by applying the reduction algorithm (Algorithm 9) only. The main idea was to reduce a power residue symbol with large input to many power residue symbols with small input. Power residue symbols with small input – so was my thought – can then be factored easily. I considered an element to be small when the absolute values of its coefficients are all bounded by some constant C. However, for varying degree n, elements $\alpha \in R$ with coefficients around C have norm $N(\alpha) \approx C^n$, a number with approximately $\log_{10}(C) \cdot n$ digits. There is no known algorithm that factors numbers of this size, within polynomial time (in the size of the input).

So reduction alone is not enough; that is why I also made a probabilistic 'evaluation algorithm' (Algorithm 10), which turned out to perform much better than the reduction algorithm. ◀

Remark 5.12. As already pointed out in the analysis of the reduction algorithm in subsection 4.6.2, it was also possible to let the reduction algorithm terminate earlier, when the reduced elements are of size n. This might be an adaptation that makes the reduction part polynomial time. ◀

Remark 5.13. It seems to be a good idea to run a combination of the reduction- and the evaluation algorithms, as already pointed out in the conclusion. Where exactly one has to switch from the reduction algorithm to the evaluation algorithm is not investigated in this thesis and is an interesting problem on its own. ◀

Remark 5.14. The results in Figure 5.1, 5.2 and 5.3 might not be fully reliable, because of the following reasons:

- The heuristic algorithm is tested only for some cyclotomic fields. In an algorithmic sense, those fields are really nice, for example, because they have a known ring of integers.
- The heuristic algorithm is tested for ring of integers only, not for general number rings. The reason for this is that the 'escapes' when one encounters singular primes are not implemented yet – also, expanding a number ring is quite a technical job.
- The heuristic algorithm is tested only for 'quite small' fields, with degree below 100. For degrees above 100 the computation of one single power residue symbol can last for several weeks.

Further research can be done in non-cyclotomic and even non-Galois number fields; one has to include expanding of number rings in those cases. ◀

Remark 5.15. I did not test the differences between two-sided reduction, q-ary two-sided reduction with the unweighted norm, and q-ary two-sided reduction with the weighted norm thoroughly. It is possible that one of these three gives better results in larger fields. ◀

Appendices

APPENDIX A

Data

Variable	Explanation
m	Indicates that the computation takes place in the cyclotomic field $\mathbb{Q}(\zeta_m)$
n	The degree $[\mathbb{Q}(\zeta_m) : \mathbb{Q}] = \phi(m)$ of the field
coef. size	The coefficient size of α and β, written in the fixed chosen basis of $\mathbb{Z}[\zeta_m]$ (see the introduction)
inp. size	The input size, calculated by multiplying the input size and the degree
Evaluation algorithm	
loop (s)	The running time of the loop part (line 3-12) of the evaluation Algorithm 10, in seconds
calc. time (s)	The running time of the 'calculation' part (line 13-16) of the evaluation algorithm, in seconds
total time (s)	The total running time of the evaluation algorithm, in seconds
Reduction algorithm	
Umkehr calls	The number of calls to the Umkehrfaktor algorithm
'small' calls	The number of calls to the evaluation Algorithm 10, to compute small power residue symbols
time (s)	The running time of the reduction algorithm (excluded the Umkehrfaktor, and evaluation, as described in the method), in seconds

Table A.1: Explanation of the variables of Table A.2 and A.3

m	n	coef. size	inp. size	Umkehr calls	'small' calls	time (s)
15	8	20	160	29	8	0.29
15	8	40	320	61	16	0.71
15	8	80	640	125	32	1.83
15	8	160	1 280	253	64	15.2
45	24	20	480	29	8	3.89
45	24	40	960	61	16	13.8
45	24	80	1 920	125	32	39.63
45	24	160	3 840	253	64	136.08
35	24	20	480	29	8	4.2
35	24	40	960	61	16	12.97
35	24	80	1 920	125	32	44.28
35	24	160	3 840	253	64	118.73
63	36	20	720	45	12	29.8
63	36	40	1 440	93	24	93.93
63	36	80	2 880	189	48	274.28
63	36	160	5 760	389	98	812.27
99	60	20	1 200	297	75	2 246.17
99	60	40	2 400	801	201	6 191.22
99	60	80	4 800	1 589	398	16 535.12
99	60	160	9 600	3 553	889	44 812.02
105	48	20	960	85	22	151.24
105	48	40	1 920	185	47	455.03
105	48	80	3 840	381	96	1 220.06
105	48	160	7 680	777	195	3 721.93
9	6	20	120	29	8	0.29
9	6	40	240	61	16	0.45
9	6	80	480	125	32	2.31
9	6	160	960	253	64	7.9
25	20	20	400	29	8	1.52
25	20	40	800	61	16	5.46
25	20	80	1 600	125	32	15.57
25	20	160	3 200	253	64	52.45
49	42	20	840	57	15	63.98
49	42	40	1 680	113	29	214.52
49	42	80	3 360	253	64	622.94
49	42	160	6 720	505	127	1 565.5
27	18	20	360	29	8	1.09
27	18	40	720	61	16	3.3
27	18	80	1 440	125	32	11.6
27	18	160	2 880	253	64	35.16
81	54	20	1 080	97	25	242.89
81	54	40	2 160	209	53	988.68
81	54	80	4 320	449	113	2 080.81
81	54	160	8 640	905	227	6 149.33

