Computing with adèles and idèles

Mathé Hertogh

August 27, 2021

Master's thesis

Supervisor: Marco Streng

Mathematical Institute
Leiden University

Acknowledgements

Ik wil mijn begeleider, Marco Streng, heel hartelijk bedanken voor zijn begeleiding. Zonder zijn ideeën, suggesties en commentaar was deze scriptie bij lange na niet wat het nu is. Bovendien was het initiële onderzoeksvoorstel zijn idee. Hij heeft mij op vele vlakken geholpen, van uitleg over theorie tot het ontwerpen van algoritmen en van technische (SageMath) hulp tot schrijf-advies. Bovenal wil ik Marco bedanken voor al onze scriptie-bijeenkomsten. Ik realiseerde me tijdens die bijeenkomsten meestal weer wat bewuster hoe leuk de wiskunde is waar we mee bezig waren en ik liep vrijwel altijd weg uit die bijeenkomsten met extra veel zin in het project.

Contents

1 Introduction

In computational algebraic number theory, ideals are widely used. This is because they have good theoretical properties as well as concrete representations, on which arithmetic can be performed efficiently.

In theoretical algebraic number theory, adèles and idèles are nowadays frequently used instead of ideals. Their ability to bundle information at all primes of a number field enables them to facilitate cleaner and more coherent theorems. Think for example of class field theory. Because of their "infinite nature", it is however hard to write them down in a concrete and finite manner. Therefore adèles and idèles are a theoretical tool which are avoided in concrete computations.

In this thesis we will build the foundations of computing with adèles and idèles. In analogy to floating point numbers in a computer "representing" real numbers, we will formally define *representations of adèles and idèles*. These will be finite objects that one can explicitly write down and perform arithmetic on, for example in a computer, and they will "represent" actual adèles and idèles.

We have implemented these representations of adèles and idèles in the open source mathematical software SageMath [19]. During development we aimed for our implementation to be included into SageMath. This would for example enable students who learn for the first time about adèles and idèles to play around with them in SageMath in an easy to use, accessible manner. It would also enable other researchers to perform computations with adèles and idèles on a computer.

Furthermore, we will discuss two applications of our representation of adèles and idèles, both of which we have implemented as well. The first application is an interactive version of the profinite Fibonacci graph as discussed in [13]. This application is a bit of a toy example, as it only uses rational integral finite adèles. We think this is a good application to better understand how our adèles work for people new to the subject, for example SageMath users trying out our adèles and idèles for the first time.

The second application can be seen as a proof of concept, showing that our adèles and idèles can be used in practice for non-trivial computations. It does however require substantially more background knowledge in number theory. Our second application is the computation of Hilbert class fields of imaginary quadratic number fields using the idèlic version of Shimura's reciprocity law. Numerous articles, such as [4], [5], [22], [23], have been written on this computation. In each of them some translation is performed from the idèlic language in which the reciprocity law is stated to the language of ideals to perform their computations. We will obtain the same results, while doing the computation directly with adèles and idèles.

For our second application, we developed an algorithm that factors a matrix $M \in \mathrm{GL}_2(\widehat{\mathbb{Q}})$ as $M = BA$ for $B \in \mathrm{GL}_2(\widehat{\mathbb{Z}})$ and $A \in \mathrm{GL}_2^+(\mathbb{Q})$. We generalized this algorithm to perform the factorization $\mathrm{GL}_n(\widehat{\mathbb{Q}}) = \mathrm{GL}_n(\widehat{\mathbb{Z}})\,\mathrm{GL}_n^+(\mathbb{Q})$ for any $n \in \mathbb{Z}_{>0}$. Furthermore we exhibited an algorithm to perform the factorization $\mathrm{GSp}_{2g}(\widehat{\mathbb{Q}}) = \mathrm{GSp}_{2g}(\widehat{\mathbb{Z}})\,\mathrm{GSp}_{2g}^+(\mathbb{Q})$, for $g \in \mathbb{Z}_{>0}$ in general symplectic groups. Although we developed these adèlic matrix factorization algorithms for our second application, they can be of independent interest.

The software written for this thesis should be viewed as part of this thesis and can be found at [6].

This thesis is structured as follows. After a preliminary chapter, we introduce our representations of adèles and idèles in Chapter 3. In Chapter 4 we define conversions from and to our representations, such as how to convert an idèle representation to an adèle representation. We elaborate on our implementation of these representations in SageMath in Chapter 5. Chapter 6 is devoted to alternative representations, which we considered during the design of our representations, but which we did not pick. Our first application is discussed in Chapter 7. In Chapter 8 we develop our adèlic matrix factorization algorithms, which we use for our second application in Chapter 9.

2 Preliminaries

In this chapter we will introduce the basic concepts and notation the reader should be familiar with. The first section is concerned with mathematical background knowledge. An elaborate treatment of this material can be found in many books on algebraic number theory. The second section introduces the mathematical software package SageMath.

2.1 Mathematics

We will write $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ for respectively the ring of rational integers and the fields of rational, real and complex numbers. The upper half plane in $\mathbb{C}$ is denoted $\mathbb{H}$.

Let K be a number field. We shall write the ring of integers of K as $\mathcal{O}_K$, or $\mathcal{O}$ if the ambient field K is clear from the context. By a *finite prime of* K we shall mean a prime ideal of $\mathcal{O}$, excluding zero. Recall that the set of finite primes of K corresponds bijectively with the set of equivalence classes of non-archimedean absolute values on K. By an *infinite prime of* K we shall mean an equivalence class of archimedean absolute values on K. If r is the number of real embeddings of K and s the number of pairs of conjugated complex embeddings of K, then K has $r + s$ infinite primes. Also (r, s) is called the *signature of* K and the degree of K over $\mathbb{Q}$ is equal to $r + 2s$. By a *prime of* K will shall mean either a finite or an infinite prime of K.

Let $\mathfrak{p}$ be a finite prime of K. By $K_\mathfrak{p}$ we denote the completion of K at $\mathfrak{p}$, which we also call the *field of* $\mathfrak{p}$*-adic numbers*. We denote the normalized valuation on $K_\mathfrak{p}$ by $\mathrm{ord}_\mathfrak{p} : K_\mathfrak{p} \to \mathbb{Z} \cup \{\infty\}$ setting $\mathrm{ord}_\mathfrak{p}(0) = \infty$. We denote the corresponding valuation ring (consisting of the elements with non-negative valuation, including zero) by $\mathcal{O}_\mathfrak{p}$ and we call it the *ring of* $\mathfrak{p}$*-adic integers*. Elements in $\mathcal{O}_\mathfrak{p}$ are called *integral* elements of $K_\mathfrak{p}$. Elements of the unit group $\mathcal{O}_\mathfrak{p}^*$ of $\mathcal{O}_\mathfrak{p}$ are called $\mathfrak{p}$*-adic units* or *units at* $\mathfrak{p}$. For $n \in \mathbb{Z}_{\geq 0}$ the *n-th multiplicative subgroup* $U_\mathfrak{p}^n$ *at* $\mathfrak{p}$ is defined by

$$U_\mathfrak{p}^n = \begin{cases} \mathcal{O}_\mathfrak{p}^* & \text{if } n = 0; \\ 1 + \mathfrak{p}^n \mathcal{O}_\mathfrak{p} & \text{if } n > 0. \end{cases}$$

A basis of open neighborhoods of $1 \in K_\mathfrak{p}^*$ is given by $\{U_\mathfrak{p}^n \mid n \in \mathbb{Z}_{\geq 0}\}$. When $K = \mathbb{Q}$, we usually write $\mathbb{Z}_p$ for the ring of p-adic integers and $\mathbb{Q}_p$ for the field of p-adic numbers, where p is a prime number.

For $\mathfrak{p}$ an infinite prime of K, we also write $K_\mathfrak{p}$ for the completion of K at $\mathfrak{p}$. In this case we have $K_\mathfrak{p} \cong \mathbb{R}$ or $K_\mathfrak{p} \cong \mathbb{C}$ and we call $\mathfrak{p}$ *real* or *complex*

respectively. For ease of speech, we define $\mathcal{O}_\mathfrak{p}$ to be equal to $K_\mathfrak{p}$. One could say that we consider all elements of $K_\mathfrak{p}$ to be integral.

Let I be an index set and let for every $i \in I$ a topological space X_i and an open subset $U_i \subseteq X_i$ be given. Then the *restricted product of* $(X_i)_{i\in I}$ *with respect to* $(U_i)_{i\in I}$ is the topological space $\prod_{i\in I}(X_i; U_i)$ with set of points

$$\left\{ x \in \prod_{i\in I} X_i \;\middle|\; x_i \in U_i \text{ for all but finitely many } i \in I \right\}$$

and for which a basis of the topology consists of all sets of the form

$$\prod_{i\in S} V_i \times \prod_{i\in I\setminus S} U_i$$

with S a finite subset of I and V_i an open subset of X_i for each $i \in S$. In case the X_i are topological groups/rings and the U_i are open subgroups/rings, the restricted product is a topological group/ring under component-wise addition and/or multiplication.

We define the *adèle ring* $\mathbb{A}_K$ *of* K as the restricted product $\prod_\mathfrak{p}(K_\mathfrak{p}; \mathcal{O}_\mathfrak{p})$, where $\mathfrak{p}$ runs over all primes of K. Its elements are called K-*adèles*, or simply *adèles* if the field K is clear from the context. We define a *finite* K-*adèle* to be an element of the *ring of finite* K-*adèles* $\mathbb{A}_K^0$, which is the restricted product $\prod_\mathfrak{p}(K_\mathfrak{p}; \mathcal{O}_\mathfrak{p})$ with $\mathfrak{p}$ running over the finite primes of K only. Note that $\mathbb{A}_K$ and $\mathbb{A}_K^0$ are topological rings as well as K-algebras. The *idèle group* J_K *of* K is defined to be the topological group $\prod_\mathfrak{p}(K_\mathfrak{p}^*; \mathcal{O}_\mathfrak{p}^*)$, with $\mathfrak{p}$ running over all primes of K. Similarly as for adèles, we define the *finite idèle group* to be $\prod_\mathfrak{p}(K_\mathfrak{p}^*; \mathcal{O}_\mathfrak{p}^*)$, with $\mathfrak{p}$ only running over the finite primes of K. The elements of J_K and J_K^0 are called K-*idèles* and *finite* K-*idèles* respectively. Note that *as a group* J_K is equal to the unit group $\mathbb{A}_K^*$ of the adèle ring, but the topology on J_K is strictly finer than the subspace topology of $\mathbb{A}_K^*$ relative to $\mathbb{A}_K$. A similar statement holds for J_K^0 with respect to $\mathbb{A}_K^0$.

The *ring of profinite* K-*integers*, denoted $\widehat{\mathcal{O}}$, is the projective limit $\varprojlim_I \mathcal{O}/I$ where I runs over all non-zero ideals of $\mathcal{O}$. Note that this is a topological ring and an $\mathcal{O}$-algebra. The Chinese Remainder Theorem induces a natural isomorphism $\widehat{\mathcal{O}} \xrightarrow{\sim} \prod_\mathfrak{p} \mathcal{O}_\mathfrak{p}$ of both topological rings and $\mathcal{O}$-algebras, where $\mathfrak{p}$ runs over all finite primes of K. We define the *ring of profinite* K-*numbers*, denoted $\widehat{K}$, to be the ring of fractions of $\widehat{\mathcal{O}}$ with respect to $\mathcal{O}\setminus\{0\}$. So a profinite K-number is a fraction a/b with $a \in \widehat{\mathcal{O}}$ and $b \in \mathcal{O}\setminus\{0\}$. This makes $\widehat{K}$ into a K-algebra. We endow $\widehat{K}$ with the unique topology which restricts to the natural topology on $\widehat{\mathcal{O}}$ and that makes $\widehat{K}$ a topological ring. The isomorphism $\widehat{\mathcal{O}} \xrightarrow{\sim} \prod_\mathfrak{p} \mathcal{O}_\mathfrak{p}$ above extends naturally to an isomorphism $\widehat{K} \xrightarrow{\sim} \mathbb{A}_K^0$ of K-algebras and topological rings, with $\mathfrak{p}$ again running over the finite primes of K.

For L/K a field extension, we have natural and compatible embeddings $\widehat{\mathcal{O}}_K \to \widehat{\mathcal{O}}_L$, $\widehat{K} \to \widehat{L}$, $\mathbb{A}_K \to \mathbb{A}_L$ and $J_K \to J_L$. We naturally have $\mathbb{A}_L \cong \mathbb{A}_K \otimes_K L$ and a basis of L over K is also a basis of $\mathbb{A}_L$ over $\mathbb{A}_K$.

For R a ring and $n, m \in \mathbb{Z}_{>0}$, we denote the ring of $m \times n$-matrices with entries in R by $R^{m\times n}$. We denote the general linear group of degree n over R by $\mathrm{GL}_n(R)$, consisting of invertible $n\times n$-matrices. Whenever it makes sense for elements of R to be positive, the subgroup of matrices with positive determinant

is denoted $\mathrm{GL}_n^+(R)$. The subgroup of matrices with determinant one is denoted $\mathrm{SL}_n(R)$, i.e. the special linear group of degree n over R. For $a_1, ..., a_n \in R$ we write $\mathrm{diag}(a_1, ..., a_n)$ for the diagonal matrix A with $A_{ii} = a_i$ for $1 \leq i \leq n$.

2.2 SageMath

SageMath is a free to use, open source mathematical software package, built on top of the programming language Python [19]. It makes use of many other mathematical software packages such as Pari and Maxima, providing a unified interface for all of them.

Our adèle and idèle software is written in Python, making heavy use of the standard SageMath library. We aimed for our adèle and idèle code to be included into SageMath. We elaborate on this in Chapter 5.

Here is an example of a SageMath prompt in which a user typed commands (on the lines with the prefix "sage:"), together with the response provided by SageMath (the lines not starting with "sage:").

```
sage: R.<X> = PolynomialRing(ZZ)
sage: K.<a> = NumberField(X^2+5)
sage: b = 3*a-6; b
3*a - 6
sage: I = K.ideal(6, 2*a+2); I
Fractional ideal (6, 2*a + 2)
sage: K.ideal(b) + I # are b and I coprime?
Fractional ideal (3, a + 1)
```

As you can see, in SageMath users can create mathematical objects, such as polynomial rings, number fields and (fractional) ideals. In turn these can be used to perform computations. The code above does the following: (1) create the polynomial ring $\mathbb{Z}[X]$, naming it `R`, and create the variable `X` representing X; (2) create the number field $\mathbb{Q}[X]/(X^2+5)$, naming it `K`, and create the variable `a` representing X mod X^2+5; (3) create the element $\mathtt{b} = 3\mathtt{a} - 6$ of `K`; (4) create the fractional $\mathcal{O}_{\mathtt{K}}$-ideal `I`, generated by 6 and $2\mathtt{a}+2$; (5) compute the sum of the fractional ideal generated by `b` and `I`. The output of SageMath in the last line means that this sum is equal to the fractional $\mathcal{O}_{\mathtt{K}}$-ideal generated by 3 and $\mathtt{a}+1$. Text after a hash symbol (`#`) is interpreted as a comment and so ignored by SageMath.

Knowing only the above about SageMath is sufficient for reading this thesis. Nevertheless we recommend readers unfamiliar with SageMath to get acquainted with it quickly using the material provided on the SageMath website [19]. For example: read the quickstart, follow the tutorial, and/or follow a thematic tutorial on number fields or p-adic numbers. Number fields are heavily used in our code and the implementation of p-adic numbers expresses ideas very similar to the ideas used in our implementation of adèles and idèles.

Throughout this thesis we will show samples of SageMath code as we did above, both to give concrete examples of the concepts explained in the text and to illustrate the capabilities of our adèle and idèle software.

3 Representations

To be able to do computations with adèles and idèles, we need to be able to write them down in a finite manner. As writing down a single adèle or idèle requires an infinite amount of data in general, we will have to work with approximations. In this chapter we formalize these approximations of adèles and idèles as so-called *representations*.

During the design of these representations we usually had multiple options. We kept the following general design goals in mind when choosing between such options. We want to perform arithmetic with these representations of adèles and idèles relatively efficiently. A computation using these representations should give a result which is guaranteed to be correct. This means in practice that explicit error bounds are included in every result. These representations should enable us to perform non-trivial computations in number theory. In particular we want to be able to compute Hilbert class fields of imaginary quadratic number fields using Shimura's reciprocity law, as we discuss in Chapter 9. And lastly, we want these representations to provide SageMath users with an easy to use and intuitive way of computing with adèles and idèles. This chapter only describes our final representations. Chapter 6 is devoted to the alternatives that we came up with, but did not choose.

We implemented the representations discussed in this chapter in SageMath. Their validity and usefulness is however not restricted to SageMath: they could be implemented in other software environments as well, or be used in purely mathematical contexts. Details on our implementation in SageMath can be found in Chapter 5.

3.1 Representations of profinite integers

For this whole chapter, let K be a number field with ring of integers $\mathcal{O}$ and let Ω be a $\mathbb{Z}$-basis of $\mathcal{O}$.

Denote the degree of K over $\mathbb{Q}$ by d and write $\Omega = (\omega_1, ..., \omega_d)$. Let I be a fractional $\mathcal{O}$-ideal in K and let $B = (B_{ij})_{ij}$ be the Hermite Normal Form (HNF) of I with respect to Ω (cf. [2], section 4.7.1). Let $x \in K$ and write x uniquely as $x = \sum_{i=1}^{d} x_i \omega_i$ with $x_i \in \mathbb{Q}$. We say that x is *HNF-reduced modulo* I if $0 \leq x_i < B_{ii}$ for every $i \in \{1, ..., n\}$. Note that this notion depends on the integral basis Ω.

Given x and I as above, there exists a unique $y \in \mathcal{O}$ such that $x - y \in I$ and y is HNF-reduced modulo I. We call such a y the *HNF-reduction of x modulo I*. An efficient algorithm to compute HNF-reductions is given in [3] as Algorithm 1.4.12.

We define a *representation of profinite K-integers* to be a pair $a = (x, I)$ with $x \in \mathcal{O}$ and I an ideal of $\mathcal{O}$ such that x is HNF-reduced modulo I. We define the *represented subset of a*, denoted $\mathcal{R}(a)$, to be the subset $x + I\widehat{\mathcal{O}}$ of $\widehat{\mathcal{O}}$. Here $I\widehat{\mathcal{O}}$ denotes the ideal in $\widehat{\mathcal{O}}$ generated by I. For any $\alpha \in \mathcal{R}(a)$ we also say that *α is represented by a* and *a is a representation of α*. Note that HNF-reduction makes sure that a representation of profinite K-integers is uniquely determined by its represented subset. We call x the *value of a* and we denote it by $v(a)$. The ideal I is called the *modulus of a* and is denoted by $m(a)$. We usually denote a as $x \bmod I$. When $I = (m)$ is principal, we also sometimes

write $a = x \bmod m$. We write $\mathfrak{R}(\widehat{\mathcal{O}})$ for the set of representations of profinite K-integers.

Note that $\mathcal{R}(a)$ is a coset of the ideal $I\widehat{\mathcal{O}}$ of $\widehat{\mathcal{O}}$. It is a closed subset of $\widehat{\mathcal{O}}$ and it is open unless the modulus is zero. If the modulus is zero, then $\mathcal{R}(a)$ is the singleton containing the value of a. This enables us to represent elements of $\mathcal{O}$ exactly using representations of profinite integers.

Above we only defined $x \bmod I$ for I an $\mathcal{O}$-ideal and $x \in \mathcal{O}$ HNF-reduced modulo I. For convenience we extend this notation to arbitrary $x \in \mathcal{O}$ and I an $\mathcal{O}$-ideal: we write $x \bmod I$ for the representation of profinite K-integers with modulus I and with value the HNF-reduction of x modulo I. For example, we have $8 \bmod 6 \in \mathfrak{R}(\widehat{\mathbb{Z}})$ and its value is 2, not 8 (for $\Omega = (1)$).

Arithmetic. Let a and b be representations of profinite K-integers. We define the *sum, difference and product of a and b*, denoted $a + b$, $a - b$ and ab, to be the unique representations of K-integers with smallest represented subsets (with respect to inclusion) satisfying

$$\mathcal{R}(a) + \mathcal{R}(b) \subseteq \mathcal{R}(a+b), \quad \mathcal{R}(a) - \mathcal{R}(b) \subseteq \mathcal{R}(a-b), \quad \mathcal{R}(a)\mathcal{R}(b) \subseteq \mathcal{R}(ab).$$

Note that we do addition, subtraction and multiplication of *sets* on the left hand side of these equations, so for example

$$\mathcal{R}(a)\mathcal{R}(b) = \{\alpha\beta \mid \alpha \in \mathcal{R}(a), \beta \in \mathcal{R}(b)\}.$$

Explicitly $a + b$ can be given as

$$a + b = (v(a) + v(b)) \bmod \gcd(m(a), m(b)),$$

$a - b$ can be given as

$$a - b = (v(a) - v(b)) \bmod \gcd(m(a), m(b))$$

and ab can be given as

$$ab = v(a)v(b) \bmod \gcd(v(a)m(b), v(b)m(a), m(a)m(b)).$$

For sums and differences we actually have

$$\mathcal{R}(a) + \mathcal{R}(b) = \mathcal{R}(a+b) \quad \text{and} \quad \mathcal{R}(a) - \mathcal{R}(b) = \mathcal{R}(a-b).$$

We sometimes identify elements $\alpha \in \mathcal{O}$ with their representations $\alpha \bmod 0$. So for a representation a of profinite K-integers, we write $a+3$ for $a+(3 \bmod 0)$.

Example. Take $K = \mathbb{Q}$, $\mathcal{O} = \mathbb{Z}$ and $\Omega = (1)$. Suppose we want to use an element $\alpha \in \prod_p \mathbb{Z}_p = \widehat{\mathbb{Z}}$ in our computation which has value 7 at 2, value 9 at 5 and value 1 at all other primes. As $\alpha \notin \mathbb{Z}$, we cannot represent α exactly using a representation of profinite integers. Instead we will specify an open subset around α in $\widehat{\mathbb{Z}}$. Let us pick 300 as our modulus. We have

$$\begin{aligned} -41 &\equiv 7 \mod 4 \\ -41 &\equiv 1 \mod 3 \\ -41 &\equiv 9 \mod 25 \end{aligned}$$

and therefore $\alpha \in -41 + 300\widehat{\mathbb{Z}}$. Hence we pick $a = -41 \bmod 300$ as our representation of α. Suppose we want to compute $(\alpha+2)(\alpha-10)$. Then we can instead compute $(a+2)(a-10)$, for example with a computer. We will obtain 189 mod 900 as the result. The inclusions of represented subsets we require in our definitions of addition, subtraction and multiplication precisely guarantee that we now have $(\alpha+2)(\alpha-10) \in \mathcal{R}((a+2)(a-10))$. Hence the upshot of the computation is that we obtained $(\alpha+2)(\alpha-10) \equiv 189 \mod 900\widehat{\mathbb{Z}}$. Above, the modulus of the result was bigger than the modulus of a. When using different moduli in your computation, the modulus of the result will usually drop. For example, if $b = 1 \bmod 290$, then we have $a + b = 0 \bmod 10$.

Implementation. We have implemented representations of profinite integers as the Python class `ProfiniteInteger` and $\mathfrak{R}(\widehat{\mathcal{O}})$ as `ProfiniteIntegers`. Below we illustrate their usage. We choose to work with the $\mathbb{Z}$-basis of $\mathcal{O}$ that is computed by the method `integral_basis()` of number fields in SageMath.

```
sage: Zhat = ProfiniteIntegers(QQ)
sage: a = Zhat(-41, 300); a
259 mod 300
sage: b = Zhat(1, 290); b
1 mod 290
sage: a.value()
259
sage: a.modulus()
300
sage: (a+2)*(a-10)
189 mod 900
sage: a*a+2*a-a*10-2*10
189 mod 300
sage: a + b
0 mod 10
```

Note that the user is allowed to specify a value-modulus pair which is not HNF-reduced: our code reduces the value automatically.

Properties. Addition and multiplication of representations of profinite integers are associative and commutative. Also an additive identity element (0 mod 0) exists as well as a multiplicative identity element (1 mod 0). Additive inverses do not in general exist however; they only exist for representations with modulus zero. Also the distributive law does not hold in $\mathfrak{R}(\widehat{\mathcal{O}})$, as shown by the example above. These statements can be summarized by saying that $\mathfrak{R}(\widehat{\mathcal{O}})$ forms a commutative monoid under both addition and multiplication.