Table A.2: Data of the charts in Figure 5.2 and 5.3, of the reduction algorithm

m	n	coef. size	inp. size	loop (s)	calc. time (s)	total time (s)
15	8	4	32	0	0.01	0.01
15	8	8	64	0.05	0	0.05
15	8	16	128	0.16	0	0.16
45	24	4	96	0.2	0.2	0.4
45	24	8	192	1.23	1.39	2.62
45	24	16	384	0.67	62.75	63.42
35	24	4	96	0.07	0.14	0.21
35	24	8	192	0.48	1.46	1.94
35	24	16	384	0.92	81.58	82.51
63	36	4	144	0.04	0.83	0.87
63	36	8	288	0.52	18.19	18.71
63	36	16	576	5.49	242.14	247.64
99	60	4	240	0.72	29	29.73
99	60	8	480	5.22	158.58	163.81
99	60	16	960	4.33	4 301.64	4 306
105	48	4	192	0.61	3.24	3.86
105	48	8	384	2.28	119.95	122.25
105	48	16	768	6.47	3 054.8	3 061.31
330	80	4	320	2.39	84.52	86.92
330	80	8	640	6.92	855.43	862.38
330	80	16	1 280	366.13	18 194.55	18 560.77
9	6	4	24	0.01	0	0.01
9	6	8	48	0.04	0	0.04
9	6	16	96	0.08	0	0.08
25	20	4	80	0.14	0	0.15
25	20	8	160	0.08	0	0.08
25	20	16	320	1.26	0	1.26
49	42	4	168	0.26	1.47	1.74
49	42	8	336	3.16	51.8	54.96
49	42	16	672	1.02	1 028.9	1 029.95
27	18	4	72	0.01	0.01	0.02
27	18	8	144	0.08	0.01	0.09
27	18	16	288	0.72	0	0.72
73	72	4	288	1.19	31.77	32.98
73	72	8	576	6.93	466.76	473.7
73	72	16	1 152	132.47	8 436.16	8 568.69
81	54	4	216	1.66	15.59	17.26
81	54	8	432	27.06	124.64	151.71
81	54	16	864	11.46	1 489.5	1 500.99
83	82	4	328	11.35	60.72	72.08
83	82	8	656	4.13	1 229.7	1 233.85
125	100	4	400	9.57	129.93	139.53
125	100	8	800	41.09	3 748.06	3 789.19
390	96	4	384	45.42	143.73	189.18
390	96	8	768	2.53	3 179.45	3 182.02

Table A.3: Data of charts in Figure 5.1, of the evaluation algorithm. Loop time refers to lines 3-12 of Algorithm 10, whereas calc. time refers to lines 13-16.

APPENDIX B

Notes

B.1 Introduction

This chapter contains two main subjects. The first subject is the QSDL-conjecture, which I invented to prove that the reduction Algorithm 9 also works well when the input vectors are small. I did not succeed in completing such a proof, and I could not give many plausible arguments why the QSDL-conjecture should be true, and there is even some mathematical evidence that it could be false. Nonetheless I include a section (section B.2) about the QSDL-conjecture, partially because my supervisor advised me to, partially because I think it might perhaps be useful in the future.

The second subject is about finding an algorithm for the power residue symbol, and mainly about how *not* to compute it. Because my research about the power residue symbol lasted about a year, I tried many other (wrong) ways to compute the power residue symbol. In order to help future researchers not to make the same mistakes as I did, I included a section about algorithms that don't work, with arguments why they do not work. This is section B.3.

B.2 The QSDL-conjecture

Conjecture B.1 (q-ary square-dense lattice)**.** *There exists a fixed $\epsilon > 0$ such that there is an algorithm that accepts as input n-dimensional square-dense q-ary ideal lattices L with standard Euclidean norm, and returns in polynomial time a vector in $\ell \in L$ with:*

$$\|\ell\|_2 \leq \Delta(L)^{\frac{2-\epsilon}{n}} = q^{1-\frac{\epsilon}{2}} \tag{B.1}$$

where $\|\cdot\|_2$ is the Euclidean length with respect to the chosen basis on L, see Notation 2.4.

As a conjecture should be plausible, I want to give some argumentation why I expect the above conjecture – in the following text abbreviated to QSDL – to be true. The first argument will cite Minkowski's second theorem, which makes a statement about how short vectors can be in lattices. The second argument shows that, without any effort, one is able to find vectors only really near de bound in (B.1). The third argument uses LLL to show that for fixed n, the QSDL-conjecture is true for all but finitely cases. For varying n, this is not the case. The last argument is about the fact that one can see a q-ary lattice as a $\mathbb{F}_q$-vector space, and that this might imply that 'reducing' in such lattices is somewhat less hard.

Minkowski's theorem

As a consequence of Minkowski's convex body theorem [MG02, p. 12], we have the following result:

Theorem B.2. *Let L be an n-dimensional lattice. Then there exists an element $\ell \in L$ that satisfies:*

$$\|\ell\|_2 \leq \sqrt{n} \cdot \Delta(L)^{1/n}$$

Applying the convex body theorem to the QSDL-conjecture, we obtain that in the q-ary lattice L there exists an element ℓ, such that $|\ell|_\infty \leq \sqrt{n} \cdot \Delta(L)^{\frac{1}{n}}$. It is good to know that there are short vectors in L, but it is well-known that there is no efficient algorithm that find such short vectors (near the Minkowski bound) in general lattices [EB81], [Ajt98].

Vectors near the QSDL-bound

The lattice in the QSDL conjecture is a q-ary and square-dense ideal lattice, which allows us to find not too large vectors: since $L_q = q\mathbb{Z}^n \subseteq L$, one can easily reduce every entry modulo q and take a representative inside $\{-\frac{q-1}{2}, \ldots, 0, \ldots, \frac{q-1}{2}\}$. This gives us a set B (not necessarily a basis) of n elements in L such that every element $\mathbf{b}_i$ satisfies:

$$\|\mathbf{b}_i\|_2 \leq \sqrt{n} \cdot \Delta(L_q)^{1/n} = \sqrt{n} \cdot \Delta(L)^{2/n}$$

So, without any effort, one is able to find multiple vectors that are only slightly bigger than the bound in equation (B.1) of Conjecture B.1.

LLL finds short vectors

Note that applying LLL to a given q-ary square dense lattice L, we have, for the uppermost basis vector $\mathbf{b}_1$, found by LLL (see Theorem 2.24)

$$\|\mathbf{b}_1\|_2 \leq \rho^{\frac{n-1}{4}} \cdot \Delta(L)^{\frac{1}{n}}$$

So, when $\rho^{\frac{n-1}{4}} \cdot \Delta(L)^{\frac{1}{n}} \leq \Delta(L)^{\frac{2-\epsilon}{n}}$, one has found a vector in λ, satisfying equation (B.1). This is precisely the case when $\rho^{\frac{n(n-1)}{4}} \leq \Delta(L)^{1-\epsilon}$.

Therefore, for fixed n, only lattices L with a discriminant less than $\rho^{\frac{n(n-1)}{4} \cdot \frac{1}{1-\epsilon}}$ might violate the QSDL-bound (B.1).

Remark B.3. This particular phenomenon, where LLL-reduction finds relatively short vectors in superexponential instances of L, but where for subexponential L finding short vectors might fail, is called the *exponential gap* of LLL. This exponential gap is the main cause why I cannot prove that the main algorithm runs in polynomial time, for varying field degree n. Also Squirrel [Squ, §V.2, §V.3] struggles with this exponential gap problem, and tries to solve it with precomputations that consists of calculating very large tables of precomputed power residue symbols, with input sizes that are exponential in the field degree n. ◀

q-ary lattices as $\mathbb{F}_q$-vector spaces

We can 'embed' the quotient L/L_q of a q-ary lattice L as a subgroup of $\mathbb{Z}^n/q\mathbb{Z}^n$, which can be seen as a $\mathbb{F}_q$-vector space; so L/L_q is then a subspace of $\mathbb{F}_q^n$. Most lattice-reduction algorithms like LLL, BKZ, Primal Dual reduction and Random Sampling reduction do not have a special subprotocol for 'finite field lattices' [NV10, Ch. 4]. Despite the fact that one cannot assume that such lattices are more simple, there is no algorithm yet that attacks the specific problem of reducing in lattices 'over finite fields'.

Remark B.4. In an article of Micciancio and Peikert [MP13], a stronger version of QSDL is proven to be hard – in the sense that one can reduce other presumably hard (worst-case) lattice problems to it. However, the lattices in this paper are q-ary, but are not ideal lattices, and also not square-dense. From [Mic16], I know that the problem of finding vectors in these specific ideal lattices satisfying (B.1) in Conjecture B.1 is still an open problem. However, it was suggested [Mic16] that, in theory, the chances that Conjecture B.1 is true, are pretty slim. However, in practice – since lattice algorithms work quite well – an algorithm based on the QSDL-conjecture might work [Mic16]. ◀

Corollary B.5. *Assuming the QSDL-conjecture, then one can effectively find – given $\alpha, \beta \in R$ coprime, and q prime – $\gamma_1, \gamma_2 \in R$ such that:*

$$\frac{\|\gamma_1\alpha + \gamma_2\beta\|_2}{q} + \|\gamma_1\|_2 < \frac{\sqrt{2n}}{q^\epsilon}\left(\max_{\sigma\in G}\left(|\sigma(\alpha)|, |\sigma(\beta)|\right) + \frac{q}{2}\right)$$

Assuming the QSDL-conjecture, one can find elements in the $2n$-dimensional lattice $L^q_{\alpha,\beta}$ with (Euclidean) length bounded by $q^{1-\epsilon}$, see (B.1). Applying this bound in the reasoning of Remark 2.42, one obtains above result.