Let $\alpha \in \widehat{\mathcal{O}}$. A fundamental system of open neighborhoods of α in $\widehat{\mathcal{O}}$ is given by

$$\{\alpha + I\widehat{\mathcal{O}}; I \text{ a non-zero ideal of } \mathcal{O}\}.$$

This together with the fact that $\mathcal{O}$ lies dense in $\widehat{\mathcal{O}}$ results in the following *representability property*: for any neighborhood $U \subseteq \widehat{\mathcal{O}}$ of α, there exists a representation a of profinite integers such that $\alpha \in \mathcal{R}(a) \subseteq U$. One could say that we can approximate any $\alpha \in \widehat{\mathcal{O}}$ arbitrarily closely by representations of

profinite integers.

Let us order moduli of representations by divisibility: for $a, b \in \mathfrak{R}(\widehat{\mathcal{O}})$, we set $m(a) \leq m(b)$ if and only if $m(a) \mid m(b)$. We also refer to the modulus as the *precision* and so we can say that 2 mod 12 has higher precision than 5 mod 6, while 3 mod 16 does not have higher nor equal nor lower precision than 1 mod 17. Note that a having higher precision than b also means that $\mathcal{R}(a)$ is smaller than $\mathcal{R}(b)$ in the sense that a translate of $\mathcal{R}(a)$ is strictly included in $\mathcal{R}(b)$. In terms of precision, the representability property says that we can represent any $\alpha \in \widehat{\mathcal{O}}$ by a representation of profinite integers of arbitrarily high non-zero precision. Note that non-zero here means that we cannot in general represent α exactly.

By inspecting the explicit formulas for addition, subtraction and multiplication in $\mathfrak{R}(\widehat{\mathcal{O}})$, one sees that the following holds. For any modulus N (i.e. N an ideal of $\mathcal{O}$) and for $a, b \in \mathfrak{R}(\widehat{\mathcal{O}})$ having precision at least N, we have that $a+b$, $a-b$ and ab all have precision at least N as well. We refer to this result as the *continuity of arithmetic in* $\mathfrak{R}(\widehat{\mathcal{O}})$.

From the representability property and the continuity of arithmetic together one could informally conclude: we can perform arithmetic in $\widehat{\mathcal{O}}$ with arbitrarily high, albeit non-exact, precision using representations of profinite integers.

Design choices. Lastly we discuss an alternative design option that we considered. Instead of HNF-reduction, one could opt for LLL-reduction, cf. [3], 1.4.13. LLL-reduction usually gives smaller elements than HNF-reduction [3]. This is beneficial for performance when calculating with the representations later on, as well as that users usually like smaller elements over larger ones. The advantages of HNF-reduction over LLL-reduction are that it gives a unique representative (which LLL does not) and it is usually faster to perform HNF-reduction compared to LLL-reduction [3]. Note that this can have big impact on performance: we reduce on every arithmetic operation. Based on this, we choose to use HNF-reduction. It could however be the case that, maybe in certain applications, LLL-reduction gives much better performance. We choose not to invest too much time in this investigation. This could be part of future research though.

Our first application, the profinite Fibonacci graph, solely uses representations of profinite integers. Moreover the application itself gives insight in the ring $\widehat{\mathbb{Z}}$ in a graphical way. Hence the reader could skip ahead to Chapter 7, to see these representations in action and get more familiar with $\widehat{\mathbb{Z}}$ before continuing in this chapter.

3.2 Representations of profinite numbers

We call a pair $(n, d) \in \mathfrak{R}(\widehat{\mathcal{O}}) \times \mathbb{Z}_{>0}$ *reduced* if no prime number exists dividing d, $v(n)$ and $m(n)$. We define a *representation of profinite K-numbers* to be a reduced pair $a = (n, d) \in \mathfrak{R}(\widehat{\mathcal{O}}) \times \mathbb{Z}_{>0}$ and we usually denote a as n/d. We call n and d the *numerator* and *denominator of* a respectively, denoted $\mathrm{num}(a)$ and $\mathrm{den}(a)$. We define the *represented subset* $\mathcal{R}(a)$ *of* a to be the subset $\mathcal{R}(n)/d$ of $\widehat{K}$. Representations of profinite K-numbers are uniquely determined by their represented subsets. The set of all representations of profinite K-numbers is denoted $\mathfrak{R}(\widehat{K})$.

We define the *value of* a, denoted $v(a)$, to be $v(n)/d$ and we define the *modulus of* a, denoted $m(a)$, to be $m(n)/d$. Note that $\mathcal{R}(a)$ is a coset of $m(a)\widehat{\mathcal{O}}$ inside $\widehat{K}$. Moreover $\mathcal{R}(a)$ is a closed subset of $\widehat{K}$ and if $m(a) \neq 0$ then it is also open. In case $m(a) = 0$, we have $\mathcal{R}(a) = \{v(a)\}$. Hence we can represent any element of K exactly using a representation of profinite K-numbers. Next to writing $a = n/d$, we sometimes also write $a = v(a) \bmod m(a)$. This makes sense since a is uniquely determined by its value and modulus: a is determined by $\mathcal{R}(a)$ and we have $\mathcal{R}(a) = v(a) + m(a)\widehat{\mathcal{O}}$.

We order moduli as follows: for $a, b \in \mathfrak{R}(\widehat{K})$, we set $m(a) \leq m(b)$ if and only if $\mathrm{ord}_{\mathfrak{p}}(m(a)) \leq \mathrm{ord}_{\mathfrak{p}}(m(b))$ for all finite primes $\mathfrak{p}$ of K. We also refer to $m(a)$ as the *precision* of a.

Let $n \in \mathfrak{R}(\widehat{\mathcal{O}})$ and let $d \in \mathcal{O} \setminus \{0\}$. Then there exists a unique $a = (\tilde{n}, \tilde{d}) \in \mathfrak{R}(\widehat{K})$ such that $\mathcal{R}(a) = \mathcal{R}(n)/d$. We call a the *reduction of* (n, d) and it can be computed as follows. Let g be the $\mathcal{O}$-ideal $d\mathcal{O} + v(n)\mathcal{O} + m(n)$. Then $\tilde{d}$ is the smallest positive integer in the ideal d/g. We have $x := \tilde{d}v(n)/d \in \mathcal{O}$ and $I := \tilde{d}m(n)/d$ is an ideal of $\mathcal{O}$. Now $\tilde{n}$ is given by $x \bmod I$. Note that this computation ensures that $(\tilde{n}, \tilde{d})$ is reduced. We denote the reduction of (n, d) by n/d. This is consistent with denoting $b \in \mathfrak{R}(\widehat{K})$ as $\mathrm{num}(b)/\mathrm{den}(b)$: the reduction of $b \in \mathfrak{R}(\widehat{K})$ is just b itself. For example, we have $c = (20 \bmod 35)/10 \in \mathfrak{R}(\widehat{\mathbb{Q}})$ and the denominator of c is 2 (not 10): $c = (20 \bmod 35)/10 = (4 \bmod 7, 2) = 2 \bmod 7/2$.

Similarly for $x \in K$ and I a (possibly zero) fractional $\mathcal{O}$-ideal in K, we write $x \bmod I$ for the unique representation of profinite K-numbers with represented subset $x + I\widehat{\mathcal{O}}$. It may be the case that $v(x \bmod I) \neq x$.

We have a canonical map from $\mathfrak{R}(\widehat{\mathcal{O}})$ to $\mathfrak{R}(\widehat{K})$ sending n to $n/1$. We also have a natural map from K to $\mathfrak{R}(\widehat{K})$ sending x to $x \bmod 0$. We will use these maps to view both $\mathfrak{R}(\widehat{\mathcal{O}})$ and K as subsets of $\mathfrak{R}(\widehat{K})$. Note that this is consistent with our embedding of $\mathcal{O}$ into $\mathfrak{R}(\widehat{\mathcal{O}})$ in the sense that the diagram

$$\begin{array}{ccc} \mathcal{O} & \longrightarrow & K \\ \downarrow & & \downarrow \\ \mathfrak{R}(\widehat{\mathcal{O}}) & \longrightarrow & \mathfrak{R}(\widehat{K}) \end{array}$$

with natural maps commutes.

Arithmetic. Let $a, b \in \mathfrak{R}(\widehat{K})$. We define the *sum, difference and product of* a *and* b to be the unique representations of profinite K-numbers $a + b$, $a - b$ and ab with the smallest represented subsets satisfying

$$\mathcal{R}(a) + \mathcal{R}(b) \subseteq \mathcal{R}(a + b), \quad \mathcal{R}(a) - \mathcal{R}(b) \subseteq \mathcal{R}(a - b), \quad \mathcal{R}(a)\mathcal{R}(b) \subseteq \mathcal{R}(ab).$$

Explicitly the sum can be given as

$$a + b = \frac{\mathrm{den}(b)\,\mathrm{num}(a) + \mathrm{den}(a)\,\mathrm{num}(b)}{\mathrm{den}(a)\,\mathrm{den}(b)},$$

the difference as

$$a - b = \frac{\mathrm{den}(b)\,\mathrm{num}(a) - \mathrm{den}(a)\,\mathrm{num}(b)}{\mathrm{den}(a)\,\mathrm{den}(b)}$$

and the product as

$$ab = \frac{\mathrm{num}(a)\,\mathrm{num}(b)}{\mathrm{den}(a)\,\mathrm{den}(b)}.$$

This actually results in the equalities

$$\mathcal{R}(a) + \mathcal{R}(b) = \mathcal{R}(a+b), \quad \mathcal{R}(a) - \mathcal{R}(b) = \mathcal{R}(a-b)$$

holding. We *only* define the *quotient* a/b *of* a *by* b in case $m(b) = 0$ and $v(b) \neq 0$. In that case, we define a/b to be $v(b)^{-1}a$. Note that inverting $v(b)$ happens inside K^* and we make use of the multiplication defined above via the embedding of K into $\mathfrak{R}(\widehat{K})$. Now a/b satisfies $\mathcal{R}(a)/\mathcal{R}(b) = \mathcal{R}(a/b)$.

We emphasize that we do not define a/b if $v(b) = 0$ or $m(b) \neq 0$. Let us explain why. If $v(b) = 0$, then b represents zero and so division by b does not make much sense. The best option might be to give back a representation c with $\mathcal{R}(c) = \widehat{K}$, but no such representation exists. If $m(b) \neq 0$, then there does not exist any $c \in \mathfrak{R}(\widehat{K})$ such that $\mathcal{R}(a)/\mathcal{R}(b) \subseteq \mathcal{R}(c)$, unless $\mathcal{R}(a) = \{0\}$. This can be seen as follows. For every $c \in \mathfrak{R}(\widehat{K})$, the set $\mathcal{R}(c)$ has *bounded denominators*: there exists $N \in \mathbb{Z}_{>0}$ such that $\mathcal{R}(c) \subseteq \widehat{\mathbb{Z}}/N$. If $m(b) \neq 0$, then $\mathcal{R}(b)$ has *unbounded size*: for every $M \in \mathbb{Z}_{>0}$ there exists $\gamma \in \mathcal{R}(c) \cap \mathbb{Z}$ such that $\gamma > M$. Hence if $\mathcal{R}(a) \neq \{0\}$, the set $\mathcal{R}(a)/\mathcal{R}(b)$ will not have bounded denominators and so cannot be contained in the represented subset of any $c \in \mathfrak{R}(\widehat{K})$.

Implementation. We implemented representations of profinite K-numbers as the Python class `ProfiniteNumber` and $\mathfrak{R}(\widehat{K})$ as `ProfiniteNumbers`. Below we show examples of their usage.

```
sage: R.<X> = PolynomialRing(ZZ)
sage: K.<a> = NumberField(X^2 + 5)
sage: Ohat = ProfiniteIntegers(K)
sage: Khat = ProfiniteNumbers(K)
sage: n = Ohat(2*a, K.ideal(4, 2*a+2))
sage: b = Khat(n, 6); b # create n/6 in Khat
1/3*a mod (2, a + 1)
sage: b.numerator(), b.denominator() # are reduced
(a mod (6, 3*a + 3), 3)
sage: # Initializing using a value/modulus pair:
sage: c = Khat(1, K.ideal(2, a+1)/3); c
1/3*a mod (2/3, 1/3*a + 1/3)
sage: b-c
0 mod (2/3, 1/3*a + 1/3)
sage: b*c
1/9*a mod (2/9, 1/9*a + 1/9)
```

Properties. For $\mathfrak{R}(\widehat{K})$ similar properties hold as for $\mathfrak{R}(\widehat{\mathcal{O}})$. Namely, $\mathfrak{R}(\widehat{K})$ forms a commutative monoid under both addition and multiplication, while the ring axioms of distributivity and existence of additive inverses do not hold. Also the *representability property* holds for $\mathfrak{R}(\widehat{K})$: for any $\alpha \in \widehat{K}$ and any neighborhood $U \subseteq \widehat{K}$ of α, there exists $a \in \mathfrak{R}(\widehat{K})$ such that $\alpha \in \mathcal{R}(a) \subseteq U$.

Lastly we state the following result. Let $\alpha, \beta \in \widehat{K}$ and let Q be any precision (i.e. a fractional $\mathcal{O}$-ideal in K). Then there exist $a, b \in \mathfrak{R}(\widehat{K})$ representing α

and β respectively such that $a+b$, $a-b$ and ab have precision at least Q. A similar statement holds for division if $\beta \in K^*$. We call this result the *continuity of arithmetic in* $\mathfrak{R}(\widehat{K})$.

Design choices. We defined a representation of profinite numbers as a numerator-denominator pair. We already mentioned that such a representation is also uniquely determined by its value-modulus pair and we could use this as our definition instead. The current definition is more similar to our definition of $\widehat{K}$, as a ring of fractions of $\widehat{\mathcal{O}}$. We accept both formats from a SageMath user and we default to printing the representations as value-modulus pairs in SageMath. We store the representations as denominator-value pairs in SageMath.

3.3 Representations of real and complex numbers

Let $\mathbb{F}_{\mathbb{R}}$ be a finite subset of $\mathbb{R}$ containing 0 and 1. Let $\mathbb{F}_s$ be a finite set of symbols, including $-\infty$ and ∞. Define $\mathbb{F}$ to be the disjoint union $\mathbb{F}_{\mathbb{R}} \sqcup \mathbb{F}_s$, which we call our *set of machine representable reals*. An important example of such an $\mathbb{F}$ is the set of representable floating-point numbers in a computer.

We assume there is a total order on $\mathbb{R} \sqcup \mathbb{F}_s$ extending the usual order on $\mathbb{R}$, such that for all $\alpha \in \mathbb{R}$, the inequalities $-\infty < \alpha$ and $\alpha < \infty$ hold. Then for $x, y \in \mathbb{F}$, we define the *interval with endpoints x and y* to be

$$[x, y] = \{\alpha \in \mathbb{R} \mid x \leq \alpha \leq y\}.$$

Let S be a subset of $\mathbb{R}$. We define the $\mathbb{F}$-lower bound of S to be

$$\underline{S} = \max\{z \in \mathbb{F} \mid \forall s \in S : z \leq s\}.$$

When using the notation $\underline{S}$ the set $\mathbb{F}$ will be clear from the context. We also define the $\mathbb{F}$-upper bound of S to be

$$\overline{S} = \min\{z \in \mathbb{F} \mid \forall s \in S : s \leq z\}.$$

Note that the finiteness of $\mathbb{F}$ and the existence of $-\infty$ and ∞ ensure that $\underline{S}$ and $\overline{S}$ always exist. Lastly, we define the $\mathbb{F}$-*enclosure of* S to be the pair $(\underline{S}, \overline{S}) \in \mathbb{F}^2$.

For S and T two subsets of $\mathbb{R}$, we define their sum, difference, product and quotient to be

$$S \circ T = \{s \circ t \mid s \in S, t \in T\}, \quad \circ \in \{+, -, \cdot, /\},$$

except when $\circ$ is $/$ and $0 \in T$, in which case we define S/T to be $\mathbb{R}$.

Now we define a *representation of real numbers* to be a pair $a = (x, y) \in \mathbb{F} \times \mathbb{F}$ such that $[x, y] \neq \emptyset$. We define the *represented subset of* a to be the interval $[x, y]$, which we also denote by $\mathcal{R}(a)$. Note that a representation of real numbers may not be uniquely determined by its represented subset: for example $\mathcal{R}((0,0)) = \mathcal{R}((0,\varepsilon))$ if $\varepsilon \in \mathbb{F}_s$ satisfies $0 < \varepsilon < \alpha$ for all $\alpha \in \mathbb{R}_{>0}$. We call x and y the *left and right endpoints of* a respectively. The set of representations of real numbers is denoted $\mathfrak{R}(\mathbb{R})$. For $a, b \in \mathfrak{R}(\mathbb{R})$ and $\circ \in \{+, -, \cdot, /\}$, we define $a \circ b$ to be the $\mathbb{F}$-enclosure of $\mathcal{R}(a) \circ \mathcal{R}(b)$ and call it the *sum, difference, product or quotient of a and b*, depending on $\circ$.

Next up are the complex numbers. A *representation of complex numbers* is defined to be a pair of representations $c = (a, b)$ of real numbers. We usually write $c = a + bi$. We define the *represented subset of* c, denoted $\mathcal{R}(c)$, to be the subset

$$\mathcal{R}(a) + \mathcal{R}(b)i = \{\alpha + \beta i \mid \alpha \in \mathcal{R}(a), \beta \in \mathcal{R}(b)\}$$

of $\mathbb{C}$. We call a the *real part of* c and denote it by $\mathrm{Re}(c)$. Similarly we call b the *imaginary part of* c, denoted $\mathrm{Im}(c)$. We denote by $\mathfrak{R}(\mathbb{C})$ the set of representations of complex numbers.

For a subset S of $\mathbb{C}$, we let the *real part* $\mathrm{Re}(S)$ and the *imaginary part* $\mathrm{Im}(S)$ *of* S be defined by

$$\mathrm{Im}(S) = \{\mathrm{Im}(s) \mid s \in S\}, \quad \mathrm{Re}(S) = \{\mathrm{Re}(s) \mid s \in S\}.$$

respectively. Let $c, d \in \mathfrak{R}(\mathbb{C})$ and let $\circ \in \{+, -, \cdot, /\}$. Then we define $c \circ d$ (called the *sum, difference, product or quotient of* c *and* d depending on $\circ$) to be the representation of complex numbers whose real and imaginary part are the $\mathbb{F}$-enclosure of the real and imaginary part of $\mathcal{R}(c) \circ \mathcal{R}(d)$ respectively.

Implementation. As far as we know, the multiple precision interval arithmetic library MPFI [16] implements exactly $\mathfrak{R}(\mathbb{R})$ and $\mathfrak{R}(\mathbb{C})$ (and more) with $\mathbb{F}_s = \{-\infty, \infty\}$ for certain $\mathbb{F}_\mathbb{R}$. We could however not verify this statement via for example documentation of MPFI. SageMath provides the functionality of MPFI via the classes `RealIntervalField` and `ComplexIntervalField`. Below we show an example usage of `RealIntervalField`.

```
sage: R = RealIntervalField(prec=40); R
Real Interval Field with 40 bits of precision
sage: a = R(3.1, 3.2)
sage: b = R(1.0, 2.0)
sage: (a * b).str(style='brackets')
'[3.0999999999985 .. 6.4000000000015]'
sage: (a - 2*b).str(style='brackets')
'[-0.90000000000146 .. 1.2000000000008]'
sage: (b/(a - 2*b)).str(style='brackets')
'[-infinity .. +infinity]'
```

Properties. We have a natural map $\mathbb{R} \to \mathfrak{R}(\mathbb{R})$ sending an element x to the enclosure of $\{x\}$. Also we have a natural map $\mathbb{C} \to \mathfrak{R}(\mathbb{C})$ sending y to the pair of enclosures of $\{\mathrm{Re}(x)\}$ and $\{\mathrm{Im}(x)\}$. Note that 0 and 1 can be represented exactly in $\mathfrak{R}(\mathbb{R})$ and $\mathfrak{R}(\mathbb{C})$, as $0, 1 \in \mathbb{F}_\mathbb{R}$. Hence there exist additive and multiplicative identity elements in $\mathfrak{R}(\mathbb{R})$ and $\mathfrak{R}(\mathbb{C})$ and we also denote them by 0 and 1. Addition and multiplication are commutative, but no other ring axioms hold in general in $\mathfrak{R}(\mathbb{R})$ nor $\mathfrak{R}(\mathbb{C})$.

In contrast to representations of profinite integers/numbers, no analogue of the representability property holds in $\mathfrak{R}(\mathbb{R})$ or $\mathfrak{R}(\mathbb{C})$. Our choice of the finite set $\mathbb{F}$ of machine representable reals induces a maximum precision one can work with in $\mathfrak{R}(\mathbb{R})$ and $\mathfrak{R}(\mathbb{C})$. See below for an explanation of this choice.

Design choices. As mentioned, our choice to pick a finite set of machine representable reals induces a maximum precision one can work with in $\mathfrak{R}(\mathbb{R})$

and $\mathfrak{R}(\mathbb{C})$. An alternative could for example be to choose $\mathbb{F}_{\mathbb{R}} = \mathbb{Q}$. In this thesis we focus on computing with adèles and idèles. Part of that is computing with real and complex numbers, but this has been an area of research for decades. Therefore, we chose to use the existing software package MPFI to handle our real/complex arithmetic, instead of creating our own. Moreover, as we shall see in Chapters 7 and 9, for our applications we essentially only use finite adèles and idèles. So for our applications our choice of representations of real and complex numbers is irrelevant.

3.4 Representations of adèles

For the rest of this chapter, let a total order on the infinite primes of K be given, such that any real prime is smaller than any complex prime. We also fix an isomorphism $\varphi_{\mathfrak{p}} : K_{\mathfrak{p}} \xrightarrow{\sim} \mathbb{C}$ of topological rings for each complex prime $\mathfrak{p}$ of K, for the rest of this chapter.

In our adèle SageMath software the choices above are given by `K.places()`, where `K` is a number field. This returns the set of infinite primes of `K` as an ordered list of embeddings of `K` into $\mathbb{C}$ (so a choice of embedding is made for the complex primes). In SageMath a number field `K` is always given together with an explicit generator α of `K` over $\mathbb{Q}$. The embeddings $\phi : \mathtt{K} \to \mathbb{C}$ returned by `K.places()` are precisely those satisfying $\mathrm{Im}(\phi(\alpha)) \geq 0$. They are sorted based on the complex numbers $\phi(\alpha)$ as described in the SageMath documentation of `sort_complex_numbers_for_display()`: "First come the real numbers (with zero imaginary part), then the complex numbers sorted according to their real part. If two complex numbers have the same real part, then they are sorted according to their imaginary part." Here the natural order on $\mathbb{R}$ is used.

Note that for each real prime $\mathfrak{p}$ of K there exists a unique isomorphism $\varphi_{\mathfrak{p}} : K_{\mathfrak{p}} \xrightarrow{\sim} \mathbb{R}$ of topological rings. Let (r, s) be the signature of K and write the infinite primes of K as $v_1, ..., v_{r+s}$ in ascending order. Write φ_0 for the natural isomorphism $\mathbb{A}_K^0 \xrightarrow{\sim} \widehat{K}$. Now we define $\varphi = (\prod_{i=1}^{r+s} \varphi_{v_i}) \times \varphi_0$. This makes φ into an isomorphism of topological rings as well and its domain is equal to $\mathbb{A}_K$. Using φ we identify $\mathbb{A}_K$ with $(\prod_{i=1}^{r} \mathbb{R}) \times (\prod_{j=1}^{s} \mathbb{C}) \times \widehat{K}$.

We define a *representation of K-adèles* to be a pair $a = (x, y)$ where x is a tuple $(x_1, ..., x_{r+s})$ such that $x_i \in \mathfrak{R}(\mathbb{R})$ if $1 \leq i \leq r$ and $x_i \in \mathfrak{R}(\mathbb{C})$ if $r < i \leq r + s$ and y is a representation of profinite K-numbers. We call x the *infinite part of a* and denote it by a_∞. The representation y is also denoted by a_0 and is called the *finite part of a*. For v an infinite prime, we also write a_v for y_i, where $i \in \{1, ..., r + s\}$ satisfies $v = v_i$. We define the *represented subset of a*, denoted $\mathcal{R}(a)$, to be the subset

$$\left(\prod_{i=1}^{r+s} \mathcal{R}(a_{v_i})\right) \times \mathcal{R}(a_0)$$

of $\mathbb{A}_K$. We define $\mathfrak{R}(\mathbb{A}_K)$ to be the set of representations of K-adèles.

Arithmetic. Sums, differences, products and quotients of representations of adèles are defined component-wise. So for example for a and b two representations of K-adèles, we set $a + b$ to be $((a_{v_i} + b_{v_i})_{i=1}^{r+s}, a_0 + b_0)$. We end up, as before, with $a + b$ having the property that its represented subset is minimal while containing $\mathcal{R}(a) + \mathcal{R}(b)$. Similar properties hold for the other operations.

Note that division is only defined when the finite part of b has zero modulus and non-zero value.

Implementation. We implemented these representations in the class `Adele` and $\mathfrak{R}(\mathbb{A}_K)$ as `Adeles`. Let us demonstrate their usage.

```
sage: A = Adeles(QQ); A
Adele Ring of Rational Field
sage: a = A(9.7, Qhat(1/2, 12/5)); a
(9.6999999999999993?, 1/2 mod 12/5)
sage: b = A(5.3, Qhat(4, 12)); b
(5.2999999999999999?, 4 mod 12)
sage: a + b
(14.999999999999999?, 21/10 mod 12/5)
sage: c = A(2.0, 1/2)
sage: a/c
(4.8499999999999997?, 1 mod 24/5)
```

Properties. The theoretical properties of $\mathfrak{R}(\mathbb{A}_K)$ are completely determined by those of $\mathfrak{R}(\mathbb{R})$, $\mathfrak{R}(\mathbb{C})$ and $\mathfrak{R}(\widehat{K})$. The only properties holding in each of them, and hence in $\mathfrak{R}(\mathbb{A}_K)$, are that addition and multiplication are commutative. When only considering the infinite or finite parts of representations of adèles, more can be said of course, see Sections 3.2 and 3.3.

3.5 Representations of multiplicative $\mathfrak{p}$-adics

Let $\mathfrak{p}$ be a finite prime of the number field K. Recall that $\mathcal{O}_\mathfrak{p}$ denotes the ring of $\mathfrak{p}$-adic integers and that for $n \in \mathbb{Z}_{\geq 0}$ the *n-th multiplicative subgroup at $\mathfrak{p}$* is given by

$$U_\mathfrak{p}^n = \begin{cases} \mathcal{O}_\mathfrak{p}^* & \text{if } n = 0; \\ 1 + \mathfrak{p}^n \mathcal{O}_\mathfrak{p} & \text{if } n > 0, \end{cases}$$

which is an open subgroup of $K_\mathfrak{p}^*$.

We define a *representation of multiplicative $\mathfrak{p}$-adics* to be a pair $a = (x, n) \in K^* \times \mathbb{Z}_{\geq 0}$ such that x is HNF-reduced modulo $\mathfrak{p}^{\max(1,n)+\mathrm{ord}_\mathfrak{p}(x)}$ (cf. Section 3.1). The open and closed subset $xU_\mathfrak{p}^n$ of $K_\mathfrak{p}^*$ is called the *represented subset of a* and denoted by $\mathcal{R}(a)$. We refer to x as the *center of a*, denoted $c(a)$, and we call n the *precision of a*, denoted $p(a)$. All $\alpha \in \mathcal{R}(a)$ have the same valuation at $\mathfrak{p}$ and we call this unique valuation the *valuation of a*, denoted $\mathrm{ord}_\mathfrak{p}(a)$. Note that a is *not* uniquely determined by its represented subset: if $n = 0$, there exist $N(\mathfrak{p})-1$ representations b of multiplicative $\mathfrak{p}$-adics such that $\mathcal{R}(a) = \mathcal{R}(b)$, where $N(\mathfrak{p})$ denotes the norm of $\mathfrak{p}$. The set of representations of multiplicative $\mathfrak{p}$-adics is denoted $\mathfrak{R}(K_\mathfrak{p}^*)$.

For any $(x, n) \in K^* \times \mathbb{Z}_{\geq 0}$ we define the *representation of multiplicative $\mathfrak{p}$-adics associated to* (x, n) to be the $a \in \mathfrak{R}(K_\mathfrak{p}^*)$ with precision n and with center the HNF-reduction of x modulo $\mathfrak{p}^{\max(1,n)+\mathrm{ord}_\mathfrak{p}(x)}$.

Arithmetic. Let $a, b \in \mathfrak{R}(K_\mathfrak{p}^*)$. We define the *product ab of a and b* to be the representation of multiplicative $\mathfrak{p}$-adics associated to

$$(c(a)c(b), \min(p(a), p(b))).$$

We define the *inverse* a^{-1} *of* a to be the representation of multiplicative $\mathfrak{p}$-adics associated to

$$(c(a)^{-1}, p(a)).$$

These definitions ensure $\mathcal{R}(a)\mathcal{R}(b) = \mathcal{R}(ab)$ and $\mathcal{R}(a)^{-1} = \mathcal{R}(a^{-1})$ as $U_{\mathfrak{p}}^m$ is a subgroup of $U_{\mathfrak{p}}^n$ whenever $m \geq n$. We define division by setting a/b to be ab^{-1}.

Implementation. These representations of multiplicative $\mathfrak{p}$-adics are implemented in the class `MultiplicativePAdic` and the set $\mathfrak{R}(K_{\mathfrak{p}}^*)$ in the class `MultiplicativePAdics`. Below we show an example of their usage.

```
sage: R.<X> = PolynomialRing(ZZ)
sage: K.<a> = NumberField(X^3-2)
sage: p3 = K.prime_above(3)
sage: M = MultiplicativePAdics(K, p3); M
Group of multiplicative (3, a + 1)-adics of Number
   Field in a with defining polynomial X^3 - 2
sage: u = M(a^2, 10); u
a^2 * U(10)
sage: v = M(-a^10, 6); v # note the reduction
a * U(6)
sage: u*v
2 * U(6)
sage: v/u
1/2*a^2 * U(6)
```

Properties. The fact that the groups $U_{\mathfrak{p}}^n$ for $n \in \mathbb{Z}_{\geq 0}$ form a basis of open neighborhoods of 1 in $K_{\mathfrak{p}}^*$ together with the fact that K^* lies dense in $K_{\mathfrak{p}}^*$ gives us the following *representability property*: for any $\alpha \in K_{\mathfrak{p}}^*$ and for any neighborhood $U \subseteq K_{\mathfrak{p}}^*$ of α, there exists $a \in \mathfrak{R}(K_{\mathfrak{p}}^*)$ such that $\alpha \in \mathcal{R}(a) \subseteq U$.

Multiplication in $\mathfrak{R}(K_{\mathfrak{p}}^*)$ is associative and commutative. No identity element exists however, so $\mathfrak{R}(K_{\mathfrak{p}}^*)$ is not a group. One should also realize that the inverse a^{-1} of a representation a of multiplicative $\mathfrak{p}$-adics is not an inverse in the sense of a group. What we do have is $1 \in \mathcal{R}(aa^{-1})$.

3.6 Representations of idèles

Recall the choices made at the start of Section 3.4. Let (r, s) be the signature of K and denote the infinite primes of K in ascending order by $v_1, ...v_{r+s}$. The unique isomorphisms $\varphi_{v_i} : K_{\mathfrak{p}} \xrightarrow{\sim} \mathbb{R}$ for real primes v_i of K restrict to isomorphisms $\varphi_{v_i}^* : K_{\mathfrak{p}}^* \xrightarrow{\sim} \mathbb{R}^*$ of topological groups. Similarly the chosen isomorphisms $\varphi_{v_j} : K_{\mathfrak{p}} \xrightarrow{\sim} \mathbb{C}$ for complex primes v_j restrict to isomorphisms $\varphi_{v_j}^* : K_{\mathfrak{p}}^* \xrightarrow{\sim} \mathbb{C}^*$ of topological groups. We define $\psi = (\prod_{i=1}^{r+s} \varphi_{v_i}^*) \times \mathrm{Id}_{J_K^0}$, which is an isomorphism of topological groups as well. We will use ψ to view J_K as $(\prod_{i=1}^{r} \mathbb{R}^*) \times (\prod_{j=1}^{s} \mathbb{C}^*) \times J_K^0$.

We define $\mathfrak{R}(\mathbb{R}^*)$ and $\mathfrak{R}(\mathbb{C}^*)$ to be the subsets of $\mathfrak{R}(\mathbb{R})$ and $\mathfrak{R}(\mathbb{C})$ respectively consisting of representations whose represented subset is not equal to $\{0\}$. Then $\mathfrak{R}(\mathbb{R}^*)$ and $\mathfrak{R}(\mathbb{C}^*)$ are closed under multiplication and division. Denote the set of finite primes of K by $\mathcal{P}$. We define

$$\mathcal{F} = \left\{ f : \mathcal{Q} \to K \times \mathbb{Z}_{\geq 0} \;\middle|\; \begin{array}{l} \mathcal{Q} \text{ is a finite subset of } \mathcal{P} \text{ and} \\ f(\mathfrak{p}) \in \mathfrak{R}(K_{\mathfrak{p}}^*) \text{ for each } \mathfrak{p} \in \mathcal{Q} \end{array} \right\}.$$

Now we define a *representation of K-idèles* to be a pair $a = (x, f)$ such that $x \in \prod_{i=1}^{r} \mathfrak{R}(\mathbb{R}^*) \times \prod_{j=1}^{s} \mathfrak{R}(\mathbb{C}^*)$ and $f \in \mathcal{F} \sqcup K^*$ (where $\sqcup$ denotes disjoint union). We define the *represented subset* $\mathcal{R}(a)$ *of* a by

$$\mathcal{R}(a) = \begin{cases} \left(\prod_{i=1}^{r+s} (\mathcal{R}(x_i) \setminus \{0\})\right) \times \left(\prod_{\mathfrak{p} \in \mathcal{Q}} \mathcal{R}(f(\mathfrak{p}))\right) \times \left(\prod_{\mathfrak{p} \in \mathcal{P} \setminus \mathcal{Q}} \mathcal{O}_{\mathfrak{p}}^*\right) & \text{if } f \in \mathcal{F}; \\ \left(\prod_{i=1}^{r+s} (\mathcal{R}(x_i) \setminus \{0\})\right) \times \left(\prod_{\mathfrak{p} \in \mathcal{P}} \{f\}\right) & \text{if } f \in K^*, \end{cases}$$

where $\mathcal{Q}$ is the domain of f if $f \in \mathcal{F}$. Note that $\mathcal{R}(a)$ is a non-empty subset of J_K. We call $x = (x_1, ..., x_{r+s})$ the *infinite part of* a and denote it by a_∞. For $i \in \{1, ..., r+s\}$ we also write a_{v_i} for x_i. We denote f by a_0 and call it the *finite part of* a. If $a_0 \in \mathcal{F}$ then we denote the domain of f by $\mathcal{P}(a)$ and call its elements the *stored primes of* a. For $\mathfrak{p} \in \mathcal{P}(a)$ we write $a_\mathfrak{p}$ for the representation $f(\mathfrak{p})$ of multiplicative $\mathfrak{p}$-adics. We extend this notation $a_\mathfrak{p}$ to all primes of K: for $\mathfrak{p} \in \mathcal{P} \setminus \mathcal{P}(a)$ we define $a_\mathfrak{p} = (1, 0) \in \mathfrak{R}(K_\mathfrak{p}^*)$. If $a_0 \in K^*$ then we say that a *has exact finite part*. In this case for every $\mathfrak{p} \in \mathcal{P}$ we set $a_\mathfrak{p} = a_0$ and $\mathcal{R}(a_\mathfrak{p}) = \{a_0\}$. This notation gives the single equation

$$\mathcal{R}(a) = \left(\prod_{i=1}^{r+s} (\mathcal{R}(a_{v_i}) \setminus \{0\})\right) \times \prod_{\mathfrak{p} \in \mathcal{P}} \mathcal{R}(a_\mathfrak{p})$$

for the represented subset of a. We write $\mathcal{R}_0(a)$ for $\prod_{\mathfrak{p} \in \mathcal{P}} \mathcal{R}(a_\mathfrak{p})$, i.e. the image of $\mathcal{R}(a)$ under the projection $J_K \to J_K^0$. For $\mathfrak{p}$ a finite prime of K, we define the *valuation* $\mathrm{ord}_\mathfrak{p}(a)$ *of* a *at* $\mathfrak{p}$ to be $\mathrm{ord}_\mathfrak{p}(a_\mathfrak{p})$. The set of representations of K-idèles is denoted $\mathfrak{R}(J_K)$.

Arithmetic. We define a commutative multiplication on $\mathfrak{R}(J_K)$ in the following manner. Let $a, b \in \mathfrak{R}(J_K)$. We define the *product* ab *of* a *and* b to be the $ab \in \mathfrak{R}(J_K)$ defined as follows. We set $(ab)_{v_i} = a_{v_i} b_{v_i}$ for $i \in \{1, ..., r+s\}$. If $a_0, b_0 \in K^*$, then we set $(ab)_0 = a_0 b_0$. If $a_0, b_0 \in \mathcal{F}$, then we define $(ab)_0 \in \mathcal{F}$ to have domain $\mathcal{P}(a) \cup \mathcal{P}(b)$ and for $\mathfrak{p} \in \mathcal{P}(a) \cup \mathcal{P}(b)$ we set $(ab)_0(\mathfrak{p}) = a_\mathfrak{p} b_\mathfrak{p}$. And lastly if $a_0 \in K^*$ and $b_0 \in \mathcal{F}$, then we define $(ab)_0 \in \mathcal{F}$ to have domain $\{\mathfrak{p} \in \mathcal{P} \mid \mathrm{ord}_\mathfrak{p}(a_0) \neq 0\} \cup \mathcal{P}(b)$ and for $\mathfrak{p} \in \mathcal{P}(ab)$ we let $(ab)_0(\mathfrak{p})$ be the representation of multiplicative $\mathfrak{p}$-adics associated to $(a_0 c(b_\mathfrak{p}), p(b_\mathfrak{p}))$. The case $a_0 \in \mathcal{F}$ and $b_0 \in K^*$ is handled by commutativity.

In $\mathfrak{R}(J_K)$ we define the *inverse* a^{-1} *of* a by setting $a^{-1} = ((a_{v_i}^{-1})_{i=1}^{r+s}, a_0^{-1})$, where for $a_0 \in \mathcal{F}$ we define $a_0^{-1} : \mathcal{P}(a) \to K^* \times \mathbb{Z}_{\geq 0}$ by $a_0^{-1}(\mathfrak{p}) = a_\mathfrak{p}^{-1}$ for $\mathfrak{p} \in \mathcal{P}(a)$.

These definitions ensure that the represented subsets of ab and a^{-1} are minimal while satisfying $\mathcal{R}(a)\mathcal{R}(b) \subseteq \mathcal{R}(ab)$ and $\mathcal{R}(a)^{-1} \subseteq \mathcal{R}(a^{-1})$. And we even have $\mathcal{R}_0(a)\mathcal{R}_0(b) = \mathcal{R}_0(ab)$ and $\mathcal{R}_0(a)^{-1} = \mathcal{R}_0(a^{-1})$. But ab and a^{-1} are not uniquely determined by these properties.

Division in $\mathfrak{R}(J_K)$ is defined by setting $a/b = ab^{-1}$.

Implementation. We have implemented $\mathfrak{R}(J_K)$ as `Ideles` and its elements as `Idele` in SageMath. The infinite primes $v_1, ..., v_{r+s}$ are called `infinity_0`, `infinity_1`, etc. in our code. Let us illustrate our `Ideles` in an example.

```
sage: R.<X> = PolynomialRing(QQ)
sage: K.<a> = NumberField(X^2+5)
sage: J = Ideles(K); J
Idele Group of Number Field in a with defining
   polynomial X^2 + 5
sage: p2, p3 = K.prime_above(2), K.prime_above(3)
sage: u = J([1+I], {p2: (a, 6), p3: (a/2-7, 1)}); u
Idele with values:
  infinity_0:   1 + 1*I
  (2, a + 1):   a * U(6)
  (3, a + 1):   3/2*a * U(1)
  other primes: 1 * U(0)
sage: u[p2]
a * U(6)
sage: u[Infinity]
1 + 1*I
sage: u[K.prime_above(97)]
1 * U(0)
sage: u.stored_primes()
[Fractional ideal (2, a + 1),
 Fractional ideal (3, a + 1)]
sage: v = J([-2], 2*a); v
Idele with values:
  infinity_0:   -2
  other primes: 2*a
sage: u.has_exact_finite_part()
True
sage: v[p3]
2*a
sage: u*v
Idele with values:
  infinity_0:   -2 - 2*I
  (2, a + 1):   6 * U(6)
  (3, a + 1):   -3*a * U(1)
  (5, a):       2*a * U(0)
  other primes: 1 * U(0)
sage: u.inverse()
Idele with values:
  infinity_0:   0.50000000000000000? -
     0.50000000000000000?*I
  (2, a + 1):   -1/5*a * U(6)
  (3, a + 1):   -2/15*a * U(1)
  other primes: 1 * U(0)
sage: 1/v
Idele with values:
  infinity_0:   -0.50000000000000000?
  other primes: -1/10*a
```

Properties. For the finite parts of representations of idèles, multiplication is associative and commutative. But as multiplication in $\mathfrak{R}(\mathbb{R})$ and $\mathfrak{R}(\mathbb{C})$ is

not associative in general, all we can say about $\mathfrak{R}(J_K)$ is that multiplication is commutative. In the same vain a representability result only holds at the finite idèles: for any $\alpha \in J_K^0$ and any neighborhood $U \subseteq J_K^0$ of α, there exists $a \in \mathfrak{R}(J_K)$ such that $\alpha \in \mathcal{R}_0(a) \subseteq U$. We call this the *finite representability property of* $\mathfrak{R}(J_K)$.

Design choices. Let us explain the choice to include the possibility of a representation having exact finite part. It originated from the desire to convert elements of K^* to representations of K-idèles. Suppose we only defined representations which do not have exact finite part. Take an element $\alpha \in K^*$. What should the image a of α in $\mathfrak{R}(J_K)$ be? It is unclear what $\mathcal{P}(a)$ should be and for each $\mathfrak{p} \in \mathcal{P}(a)$ it is unclear what the precision of $a_\mathfrak{p}$ should be. We want $\mathcal{P}(a)$ and $p(a_\mathfrak{p})$ as large as possible, but they must be finite. The second problem, choosing $p(a_\mathfrak{p})$, could be addressed in two ways. In the definition of representations of multiplicative $\mathfrak{p}$-adics we could allow ∞ as a precision. For $a \in \mathfrak{R}(K_\mathfrak{p}^*)$ with $p(a) = \infty$ we would set $\mathcal{R}(a) = \{c(a)\}$. Another option would be to only work with a set $\mathfrak{R}(K_\mathfrak{p}^*; n_{\max})$ of representations of multiplicative $\mathfrak{p}$-adics with some maximum precision $n_{\max}$. Neither of these solve the first problem though: what should $\mathcal{P}(a)$ be? No single prime can be considered to have more priority to be stored by a than another. Hence no sensible choice of $\mathcal{P}(a)$ (which must be finite) could be made. This lead us to include the option of a having exact finite part.

3.7 Recap of the choices made

In defining representations, certain choices had to be made. We summarize these choices here, for clarity.

The definitions of representations of elements of $\widehat{\mathcal{O}}$, $\widehat{K}$, $\mathbb{A}_K$, $K_\mathfrak{p}^*$ and J_K all depend on a choice of $\mathbb{Z}$-basis Ω of $\mathcal{O}$. The definitions of $\mathfrak{R}(\mathbb{R})$ and $\mathfrak{R}(\mathbb{C})$, and therefore also of $\mathfrak{R}(\mathbb{A}_K)$ and $\mathfrak{R}(J_K)$, depend on a choice of a set of machine representable reals. The definitions of $\mathfrak{R}(\mathbb{A}_K)$ and $\mathfrak{R}(J_K)$ furthermore require a choice of a total ordering of the infinite primes of K, such that any real prime is smaller than any complex prime, together with a choice of topological ring isomorphisms $K_\mathfrak{p} \xrightarrow{\sim} \mathbb{C}$ for every complex prime $\mathfrak{p}$.

In the rest of this thesis, whenever talking about any of these representations, we will assume these choices to be made. Moreover, given these choices and writing (r, s) for the signature of K, we will always view $\mathbb{A}_K$ and J_K as

$$\mathbb{A}_K = \left(\prod_{i=1}^{r} \mathbb{R}\right) \times \left(\prod_{j=1}^{s} \mathbb{C}\right) \times \widehat{K} \quad \text{and} \quad J_K = \left(\prod_{i=1}^{r} \mathbb{R}^*\right) \times \left(\prod_{j=1}^{s} \mathbb{C}^*\right) \times J_K^0$$

as described in Section 3.4 and 3.6, unless explicitly stated otherwise.

4 Conversions

In Chapter 3 we defined representations of elements of $\widehat{\mathcal{O}}_K$, $\widehat{K}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{A}_K$, $K_\mathfrak{p}^*$ and J_K, for K a number field and $\mathfrak{p}$ a finite prime of K. The underlying algebraic structures of these representations are related to each other via natural maps. Based on these natural maps, we will define so-called *conversions* in this chapter,

which are maps on the corresponding sets of representations. For example, the natural embedding $J_K \to \mathbb{A}_K$ will induce a conversion $\mathfrak{R}(J_K) \to \mathfrak{R}(\mathbb{A}_K)$.

We will also define conversions between the sets of representations mentioned above and some exact algebraic structures, namely K, (quotients of) $\mathcal{O}_K$, K^* and ray class groups of K. For ease of speech we define $\mathfrak{R}(K) = K$, call an element $x \in \mathfrak{R}(K)$ a K-representation and set the *represented subset* $\mathcal{R}(x)$ *of* x to be $\{x\}$; and we take similar definitions for the other exact algebraic structures.

Let A and B be sets for which representation sets $\mathfrak{R}(A)$ and $\mathfrak{R}(B)$ are defined. We define a *conversion from* A *to* B to be a map $f : D \to \mathfrak{R}(B)$ where $D \subseteq \mathfrak{R}(A)$. Let $\varphi : A \to B$ be a map. We say that f is *based on* φ if for all $a \in D$ we have $\varphi(\mathcal{R}(a)) \subseteq \mathcal{R}(f(a))$. If moreover $f(a)$ has minimal represented subset under this requirement, then we call f *sharp*. If equality always holds, i.e. $\varphi(\mathcal{R}(a)) = \mathcal{R}(f(a))$ for all $a \in D$, then we call f *exact*. Now let $\psi : B \to A$ be a map. Then we call f *based on* ψ if for all $a \in D$ we have $\psi^{-1}(\mathcal{R}(a)) \subseteq \mathcal{R}(f(a))$. Similar to before we call f *sharp* and *exact* if this inclusion is always minimal or an equality respectively.

In this chapter we let K be a number field with signature (r, s) and ring of integers $\mathcal{O}$.

4.1 Base embeddings

We have actually already encountered some conversions. In Sections 3.1 and 3.2 we defined natural maps making the diagram

$$\begin{array}{ccc} \mathcal{O} & \longrightarrow & K \\ \downarrow & & \downarrow \\ \mathfrak{R}(\widehat{\mathcal{O}}) & \longrightarrow & \mathfrak{R}(\widehat{K}) \end{array}$$

commute. Each of these four maps is an exact conversion based on the natural embeddings

$$\begin{array}{ccc} \mathcal{O} & \longrightarrow & K \\ \downarrow & & \downarrow \\ \widehat{\mathcal{O}} & \longrightarrow & \widehat{K}. \end{array}$$

Let $D \subseteq \mathfrak{R}(\widehat{K})$ consist of the representations with denominator equal to 1. We also have the *conversion from* $\widehat{K}$ *to* $\widehat{\mathcal{O}}$ which is the map $D \to \mathfrak{R}(\widehat{\mathcal{O}})$ that sends a representation to its numerator. This is an exact conversion based on the embedding $\widehat{\mathcal{O}} \to \widehat{K}$.

Let $\varepsilon : K \to \mathfrak{R}(\widehat{K})$ denote the conversion from K to $\widehat{K}$. For $i \in \{1, ..., r\}$ let $\pi_i : \mathbb{A}_K \to \mathfrak{R}(\mathbb{R})$ denote the composition of the projection from $\mathbb{A}_K$ to the i-th $\mathbb{R}$ in the product $\mathbb{A}_K = (\prod_{i=1}^{r} \mathbb{R}) \times (\prod_{j=1}^{s} \mathbb{C}) \times \widehat{K}$ and the natural map $\mathbb{R} \to \mathfrak{R}(\mathbb{R})$ from Section 3.3. Similarly for $j \in \{1, ..., s\}$ let $\pi_{r+j} : \mathbb{A}_K \to \mathfrak{R}(\mathbb{C})$ denote the composition of the projection from $\mathbb{A}_K$ to the j-th $\mathbb{C}$ and the natural map $\mathbb{C} \to \mathfrak{R}(\mathbb{C})$ from Section 3.3. Let $\iota : K \to \mathbb{A}_K$ be the natural embedding. We define the *conversion from* K *to* $\mathbb{A}_K$ to be the map sending $x \in K$ to the

representation $(((\pi_i \circ \iota)(x))_{i=1}^{r+s}, \varepsilon(x))$ of K-adèles. We obtain a sharp conversion based on ι.