B.3 Other attempts to compute the power residue symbol

In this section, some other methods I considered to compute the power residue symbols are described. These methods were all not feasible, and I will try to explain why.

- **(Factorization is hard)** Since Property 3.35, 3.36 and 3.37 are multiplicative, they seem to require some kind of factorization, which is believed to be hard in the rational integers [1] in its non-modular form.

[1] The presumed hardness of factoring is widely used in cryptographic protocols like RSA [RSA78], and no effective polynomial time non-quantum algorithms are known [CP05]. The best known heuristic estimation for the asymptotic running time of factoring the number n is $L_n[\frac{1}{3}, \sqrt[3]{64/9}]$, according to [Pom96] and [LL93], using the number field sieve.

Moreover, for Property 3.36, one would like factoring of ideals in the denominator, and for Property 3.35 factoring of elements in the numerator. Factoring of ideals is at least as hard as factoring in $\mathbb{Z}$, since a factorization of an ideal gives a factorization of its norm in prime powers .

Factoring an element α of a number ring as a product of more elements seems to be harder than factoring in $\mathbb{Z}$, since the straightforward way to compute such a factorization is by computing the ideal factorization of (α) into prime ideals first, and then using the class group to generate principal ideals that divide α . Note that after finding those principal ideals, one still needs to find small generators of these ideals, which is believed to be difficult problem too [Cra+15].

- **(Smooth numbers are rare)** One should remark that the factorizations needed for Property 3.35, 3.36 and 3.37 are modular, in the sense that they are modulo the denominator $\mathfrak{b}$. So, in this particular context it is enough, given $\alpha \in K$ and $\mathfrak{b}$ an ideal of $\mathcal{O}_K$, to find 'quite small' $\delta_1, \ldots, \delta_s, \eta_1, \ldots, \eta_t \in K$ but arbitrary $\gamma_1, \gamma_2 \in K$ such that

$$\gamma_1^m \delta_1 \cdots \delta_s \alpha \equiv \gamma_2^m \eta_1 \cdots \eta_t \bmod \mathfrak{b}. \tag{B.2}$$

The computation of $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$, using equation (B.2), is then as follows[2].

$$\left(\frac{\alpha}{\mathfrak{b}}\right)_m = \frac{\prod_{i=1}^t \left(\frac{\eta_i}{\mathfrak{b}}\right)_m}{\prod_{i=1}^s \left(\frac{\delta_i}{\mathfrak{b}}\right)_m}. \tag{B.3}$$

This 'modular factorization' (B.2) appears to me as being somewhat easier than a non-modular factorization. However, note that similar modular equations as in (B.2) are searched for in the so-called index calculus algorithm [Adl79], which computes discrete logarithms in the multiplicative group of the integers modulo a prime p. In the index calculus algorithm, one searches for so-called B-smooth representatives of numbers (mod p); in order to find enough such representatives to compute the discrete logarithm, B must be quite big [Pom94, §2].

I expect it to be not much different in our number field analogy, equation (B.2), in spite of the fact that we only need óne such modular equation (instead of many, as in the index calculus algorithm). Also, note that testing for smoothness in $\mathbb{Z}$ can be done by trial division or elliptic curve factorization [Len87]. In a number ring, element factorization is needed, which is only feasible (by part (i)) when really smooth[3] representatives (i.e. B must be small) can be found, hopefully yielding factorization in small elements. Because of this relation with the index calculus algorithm, I do not expect to find modular equations like (B.2) easily.

[2]

$$\begin{aligned}\prod_{i=1}^{s}\left(\frac{\delta_i}{\mathfrak{b}}\right)_m \cdot \left(\frac{\alpha}{\mathfrak{b}}\right)_m &= \left(\frac{\delta_1 \cdots \delta_s \alpha}{\mathfrak{b}}\right)_m = \left(\frac{\gamma_1^m \delta_1 \cdots \delta_s \alpha}{\mathfrak{b}}\right)_m \\ &= \left(\frac{\gamma_2^m \eta_1 \cdots \eta_t}{\mathfrak{b}}\right)_m = \left(\frac{\eta_1 \cdots \eta_t}{\mathfrak{b}}\right)_m = \prod_{i=1}^{t}\left(\frac{\eta_i}{\mathfrak{b}}\right)_m\end{aligned}$$

[3] I.e., smooth norm.

As an additional argument, I expect finding a smooth algebraic integer in the subset $\alpha + \mathfrak{b} \subseteq \mathcal{O}_K$ to be hard, because the analogous problem in $\mathbb{Z}$, finding smooth numbers in arithmetic progressions, does not give really much hope. According to [BP92], smooth numbers in arithmetic progressions are very rare.