We define the *conversion from K^* to J_K* as the map sending $x \in K^*$ to the representation $(((\pi_i \circ \iota)(x))_{i=1}^{r+s}, x)$ of K-idèles. This is a sharp conversion based on the natural embedding $K^* \to J_K$.

4.2 Quotients of $\mathcal{O}$

Let I be an ideal of $\mathcal{O}$. Then we have the natural map $\pi : \widehat{\mathcal{O}} \to \mathcal{O}/I$. Let $D \subseteq \mathfrak{R}(\widehat{\mathcal{O}})$ consist of the representations with precision at least I, that is, having modulus divisible by I. The *conversion from $\widehat{\mathcal{O}}$ to $\mathcal{O}/I$* is the map $D \to \mathcal{O}/I$ sending $a \in D$ to $v(a) + I$. This is an exact conversion based on π.

The *conversion from $\mathcal{O}/I$ to $\widehat{\mathcal{O}}$* sends $x + I \in \mathcal{O}/I$ to the representation x mod I of profinite K-integers. This is an exact conversion based on π.

Now we consider the unit group $(\mathcal{O}/I)^*$. We have the natural surjection $\varphi : \widehat{\mathcal{O}}^* \to (\mathcal{O}/I)^*$ whose kernel is $W_I = \prod_{\mathfrak{p}} U_{\mathfrak{p}}^{\mathrm{ord}_{\mathfrak{p}}(I)}$, where the product ranges over the finite primes of K and $U_{\mathfrak{p}}^n$ denotes the n-th multiplicative subgroup at $\mathfrak{p}$. Denote the set of finite primes of K by $\mathcal{P}$. For $\mathfrak{p} \in \mathcal{P}$ and $u \in \mathfrak{R}(J_K)$ define $p(u_{\mathfrak{p}})$ to be ∞ if u has exact finite part. Let

$$E = \{u \in \mathfrak{R}(J_K) \mid \mathcal{R}_0(u) \subseteq \widehat{\mathcal{O}}^* \text{ and } p(u_{\mathfrak{p}}) \geq \mathrm{ord}_{\mathfrak{p}}(I) \text{ for every } \mathfrak{p} \in \mathcal{P}\}.$$

This ensures that E consists of the $u \in \mathfrak{R}(J_K)$ such that $\varphi(\mathcal{R}_0(u))$ is a singleton. We define the *conversion from J_K to $(\mathcal{O}/I)^*$* to be the map $E \to (\mathcal{O}/I)^*$ sending $u \in E$ to the unique element in $\varphi(\mathcal{R}_0(u))$.

We define the *conversion from $(\mathcal{O}/I)^*$ to J_K* to be the map sending $x + I \in (\mathcal{O}/I)^*$ to $u \in \mathfrak{R}(J_K)$ such that on real/complex primes $\mathfrak{p}$ we have $u_{\mathfrak{p}} = (-\infty, \infty)$ and $u_{\mathfrak{p}} = ((-\infty, \infty), (-\infty, \infty))$ respectively and for $\mathfrak{p} \in \mathcal{P}(u) = \{\text{finite primes dividing } I\}$ we have that $u_{\mathfrak{p}}$ is the representation of multiplicative $\mathfrak{p}$-adics associated to $(x, \mathrm{ord}_{\mathfrak{p}}(I))$. Note that for this u we have $\mathcal{R}_0(u) = xW_I = \varphi^{-1}\{x + I\}$.

4.3 Adèles and idèles

Let us call $u \in \mathfrak{R}(J_K)$ *very integral* if either $u_0 \in \mathcal{O}$ or u does not have exact finite part and for each $\mathfrak{p} \in \mathcal{P}(u)$ we have $c(u_{\mathfrak{p}}) \in \mathcal{O}$. Note that this is stronger than u being integral in the sense of $\mathcal{R}_0(u) \subseteq \widehat{\mathcal{O}}$.

The *conversion from J_K to $\mathbb{A}_K$* is the map $f : \mathfrak{R}(J_K) \to \mathfrak{R}(\mathbb{A}_K)$ defined as follows. Let $u \in \mathfrak{R}(J_K)$. For $\mathfrak{p}$ a real/complex prime of K, let $a_{\mathfrak{p}} \in \mathfrak{R}(\mathbb{R})$ or $a_{\mathfrak{p}} \in \mathfrak{R}(\mathbb{C})$ respectively be the enclosure of $\mathcal{R}(u_{\mathfrak{p}}) \setminus \{0\}$. Write $a_\infty = (a_{\mathfrak{p}_1}, ..., a_{\mathfrak{p}_{r+s}})$ where $\mathfrak{p}_1, ..., \mathfrak{p}_{r+s}$ denote the infinite primes of K in ascending order. If u has exact finite part, then we set $f(u) = (a_\infty, \varepsilon(u_0))$, where $\varepsilon : K \to \mathfrak{R}(\widehat{K})$ denotes the conversion from K to $\widehat{K}$. Otherwise, we first assume that u is very integral. Write $\mathcal{P}$ for the set of finite primes of K and for $\mathfrak{p} \in \mathcal{P}$ set

$$n_{\mathfrak{p}} = \begin{cases} \max(1, p(u_{\mathfrak{p}})) & \text{if } \mathcal{N}(\mathfrak{p}) = 2; \\ p(u_{\mathfrak{p}}) & \text{otherwise} \end{cases}$$

and $e_{\mathfrak{p}} = n_{\mathfrak{p}} + \mathrm{ord}_{\mathfrak{p}}(c(u_{\mathfrak{p}}))$, where $\mathcal{N}$ denotes the absolute ideal norm. Note that for $\mathfrak{p} \in \mathcal{P}$, if $\mathcal{N}(\mathfrak{p}) = 2$ then $U_{\mathfrak{p}}^0 = U_{\mathfrak{p}}^1$. Hence for $\mathfrak{p} \in \mathcal{P}$ with $n_{\mathfrak{p}} > 0$ we

have $\mathcal{R}(u_\mathfrak{p}) = c(u_\mathfrak{p}) + \mathfrak{p}^{e_\mathfrak{p}}\mathcal{O}_\mathfrak{p}$. Define $\mathcal{Q} = \{\mathfrak{p} \in \mathcal{P} \mid e_\mathfrak{p} \geq 1\}$. By the Chinese Remainder Theorem, there exists $x \in \mathcal{O}$ satisfying

$$x \equiv c(u_\mathfrak{p}) \mod \mathfrak{p}^{e_\mathfrak{p}} \qquad \text{for all } \mathfrak{p} \in \mathcal{Q}.$$

Write $I = \prod_{\mathfrak{p}\in\mathcal{Q}} \mathfrak{p}^{e_\mathfrak{p}}$ and let y be the HNF-reduction of x modulo I. Then y is uniquely determined by u. Now we define $f(u) = (a_\infty, y \bmod I) \in \mathfrak{R}(\mathbb{A}_K)$. Lastly if u is not very integral, then we take some $d \in \mathbb{Z}_{>0}$ such that du is very integral and we set $f(u) = (a_\infty, f(du)_0/d)$. We claim that this yields a sharp conversion based on the natural embedding $J_K \to \mathbb{A}_K$.

Proof claim. For a very integral $u \in \mathfrak{R}(J_K)$ one can check that $f(du)_0 = d \cdot f(u)_0$ for any $d \in \mathbb{Z}_{>0}$. It follows that f is a well-defined map. We view $J_K \subseteq \mathbb{A}_K$. Our definition of enclosure ensures that the conversion is sharp at the infinite primes. Let $u \in \mathfrak{R}(J_K)$. If u has exact finite part we clearly have $\mathcal{R}(f(u)_0) = \mathcal{R}_0(u)$. Now suppose $u_0 \notin K^*$ and u is very integral. We use the notation from the definition. Let $\mathfrak{p} \in \mathcal{Q}$. The projection $\mathcal{R}_\mathfrak{p}(f(u)_0)$ of $\mathcal{R}(f(u)_0)$ to $K_\mathfrak{p}$ satisfies:

$$\mathcal{R}_\mathfrak{p}(f(u)_0) = y + I\mathcal{O}_\mathfrak{p} = x + \mathfrak{p}^{e_\mathfrak{p}}\mathcal{O}_\mathfrak{p} = c(u_\mathfrak{p}) + \mathfrak{p}^{e_\mathfrak{p}}\mathcal{O}_\mathfrak{p}.$$

So if $n_\mathfrak{p} > 0$, then $\mathcal{R}_\mathfrak{p}(f(u)_0) = \mathcal{R}(u_\mathfrak{p})$. And if $n_\mathfrak{p} = 0$, then $e_\mathfrak{p} = \mathrm{ord}_\mathfrak{p}(c(u_\mathfrak{p}))$ and so $\mathcal{R}_\mathfrak{p}(f(u)_0) = c(u_\mathfrak{p})\mathcal{O}_\mathfrak{p}$. For $\mathfrak{p} \in \mathcal{P} \setminus \mathcal{Q}$, we have $\mathcal{R}_\mathfrak{p}(f(u)_0) = \mathcal{O}_\mathfrak{p} = c(u_\mathfrak{p})\mathcal{O}_\mathfrak{p}$. Now let $\mathfrak{p} \in \mathcal{P}$ such that $n_\mathfrak{p} = 0$. Above we saw that $\mathcal{R}_\mathfrak{p}(f(u)_0) = c(u_\mathfrak{p})\mathcal{O}_\mathfrak{p}$, while $\mathcal{R}(u_\mathfrak{p}) = c(u_\mathfrak{p})\mathcal{O}_\mathfrak{p}^*$. As $\mathcal{N}(\mathfrak{p}) > 2$, the unit group $\mathcal{O}_\mathfrak{p}^* = \mathcal{O}_\mathfrak{p} \setminus \mathfrak{p}\mathcal{O}_\mathfrak{p}$ consists of more than one coset of $\mathfrak{p}\mathcal{O}_\mathfrak{p}$ in $\mathcal{O}_\mathfrak{p}$. Hence any $a \in \mathfrak{R}(\mathbb{A}_K)$ such that $\mathcal{O}_\mathfrak{p}^* \subseteq \mathcal{R}_\mathfrak{p}(a)$ satisfies $\mathrm{ord}_\mathfrak{p}(m(a)) \leq 0$. This shows that $\mathcal{R}_\mathfrak{p}(f(u)_0)$ is minimal while containing $\mathcal{R}(u_\mathfrak{p})$. This shows sharpness in the very integral case. The non-very integral case follows from the fact that for $d \in \mathbb{Z}_{>0}$ and $a \in \mathfrak{R}(\widehat{K})$ we have $\mathcal{R}(a)/d = \mathcal{R}(a/d)$. □

We will not define a conversion from A_K to J_K, because of the following obstruction. Let $a \in \mathfrak{R}(\mathbb{A}_K)$ whose finite part has non-zero modulus. Then for all but finitely many primes $\mathfrak{p}$ of K the projection of $\mathcal{R}(a)$ to $K_\mathfrak{p}$ equals $\mathcal{O}_\mathfrak{p}$. On the other hand, for any $u \in \mathfrak{R}(J_K)$ the projection of $\mathcal{R}(u)$ to $K_\mathfrak{p}^*$ is a singleton or equals $\mathcal{O}_\mathfrak{p}^*$ for all but finitely many primes $\mathfrak{p}$. Hence there does not exist any $u \in \mathfrak{R}(J_K)$ such that $J_K \cap \mathcal{R}(a) \subseteq \mathcal{R}(u)$.

4.4 Profinite rational vectors

Let $\alpha \in \mathcal{O}$ be a generator of K over $\mathbb{Q}$, i.e. $K = \mathbb{Q}(\alpha)$. Write d for the degree of K over $\mathbb{Q}$. Then $B = (1, \alpha, \alpha^2, ..., \alpha^{d-1})$ is a $\mathbb{Q}$-basis of K. Therefore B is also a $\widehat{\mathbb{Q}}$-basis of $\widehat{K}$ and so we have the isomorphism of $\widehat{\mathbb{Q}}$-vector spaces $\varphi : \widehat{K} \to \widehat{\mathbb{Q}}^d$ sending $x \in \widehat{K}$ to the unique $y = (y_0, y_1, ..., y_{d-1}) \in \widehat{\mathbb{Q}}^d$ such that $x = \sum_{i=0}^{d-1} y_i\alpha^i$.

We define the *conversion from* $\widehat{K}$ *to* $\widehat{\mathbb{Q}}^d$ to be the map $f : \mathfrak{R}(\widehat{K}) \to \mathfrak{R}(\widehat{\mathbb{Q}})^d$ defined as follows. Let $a \in \mathfrak{R}(\widehat{K})$. If $m(a)$ is zero, then set $n = 0$. Otherwise let n be the largest rational number such that $m(a) \subseteq n\mathcal{O}$. Let h denote the index of the order $\mathbb{Z}[\alpha]$ in $\mathcal{O}$. Then we define $f(a) = (\varphi(v(a))_i \bmod n/h)_{i=0}^{d-1} \in \mathfrak{R}(\widehat{\mathbb{Q}})^d$. We claim that this gives us a conversion based on φ.

Proof claim. We keep the notation introduced above. For $i \in \{0, 1, ..., d-1\}$ we define $\varphi_i : \widehat{K} \to \widehat{\mathbb{Q}}$ to be φ composed with the projection to the i-th $\widehat{\mathbb{Q}}$. What we want to show is $\varphi_i(\mathcal{R}(a)) \subseteq \mathcal{R}(f(a)_i)$ for every $i \in \{0, 1, ..., d-1\}$.

Since φ is additive we have $\varphi(\mathcal{R}(a)) = \varphi(v(a)) + \varphi(m(a)\widehat{\mathcal{O}})$. For every $i \in \{0, 1, ..., d-1\}$ we have $\mathcal{R}(f(a)_i) = \varphi_i(v(a)) + n/h\widehat{\mathbb{Z}}$. Hence it suffices to show that $\varphi_i(m(a)\widehat{\mathcal{O}}) \subseteq n/h\widehat{\mathbb{Z}}$ for every $i \in \{0, 1, ..., d-1\}$.

As groups $\mathcal{O}/\mathbb{Z}[\alpha]$ and $\widehat{\mathcal{O}}/\widehat{\mathbb{Z}}[\alpha]$ are isomorphic. So since the index of $\mathbb{Z}[\alpha]$ in $\mathcal{O}$ is h, the index of $\widehat{\mathbb{Z}}[\alpha]$ in $\widehat{\mathcal{O}}$ is also h and we have $\widehat{\mathcal{O}} \subseteq \frac{1}{h}\widehat{\mathbb{Z}}[\alpha]$. It follows that $\varphi(\widehat{\mathcal{O}}) \subseteq (\frac{1}{h}\widehat{\mathbb{Z}})^d$.

Hence by definition of n and the fact that φ is a $\mathbb{Q}$-linear map, we have

$$\varphi\left(m(a)\widehat{\mathcal{O}}\right) \subseteq \varphi\left(n\widehat{\mathcal{O}}\right) = \varphi\left(\frac{n}{h}\widehat{\mathbb{Z}}[\alpha]\right) \subseteq \frac{n}{h}\varphi\left(\widehat{\mathbb{Z}}[\alpha]\right) = \left(\frac{n}{h}\widehat{\mathbb{Z}}\right)^d.$$

This proves our claim. □

Viewing $\mathbb{A}_K$ as a free $\mathbb{A}_{\mathbb{Q}}$-algebra with basis $(1, \alpha, ...\alpha^{d-1})$ we implemented a similar conversion from $\mathbb{A}_K$ to $\mathbb{A}_{\mathbb{Q}}^d$. This uses the conversion above on finite parts and performs some linear algebra over $\mathbb{R}$ and $\mathbb{C}$ on infinite parts. For details, see the method `to_rational_vector()` of the class `Adele` at [6].

4.5 Ray class groups and idèles

For $\mathfrak{p}$ a real prime of K we define $U_{\mathfrak{p}}^0 = \mathbb{R} = K_{\mathfrak{p}}$ and $U_{\mathfrak{p}}^1 = \mathbb{R}_{>0} \subseteq K_{\mathfrak{p}}$. For a cycle $\mathfrak{f}$ of K we write $W_{\mathfrak{f}} = \prod_{\mathfrak{p}} U_{\mathfrak{p}}^{\mathrm{ord}_{\mathfrak{p}}(\mathfrak{f})}$ with $\mathfrak{p}$ running over all primes of K.

Let $\mathfrak{f}$ be a cycle of K and write $Cl_{\mathfrak{f}}$ for the ray class group of K modulo $\mathfrak{f}$. We have the canonical homomorphism $\varphi : J_K \to Cl_{\mathfrak{f}}$ with kernel $K^*W_{\mathfrak{f}}$ such that for every finite prime $\mathfrak{p}$ that does not divide $\mathfrak{f}$ and for every $\pi \in K_{\mathfrak{p}}^*$ with $\mathrm{ord}_{\mathfrak{p}}(\pi) = 1$ the K-idèle $x = (1, 1, ..., 1, \pi, 1, 1, ...)$ with $\mathrm{ord}_{\mathfrak{p}}(x) = 1$ is send to the class of $\mathfrak{p}$ in $Cl_{\mathfrak{f}}$.

Let $D \subseteq \mathfrak{R}(J_K)$ consist out of those representations u for which $\mathcal{R}(u) \subseteq yW_{\mathfrak{f}}$ for some $y \in J_K$. So for $u \in \mathfrak{R}(J_K)$ we have $u \in D$ if and only if (1) for each infinite prime $\mathfrak{p}$ dividing $\mathfrak{f}$, we have $\mathcal{R}(u_{\mathfrak{p}}) \subseteq \mathbb{R}_{>0}$ or $\mathcal{R}(u_{\mathfrak{p}}) \subseteq \mathbb{R}_{<0}$ and (2) either u has exact finite part or for each finite prime $\mathfrak{p}$ dividing $\mathfrak{f}$, we have $p(u_{\mathfrak{p}}) \geq \mathrm{ord}_{\mathfrak{p}}(\mathfrak{f})$.

We let the *conversion from J_K to $Cl_{\mathfrak{f}}$* to be the map $f : D \to Cl_{\mathfrak{f}}$ that sends $u \in D$ to the unique $\alpha \in Cl_f$ such that $\varphi(\mathcal{R}(u)) = \{\alpha\}$. This is an exact conversion based on φ. For details on how to compute such α, see the method `_from_ray_class()` of the class `Ideles` in `idele.py` at [6].

Suppose that at each finite prime $\mathfrak{p}$ of K a uniformizer $\pi_{\mathfrak{p}} \in K$ at $\mathfrak{p}$ is given. Also for each ray class $c \in Cl_{\mathfrak{f}}$ let a representative ideal I_c be given. The *conversion from $Cl_{\mathfrak{f}}$ to J_K* is the map $f : Cl_{\mathfrak{f}} \to \mathfrak{R}(J_K)$ sending $c \in Cl_{\mathfrak{f}}$ to the $u \in \mathfrak{R}(J_K)$ defined as follows. At each complex prime $\mathfrak{p}$ of K we put $u_{\mathfrak{p}} = ((-\infty, \infty), (-\infty, \infty))$. At real primes $\mathfrak{p}$ dividing $\mathfrak{f}$ we set $u_{\mathfrak{p}} = (0, \infty)$. At the other real primes $\mathfrak{p}$ we set $u_{\mathfrak{p}} = (-\infty, \infty)$. We define the set of stored primes of u by $\mathcal{P}(u) = \{\mathfrak{p} \mid \mathrm{ord}_{\mathfrak{p}}(I_c) \neq 0 \text{ or } \mathrm{ord}_{\mathfrak{p}}(\mathfrak{f}) \neq 0\}$. For $\mathfrak{p} \in \mathcal{P}(u)$ we define

$$u_{\mathfrak{p}} = \begin{cases} (1, \mathrm{ord}_{\mathfrak{p}}(\mathfrak{f})) & \text{if } \mathfrak{p} \mid \mathfrak{f}; \\ (\pi_{\mathfrak{p}}^{\mathrm{ord}_{\mathfrak{p}}(I_c)}, 0) & \text{if } \mathfrak{p} \mid I_c. \end{cases}$$

This yields a conversion satisfying $\mathcal{R}(f(c)) \subseteq \varphi^{-1}\{c\}$. Note that this conversion is not based on φ: the inclusion is the wrong way around. It can be seen as an implementation of a section of φ. We will put this conversion to good use in Chapter 9.

4.6 p-adic numbers

There is an implementation in SageMath of p-adic numbers, for (rational) prime numbers p. We will not go into the details of this implementation here, but we did implement conversions between our representations of profinite integers/numbers and the implementation of p-adic numbers in SageMath. For p a prime number we implemented exact conversions from $\widehat{\mathbb{Z}}$ to $\mathbb{Z}_p$ and from $\widehat{\mathbb{Q}}$ to $\mathbb{Q}_p$ based on the natural projections $\widehat{\mathbb{Z}} \to \mathbb{Z}_p$ and $\widehat{\mathbb{Q}} \to \mathbb{Q}_p$. For $p_1, p_2, ..., p_k$ distinct prime numbers we also implemented an exact conversion from $\prod_{i=1}^{k} \mathbb{Z}_p$ to $\widehat{\mathbb{Z}}$ based on the projection $\widehat{\mathbb{Z}} \to \prod_{i=1}^{k} \mathbb{Z}_p$. We extended that conversion to a conversion from $\widehat{\mathbb{Q}}$ to $\prod_{i=1}^{k} \mathbb{Q}_p$, although this does not result in a conversion based on the projection $\widehat{\mathbb{Q}} \to \prod_{i=1}^{k} \mathbb{Q}_p$. See files `profinite_integer.py` and `profinite_number.py` at [6] for details.

5 Implementation in SageMath

We have implemented the representations of profinite integers, profinite numbers, adèles, multiplicative $\mathfrak{p}$-adics and idèles defined in Chapter 3 in SageMath. We describe our implementation in this chapter. The source code can be found at [6].

5.1 Elements, parents and categories

SageMath implements many mathematical objects and algorithms. SageMath is built on top of the programming language Python (and many other open source packages) and uses the Object Oriented Programming design pattern. In practice this means that each mathematical object is implemented in SageMath as (an instance of) a Python class. Algorithms are in turn implemented as methods of these classes. SageMath uses Elements, Parents and Categories to organize these classes. We capitalize these words to distinguish their meaning in SageMath from their usual meaning in mathematics. A Category is a Python object modeling a (mathematical) category, e.g. the category of groups or the category of $\mathbb{Q}$-algebras. Both Elements and Parents are also Python objects, modeling (mathematical) elements of sets and sets of elements respectively. A Parent usually models a set with additional (algebraic) structure. This is encoded into SageMath by specifying that the Parent lies in certain Categories. For example: in SageMath the category of rings is implemented as `Rings()`. There is a Parent implementing the ring of integers called `ZZ` and it lies in `Rings()`. The Elements of `ZZ` are instances of the class `Integer`.

Categories provide an efficient and clearly organized way of using generic code as well. For example, computing greatest common divisors can be done in any Euclidean domain. In SageMath, a generic algorithm is implemented for computing greatest common divisors in Euclidean domains. Whenever a

Parent belongs to the category `EuclideanDomains()`, a method `gcd()` becomes available for the corresponding Elements. This does however sometimes require the implementation of other methods first. In the case of `EuclideanDomains()`, a method `quo_rem()` performing division with remainder must be implemented before `gcd()` works correctly. For more details on the Element, Parent and Category framework of SageMath, see [17].

We used this framework for implementing our representations of adèles and idèles. For example, we implemented $\mathfrak{R}(\widehat{\mathcal{O}}_K)$, for K a number field, as a Parent, namely `ProfiniteIntegers(K)`. Its Elements are instances of the class `ProfiniteInteger`, which implements representations of profinite K-integers. We put our parent in the Category `CommutativeAlgebras(O)`, for `O` the ring of integers of K. Recall from Section 3.1 that $\mathfrak{R}(\widehat{\mathcal{O}}_K)$ does *not* satisfy all ring axioms and therefore is *not* an $\mathcal{O}_K$-algebra. Although one could view this as an inconsistency, this is compliant with the intended use of Categories. For example, floating point arithmetic does not satisfy all ring axioms either, while the Parent `RealField` implementing floating point arithmetic does belong to `Rings()` in SageMath. Similarly we declared the Parents `ProfiniteNumbers(K)` and `Adeles(K)` to be in the Category `CommutativeAlgebras(K)` and we declared `Ideles(K)` to be in `CommutativeGroups()`. We made sure that all methods that these Categories expect to be implemented (like the `quo_rem()` example above), are implemented for our algebraic structures whenever this makes sense.

5.2 Implementation of representations

The implementations of our representations exactly mimic the definitions in Chapter 3. Let us give two examples to illustrate this. An instance of the class `ProfiniteInteger` has attributes `_value` and `_modulus`, which are an algebraic integer and an integral ideal respectively. An instance of the class `Idele` has attributes `_infinite` and `_finite`. Here `_infinite` is a Python list of Elements of `RealIntervalField` and `ComplexIntervalField`. And `_finite` is either a number field element or a Python dictionary with prime ideals as keys and instances of `MultiplicativePAdic` as values.

All operations defined on representations are implemented precisely as stated in Chapter 3. The formula $m(a+b) = \gcd(m(a), m(b))$ for $a, b \in \mathfrak{R}(\widehat{\mathcal{O}}_K)$ and K a number field is for example used to implement addition for the class `ProfiniteInteger`.

Also we implemented notions related to representations that were defined in Chapter 3 as methods of the corresponding classes. For example the class `Idele` has the methods `has_exact_finite_part()` and `stored_primes()`.

5.3 Conversions and coercions

In Chapter 4 we defined conversions between our representations. All of these conversions are implemented as well. For example we can do the following.

```
sage: Qhat(7/2) # convert from Q to Qhat
7/2 mod 0
sage: x = Zmod(500)(79) # image of 79 in Z/500Z
sage: a = Zhat(x); a # convert from Z/500Z to Zhat
79 mod 500
```

```
sage: a[5] # from Zhat to the 5-adic integers
4 + 3*5^2 + O(5^3)
```

In SageMath, some conversions can be declared to be a *coercion*. Coercions are used automatically by SageMath to enable arithmetic between Elements of different Parents. For example, we declared our conversion from idèles to adèles to be a coercion and hence we can do the following.

```
sage: A = Adeles(QQ)
sage: a = A(-1, Qhat(1/2, 50))
sage: J = Ideles(QQ)
sage: u = J(2.5, {2: (1, 1), 5: (-1, 2)})
sage: a + u
(1.5000000000000000?, 99/2 mod 50)
```

The representation of idèles `u` is implicitly converted (also called *coerced*) to a representation of adèles and the result is added to `a` inside $\mathfrak{R}(\mathbb{A}_{\mathbb{Q}})$.

Not every conversion may be declared to be a coercion. For example, the domain of a coercion must always be the whole Parent, hence our conversion from $\widehat{K}$ to $\widehat{\mathcal{O}}_K$, with K a number field, cannot be declared to be a coercion. For the other rules we refer to the SageMath documentation, for example Section "Coercion – the basics" of [18]. The conversions that we have declared to be coercions are the conversions from $\mathcal{O}_K$ to $\widehat{\mathcal{O}}_K$, from $\mathcal{O}_K/I$ to $\widehat{\mathcal{O}}_K$, from K to $\widehat{K}$, from $\widehat{\mathcal{O}}_K$ to $\widehat{K}$, from K to $\mathbb{A}_K$, from K^* to J_K and from J_K to $\mathbb{A}_K$, where K denotes a number field and I an $\mathcal{O}_K$-ideal.

SageMath can even construct new Parents which are appropriate for performing arithmetic.

```
sage: R.<X> = PolynomialRing(ZZ)
sage: f = X^2 - 7
sage: a = Zhat(8, 14)
sage: f+a
X^2 + 1 mod 14
sage: (f+a).parent()
Univariate Polynomial Ring in X over Profinite
    Integers of Rational Field
```

In the example above, the polynomial `f` has Parent `R` $= \mathbb{Z}[X]$ and `a` has Parent `Zhat` $= \mathfrak{R}(\widehat{\mathbb{Z}})$. Both Parents can be coerced into the Parent `S` $= \mathfrak{R}(\widehat{\mathbb{Z}})[X]$. SageMath detects this, constructs the Parent `S`, coerces `f` and `a` to `S` and then performs the addition inside `S`. SageMath is able to do this because we implemented so-called *construction functors* for our Parent classes, see Section "Coercion – the advanced parts" of [18] for details.

5.4 Equivalence of representations

We needed to implement the equality operator `==` for our classes. Consider the following example.

```
sage: Zhat = ProfiniteIntegers(QQ)
sage: Zhat(7, 9) == Zhat(7, 27)
```

Should the last line return `True` or `False`?

To determine this we needed to define a notion of *equivalence of representations*. Let K be a number field, let $\mathfrak{p}$ be a finite prime of K and let $A \in \{\widehat{\mathcal{O}}_K, \widehat{K}, \mathbb{A}_K, K_{\mathfrak{p}}^*, J_K\}$. We considered the following three different notions of equivalence in $\mathfrak{R}(A)$. Let $a, b \in \mathfrak{R}(A)$.

- We call a and b *strictly equivalent* if there exists $\alpha \in A$ such that $\mathcal{R}(a) = \{\alpha\} = \mathcal{R}(b)$, i.e. a and b are both exact and represent the same unique value.
- We call a and b *represented subset equivalent* if $\mathcal{R}(a) = \mathcal{R}(b)$, i.e. a and b represent the same subset of A.
- We call a and b *loosely equivalent* if $\mathcal{R}(a) \cap \mathcal{R}(b) \neq \emptyset$, i.e. there exists an element $\alpha \in A$ that a and b both represent.

Strict equivalence implies represented subset equivalence and represented subset equivalence implies loose equivalence. Strict and represented subset equivalence define equivalence relations, while loose equivalence fails to be transitive in general. The implementations of $\mathfrak{R}(\mathbb{R})$ and $\mathfrak{R}(\mathbb{C})$ in SageMath use strict equivalence for the `==` operator. On the other hand, the implementation of p-adic numbers in SageMath, for p a prime number, uses loose equivalence for comparison using `==`. Hence a consistent choice throughout SageMath was not possible. We did however want a single choice for all our own representations. When working with these representations during development, the question whether or not two representations *could* represent the same element came up more often than the question whether or not they certainly did represent the same element. Hence we ended up implementing loose equivalence for `==` comparison of our own representations. The other options could have been implemented as well and they might be better for certain applications. We implemented `!=` comparison as *not* being loosely equivalent, i.e., having disjoint represented subsets.

5.5 Case distinction for $\mathbb{Q}$

From a mathematical point of view, the field $\mathbb{Q}$ is a number field. Unfortunately, in SageMath this is not the case: the implementation `QQ` of $\mathbb{Q}$ does not inherit from the class `NumberField_generic` and misses many methods that number fields have in SageMath. Also, methods with the same name may have different meanings. For example when `K` is a number field in SageMath, the command `K.ideal(10/3)` gives the fractional $\mathcal{O}_K$-ideal in K generated by $10/3$, while `QQ.ideal(10/3)` gives the ideal of $\mathbb{Q}$ generated by $10/3$ (i.e. the unit ideal). Fractional ideals of $\mathbb{Q}$ are not implemented in SageMath, while we do want to use those fractional ideals.

A possible solution would be to use `NumberField(X-1)` instead of `QQ`. This gives the number field $\mathbb{Q}[X]/(X-1)$, which is canonically isomorphic to $\mathbb{Q}$. But `QQ` and its Elements do not only lack certain number field (element) methods in SageMath; they also have many methods that number field(s) (elements) do not have. Elements of `QQ` (currently) even have more methods than number field elements in SageMath. Hence we did not want to prohibit users from using `QQ` itself as a base field.

Therefore we choose to distinguish two cases in our implementation: the base field being equal to `QQ` or the base field being a SageMath number field. In many places in our code these cases are handled separately.

A user of our software will notice this case distinction in the following ways. The modulus of a `ProfiniteInteger` over `QQ` is a non-negative integer, instead of a $\mathbb{Z}$-ideal. Similarly, the modulus of a `ProfiniteNumber` over `QQ` is a non-negative rational number, as opposed to a fractional $\mathbb{Z}$-ideal in $\mathbb{Q}$. The stored primes of an `Idele` over `QQ` are prime numbers, while they are prime ideals in the general case.

Note that a user still has the option to use `NumberField(x-1)` instead of `QQ`, if he/she wants the moduli and stored primes to be ideals.

5.6 Inclusion into SageMath

We aim for our implementation of adèles and idèles to be included into SageMath.

As described in 5.1, we incorporated our implementation in the Elements, Parents and Category framework. Also we implemented various conversions and coercions between both our own representations and already existing SageMath objects, as described in 5.3. This ensures that our code fits well into the existing SageMath code base and makes it easier for the SageMath community to maintain the code in the future. It also provides the user with an interface very similar to the rest of SageMath.

We also ensured that our code adheres to the coding guidelines described in [20]. In particular, every module, class and function is well documented and doc-tested. We also followed the guidelines mentioned in the thematic tutorials [18] and [25].

Furthermore, we tried to make our implementation complete. The fact that we implemented construction functors and many abstract methods such as `ProfiniteIntegers(K).krull_dimension()` do not contribute much to the "foundations of computing with adèles and idèles" that this thesis is about. They do however improve the SageMath user experience.

We hope that the careful design of our adèle and idèle functionality, together with the coding efforts mentioned above, will lead to the inclusion of our code into SageMath in the near future.

6 Alternative representations

During the design of our adèle and idèle representations, we considered multiple options. In Chapter 3 we described the ones that we chose and implemented. This chapter is devoted to the alternative options that we did not choose. We will briefly describe the alternative designs and indicate why we implemented the representations from Chapter 3 and not these.

6.1 Real and complex numbers

The representations of real and complex numbers from Section 3.3 and their arithmetic are usually referred to as *(real/complex) intervals* and *interval arithmetic*.

The most standard representations of real/complex numbers are floating point numbers. Floating point arithmetic is not exact: errors are introduced by arithmetic operations. These errors are not tracked or stored by floating point

numbers. Hence using bare floating point numbers for representing adèles or idèles did not fit our design aim of providing explicit error margins for computational results.

Another widely used way of computing with real/complex numbers is called *ball arithmetic*. Where interval arithmetic stores pairs (x, y), usually of floating point numbers, to represent intervals $[x, y] \subseteq \mathbb{R}$, ball arithmetic stores pairs (m, r), also usually of floating point numbers, to represent balls in $\mathbb{R}$ with midpoint m and radius r. The arithmetic of balls is very similar to that of intervals. The main advantage of balls over intervals is that for high precision computations, only the midpoint needs to be stored with high precision, while the radius can be stored with low precision. In contrast, for high precision interval arithmetic, both of the endpoints need to be stored with high precision. As a result, ball arithmetic can be approximately twice as fast and requires approximately half as much space as interval arithmetic for high precision computations, cf. [7].

We choose to use interval arithmetic for the following reason. We wanted to implement conversions between idèle groups and ray class groups (cf. Section 4.5) and for this we wanted to be able to represent $\mathbb{R}_{>0}$ and $\mathbb{R}_{<0}$, which is possible with intervals, but not with balls.

6.2 $\mathfrak{p}$-adic numbers

For p a prime number, SageMath implements p-adic numbers. There is however no implementation of $\mathfrak{p}$-adic numbers in SageMath, for $\mathfrak{p}$ a finite prime of a number field. As we will describe in Section 6.3, we considered representing adèles using $\mathfrak{p}$-adic numbers. So we needed to be able to represent $\mathfrak{p}$-adic numbers themselves. Below we will show three different ideas we had on how to do this. We let K denote a number field and $\mathfrak{p}$ a finite prime of K.

We define a *value-modulus representation of $\mathfrak{p}$-adics numbers* to be a representation of profinite K-numbers whose modulus is zero or has valuation zero at every finite prime not equal to $\mathfrak{p}$. The set of value-modulus representations of $\mathfrak{p}$-adic numbers is denoted $\mathfrak{R}_{\mathrm{vm}}(K_{\mathfrak{p}})$. For $a \in \mathfrak{R}_{\mathrm{vm}}(K_{\mathfrak{p}})$ we define the *represented subset $\mathcal{R}(a)$ of a* to be the projection to $K_{\mathfrak{p}}$ of the represented subset of a considered as a representation of profinite K-numbers. Arithmetic in $\mathfrak{R}_{\mathrm{vm}}(K_{\mathfrak{p}})$ can be performed as in $\mathfrak{R}(\widehat{\mathcal{O}})$.

We did not explicitly implement $\mathfrak{R}_{\mathrm{vm}}(K_{\mathfrak{p}})$ as some sort of "`pAdics`" class. Our implementation of $\mathfrak{R}(\widehat{\mathcal{O}})$ can however be seen as an implementation of $\mathfrak{R}_{\mathrm{vm}}(K_{\mathfrak{p}})$ of course.

Our next idea is to consider $K_{\mathfrak{p}}$ as a field extension of $\mathbb{Q}_p$, where p is the (rational) prime number lying below $\mathfrak{p}$. Suppose we already have a set $\mathfrak{R}(\mathbb{Q}_p)$ of representations of p-adic numbers at our disposal, within which we can perform arithmetic and whose elements $a \in \mathfrak{R}(\mathbb{Q}_p)$ have associated represented subsets $\mathcal{R}(a) \subseteq \mathbb{Q}_p$. Our idea was to use the existing p-adic numbers of SageMath, but for clarity one may also take $\mathfrak{R}(\mathbb{Q}_p) = \mathfrak{R}_{\mathrm{vm}}(\mathbb{Q}_{(p)})$ as defined above. Let K be given as $K = \mathbb{Q}[X]/(f)$ for some irreducible $f \in \mathbb{Q}[X]$. Let g be the monic irreducible factor of f in $\mathbb{Q}_p[X]$ such that $K_{\mathfrak{p}} \cong \mathbb{Q}_p[X]/(g)$ with $(X \bmod (f)) \in K \subseteq K_{\mathfrak{p}}$ corresponding to $(X \bmod (g)) \in \mathbb{Q}_p[X]/(g)$. We give $X \bmod (g)$ a name, say α, such that $K_{\mathfrak{p}} = \mathbb{Q}_p(\alpha)$. Let $c_0, c_1, ..., c_d \in \mathfrak{R}(\mathbb{Q}_p)$ represent the

coefficients $g_0, g_1, ..., g_d \in \mathbb{Q}_p$ of g. Suppose that we can uniquely derive g from only f and $c_0, ..., c_d$. Then f and $c_0, ..., c_d$ will enable us to perform arithmetic on the following representations of $\mathfrak{p}$-adic numbers.

Note that a $\mathbb{Q}_p$-basis of $K_\mathfrak{p}$ is given by $(1, \alpha, \alpha^2, ..., \alpha^{d-1})$. We define a *$\mathbb{Q}_p$-vector representation of $\mathfrak{p}$-adic numbers* to be a tuple $a = (a_0, ..., a_{d-1})$ of representations of p-adic numbers and we define the represented subset of a to be

$$\mathcal{R}(a) = \mathcal{R}(a_0) + \alpha\mathcal{R}(a_1) + \alpha^2\mathcal{R}(a_2) + ... + \alpha^{d-1}\mathcal{R}(a_{d-1}).$$

This version of representations of $\mathfrak{p}$-adic numbers is the most similar to the already existing $\mathfrak{p}$-adic functionality of SageMath. SageMath does not have an implementation of $K_\mathfrak{p}$ in general, but only when $\mathfrak{p}$ is unramified or totally ramified. Both cases are constructed as finite extensions of $\mathbb{Q}_p$ in SageMath.

Choose a uniformizer $\pi \in K$ at $\mathfrak{p}$ (i.e. $\mathrm{ord}_\mathfrak{p}(\pi) = 1$) and choose a set of representatives $D \subseteq \mathcal{O}$ of $\mathcal{O}_\mathfrak{p}/\pi\mathcal{O}_\mathfrak{p}$ with $0 \in D$. We call D our *set of digits*. Now we have

$$K_\mathfrak{p} = \left\{ \sum_{i=k}^{\infty} d_i \pi^i \mid k \in \mathbb{Z}, d_i \in D \text{ for } i \geq k \right\}.$$

Choose some *maximum relative precision* $n \in \mathbb{Z}_{>0}$. Now we define a *power series representation of $\mathfrak{p}$-adic numbers* to be a tuple $a = (k, d_k, d_{k+1}, ..., d_{k+n-1}) \in \mathbb{Z} \times D^n$ with $d_k \neq 0$. Writing $x = \sum_{i=k}^{k+n-1} d_i \pi^i$ we define the *represented subset of a* to be $\mathcal{R}(a) = x + \mathfrak{p}^{k+n}\mathcal{O}_\mathfrak{p}$. By $\mathfrak{R}_{\mathrm{pow}}(K_\mathfrak{p})$ we denote the set of power series representations of $\mathfrak{p}$-adic numbers. The need for a choice of uniformizer π and a set of digits D is a disadvantage of these representations: no natural choice seems to be available in general. Rational p-adic numbers are by default printed as power series in p in SageMath. Therefore $\mathfrak{R}_{\mathrm{pow}}(K_\mathfrak{p})$ might be the most intuitive implementation of $K_\mathfrak{p}$ for many SageMath users.

6.3 Adèles

In this section we describe two alternative representations of adèles. We think the first one, a prime-wise representation, is a viable alternative to the representation described in Section 3.4 and so we compare them in detail. The second alternative, $\mathbb{A}_\mathbb{Q}$-vector representations, is one that we did consider, but ultimately decided to be clearly less viable than the other two. We will briefly explain why.

Let K be a number field with signature (r, s). Denote the set of finite primes of K by $\mathcal{P}$. Define

$$\mathcal{G} = \left\{ f : \mathcal{Q} \to \mathfrak{R}(\widehat{\mathcal{O}}) \,\middle|\, \begin{array}{c} \mathcal{Q} \text{ is a finite subset of } \mathcal{P} \text{ and} \\ f(\mathfrak{p}) \in \mathfrak{R}_{\mathrm{vm}}(K_\mathfrak{p}) \text{ for each } \mathfrak{p} \in \mathcal{Q} \end{array} \right\}.$$

We define a *prime-wise representation of K-adèles* to be a pair $a = (x, f)$ with $x \in (\prod_{i=1}^{r} \mathfrak{R}(R)) \times (\prod_{j=1}^{s} \mathfrak{R}(\mathbb{C}))$ and $f \in K \sqcup \mathcal{G}$.

If $f \in \mathcal{G}$, we define the *set of stored primes of a* to be the domain of f, which we denote by $\mathcal{P}(a)$. Now we define the *represented subset of a* to be

$$\mathcal{R}(a) = \begin{cases} \left(\prod_{i=1}^{r+s} \mathcal{R}(x_i)\right) \times \left(\prod_{\mathfrak{p} \in \mathcal{P}(a)} \mathcal{R}(f(\mathfrak{p}))\right) \times \left(\prod_{\mathfrak{p} \in \mathcal{P} \setminus \mathcal{P}(a)} \mathcal{O}_\mathfrak{p}\right) & \text{if } f \in \mathcal{G}; \\ \left(\prod_{i=1}^{r+s} \mathcal{R}(x_i)\right) \times \left(\prod_{\mathfrak{p} \in \mathcal{P}} \{f\}\right) & \text{if } f \in K. \end{cases}$$

Note that we have $\mathcal{R}(a) \subseteq \mathbb{A}_K = (\prod_{i=1}^{r} \mathbb{R}) \times (\prod_{j=1}^{s} \mathbb{C}) \times \mathbb{A}_K^0$. We call x and f the *finite* and *infinite part of* a respectively. We denote the set of prime-wise representations of K-adèles by $\mathfrak{R}_{\mathrm{pw}}(\mathbb{A}_K)$.

For $a, b \in \mathfrak{R}_{\mathrm{pw}}(\mathbb{A}_K)$ the *sum* $a + b$ can be defined in such a way that $a + b$ has minimal represented subset satisfying $\mathcal{R}(a) + \mathcal{R}(b) \subseteq \mathcal{R}(a + b)$. Similar statements hold for $a - b$, ab and a/b, where in the last case b is assumed to have finite part lying in K^*.

The same representability in $\mathbb{A}_K$ is achieved by $\mathfrak{R}_{\mathrm{pw}}(\mathbb{A}_K)$ and $\mathfrak{R}(\mathbb{A}_K)$, in the sense that a subset of $\mathbb{A}_K$ is of the form $\mathcal{R}(a)$ for some $a \in \mathfrak{R}_{\mathrm{pw}}(\mathbb{A}_K)$ if and only if it is of the form $\mathcal{R}(b)$ for some $b \in \mathfrak{R}(\mathbb{A}_K)$. Hence from a mathematical point of view, these two representations could be called equivalent. From a practical point of view there are however advantages of each representation over the other, which we will discuss below. For this discussion, let us call elements of $\mathfrak{R}(\mathbb{A}_K)$ *bundled representations*, to clearly distinguish them from prime-wise representations.

Implementing bundled representations can be done with less code than prime-wise representations. For prime-wise representations computations must be performed for each prime individually and quite some case distinctions need to be made in the code, to handle exact finite parts correctly. No explicit case distinctions need to be made when implementing bundled representations, as they can be given a finite part with modulus zero. Formulas such as $m(a+b) = \gcd(m(a), m(b))$ for $a, b \in \mathfrak{R}(\widehat{\mathcal{O}})$ hold in general (both for zero and non-zero moduli) and they can be used to perform the computation for all finite primes at once.

When doing arithmetic, both in $\mathfrak{R}(\mathbb{A}_K)$ and $\mathfrak{R}_{\mathrm{pw}}(\mathbb{A}_K)$, we use integer arithmetic and in particular integer multiplication. The best known integer multiplication algorithm has worse than linear time complexity. Hence performing k multiplications of n-digit integers is faster than doing a single multiplication of kn-digit integers for $k \in \mathbb{Z}_{>1}$ and large $n \in \mathbb{Z}$. Suppose we have $a \in \mathfrak{R}(\mathbb{A}_K)$ and $b \in \mathfrak{R}_{\mathrm{pw}}(\mathbb{A}_K)$ such that $\mathcal{R}(a) = \mathcal{R}(b)$ and $k := \#\mathcal{P}(b) > 1$. Computing with b requires approximately k times as many operations as computing with a. But each single such operation is performed on smaller numbers for b: on average approximately k times smaller in bit size. The total number of bits needed to store a and b is roughly the same. Hence if this total number of bits is large, then we expect arithmetic with b to be faster than arithmetic with a. We summarize this by saying that we expect arithmetic in $\mathfrak{R}_{\mathrm{pw}}(\mathbb{A}_K)$ to be asymptotically faster than arithmetic in $\mathfrak{R}(\mathbb{A}_K)$.

On the other hand, in the situation above, if the number of bits needed to store a and b is very small, then we expect arithmetic in with a to be faster than arithmetic with b. If for example all integers involved can be stored in a single machine word on a modern computer (say integers of absolute value less than 2^{63}), then performing k multiplications simply costs k times as much as performing a single multiplication.

Conversion between the two representations is possible of course. Given $a \in \mathfrak{R}(\mathbb{A}_K)$ and a finite prime $\mathfrak{p}$ of K, it is very cheap to compute the projection of a to $\mathfrak{p}$, i.e. the representation $a_{\mathfrak{p}}$ of $\mathfrak{p}$-adic numbers such that $\mathcal{R}(a_{\mathfrak{p}})$ equals the projection of $\mathcal{R}(a)$ to $K_{\mathfrak{p}}$. It is however very expensive to find $b \in \mathfrak{R}_{\mathrm{pw}}(\mathbb{A}_K)$ with $\mathcal{R}(a) = \mathcal{R}(b)$: this requires factoring the modulus of a_0. For big moduli this is practically impossible. The other way around is simpler: to compute a

from b one needs to solve the Chinese Remainder Problem over K, for which polynomial time algorithms exist.

Before we can compute with representations of adèles, we need to create them first. The input data can have many formats. Some logical formats could be: a list of $\mathfrak{p}$-adic numbers, an element of K or an element of a quotient $\mathcal{O}/I$ of $\mathcal{O}$. From these input data both prime-wise and bundled representations can be created efficiently, except for one combination: converting $x \in O/I$ to $\mathfrak{R}_{\text{pw}}(\mathbb{A}_K)$. For this it is necessary to factor I, which can be very expensive for big I. When using bundled representations, no problem would arise: we simply store I in our bundled representation and upon doing arithmetic in $\mathfrak{R}(\mathbb{A}_K)$ there is never a need to factor I.

One could try to get the best out of both worlds by letting bundled representations *cache moduli factorizations*. By this we mean that we store the factorization of the modulus of the finite part of $a \in \mathfrak{R}(\mathbb{A}_K)$, if we know it, together with (or in) a. For example, if the user creates representations of adèles by giving lists of $\mathfrak{p}$-adic numbers, then we know the factorization of the modulus and can store it immediately. And if two representations, whose moduli factorizations we stored, are added or multiplied then we can efficiently compute the factorization of the modulus of the result as well, and store it. Using this approach, we *could* do our computations per prime if we want to. *During computation* the implementation could choose whether to perform computations bundled or prime-wise. When dealing with very small values or if big moduli are involved whose factorization we do not know we should probably opt for bundled computation. But in situations with known moduli factorizations and big values, prime-wise computation could be chosen.

As described in Section 3.4, we have implemented bundled representations. We did not implement prime-wise implementations, nor did we implement moduli factorization caching. We do think they could be useful, but experimental future research would have to tell. We anticipate the biggest difficulty of such a research project to be the determination of applications used for comparing the performance of these two representations. Our second application, described in Chapter 9, may provide one such application.