Remark B.6. Note that, when $\mathfrak{b} = (\beta)$ in equation (B.2), one can use reciprocity in (B.3), and repeat the procedure. If modular equations as in (B.2) were easy to find, the principal power residue symbol could be computed effectively. ◀

Remark B.7. Note that the power residue symbol has a link with m-th residuosity, which is being able to decise whether α is an m-th power modulo $\mathfrak{b}$ or not. If residuosity can be computed effectively, the principal power residue can too, since one can test $\gamma\alpha \bmod \beta$ for residuosity, for varying, very small γ. After a few tries, one finds a γ_0 such that $\gamma_0\alpha$ is an m-th power modulo β. Then, one computes $\left(\frac{\alpha}{\beta}\right)_m$ by calculating $\left(\frac{\gamma_0}{\beta}\right)_m^{-1}$. This can be done by applying reciprocity, using the fact that γ_0 is very small, and thus easily factorizable. Higher residuosity is believed to be hard [AM82], and there are cryptosystems based on it [ZMI88]. ◀

APPENDIX C

Explanation of the picture on the front cover

The picture on the front page (and the cover) is generated by a program made by David Moore [Moo16], based on earlier work by Stephen J. Brooks. The picture visualizes the algebraic numbers in the complex plane, where the size of the circles decreases exponentially with the 'complexity' of the algebraic number. The complexity of an algebraic number α is defined as the sum of the absolute values of the coefficients of $f_\alpha(x)$, the minimum polynomial of α over $\mathbb{Z}$.

The colouring of the circle associated to α corresponds with the degree of the minimum polynomial of α.

Bibliography

[Adl79] L. Adleman. “A Subexponential Algorithm for the Discrete Logarithm Problem with Applications to Cryptography”. In: *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*. SFCS ’79. Washington, DC, USA: IEEE Computer Society, 1979, pp. 55–60. DOI: `10.1109/SFCS.1979.2`. URL: `http://dx.doi.org/10.1109/SFCS.1979.2`.

[AM82] L. M. Adleman and R. McDonnell. “An Application of Higher Reciprocity to Computational Number Theory (Abstract)”. In: *FOCS*. IEEE Computer Society, 1982, pp. 100–106. URL: `http://dblp.uni-trier.de/db/conf/focs/focs82.html#AdlemanM82`.

[AKS02] M. Agrawal, N. Kayal, and N. Saxena. “PRIMES is in P”. In: *Ann. of Math* 2 (2002), pp. 781–793.

[Ajt98] M. Ajtai. “The Shortest Vector Problem in L2 is NP-hard for Randomized Reductions (Extended Abstract)”. In: STOC (1998), pp. 10–19. DOI: `10.1145/276698.276705`. URL: `http://doi.acm.org/10.1145/276698.276705`.

[AW03] S. Alaca and K. S. Williams. *Introductory Algebraic Number Theory*. Cambridge Books Online. Cambridge University Press, 2003. ISBN: 9780511791260. URL: `http://dx.doi.org/10.1017/CBO9780511791260`.

[Apo98] T. Apostol. *Introduction to Analytic Number Theory*. Undergraduate Texts in Mathematics. Springer New York, 1998. ISBN: 9780387901633. URL: `https://books.google.nl/books?id=Il64dZELHEIC`.

[AH28] E. Artin and H. Hasse. “Die beiden Ergänzungssätze zum Reziprozitätsgesetz der l^n-ten Potenzreste im Körper der l^n-ten Einheitswurzeln.” German. In: *Abh. Math. Semin. Univ. Hamb.* 6 (1928), pp. 146–162. ISSN: 0025-5858; 1865-8784/e. DOI: `10.1007/BF02940607`.

[AM69] M. Atiyah and I. G. Macdonald. *Introduction to commutative algebra*. 1969.

[Bac90] E. Bach. “Explicit Bounds For Primality Testing And Related Problems”. In: *Mathematics of Computation* 55 (1990), pp. 355–380.

[BP92] A. Balog and C. Pomerance. “The distribution of smooth numbers in arithmetic progressions”. In: *Proceedings of the American mathematical society*. Vol. 11. May 1992, pp. 33–43.

[BS66] Z. Borevich and I. Shafarevich. *Number theory*. Pure and applied mathematics. Academic Press, 1966. URL: `https://books.google.nl/books/about/Number_theory.html?hl=nl&id=v_juAAAAMAAJ`.

[BCP97] W. Bosma, J. Cannon, and C. Playoust. “The Magma algebra system. I. The user language”. In: *J. Symbolic Comput.* 24.3-4 (1997). Computational algebra and number theory (London, 1993), pp. 235–265. ISSN: 0747-7171. DOI: `10.1006/jsco.1996.0125`. URL: `http://dx.doi.org/10.1006/jsco.1996.0125`.

[Bou16] J. Bouw. “On the computation of Hilbert norm residue symbols”. PhD thesis. Universiteit Leiden, 2016.

[Cas86] J. W. S. Cassels. *Local Fields*. Cambridge Books Online. Cambridge University Press, 1986. ISBN: 9781139171885. URL: `http://dx.doi.org/10.1017/CBO9781139171885`.