Above we described prime-wise representations for clarity as being based on value-modulus representations of $\mathfrak{p}$-adic numbers. One could of course replace value-modulus representations by $\mathbb{Q}_p$-vector representations or power series representations of $\mathfrak{p}$-adic numbers. This results in very similar representations of adèles.

Lastly we present another idea we had for representing adèles, which we did not implement. Let our number field K be given together with a generator α over $\mathbb{Q}$ and write $n = \deg(K/\mathbb{Q})$. View $\mathbb{A}_K$ as a free $\mathbb{A}_\mathbb{Q}$-algebra with the basis $(1, \alpha, \alpha^2, ..., \alpha^{n-1})$. Suppose we already have an implementation of $\mathbb{A}_\mathbb{Q}$. This could for example be a prime-wise representation using the (already implemented) rational p-adic numbers in SageMath. Now we define an *$\mathbb{A}_\mathbb{Q}$-vector representation of K-adèles* to be a tuple $a = (a_0, a_1, ..., a_{n-1})$ of representations of $\mathbb{Q}$-adèles, whose *represented subset* $\mathcal{R}(a)$ is given by

$$\mathcal{R}(a_0) + \alpha\mathcal{R}(a_1) + ... + \alpha^{n-1}\mathcal{R}(a_{n-1}).$$

A disadvantage of this approach is the following. Let $\mathfrak{p}$ and $\mathfrak{q}$ be primes of K lying over the same rational prime number p. Suppose we want to compute with

an adèle x of which we know $x_{\mathfrak{p}}$ very precisely, while we know nothing about $x_{\mathfrak{q}}$. With an $\mathbb{A}_{\mathbb{Q}}$-vector representation, the information about $x_{\mathfrak{p}}$ must be encoded in $a_0, a_1, ..., a_{n-1}$ by specifying their values at p (up to some precision). But we know nothing about $x_{\mathfrak{q}}$ and therefore we cannot say anything about the values of $a_0, ..., a_{n-1}$ at p. As a result, we cannot store the information about $x_{\mathfrak{p}}$ in an $\mathbb{A}_{\mathbb{Q}}$-vector representation. We saw this as a big disadvantage of $\mathbb{A}_{\mathbb{Q}}$-vector representations, compared to both bundled and prime-wise representations.

6.4 Idèles

Let K be a number field. Given that $J_K \subseteq \mathbb{A}_K$ and that we had already defined representations of K-adèles, a logical option for representing K-idèles would be to simply use representations of K-adèles. We could define a representation of K-idèles to be a representation of K-adèles, but with its represented subset intersected with J_K. This unification of representations of adèles and idèles sounds nice. It would however result in the failing of the finite representability property as stated in Section 3.6. This is caused by the fact that the topology on J_K is finer than the topology on $\mathbb{A}_K^*$ (induced from $\mathbb{A}_K$). For example, the set $U = \prod_{\mathfrak{p}} \mathcal{O}_{\mathfrak{p}}^*$, with $\mathfrak{p}$ running over all primes of K, is open in J_K while it is not open in $\mathbb{A}_K^*$. With our final definition of representations of K-idèles U is representable, while U would not be representable when using representations of K-adèles. Worse even: no representable subset contained in U would exist in that case. This motivated our decision to design representations of idèles separately from representations of adèles, as we did in Chapter 3.

Before choosing our representations of idèles from Chapter 3, we used a more general notion of representation of idèles. Denote the set of finite primes of K by $\mathcal{P}$ and recall our definition

$$\mathcal{F} = \left\{ f : \mathcal{Q} \to K \times \mathbb{Z}_{\geq 0} \;\middle|\; \begin{array}{l} \mathcal{Q} \text{ is a finite subset of } \mathcal{P} \text{ and} \\ f(\mathfrak{p}) \in \mathfrak{R}(K_{\mathfrak{p}}^*) \text{ for each } \mathfrak{p} \in \mathcal{Q} \end{array} \right\}.$$

An *alternative representation of K-idèles* consists of a triple $a = (x, f, e)$ with $x \in (\prod_{i=1}^{r} \mathfrak{R}(\mathbb{R}^*)) \times (\prod_{j=1}^{s} \mathfrak{R}(\mathbb{C}^*))$, $f \in \mathcal{F}$ and $e \in K^* \sqcup \{*\}$. Writing $\mathcal{P}(a)$ for the domain of f, the *represented subset of a* is be given by

$$\mathcal{R}(a) = \left(\prod_{i=1}^{r+s} (\mathcal{R}(x_i) \setminus \{0\}) \right) \times \left(\prod_{\mathfrak{p} \in \mathcal{P}(a)} \mathcal{R}(f(\mathfrak{p})) \right) \times \left(\prod_{\mathfrak{p} \in \mathcal{P} \setminus \mathcal{P}(a)} \{e\} \right)$$

if $e \in K^*$ and by the represented subset of $(x, f) \in \mathfrak{R}(J_K)$ if $e = *$. We did actually fully implement this alternative. An interested reader can request the corresponding code from the author. The advantage of this approach is better expressiveness: there are more available representations which give more representable subsets in J_K. It is however questionable how useful the extra representable subsets are: we could not think of any application where they would be useful. Our final version of representations of idèles is simpler than this alternative. This makes them easier to use and understand, in particular by SageMath users trying them out for the first time. The code of our final version is also much cleaner compared to the alternative version: fewer case distinctions need to be made. Keeping in mind our goal of including our code into SageMath,

this was an advantage as well as it makes our code better maintainable for the (future) SageMath community.

7 Application 1: profinite Fibonacci graph

Our first application is inspired by Lenstra's paper [13] on *profinite Fibonacci numbers*: generalizations of the usual Fibonacci numbers

$$0,\ 1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34,\ 55,\ 89,\ 144,\ 233,\ 377,\ 610,\ 987, ...$$

to $\widehat{\mathbb{Z}}$. The paper also contains a visualization of the graph of the profinite Fibonacci function. In [13] these subjects are treated in an informal manner. Proofs of many of the statements that are made in [13] can be found in [8] and [9]. The graph is however treated by neither.

We will use our representations of profinite numbers to compute profinite Fibonacci numbers and the corresponding graph. In Section 7.1 we introduce profinite Fibonacci numbers and we describe how to compute them using representations of profinite numbers. Section 7.2 introduces factorial digits, which are used in Section 7.3 to visualize profinite numbers. Section 7.3 formally introduces the graph of the profinite Fibonacci function. In Section 7.4 we explain how the graphs are computed in practice.

In this chapter we will view moduli of representations of profinite $\mathbb{Q}$-integers as integers: for $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$ we identify the ideal $m(a)$ of $\mathbb{Z}$ with its unique non-negative generator in $\mathbb{Z}$. For $n \in \mathbb{Z}_{>0}$ we write $n!$ for n factorial.

7.1 Profinite Fibonacci numbers

The *Fibonacci function* is the map $F_{(-)} : \mathbb{Z} \to \mathbb{Z}$ satisfying $F_0 = 0$, $F_1 = 1$ and $F_{n+2} = F_{n+1} + F_n$ for all $n \in \mathbb{Z}$. View F via the embedding of $\mathbb{Z}$ in $\widehat{\mathbb{Z}}$ as a function $F : \mathbb{Z} \to \widehat{\mathbb{Z}}$. Theorem 5.6 of [8] shows that there exists a unique continuous extension $\widehat{\mathbb{Z}} \to \widehat{\mathbb{Z}}$ of F. It is constructed using the ring $\widehat{\mathcal{O}}_K$ of profinite integers of the field $K = \mathbb{Q}[X]/(X^2 - X - 1)$. This extension is called the *profinite Fibonacci function* and we denote it by F as well. Elements in its image are called *profinite Fibonacci numbers*.

Based on the profinite Fibonacci function $F_{(-)} : \widehat{\mathbb{Z}} \to \widehat{\mathbb{Z}}$ we define the map $\tilde{F} : \mathfrak{R}(\widehat{\mathbb{Z}}) \to \mathfrak{R}(\widehat{\mathbb{Z}})$ by

$$\tilde{F}(a) = F_{v(a)} \bmod \gcd(F_{m(a)}, F_{m(a)+1} - 1) \in \mathfrak{R}(\widehat{\mathbb{Z}})$$

for $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$. This definition ensures the following.

Theorem 7.1. *For any $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$ the image of $\mathcal{R}(a)$ under F is contained in $\mathcal{R}(\tilde{F}(a))$.*

Theorem 7.2. *For any $k \in \mathbb{Z}_{>0}$ there exists $\ell \in \mathbb{Z}_{>0}$ such that for every $n \in \mathbb{Z}_{\geq \ell}$ and for every $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$ of precision $n!$, the precision of $\tilde{F}(a)$ is at least $k!$.*

We prove these theorems after the following lemma.

Lemma 7.3. *Let $c, d \in \mathbb{Z}$ and define $n = \gcd(F_d, F_{d+1} - 1)$. Then we have $F_c \equiv F_{c+d} \mod n\mathbb{Z}$.*

Proof. By definition of n we have the identities

$$\begin{aligned} F_d &\equiv 0 = F_0 \mod n\mathbb{Z}; \\ F_{d+1} &\equiv 1 = F_1 \mod n\mathbb{Z}. \end{aligned}$$

Hence the result follows by induction on c from the relation $F_{k+2} = F_{k+1} + F_k$ for $k \in \mathbb{Z}$. □

Proof of Theorem 7.1. Let $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$. In order to prove that we have $\mathcal{R}(a) = v(a) + m(a)\widehat{\mathbb{Z}} \subseteq \mathcal{R}(\tilde{F}(a))$, it suffices to prove $v(a) + m(a)\mathbb{Z} \subseteq \mathcal{R}(\tilde{F}(a))$, because F is continuous and $\mathbb{Z}$ lies dense in $\widehat{\mathbb{Z}}$. Let $\alpha = v(a) + m(a)k$ for $k \in \mathbb{Z}$. By applying Lemma 7.3 we find that $F_{v(a)} \equiv F_\alpha \mod \gcd(F_{m(a)}, F_{m(a)+1} - 1)$. Hence $F_\alpha \in \mathcal{R}(\tilde{F}(a))$ and this finishes the proof. □

Proof of Theorem 7.2. Let $k \in \mathbb{Z}_{>0}$. Set $\ell = \max(4, k)$. Let $n \in \mathbb{Z}_{\geq \ell}$ and let $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$ with $m(a) = n!$. Corollary 5.2 of [8] in particular states that $F_{n!} \equiv F_0 \mod n!\mathbb{Z}$ and $F_{n!+1} \equiv F_1 \mod n!\mathbb{Z}$. Hence $n!$ divides $\gcd(F_{n!} - F_0, F_{n!+1} - F_1) = m(\tilde{F}(a))$. Because $k \leq n$ it follows that $k!$ divides $m(\tilde{F}(a))$. □

For $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$, the definition of $\tilde{F}(a)$ is well suited for computation, as it is given by a direct formula. Theorem 7.2 ensures that the precision of $\tilde{F}(a)$ is high enough for our purposes. It may however be the case that $\tilde{F}(a)$ is *not* always the most precise representation b of profinite $\mathbb{Q}$-integers satisfying $F(\mathcal{R}(a)) \subseteq \mathcal{R}(b)$. For example we have $\tilde{F}(3 \bmod 6) = 2 \bmod 4$, but for each $n \in \mathbb{Z}$ such that $|n| \leq 10000$ we computed that $F_{3+6n} \in 2 + 32\mathbb{Z}$, based on which one could suspect $\tilde{F}(3 \bmod 6) \subseteq \mathcal{R}(2 \bmod 32)$ to hold. As our definition of $\tilde{F}$ nicely fits our purposes, we did not further investigate these suspicions.

7.2 Factorial digits

Let $\alpha \in \widehat{\mathbb{Z}}$. We define the *factorial digit sequence of* α to be the unique sequence $(d_i)_{i=1}^\infty \in \prod_{i=1}^\infty \mathbb{Z}$ satisfying $0 \leq d_k \leq k$ and $\alpha \equiv \sum_{i=1}^k d_i(i!) \mod (k+1)!\widehat{\mathbb{Z}}$ for each $k \in \mathbb{Z}_{\geq 1}$. These factorial digits can be obtained recursively as follows. For any $n \in \mathbb{Z}_{\geq 1}$ the natural isomorphism $\widehat{\mathbb{Z}}/n\widehat{\mathbb{Z}} \xrightarrow{\sim} \mathbb{Z}/n\mathbb{Z}$ induces a map $\rho_n : \widehat{\mathbb{Z}} \to \{0, 1, ..., n-1\}$ satisfying $\beta \equiv \rho_n(\beta) \mod n\widehat{\mathbb{Z}}$ for all $\beta \in \widehat{\mathbb{Z}}$. For $i \in \mathbb{Z}_{\geq 0}$ define $\alpha_i \in \widehat{\mathbb{Z}}$ and $d_i \in \mathbb{Z}$ recursively by $\alpha_0 = \alpha$, $d_0 = 0$ and

$$\begin{aligned} \alpha_k &= (\alpha_{k-1} - d_{k-1})/k && \text{for } k \in \mathbb{Z}_{\geq 1}; \\ d_k &= \rho_{k+1}(\alpha_k) && \text{for } k \in \mathbb{Z}_{\geq 1}. \end{aligned}$$

The division is performed in $\widehat{\mathbb{Q}}$ and since $\alpha_{k-1} \equiv d_{k-1} \mod k\widehat{\mathbb{Z}}$, the result will lie in $\widehat{\mathbb{Z}}$. This results in $(d_i)_{i=1}^\infty$ being the factorial digit sequence of α. For $i \in \mathbb{Z}_{\geq 1}$ we call d_i the *i-th factorial digit of* α.

For example the identities

$$\begin{aligned} 970 &= 0 \cdot 1! + 2 \cdot 2! + 1 \cdot 3! + 0 \cdot 4! + 2 \cdot 5! + 1 \cdot 6!; \\ 2021 &= 1 \cdot 1! + 2 \cdot 2! + 0 \cdot 3! + 4 \cdot 4! + 4 \cdot 5! + 2 \cdot 6! \end{aligned}$$

reveal that $(0, 2, 1, 0, 2, 1, 0, 0, 0...)$ are the factorial digits of 970 and that those of 2021 are $(1, 2, 0, 4, 4, 2, 0, 0, 0, ...)$. Non-negative integers always have only finitely many non-zero factorial digits. One can check that the factorial digits of -1 are $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...)$.

We implemented factorial digits in SageMath as well.

```
sage: a = Zhat(970, factorial(10))
sage: a.str(style='factorial')
'2*2! + 1*3! + 2*5! + 1*6! + O(10!)'
sage: a.factorial_digits()
[0, 2, 1, 0, 2, 1, 0, 0, 0]
sage: Zhat([1, 2, 0, 4, 4, 2, 0, 0])
2021 mod 362880
```

One can show that the map $\widehat{\mathbb{Z}} \to \{(d_i)_{i=1}^{\infty} \in \prod_{i=1}^{\infty} \mathbb{Z} \mid 0 \leq d_i \leq i\}$ sending a profinite integer to its factorial digits sequence is a bijection: injectivity follows form the definition of $\widehat{\mathbb{Z}}$ and surjectivity from the completeness of $\widehat{\mathbb{Z}}$.

7.3 Visualizing profinite graphs

We define the *visualization function* $\phi : \widehat{\mathbb{Z}} \to [0, 1]$ for $\alpha \in \widehat{\mathbb{Z}}$ with factorial digit sequence $(d_i)_{i=0}^{\infty}$ by

$$\phi(\alpha) = \sum_{i=1}^{\infty} \frac{d_i}{(i+1)!},$$

which is indeed an element of the unit interval. It is instructive to verify the equations $\phi(1 + 2\widehat{\mathbb{Z}}) = [1/2, 1]$ and $\phi(2 + 3\widehat{\mathbb{Z}}) = [1/6, 1/3] \cup [5/6, 1]$ and to determine for example $\phi(8 + 24\widehat{\mathbb{Z}})$.

Recall that $F_{(-)} : \widehat{\mathbb{Z}} \to \widehat{\mathbb{Z}}$ denotes the profinite Fibonacci function. The *graph of* F is the set $\mathcal{G} = \{(\alpha, F_\alpha) \mid \alpha \in \widehat{\mathbb{Z}}\} \subseteq \widehat{\mathbb{Z}} \times \widehat{\mathbb{Z}}$. We want to visualize this graph using the set

$$\tilde{\mathcal{G}} = \{(\phi(\alpha), \phi(F_\alpha)) \mid \alpha \in \widehat{\mathbb{Z}}\} \subseteq [0, 1] \times [0, 1].$$

Now F is continuous, but ϕ is not if we endow $[0, 1]$ with the usual topology. In particular $\tilde{\mathcal{G}}$ is not path-connected in $[0, 1] \times [0, 1]$ and it is difficult to draw it in the Euclidean plane. Informally, one could describe $\tilde{\mathcal{G}}$ as a cloud of scattered points in the unit square.

Instead we will visualize approximations to $\mathcal{G}$. Let $k \in \mathbb{Z}_{>0}$. For $x \in \widehat{\mathbb{Z}}/k!\widehat{\mathbb{Z}}$ we define the set

$$Y_x = \left\{ y \in \widehat{\mathbb{Z}}/k!\widehat{\mathbb{Z}} \mid F_x \cap y \neq \emptyset \right\}.$$

We call the set $\mathcal{G}_k = \{(x, Y_x) \mid x \in \widehat{\mathbb{Z}}/k!\widehat{\mathbb{Z}}\}$ the *approximation of* $\mathcal{G}$ *at precision* k. In Section 7.4 we will describe how to compute $\mathcal{G}_k$ using $\tilde{F}$. For each $(x, Y_x) \in \mathcal{G}_k$ and for each $y \in Y_x$ the set $\phi(x) \times \phi(y)$ is a closed square in $[0, 1] \times [0, 1]$ which we can draw in the plane. By *drawing* $\mathcal{G}_k$ we mean drawing each of the squares $\phi(x) \times \phi(y)$ for $x \in \widehat{\mathbb{Z}}/k!\widehat{\mathbb{Z}}$ and $y \in Y_x$ in the plane. We did this for $\mathcal{G}_3$, $\mathcal{G}_4$ and $\mathcal{G}_5$, see Figure 1.

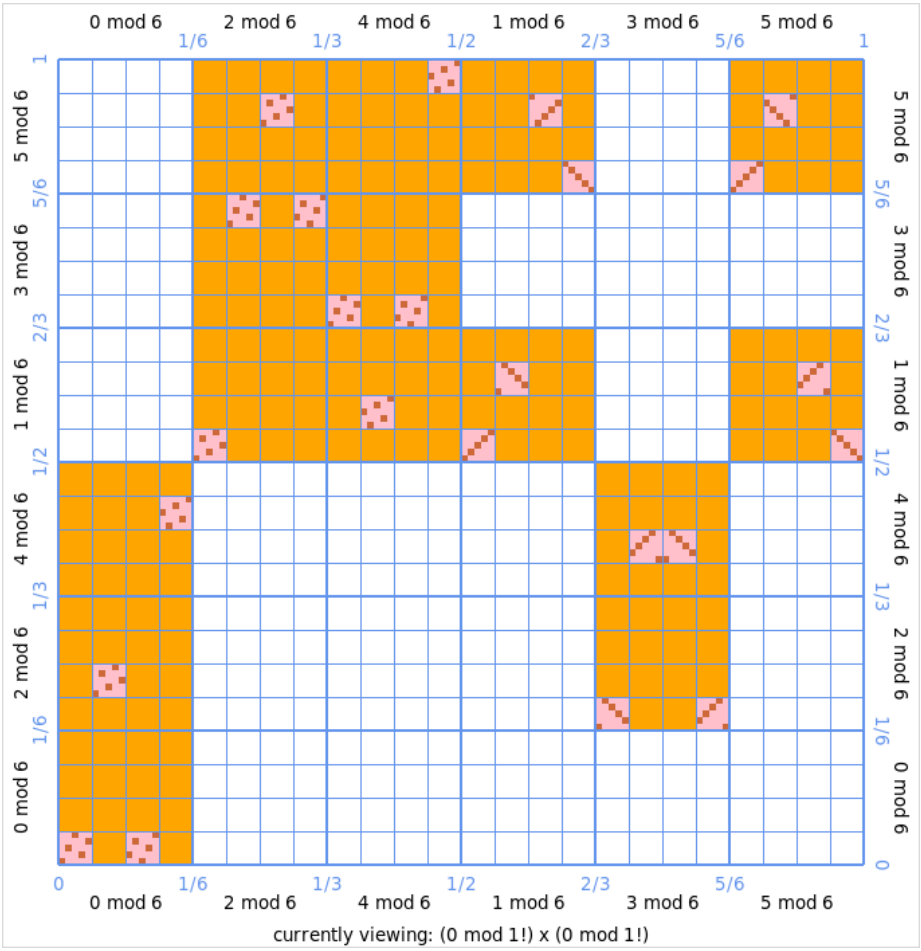

Figure 1: A visualization of the approximations $\mathcal{G}_3$ (orange), $\mathcal{G}_4$ (pink) and $\mathcal{G}_5$ (brown) of the graph of the profinite Fibonacci function F. The blue axis labels indicate coordinates in $[0,1]$ and the black axis labels indicate subsets of $\widehat{\mathbb{Z}}$. For example, the interval $[1/3, 1/2]$ corresponds via the visualization function ϕ to $4 + 6\widehat{\mathbb{Z}}$. From the orange squares we can read off that F maps $1 + 6\widehat{\mathbb{Z}}$ to $(1 + 6\widehat{\mathbb{Z}}) \cup (5 + 6\widehat{\mathbb{Z}})$ and both $1 + 6\widehat{\mathbb{Z}}$ and $5 + 6\widehat{\mathbb{Z}}$ are hit by $1 + 6\widehat{\mathbb{Z}}$ under F. Similarly there is a pink square indicating that F maps $7 + 24\widehat{\mathbb{Z}}$ to $13 + 24\widehat{\mathbb{Z}}$, which is consistent with $F_7 = 13$.

The graph in Figure 1 can be created using our SageMath code as follows.

```
sage: F = ProfiniteFibonacci()
sage: F(Zhat(6, 20)) # Evaluate F at 6 mod 20
8 mod 55
sage: G = ProfiniteGraph(F)
sage: G.plot()
```

The command `G.plot()` will open a window with the graph from Figure 1. This window is *interactive*: a user can left-click on rectangles corresponding to subsets of the form $(x + n!\widehat{\mathbb{Z}}) \times (y + n!\widehat{\mathbb{Z}})$ for $x, y, n \in \mathbb{Z}$, $n \geq 1$, to zoom in to that region. By right-clicking anywhere the user can zoom out. Upon zooming in/out, approximations of $\mathcal{G}$ of more/less precision are computed and drawn. One can for example perform the "blow-ups" described in [13] in the search for fixed points of F, as shown in Figure 2.

The colors of the graph in [13] are set as default colors for our graph. Many options of the graph can be tuned by methods of `ProfiniteGraph`, such as the colors, the title, the window sizes and some technical draw options.

Our `ProfiniteGraph` can also graph other functions $f : \widehat{\mathbb{Z}} \to \widehat{\mathbb{Z}}$, as long as there is (an implementation of) a map $\tilde{f} : \mathfrak{R}(\widehat{\mathbb{Z}}) \to \mathfrak{R}(\widehat{\mathbb{Z}})$ that satisfies Theorems 7.1 and 7.2, with f and $\tilde{f}$ in the place of F and $\tilde{F}$. This is for example the case for the map sending an $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$ to $a \cdot a$ with respect to the squaring-map $\widehat{\mathbb{Z}} \to \widehat{\mathbb{Z}}, x \mapsto x^2$. Hence the following code produces a graph of the squaring-map. See Figure 3 for the result.

```
sage: square = lambda x, _: x*x
sage: G = ProfiniteGraph(square)
sage: G.plot()
```

For details, see `profinite_graph.py` at [6].