[Cas97] J. W. S. Cassels. *An introduction to the geometry of numbers*. Classics in mathematics. Originally published as vol. 99 of : Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen. Berlin, Heidelberg, Paris: Springer, 1997. ISBN: 3-540-61788-4. URL: `http://opac.inria.fr/record=b1099534`.

[CF67] J. Cassels and A. Fröhlich, eds. *Algebraic Number Theory*. Academic Press, 1967.

[Chi07] N. Childress. *Class Field Theory*. Universitext. Dordrecht: Springer, 2007. URL: `https://cds.cern.ch/record/1315284`.

[Chi89] A. Chistov. “The complexity of constructing the ring of integers of a global field”. In: *Soviet Math. Dokl.* 39 (1989), pp. 597–600.

[CG05] H. Cohen and G. Gras. *Class Field Theory: From Theory to Practice*. Springer Monographs in Mathematics. Springer Berlin Heidelberg, 2005. ISBN: 9783540441335. URL: `https://books.google.nl/books?id=cKHXnJ2eafsC`.

[Coh93] H. Cohen. *A Course in Computational Algebraic Number Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1993. ISBN: 0-387-55640-0.

[Coh00] H. Cohen. *Advanced topics in computational number theory*. Graduate texts in mathematics. New York, N.Y. and Berlin, Heidelberg: Springer, 2000. ISBN: 0-387-98727-4. URL: `http://opac.inria.fr/record=b1096070`.

[CL84] H. Cohen and H. W. Lenstra Jr. “Heuristics on class groups of number fields”. In: *Lecture notes in Mathematics* 1068 (1984), pp. 33–62. URL: `http://stacks.iop.org/0025-5726/13/i=3/a=A03`.

[CS99] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices, and Groups.* 3rd ed. Vol. 290. Grundlehren der mathematischen Wissenschaften. New York, NY, USA: Springer-Verlag New York, Inc., 1999. ISBN: 978-0-387-98585-5.

[Cra+15] R. Cramer et al. *Recovering Short Generators of Principal Ideals in Cyclotomic Rings.* Cryptology ePrint Archive, Report 2015/313. `http://eprint.iacr.org/`. 2015.

[CP05] R. Crandall and C. Pomerance. *Prime Numbers: A Computational Perspective.* Lecture notes in statistics. Springer, 2005. ISBN: 9780387252827. URL: `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.132.7115`.

[Dab01] M. Daberkow. "On Computations in Kummer Extensions". In: *J. Symb. Comput.* 31.1/2 (2001), pp. 113–131. DOI: `10.1006/jsco.2000.1013`. URL: `http://dx.doi.org/10.1006/jsco.2000.1013`.

[EB81] P. van Emde Boas. "Another NP-complete problem and the complexity of computing short vectors in a lattice". In: *Technical Report 81-04* (1981).

[GP01] J. von zur Gathen and D. Panario. "Factoring Polynomials Over Finite Fields: A Survey". In: *Journal of Symbolic Computation* 31.1 (2001), pp. 3 –17. ISSN: 0747-7171. DOI: `http://dx.doi.org/10.1006/jsco.1999.1002`. URL: `http://www.sciencedirect.com/science/article/pii/S0747717199910027`.

[Gro03] R. Groenewegen. "Bounds for computing the tame kernel". In: *Mathematics of computation* 73.247 (July 2003), pp. 1443–1458. ISSN: S 0025-5718(03)01592-8.

[HM90] J. L. Hafner and K. S. McCurley. "Asymptotically Fast Triangulation of Matrices over Rings". In: SODA '90 (1990), pp. 194–200. URL: `http://dl.acm.org/citation.cfm?id=320176.320198`.

[Hun03] T. Hungerford. *Algebra.* Graduate Texts in Mathematics. Springer New York, 2003. ISBN: 9780387905181. URL: `https://books.google.nl/books?id=t6N\_tOQhafoC`.

[ILC] *Ideal Lattice Challenge.* `http://latticechallenge.org/ideallattice-challenge/`. (Visited on 06/06/2016).

[IR90] K. Ireland and M. Rosen. *A Classical Introduction to Modern Number Theory.* Graduate Texts in Mathematics. Springer, 1990. ISBN: 9780387973296. URL: `https://books.google.nl/books?id=jhAXHuP2y04C`.

[Iwa68] K. Iwasawa. "On explicit formulas for the norm residue symbol". In: *J. Math. Soc. Japan* 20.1-2 (Apr. 1968), pp. 151–165. DOI: `10.2969/jmsj/02010151`. URL: `http://dx.doi.org/10.2969/jmsj/02010151`.

[JB94] H. L. J.A. Buchmann. "Approximating rings of integers in number fields". In: *Journal de Théorie des Nombres de Bordeaux* 6 (1994), pp. 221–260.

[Jan96] G. Janusz. *Algebraic Number Fields.* Second. Amer. Math. Soc., 1996.

[Koc97] H. Koch. *Algebraic Number Theory*. Springer Berlin Heidelberg, 1997. ISBN: 9783540630036. URL: https://books.google.nl/books?id=JihjOAEOldgC.

[Kos14] M. Kosters. “Calculating norm residue symbols in polynomial time”. Unpublished notes. Sept. 2014.