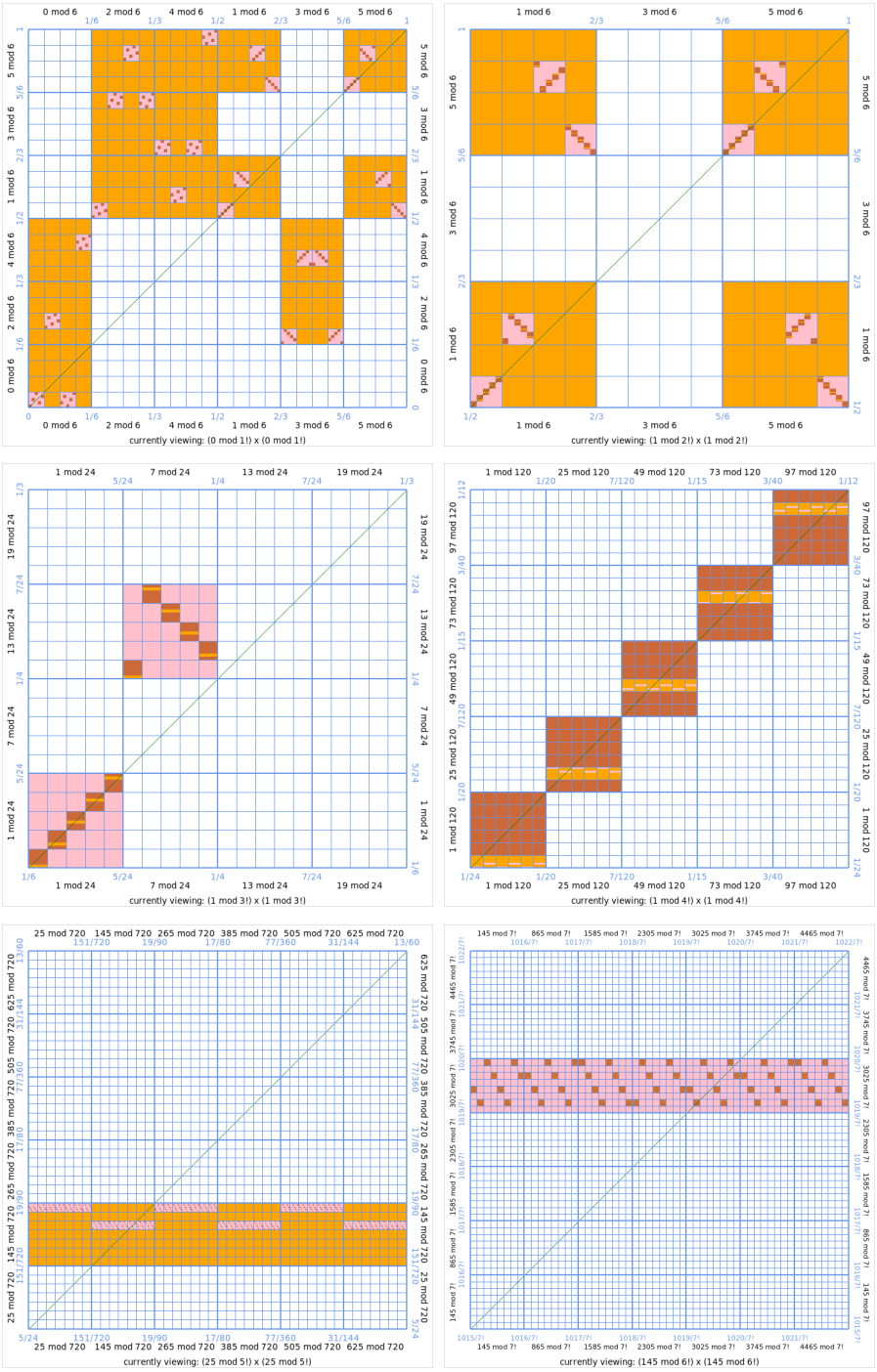

Figure 2: Searching for fixed points of the Fibonacci graph by zooming in. From top-left to bottom right we view $(x + n!\widehat{\mathbb{Z}}) \times (x + n!\widehat{\mathbb{Z}})$ for $(x, n) = (1, 1), (1, 2), (1, 3), (1, 4), (25, 5), (145, 6)$. The green lines visualizes the line $x = x$ in $\widehat{\mathbb{Z}}$, set by the method `set_identity_line()` of `ProfiniteGraph`.

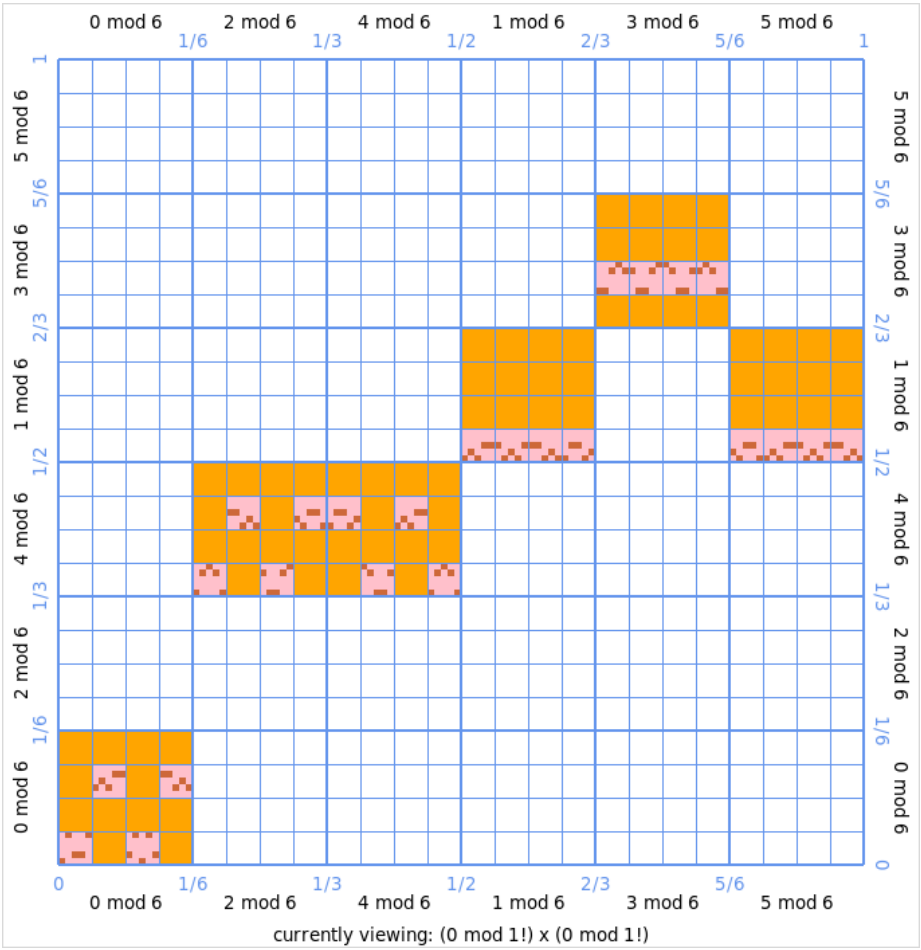

Figure 3: Approximations to the graph of the squaring-map $\widehat{\mathbb{Z}} \to \widehat{\mathbb{Z}}, x \mapsto x^2$.

7.4 Computing the graphs

Using our representations of profinite numbers and the function $\tilde{F}$, we can compute the approximation $\mathcal{G}_k$ of the graph of the profinite Fibonacci function at any precision $k \in \mathbb{Z}_{>0}$. Algorithm 7.4 gives an overview of the computation, which is not yet optimal. Below we will describe some adjustments that we implemented to achieve good performance. In our computation a coset $x \in \widehat{\mathbb{Z}}/k!\widehat{\mathbb{Z}}$ is represented by the unique representation $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$ with $x = \mathcal{R}(a)$.

Algorithm 7.4. Approximating the profinite Fibonacci graph.
INPUT: $k \in \mathbb{Z}_{>0}$.
OUTPUT: $\mathcal{G}_k$.
ALGORITHM:

1. Initialize $\mathcal{G}_k = \emptyset$.
2. For each $v_0 \in \{0, 1, ..., k! - 1\}$, do the following.
   2(a). Initialize $Y = \emptyset$, $\ell = k$ and $v = v_0$.

2(b). Construct $a = v \bmod \ell! \in \mathfrak{R}(\widehat{\mathbb{Z}})$ and compute $\tilde{F}(a)$.
2(c). If $k!$ does not divide the modulus of $\tilde{F}(a)$, then increase ℓ by 1 and go to Step 2(b).
2(d). Add the representation $y = v(\tilde{F}(a)) \bmod k! \in \mathfrak{R}(\widehat{\mathbb{Z}})$ to Y.
2(e). Increase v by $k!$. If $v < \ell!$, then go to Step 2(b).
2(f). Set $x = v_0 \bmod k! \in \mathfrak{R}(\widehat{\mathbb{Z}})$ and add (x, Y) to $\mathcal{G}_k$.
3. Output $\mathcal{G}_k$.

Theorem 7.2 ensures that Algorithm 7.4 terminates. Correctness of the output is a consequence of Theorem 7.1.

Based on the proof of Theorem 7.2, one may notice that increasing ℓ is never necessary if we simply initialize ℓ to $\max(4, k)$. This simplifies the algorithm considerably. This only holds for the Fibonacci function in general though. The algorithm above is valid for computing approximations of graphs of arbitrary maps $f : \widehat{\mathbb{Z}} \to \widehat{\mathbb{Z}}$, given an $\tilde{f} : \mathfrak{R}(\widehat{\mathbb{Z}}) \to \mathfrak{R}(\widehat{\mathbb{Z}})$ for which Theorems 7.1 and 7.2 hold.

In our application we only show part of the graph, namely the region which the user zoomed in to. Let $x, y, n, k \in \mathbb{Z}$, $k > n \geq 1$ and suppose our application needs to draw $\mathcal{G}_k$ in the region $(x + n!\widehat{\mathbb{Z}}) \times (y + n!\widehat{\mathbb{Z}})$. Then it suffices to perform Step 2 of Algorithm 7.4 only for $v_0 \in \{x + i \cdot n! \mid 0 \leq i < k!/n!\}$.

Evaluating $\tilde{F}$ in $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$ requires the computation of $F_{v(a)}$, $F_{m(a)}$ and $F_{m(a)+1}$. This is done using the identity

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$$

for $n \in \mathbb{Z}_{\geq 0}$, which can be evaluated using $O(\log_2(n))$ matrix multiplications. Computing $\tilde{F}(a)$ is costly for $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$ with a big modulus: it requires the computation of $F_{m(a)}$, which is huge for big $m(a)$. For our application, we do not need the full output precision of $\tilde{F}$ though: we are only interested in the image of $\mathcal{R}(\tilde{F}(a))$ in $\widehat{\mathbb{Z}}/k!\widehat{\mathbb{Z}}$. For this it suffices to compute the images of $F_{m(a)}$ and $F_{m(a)+1}$ in $\mathbb{Z}/k!\mathbb{Z}$ by performing the matrix exponentiation above in the ring $\mathbb{Z}/k!\mathbb{Z}$. This results in good performance: on a modern computer the Fibonacci graph can at least be zoomed in to (translates of) the region $100!\widehat{\mathbb{Z}} \times 100!\widehat{\mathbb{Z}}$ without the user having to wait more than a second per zoom operation.

Lastly drawing the rectangle in $[0, 1] \times [0, 1]$ corresponding to a pair $(a, b) \in \mathfrak{R}(\widehat{\mathbb{Z}})^2$ under the visualization function ϕ is done using the `visual()` method of a `ProfiniteInteger`. For an $a \in \mathfrak{R}(\widehat{\mathbb{Z}})$, this method computes the smallest closed interval $I \subseteq [0, 1]$ such that $v(\mathcal{R}(a)) \subseteq I$. If $m(a) = k!$ for some $k \in \mathbb{Z}_{\geq 1}$, then we have $v(\mathcal{R}(a)) = I$.

For details on the implementation of these profinite functions and graphs, see the files `profinite_function.py` and `profinite_graph.py` at [6].

8 Adèlic matrix factorization

Let g be a positive integer. Let the matrix $\Omega \in \mathbb{Z}^{2g \times 2g}$ be given in $g \times g$-blocks as

$$\Omega = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

with I the $g \times g$-identity matrix.

For a commutative ring R we define the *general symplectic group over* R by

$$\mathrm{GSp}_{2g}(R) = \{A \in R^{2g\times 2g} \mid A^{\mathsf{T}}\Omega A = \mu(A)\Omega \text{ for some } \mu(A) \in R^*\},$$

with A^{T} denoting the transpose of A. The element $\mu(A)$ is called the *multiplier of* A and we have a *multiplier homomorphism* $\mu : \mathrm{GSp}_{2g}(R) \to R^*$ sending a matrix to its multiplier. We denote the kernel of μ by $\mathrm{Sp}_{2g}(R)$, called the *symplectic group over* R. For $g = 1$ we simply have $\mathrm{GSp}_2(R) = \mathrm{GL}_2(R)$, $\mathrm{Sp}_2(R) = \mathrm{SL}_2(R)$ and $\mu = \det$. In general we always have that $\mathrm{GSp}_{2g}(R)$ is a subgroup of $\mathrm{GL}_{2g}(R)$. For $R = \mathbb{Q}$ or $R = \mathbb{Z}$ we define $\mathrm{GSp}^+_{2g}(R)$ to be the subgroup of $\mathrm{GSp}_{2g}(R)$ consisting of matrices with positive multiplier.

It has been known for a long time that the equalities

$$\mathrm{GL}_n(\widehat{\mathbb{Q}}) = \mathrm{GL}_n(\widehat{\mathbb{Z}})\,\mathrm{GL}^+_n(\mathbb{Q}) \quad \text{and} \quad \mathrm{GSp}_{2g}(\widehat{\mathbb{Q}}) = \mathrm{GSp}_{2g}(\widehat{\mathbb{Z}})\,\mathrm{GSp}^+_{2g}(\mathbb{Q})$$

hold for $n, g \in \mathbb{Z}_{\geq 1}$. These equalities are a special case of strong approximation for the algebraic groups GL_n and GSp_{2g}, which in our situation can be phrased as the fact that $\mathrm{GL}_n(\mathbb{Q})$ and $\mathrm{GSp}_{2g}(\mathbb{Q})$ lie dense in $\mathrm{GL}_n(\widehat{\mathbb{Q}})$ and $\mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$ respectively. Important results of Eichler on strong approximation in algebraic groups date back to 1938. For a proof as well as a historic account of strong approximation in algebraic groups, see Section 7.4 of Platonov and Rapinchuk [15].

So for any $M \in \mathrm{GL}_n(\widehat{\mathbb{Q}})$, we are assured of the existence of a factorization $M = BA$ with $B \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$ and $A \in \mathrm{GL}^+_n(\mathbb{Q})$. Moreover if $M \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$, then we can get $B \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Z}})$ and $A \in \mathrm{GSp}^+_{2g}(\mathbb{Q})$. In this chapter we will describe algorithms, based on our representations of profinite numbers, that can perform such factorizations in practice.

8.1 Representations of $\mathrm{GL}_n(\widehat{\mathbb{Q}})$-elements

Let n be a positive integer. We define a *representation of* $\mathrm{GL}_n(\widehat{\mathbb{Q}})$*-elements* to be a pair $\mathcal{M} = (E, \Delta)$ with $E \in \mathfrak{R}(\widehat{\mathbb{Q}})^{n\times n}$ and $\Delta \in \mathbb{Q}_{>0}$, such that the set

$$\mathcal{R}(\mathcal{M}) = \left\{M \in \mathrm{GL}_n(\widehat{\mathbb{Q}}) \mid \det(M)\widehat{\mathbb{Z}} = \Delta\widehat{\mathbb{Z}} \text{ and } M_{ij} \in \mathcal{R}(E_{ij}) \text{ for } 1 \leq i, j \leq n\right\}$$

is non-empty. The set $\mathcal{R}(\mathcal{M})$ is called the *represented subset of* $\mathcal{M}$. The entries of E are also called the *entries of* $\mathcal{M}$ and we write $\mathcal{M}_{ij} = E_{ij}$ for $i, j \in \{1, ..., n\}$. We call Δ the *determinant of* $\mathcal{M}$ and denote it by $\det(\mathcal{M})$. We call $\mathcal{M}$ *integral* if all of its entries are integral, i.e. have denominator 1. If $\mathcal{M}$ is integral, then also $\det(\mathcal{M})$ is integral and $\mathcal{R}(\mathcal{M}) \subseteq \widehat{\mathbb{Z}}^{n\times n}$. The *transpose* $\mathcal{M}^{\mathsf{T}}$ *of* $\mathcal{M}$ is defined to be the representation of $\mathrm{GL}_n(\widehat{\mathbb{Q}})$-elements (E^{T}, Δ). We define the *value matrix* $v(\mathcal{M})$ *of* $\mathcal{M}$ to be the matrix $(v(\mathcal{M}_{ij}))_{i,j=0}^n \in \mathbb{Q}^{n\times n}$. We denote the set of representations of $\mathrm{GL}_n(\widehat{\mathbb{Q}})$-elements by $\mathfrak{R}(\mathrm{GL}_n(\widehat{\mathbb{Q}}))$.

Recall that for $a \in \mathfrak{R}(\widehat{\mathbb{Q}})$, the precision $m(a)$ of a was defined to be its modulus, which is a (possibly zero) fractional $\mathbb{Z}$-ideal in $\mathbb{Q}$. We define the *precision* $m(\mathcal{M})$ *of* $\mathcal{M}$ to be the greatest common divisor of the precisions of the entries of $\mathcal{M}$, i.e. the fractional $\mathbb{Z}$-ideal having valuation $\min_{1\leq i,j\leq n}(\mathrm{ord}_p(m(\mathcal{M}_{ij})))$ at p for each prime number p. We order precisions of elements in $\mathfrak{R}(\widehat{\mathbb{Q}})$ and

$\mathfrak{R}(\mathrm{GL}_n(\widehat{\mathbb{Q}}))$ the same, namely using the following order on the set $\mathcal{I}$ of fractional $\mathbb{Z}$-ideals in $\mathbb{Q}$: for $I, J \in \mathcal{I}$, we declare $I \leq J$ if and only if for each prime number p we have $\mathrm{ord}_p(I) \leq \mathrm{ord}_p(J)$. We sometimes compare a rational number r to a precision of an element of $\mathfrak{R}(\widehat{\mathbb{Q}})$ or $\mathfrak{R}(\mathrm{GL}_n(\widehat{\mathbb{Q}}))$, in which case the reader should interpret r as $r\mathbb{Z} \in \mathcal{I}$.

As $E \in \mathfrak{R}(\widehat{\mathbb{Q}})^{n\times n}$ one can define the *determinant of* E in the usual way using the associative and commutative addition and multiplication in $\mathfrak{R}(\widehat{\mathbb{Q}})$, resulting in a $\det(E) \in \mathfrak{R}(\widehat{\mathbb{Q}})$. We emphasize that $\det(\mathcal{M})$ is *not* $\det(E)$. We will *not* use this determinant of E in this chapter and we warm the reader that it might be the case that $\det(\mathcal{M}) \notin \mathcal{R}(\det(E))$. This happens for example for $\mathcal{M} = \left(\left(\begin{smallmatrix} 1 \bmod 3 & 0 \bmod 3 \\ 0 \bmod 3 & 2 \bmod 3 \end{smallmatrix}\right), 1\right) \in \mathfrak{R}(\mathrm{GL}_2(\widehat{\mathbb{Q}}))$.

We define multiplication in $\mathfrak{R}(\mathrm{GL}_n(\widehat{\mathbb{Q}}))$ as follows: for $\mathcal{M}_1 = (E_1, \Delta_1)$ and $\mathcal{M}_2 = (E_2, \Delta_2)$ we set $\mathcal{M}_1\mathcal{M}_2 = (E_1E_2, \Delta_1\Delta_2)$, where E_1E_2 denotes usual matrix multiplication, using the associative and commutative addition and multiplication in $\mathfrak{R}(\widehat{\mathbb{Q}})$.

A matrix $A \in \mathrm{GL}_n(\mathbb{Q})$ can be viewed in a canonical way as a representation of $\mathrm{GL}_n(\widehat{\mathbb{Q}})$-elements, namely as $\mathcal{A} = (A, |\det(A)|)$ with the entries of A viewed as representations of profinite $\mathbb{Q}$-numbers in the canonical way. This gives $\mathcal{R}(\mathcal{A}) = \{A\}$.

8.2 Factorization in general linear groups

Let $\mathcal{M}$ be an integral representation of $\mathrm{GL}_n(\widehat{\mathbb{Q}})$-elements with precision at least $\det(\mathcal{M})$. Let $M \in \mathcal{R}(\mathcal{M})$. In this section we will use $\mathcal{M}$ to factor M as $M = BA$ with $B \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$ and $A \in \mathrm{GL}_n^+(\mathbb{Q})$. We start of with a lemma that gives an explicit set of generators of the lattice $\mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n M$ in $\mathbb{Q}^n$ in terms of $\mathcal{M}$.

Lemma 8.1. *The equality* $\mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n M = \mathbb{Z}^n v(\mathcal{M}) + \mathbb{Z}^n \det(\mathcal{M})$ *holds.*

Proof. We will first prove the inclusion from right to left by showing that both $\mathbb{Z}^n v(\mathcal{M})$ and $\mathbb{Z}^n \det(\mathcal{M})$ are subsets of $\mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n M$.

Write N for the matrix of cofactors of M, satisfying $MN = NM = \det(M)I$, for I the $n \times n$ identity matrix. As $\mathcal{M}$ is integral, we have $M \in \widehat{\mathbb{Z}}^{n\times n}$ and therefore $N \in \widehat{\mathbb{Z}}^{n\times n}$ as well. In turn this gives us

$$\widehat{\mathbb{Z}}^n \det(\mathcal{M}) = \widehat{\mathbb{Z}}^n \det(M) = \widehat{\mathbb{Z}}^n NM \subseteq \widehat{\mathbb{Z}}^n M.$$

As $\det(\mathcal{M}) \in \mathbb{Z}_{>0}$ this shows $\mathbb{Z}^n \det(\mathcal{M}) \subseteq \mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n M$.

By the precision assumption on $\mathcal{M}$, there exists $\lambda \in \widehat{\mathbb{Z}}^{n\times n}$ such that $M = v(\mathcal{M}) + \lambda \det(\mathcal{M})$. Using the inequality above again we obtain

$$\widehat{\mathbb{Z}}^n v(\mathcal{M}) \subseteq \widehat{\mathbb{Z}}^n M + \widehat{\mathbb{Z}}^n \lambda \det(\mathcal{M}) \subseteq \widehat{\mathbb{Z}}^n M + \widehat{\mathbb{Z}}^n \det(\mathcal{M}) \subseteq \widehat{\mathbb{Z}}^n M.$$

Since $v(\mathcal{M}) \in \mathbb{Z}^{n\times n}$ this shows $\mathbb{Z}^n v(\mathcal{M}) \subseteq \mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n M$.

Next we will prove $\mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n M \subseteq \mathbb{Z}^n v(\mathcal{M}) + \mathbb{Z}^n \det(\mathcal{M})$. Let $y \in \mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n M$ and write $y = xM$ for $x \in \widehat{\mathbb{Z}}^n$. As $\widehat{\mathbb{Z}} \det(\mathcal{M})$ is open in $\widehat{\mathbb{Z}}$ and $\mathbb{Z}$ lies dense in $\widehat{\mathbb{Z}}$,

we can find $\tilde{x} \in \mathbb{Z}^n$ and $z \in \widehat{\mathbb{Z}}^n$ such that $x = \tilde{x} + z\det(\mathcal{M})$. Now we have

$$\begin{aligned} y &= xM \\ &= (\tilde{x} + z\det(\mathcal{M}))(v(\mathcal{M}) + \lambda\det(\mathcal{M})) \\ &= \tilde{x}v(\mathcal{M}) + (zv(\mathcal{M}) + \tilde{x}\lambda + z\lambda\det(\mathcal{M}))\det(\mathcal{M}). \end{aligned}$$

From $y \in \mathbb{Q}^n$, $\tilde{x}v(\mathcal{M}) \in \mathbb{Z}^n$ and $\det(\mathcal{M}) \in \mathbb{Z} \setminus \{0\}$ it follows that $zv(\mathcal{M}) + \tilde{x}\lambda + z\lambda\det(\mathcal{M}) \in \mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n = \mathbb{Z}^n$. Hence $y \in \mathbb{Z}^n v(\mathcal{M}) + \mathbb{Z}^n \det(\mathcal{M})$. □

Write I for the $n \times n$-identity matrix and let $J \in \mathbb{Z}^{2n\times n}$ by given in $n \times n$-blocks as

$$J = \begin{pmatrix} v(\mathcal{M}) \\ \det(\mathcal{M})I \end{pmatrix}.$$

Lemma 8.1 shows that the rows of J generate the lattice $\mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n M$. We can compute the row Hermite Normal Form of J (cf. [2], 2.4.2). As I has rank n, this gives us an $n \times n$-matrix $A \in \mathbb{Z}^{n\times n}$ whose rows form a $\mathbb{Z}$-basis of $\mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n M$. Hence we actually have that $A \in \mathrm{GL}_n(\mathbb{Q}) \cap \mathbb{Z}^{n\times n}$. After optionally replacing A by $\mathrm{diag}(1, 1, ..., 1, -1)A$ we may assume $A \in \mathrm{GL}_n^+(\mathbb{Q}) \cap \mathbb{Z}^{n\times n}$. Because the rows of A form a $\mathbb{Z}$-basis of $\mathbb{Q}^n \cap M\widehat{\mathbb{Z}}^n$ we have $\mathbb{Z}^n A = \mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n M$.