[Lan03] E. Landau. “Neuer Beweis des Primzahlsatzes und Beweis des Primidealsatzes”. In: *Mathematische Annalen* 56.4 (1903), pp. 645–670. ISSN: 1432-1807. DOI: 10.1007/BF01444310. URL: http://dx.doi.org/10.1007/BF01444310.

[Lan05] S. Lang. *Algebra*. Graduate Texts in Mathematics. Springer New York, 2005. ISBN: 9780387953854. URL: https://books.google.nl/books?id=Fge-BwqhqIYC.

[Lan94] S. Lang. *Algebraic Number Theory*. Applied Mathematical Sciences. Springer, 1994. ISBN: 9780387942254. URL: https://books.google.nl/books?id=u5eGtA0YalgC.

[LG87] C. Lekkerkerker and P. Gruber. *Geometry of Numbers*. North-Holland Mathematical Library. Elsevier Science, 1987. ISBN: 9780080960234. URL: https://books.google.nl/books?id=eYeWNW8mzXMC.

[Lem00] F. Lemmermeyer. *Reciprocity Laws: From Euler to Eisenstein*. Monographs in Mathematics. Springer, 2000. ISBN: 9783540669579. URL: https://books.google.nl/books?id=EwjpPeK6GpEC.

[LLL82] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász. “Factoring polynomials with rational coefficients”. In: *Math. Ann.* 261 (1982), pp. 515–534.

[LL93] A. K. Lenstra and H. W. Lenstra Jr., eds. *The development of the number field sieve*. Vol. 1554. Lecture Notes in Mathematics. Berlin: Springer-Verlag, 1993. ISBN: 3–540–57013–6.

[Len15] H. W. Lenstra Jr. personal communication. Universiteit Leiden, Oct. 2015.

[Len16] H. W. Lenstra Jr. personal communication. Radboud University Nijmegen, July 2016.

[Len80] H. W. Lenstra Jr. “Euclidean number fields 2”. In: *The Mathematical Intelligencer* 2.2 (1980), pp. 73–77. ISSN: 0343-6993. DOI: 10.1007/BF03023376. URL: http://dx.doi.org/10.1007/BF03023376.

[Len77] H. W. Lenstra Jr. “Euclidean Number Fields of Large Degree”. In: *Inventiones Mathematicae* 38 (1977), pp. 236–254. URL: http://www.math.leidenuniv.nl/~hwl/PUBLICATIONS/1977b/art.pdf.

[Len95] H. Lenstra Jr. “Computing Jacobi Symbols in ALgebraic Number Fields”. In: (1995). URL: http://www.math.leidenuniv.nl/~hwl/PUBLICATIONS/1995g/art.pdf.

[Len87] H. Lenstra Jr. “Factoring integers with elliptic curves”. In: *Annals of Mathematics* 126 (1987), pp. 649–673. URL: http://www.math.leidenuniv.nl/~hwl/PUBLICATIONS/1995g/art.pdf.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *Advances in Cryptology – EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 – June 3, 2010. Proceedings.* Ed. by H. Gilbert. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–23. ISBN: 978-3-642-13190-5. DOI: 10.1007/978-3-642-13190-5_1. URL: http://dx.doi.org/10.1007/978-3-642-13190-5_1.

[Mic16] D. Micciancio. personal communication. May 2016.

[Mic] D. Micciancio. *Minkowski's Theorem.* URL: https://cseweb.ucsd.edu/classes/sp14/cse206A-a/lec2.pdf.

[MG02] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective.* Vol. 671. The Kluwer International Series in Engineering and Computer Science. Boston, Massachusetts: Kluwer Academic Publishers, Mar. 2002.

[MP13] D. Micciancio and C. Peikert. *Hardness of SIS and LWE with Small Parameters.* Cryptology ePrint Archive, Report 2013/069. http://eprint.iacr.org/. 2013.

[MR08] D. Micciancio and O. Regev. *Lattice-based Cryptography.* 2008. URL: http://www.math.uni-bonn.de/~saxena/courses/WS2010-ref5.pdf.

[MW01] D. Micciancio and B. Warinschi. "A Linear Space Algorithm for Computing the Hermite Normal Form". In: *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation.* ISSAC '01. London, Ontario, Canada: ACM, 2001, pp. 231–236. ISBN: 1-58113-417-7. DOI: 10.1145/384101.384133. URL: http://doi.acm.org/10.1145/384101.384133.

[Mil13] J. Milne. *Class Field Theory (v4.02).* Available at www.jmilne.org/math/. 2013.

[Min10] H. Minkowski. *Geometrie der Zahlen.* Bibliotheca mathematica Teubneriana v. 2. Teubner, 1910. URL: https://books.google.nl/books?id=OGZbAAAAcAAJ.

[MV07] H. Montgomery and R. Vaughan. *Multiplicative Number Theory I: Classical Theory.* Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2007. ISBN: 9780521849036. URL: https://books.google.nl/books?id=mrMDWlRZXRsC.

[Moo16] D. Moore. *Algebraic Numbers.* http://www.mathandcode.com/programs/index.html. 2016.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms.* New York, NY, USA: Cambridge University Press, 1995. ISBN: 0-521-47465-5, 9780521474658.

[Nar04] W. Narkiewicz. *Elementary and Analytic Theory of Algebraic Numbers.* 3rd ed. Springer Monographs in Mathematics. Springer-Verlag Berlin Heidelberg, 2004. ISBN: 978-3-540-21902-6. DOI: 10.1007/978-3-662-07001-7. URL: https://books.google.co.uk/books?id=Pw4F-EVIK-oC.

[Neu99] J. Neukirch. *Algebraic Number Theory.* Trans. by N. Schappacher. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 1999. ISBN: 9783540653998. URL: `https://books.google.nl/books?id=plE3PwAACAAJ`.

[NV10] P. Q. Nguyen and B. Vallée, eds. *The LLL algorithm : survey and applications.* Information security and cryptography. Berlin, Heidelberg: Springer, 2010. ISBN: 978-3-642-02294-4. URL: `http://opac.inria.fr/record=b1130525`.

[Ove15] M. Overholt. *A Course in Analytic Number Theory.* Graduate Studies in Mathematics Series. American Mathematical Society, 2015. ISBN: 9781470417062. URL: `https://books.google.nl/books?id=m2CuoQEACAAJ`.

[PZ89] M. Pohst and H. Zassenhaus. *Algorithmic algebraic number theory.* Encyclopedia of mathematics and its applications. Réimpr. 1990, 1993, 1997. Cambridge, Cambridgeshire: Cambridge University Press, 1989. ISBN: 0-521-59669-6. URL: `http://opac.inria.fr/record=b1086848`.

[Pom96] C. Pomerance. "A tale of two sieves". In: *Notices Amer. Math. Soc* 43 (1996), pp. 1473–1485.

[Pom94] C. Pomerance. "The role of smooth numbers in number theoretic algorithms". In: *In International Congress of Mathematicians.* 1994, pp. 411–422.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public-key Cryptosystems". In: *Commun. ACM* 21.2 (Feb. 1978), pp. 120–126. ISSN: 0001-0782. DOI: `10.1145/359340.359342`. URL: `http://doi.acm.org/10.1145/359340.359342`.

[Sha50] I. Shafarevich. "A general reciprocity law." Russian. In: *Mat. Sb., Nov. Ser.* 26 (1950), pp. 113–146.

[OEIS] N. Sloane. *The Online Encyclopedia of Integer Sequences.* A055513. Class number of cyclotomic fields.

[Squ] D. Squirrel. "An algorithm for the power residue symbol". URL: `http://douglassquirrel.com/research/AnAlgorithmForThePowerResidueSymbol.pdf`.

[SS11] D. Stehlé and R. Steinfeld. "Making NTRU As Secure As Worst-case Problems over Ideal Lattices". In: *Proceedings of the 30th Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology.* EUROCRYPT'11. Tallinn, Estonia: Springer-Verlag, 2011, pp. 27–47. ISBN: 978-3-642-20464-7. URL: `http://dl.acm.org/citation.cfm?id=2008684.2008690`.

[Ste04] W. Stein. *A Brief Introduction to Classical and Adelic Algebraic Number Theory. Lecture Notes.* 2004. URL: `http://wstein.org/129/ant/ant.pdf`.

[Ste08] P. Stevenhagen. "The arithmetic of number rings". In: *Algorithmic number theory* 44 (2008). URL: `http://www.math.leidenuniv.nl/~psh/ANTproc/08psh.pdf`.

[SD01] H. Swinnerton-Dyer. *A Brief Guide to Algebraic Number Theory.* A Brief Guide to Algebraic Number Theory. Cambridge University Press, 2001. ISBN: 9780521004237.

[Mat] *The matrix determinant formula.* `http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/proof003.html`. (Visited on 06/06/2016).

[Vos79] S. V. Vostokov. "Explicit form of the law of reciprocity". In: *Mathematics of the USSR-Izvestiya* 13.3 (1979), p. 557. URL: `http://stacks.iop.org/0025-5726/13/i=3/a=A03`.

[Was12] L. Washington. *Introduction to Cyclotomic Fields.* Graduate Texts in Mathematics. Springer New York, 2012. ISBN: 9781461219347. URL: `https://books.google.nl/books?id=27zkBwAAQBAJ`.

[Wei98] E. Weiss. *Algebraic Number Theory.* Dover Books on Mathematics. Dover Publications, 1998. ISBN: 9780486401898. URL: `https://books.google.nl/books?id=qoqVhZGbFKgC`.

[ZMI88] Y. Zheng, T. Matsumoto, and H. Imai. *Cryptographic Applications of rth-Residuosity Problem with r an Odd Integer.* Yokohama National University, Japan, 1988.

[ZH80] H. Zimmer and H. Hasse. *Number Theory.* Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 1980. ISBN: 9783540082750. URL: `https://books.google.nl/books?id=i141MQAACAAJ`.