Now define $B = MA^{-1} \in \mathrm{GL}_n(\widehat{\mathbb{Q}})$. This matrix B satisfies

$$\mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n B = \mathbb{Q}^n A^{-1} \cap \widehat{\mathbb{Z}}^n MA^{-1} = (\mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n M)A^{-1} = \mathbb{Z}^n AA^{-1} = \mathbb{Z}^n.$$

The following two lemmas prove that the property above guarantees that we have $B \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$.

Lemma 8.2. *Let U be an open subset of $\widehat{\mathbb{Q}}^n$ and assume $\mathbb{Q}^n \cap U \subseteq \mathbb{Z}^n$. Then $U \subseteq \widehat{\mathbb{Z}}^n$.*

Proof. Suppose towards a contradiction that there exists some $x \in U \setminus \widehat{\mathbb{Z}}^n$. As U is open there exists $m \in \mathbb{Z} \setminus \{0\}$ such that $x + m\widehat{\mathbb{Z}}^n \subseteq U$. Because $\mathbb{Q}^n$ lies dense in $\widehat{\mathbb{Q}}^n$ and $x + m\widehat{\mathbb{Z}}^n$ is open in $\widehat{\mathbb{Q}}^n$, there exists some $y \in \mathbb{Q}^n \cap (x + m\widehat{\mathbb{Z}}^n)$. Because x is not integral, there exists a coordinate x_i of x and a prime number p such that $\mathrm{ord}_p(x_i) < 0$. As $m \in \mathbb{Z}$ and $y \in x + m\widehat{\mathbb{Z}}^n$ it follows that also $\mathrm{ord}_p(y_i) < 0$. Hence y is not integral, which contradicts the fact that $y \in \mathbb{Q}^n \cap (x + m\widehat{\mathbb{Z}}^n) \subseteq \mathbb{Q}^n \cap U \subseteq \mathbb{Z}^n$. □

Lemma 8.3. *Let $B \in \mathrm{GL}_n(\widehat{\mathbb{Q}})$ such that $\mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n B = \mathbb{Z}^n$. Then $B \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$.*

Proof. Right-multiplication by B induces a homeomorphism $\widehat{\mathbb{Q}}^n \to \widehat{\mathbb{Q}}^n$, as $B \in \mathrm{GL}_n(\widehat{\mathbb{Q}})$. This together with the fact that $\widehat{\mathbb{Z}}^n$ is open in $\widehat{\mathbb{Q}}^n$, gives that $\widehat{\mathbb{Z}}^n B$ is also open in $\widehat{\mathbb{Q}}^n$. Now from $\mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n B = \mathbb{Z}^n$ and the lemma above it follows that $\widehat{\mathbb{Z}}^n B \subseteq \widehat{\mathbb{Z}}^n$. We also have $\mathbb{Z}^n = \mathbb{Q}^n \cap \widehat{\mathbb{Z}}^n B \subseteq \widehat{\mathbb{Z}}^n B$. Because $\mathbb{Z}^n$ lies dense in $\widehat{\mathbb{Z}}^n$ and $\widehat{\mathbb{Z}}^n B$ is open in $\widehat{\mathbb{Z}}^n$, we even have $\widehat{\mathbb{Z}}^n \subseteq \widehat{\mathbb{Z}}^n B$. Hence $\widehat{\mathbb{Z}}^n B = \widehat{\mathbb{Z}}^n$ and therefore $B \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$. □

Let us recapitulate what we did up to now. For the given integral representation of $\mathrm{GL}_n(\widehat{\mathbb{Q}})$-elements $\mathcal{M}$ having precision at least $\det(\mathcal{M})$ and for the given $M \in \mathcal{R}(\mathcal{M})$, we described above how to compute an $A \in \mathrm{GL}_n^+(\mathbb{Q}) \cap \mathbb{Z}^{n\times n}$ such that $MA^{-1} \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$.

We can generalize this method to non-integral $\mathcal{M}$ as follows. Let $\mathcal{M}$ be any representation of $\mathrm{GL}_n(\widehat{\mathbb{Q}})$-elements. For each $j \in \{1, ..., n\}$ let d_j be the least common multiple of the denominators of the entries of $\mathcal{M}$ in the j-th column. Define the *denominator matrix of* $\mathcal{M}$ to be $D = \mathrm{diag}(d_1, ..., d_n)$. Let us say that $\mathcal{M}$ has *good precision* if $\mathcal{M}D$ has precision at least $\det(\mathcal{M}D)$. This is equivalent to the condition that for each prime number p and for all $i, j \in \{1, ..., n\}$ we have $\mathrm{ord}_p(m(\mathcal{M}_{ij})d_j) \geq \mathrm{ord}_p(\det(\mathcal{M}) \prod_{k=1}^{n} d_k)$.

Now let $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_n(\widehat{\mathbb{Q}}))$ have good precision and denominator matrix D. Let $M \in \mathcal{R}(\mathcal{M})$. Then $\mathcal{M}D$ is integral and has precision at least $\det(\mathcal{M}D)$, hence we can find $A_0 \in \mathrm{GL}_n^+(\mathbb{Q})$ satisfying $MDA_0^{-1} \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$. Then $A = A_0D^{-1} \in \mathrm{GL}_n^+(\mathbb{Q})$ satisfies $MA^{-1} \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$.

We summarize the method described above in the following algorithm.

Algorithm 8.4. Matrix factorization in $\mathrm{GL}_n(\widehat{\mathbb{Q}})$
INPUT: a representation of $\mathrm{GL}_n(\widehat{\mathbb{Q}})$-elements $\mathcal{M}$ having good precision.
OUTPUT: an $A \in \mathrm{GL}_n^+(\mathbb{Q})$ such that $\mathcal{R}(\mathcal{M}A^{-1}) \subseteq \mathrm{GL}_n(\widehat{\mathbb{Z}})$.
ALGORITHM:

1. Compute the denominator matrix D of $\mathcal{M}$.
2. Construct the matrix $J \in \mathbb{Z}^{2n \times n}$ with upper $n \times n$-block equal to $v(\mathcal{M}D)$ and lower $n \times n$-block equal to $\det(\mathcal{M}D)I$, with I the identity matrix.
3. Compute the row Hermite Normal Form $A_0 \in \mathrm{GL}_n(\mathbb{Q})$ of J.
4. If $\det(A_0) < 0$, then replace A_0 by $\mathrm{diag}(1, 1, ..., 1, -1)A_0$.
5. Set $A := A_0D^{-1}$ and output A.

We implemented Algorithm 1 in SageMath as the function `factor_GLQhat()` in the file `matrix.py` at [6].

Note that we did not use the equality $\mathrm{GL}_n(\widehat{\mathbb{Q}}) = \mathrm{GL}_n(\widehat{\mathbb{Z}})\,\mathrm{GL}_n^+(\mathbb{Q})$ in the exposition above. By the representability property of $\mathfrak{R}(\widehat{\mathbb{Q}})$ any $M \in \mathrm{GL}_n(\widehat{\mathbb{Q}})$ has a representation of $\mathrm{GL}_n(\widehat{\mathbb{Q}})$-elements of good precision. Therefore Algorithm 1 (together with our discussion showing its correctness) can be seen as a constructive proof that $\mathrm{GL}_n(\widehat{\mathbb{Q}}) = \mathrm{GL}_n(\widehat{\mathbb{Z}})\,\mathrm{GL}_n^+(\mathbb{Q})$ holds.

Now we will prove a result on precision regarding Algorithm 1. For $A \in \mathbb{Q}^{n \times n}$ define the *denominator of* A, denoted $\mathrm{den}(A)$, to be the smallest positive integer d such that $dA \in \mathbb{Z}^{n \times n}$.

Theorem 8.5. *Let $M \in \mathrm{GL}_n(\widehat{\mathbb{Q}})$. Let $Q \in \mathbb{Z}_{>0}$. Let $A_0 \in \mathrm{GL}_n^+(\mathbb{Q})$ such that $MA_0^{-1} \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$. Let $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_n(\widehat{\mathbb{Q}}))$ represent M and suppose that $\mathcal{M}$ has good precision and has precision at least $Q\,\mathrm{den}(A_0^{-1})$. Then the output $A \in \mathrm{GL}_n^+(\mathbb{Q})$ of Algorithm 1 upon giving $\mathcal{M}$ as input satisfies: $\mathcal{M}A^{-1}$ has precision at least Q.*

Theorem 8.5 follows directly from Lemma 8.6 and Lemma 8.7, which we prove below.

Lemma 8.6. *Let $M \in \mathrm{GL}_n(\widehat{\mathbb{Q}})$ and let $N_1, N_2 \in \mathrm{GL}_n^+(\mathbb{Q})$. Suppose that $MN_1, MN_2 \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$. Then $\mathrm{den}(N_1) = \mathrm{den}(N_2)$.*

Proof. Write $B_1 = MN_1$ and $B_2 = MN_2$. Then we have

$$U := N_1^{-1}N_2 = B_1^{-1}MM^{-1}B_2 = B_1^{-1}B_2 \in \mathrm{GL}_n^+(\mathbb{Q}) \cap \mathrm{GL}_n(\widehat{\mathbb{Z}}) = \mathrm{SL}_n(\mathbb{Z}).$$

From $N_2 = N_1 U$ and $U \in \mathbb{Z}^{n\times n}$ it follows that $\operatorname{den}(N_1)N_2 \in \mathbb{Z}^{n\times n}$ and hence $\operatorname{den}(N_2) \leq \operatorname{den}(N_1)$. Similarly $N_1 = N_2 U^{-1}$ and $U^{-1} \in \mathbb{Z}^{n\times n}$ together give $\operatorname{den}(N_1) \leq \operatorname{den}(N_2)$. □

Lemma 8.7. *Let $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_n(\widehat{\mathbb{Q}}))$ and let $N \in \mathrm{GL}_n(\mathbb{Q})$. Then $\mathcal{M}N$ has precision at least $m(\mathcal{M})/\operatorname{den}(N)$.*

Proof. Note that within $\mathfrak{R}(\widehat{\mathbb{Q}})$ the identity $a(b+c) = ab + ac$ holds for $a \in \mathbb{Q}$ and $b, c \in \mathfrak{R}(\widehat{\mathbb{Q}})$. Let $i, j \in \{1, ..., n\}$. We have

$$(\mathcal{M}N)_{ij} = \sum_{k=1}^{n} \mathcal{M}_{ik} N_{kj} = \frac{1}{\operatorname{den}(N)} \sum_{k=1}^{n} \mathcal{M}_{ik} \operatorname{den}(N) N_{kj}.$$

For each $k \in \{1, ..., n\}$ we have $\operatorname{den}(N)N_{kj} \in \mathbb{Z}$ and hence the precision of $\mathcal{M}_{ik} \operatorname{den}(N) N_{kj}$ is at least $m(\mathcal{M}_{ik})$, which in turn is at least $m(\mathcal{M})$. It follows that the precision of the sum $\sum_{k=1}^{n} \mathcal{M}_{ik} \operatorname{den}(N) N_{kj}$ is at least $m(\mathcal{M})$ as well. So we have $m((\mathcal{M}N)_{ij}) \geq m(\mathcal{M})/\operatorname{den}(N)$ for all $i, j \in \{1, ..., n\}$. This proves the lemma. □

Above we treated the problem of factoring $M \in \mathrm{GL}_n(\widehat{\mathbb{Q}})$ as $M = BA$ with $B \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$ and $A \in \mathrm{GL}_n^+(\mathbb{Q})$. If one instead wants to find $A \in \mathrm{GL}_n^+(\mathbb{Q})$ and $B \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$ such that $M = AB$, then this can be done using Algorithm 1 as well. Take an $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_n(\widehat{\mathbb{Q}}))$ representing M such that M^{T} has good precision, apply Algorithm 1 to $\mathcal{M}^{\mathsf{T}}$ and define A to be the transpose of the output. Then $A^{-1}M \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$.

Algorithm 8.4 for $n = 2$ will play a key role in our second application, which we discuss in Chapter 9.

Recall that a representation of $\mathrm{GL}_n(\widehat{\mathbb{Q}})$-elements $\mathcal{M} = (E, \Delta)$ consists of the determinant Δ of $\mathcal{M}$ as well. Hence before we can apply Algorithm 1 to an $M \in \mathrm{GL}_n(\widehat{\mathbb{Q}})$, we must know $\det(M)\widehat{\mathbb{Z}}$. Let us explain why this information is necessary for performing our factorization. Consider the case that

$$E = \begin{pmatrix} 1 \bmod 2 & 0 \\ 0 & 1 \end{pmatrix}, \quad M_1 = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 5 & 0 \\ 0 & 1 \end{pmatrix}.$$

Now for $i, j, k \in \{1, 2\}$ we have $(M_k)_{ij} \in \mathcal{R}(E_{ij})$. But there does *not* exist an $A \in \mathrm{GL}_n^+(\mathbb{Q})$ such that $M_1 A^{-1}, M_2 A^{-1} \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$, because this would imply $3\widehat{\mathbb{Z}} = \det(M_1)\widehat{\mathbb{Z}} = \det(A)\widehat{\mathbb{Z}} = \det(M_2)\widehat{\mathbb{Z}} = 5\widehat{\mathbb{Z}}$. In general for any $E \in \mathfrak{R}(\widehat{\mathbb{Q}})^{n\times n}$ there exist $M_1, M_2 \in \mathrm{GL}_n(\widehat{\mathbb{Q}})$ such that $\det(M_1)\widehat{\mathbb{Z}} \neq \det(M_2)\widehat{\mathbb{Z}}$ and $(M_1)_{ij}, (M_2)_{ij} \in \mathcal{R}(E_{ij})$ for $1 \leq i, j \leq n$.

8.3 Factorization in general symplectic groups

Let $g \in \mathbb{Z}_{\geq 1}$ and let $M \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$. In this section we will use representations of $\mathrm{GL}_{2g}(\widehat{\mathbb{Q}})$-elements to find $A \in \mathrm{GSp}_{2g}^+(\mathbb{Q})$ such that $MA^{-1} \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Z}})$.

To this end, let $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_{2g}(\widehat{\mathbb{Q}}))$ represent M and have good precision. First of all we apply Algorithm 1 to $\mathcal{M}$ to obtain an $A_0 \in \mathrm{GL}_{2g}^+(\mathbb{Q})$ such that $B_0 = MA_0^{-1} \in \mathrm{GL}_{2g}(\widehat{\mathbb{Z}})$. Define $A_1 = \operatorname{den}(A_0)A_0 \in \mathrm{GL}_{2g}^+(\mathbb{Q}) \cap \mathbb{Z}^{2g\times 2g}$.

Now $E = A_1 \Omega A_1^{\mathsf{T}} \in \mathbb{Z}^{2g \times 2g}$ is an alternating matrix of full rank (with alternating meaning $x^{\mathsf{T}} E x = 0$ for all $x \in \mathbb{Z}^n$). In particular E has only zeros on its diagonal and we have $E = -E^{\mathsf{T}}$. SageMath provides the function `symplectic_basis_over_ZZ()`, which is based on (the constructive proof of) Theorem 18 of [10] and which does the following. Upon giving E as input, `symplectic_basis_over_ZZ()` will compute a matrix $C \in \mathrm{GL}_{2g}(\mathbb{Z})$ such that CEC^{T} is given in terms of $g \times g$-blocks as

$$CEC^{\mathsf{T}} = \begin{pmatrix} 0 & R \\ -R & 0 \end{pmatrix}$$

where $R = \mathrm{diag}(r_1, ..., r_g)$ for positive integers $r_1, ..., r_g$ satisfying the divisibility property $r_i \mid r_{i+1}$ for each $i \in \{1, ..., g-1\}$. The rows of this matrix C are said to form a *symplectic basis* for (the bilinear form represented by) the matrix E, hence the name of the function. See any book on symplectic geometry for context on symplectic bases, for example [1]. In our particular situation we have the following.

Lemma 8.8. *Let $g \in \mathbb{Z}_{\geq 1}$ and let $M \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$. Let $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_{2g}(\widehat{\mathbb{Q}}))$ represent M. Let A_0 be the output of Algorithm 1 on the input $\mathcal{M}$. Let $A_1 = \mathrm{den}(A_0)A_0$ and let $E = A_1 \Omega A_1^{\mathsf{T}}$. Let $C =$ `symplectic_basis_over_ZZ`(E) and write $CEC^{\mathsf{T}} = \mathrm{diag}(r_1, r_2, ..., r_g, r_1, r_2, ..., r_g)\Omega$. Then $r_1 = r_2 = ... = r_g$ and $r_i \widehat{\mathbb{Z}} = \mathrm{den}(A_0)^2 \mu(M) \widehat{\mathbb{Z}}$ for $1 \leq i \leq g$.*

Proof. Let $x = (1, 0, 0, ..., 0) \in \mathbb{Z}^{1 \times 2g}$ and let $y = (0, ..., 0, 1, 0, ..., 0)^{\mathsf{T}} \in \mathbb{Z}^{2g \times 1}$ with $y_{g+1,1} = 1$. Define $B_0 = MA_0^{-1}$. Then by using the explicit form of CEC^{T} and by unrolling the definition of E we see that

$$\begin{aligned} r_1 &= xCEC^{\mathsf{T}}y \\ &= x\,\mathrm{den}(A_0)^2 C B_0^{-1} M \Omega M^{\mathsf{T}} (B_0^{-1})^{\mathsf{T}} C^{\mathsf{T}} y \\ &= \mathrm{den}(A_0)^2 \mu(M) x C B_0^{-1} \Omega (B_0^{-1})^{\mathsf{T}} C^{\mathsf{T}} y. \end{aligned}$$

Above we used the fact that M^{T} is also in $\mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$, with the same multiplier as M. Since we have $B_0, C, \Omega \in \mathrm{GL}_2(\widehat{\mathbb{Z}})$ this shows that $r_1 \in \mathrm{den}(A_0)^2 \mu(M) \widehat{\mathbb{Z}}$. By the divisibility property of the r_i we then have $r_1, ..., r_g \in \mathrm{den}(A_0)^2 \mu(M) \widehat{\mathbb{Z}}$. Looking at the determinant we see

$$\begin{aligned} \prod_{i=1}^{g} r_i^2 \widehat{\mathbb{Z}} &= \det(CEC^{\mathsf{T}}) \widehat{\mathbb{Z}} \\ &= \det(\mathrm{den}(A_0)^2 \mu(M) C B_0^{-1} \Omega (B_0^{-1})^{\mathsf{T}} C^{\mathsf{T}}) \widehat{\mathbb{Z}} \\ &= \mathrm{den}(A_0)^{4g} \mu(M)^{2g} \widehat{\mathbb{Z}} \end{aligned}$$

as the determinants of B_0, C and Ω lie in $\widehat{\mathbb{Z}}^*$. Hence $r_i \widehat{\mathbb{Z}} = \mathrm{den}(A_0)^2 \mu(M) \widehat{\mathbb{Z}}$ for $1 \leq i \leq g$. This shows that $r_1 = r_2 = ... = r_g$ as the r_i are positive integers. □

Now putting $A = CA_0$, we have

$$A \Omega A^{\mathsf{T}} = CA_0 \Omega A_0^{\mathsf{T}} C^{\mathsf{T}} = \mathrm{den}(A_0)^{-2} CEC^{\mathsf{T}} = \mathrm{den}(A_0)^{-2} r_1 \Omega$$

and so $A^T \in \mathrm{GSp}_{2g}^+(\mathbb{Q})$ and hence $A \in \mathrm{GSp}_{2g}^+(\mathbb{Q})$. So we have $B = MA^{-1} \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$ as well as $B = (MA_0^{-1})C^{-1} \in \mathrm{GL}_{2g}(\widehat{\mathbb{Z}})$. We conclude that we have $B \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Q}}) \cap \mathrm{GL}_{2g}(\widehat{\mathbb{Z}}) = \mathrm{GSp}_{2g}(\widehat{\mathbb{Z}})$, as desired.

The discussion above shows the correctness of the following algorithm.

Algorithm 8.9. Factorization for $\mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$.
INPUT: $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_{2g}(\widehat{\mathbb{Q}}))$ satisfying $\mathcal{R}(\mathcal{M}) \cap \mathrm{GSp}_{2g}(\widehat{\mathbb{Q}}) \neq \emptyset$ and having good precision (as described in the paragraph above Algorithm 1 in Section 8.2).
OUTPUT: an $A \in \mathrm{GSp}_{2g}^+(\mathbb{Q})$ such that $(\mathcal{M}A^{-1}) \cap \mathrm{GSp}_{2g}(\widehat{\mathbb{Q}}) \subseteq \mathrm{GSp}_{2g}(\widehat{\mathbb{Z}})$.
ALGORITHM:

1. Perform Algorithm 1 on $\mathcal{M}$ obtaining $A_0 \in \mathrm{GL}_{2g}^+(\mathbb{Q})$.
2. Set $A_1 := \mathrm{den}(A_0)A_0$ and $E := A_1 \Omega A_1^{\mathsf{T}}$.
3. Compute $C :=$ `symplectic_basis_over_ZZ`(E).
4. Set $A = CA_0$ and output A.

We implemented Algorithm 8.9 in the SageMath function `factor_GSpQhat()`, see `matrix.py` at [6].

Similar to the case of general linear groups, Algorithm 8.9 together with the discussion above establishing its correctness can be viewed as a constructive proof of the fact that $\mathrm{GSp}_{2g}(\widehat{\mathbb{Q}}) = \mathrm{GSp}_{2g}(\widehat{\mathbb{Z}})\,\mathrm{GSp}_{2g}^+(\mathbb{Q})$ holds.

Theorem 8.10. *Let $M \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$. Let $Q \in \mathbb{Z}_{>0}$. Let $A_0 \in \mathrm{GL}_n^+(\mathbb{Q})$ such that $MA_0^{-1} \in \mathrm{GL}_n(\widehat{\mathbb{Z}})$. Let $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_n(\widehat{\mathbb{Q}}))$ represent M and suppose that $\mathcal{M}$ has good precision and has precision at least $Q\,\mathrm{den}(A_0^{-1})$. Then the output $A \in \mathrm{GSp}_{2g}^+(\mathbb{Q})$ of Algorithm 2 upon giving $\mathcal{M}$ as input satisfies: $\mathcal{M}A^{-1}$ has precision at least Q.*

Proof. This follows directly from Theorem 8.5 and the fact that the precision of a representation of $\mathrm{GL}_{2g}(\widehat{\mathbb{Q}})$-elements does not decrease upon multiplication by an element of $\mathrm{GL}_{2g}(\mathbb{Z})$. □

Just as for Algorithm 8.4, we can also use Algorithm 8.9 to factor an $M \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$ as $M = AB$ instead of $M = BA$ with $A \in \mathrm{GSp}_{2g}^+(\mathbb{Q})$ and $B \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Z}})$. This is done by taking an $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_n(\widehat{\mathbb{Q}}))$ representing M such that $\mathcal{M}^{\mathsf{T}}$ has good precision, applying Algorithm 2 to $\mathcal{M}^{\mathsf{T}}$ and taking the transpose of the output.

Algorithm 8.9 will be import for generalizations of our second application, which we will discuss in Section 9.6.

9 Application 2: Hilbert class field computation

In this chapter we will apply our representations of adèles and idèles to compute Hilbert class fields of imaginary quadratic number fields using Shimura's reciprocity law. Several articles, such as [4], [5], [22], [23], have already been written on this computation. All of these articles rephrase the idèlic results given by Shimura's reciprocity law in terms of ideals, avoiding idèles in their calculations. In contrast, we will perform this computation in a direct idèlic manner, obtaining the same results.

The theoretical background for this computation is substantially more than basic algebraic number theory and knowledge of adèles and idèles. In Section 9.1 we will concisely formulate all the main results that we use in the computation.

Let K be an imaginary quadratic number field. Consider the problem of computing the Hilbert class field H of K, i.e. computing a polynomial $h \in K[X]$ such that $H \cong K[X]/(h)$. One can find such an h using the j-invariant (cf. [11], Chapter 3), as the minimal polynomial of $j(\theta)$, for θ a generator of $\mathcal{O}_K$ (cf. [11], Chapter 10, Theorem 1). The problem with this approach is that the resulting polynomial has huge coefficients, even for K with small discriminant. In order to speed up computations, we want to find an h with small(er) coefficients. A way to obtain such h is by replacing the j-invariant with a modular function f of higher level. This way one hopes to obtain values $f(\theta)$ which also generate H but have (much) smaller height then $j(\theta)$. Such a value $f(\theta)$ is called a *class invariant*. The computation of such class invariants can be done using Shimura's reciprocity law. This chapter is devoted to performing this computation using representations of adèles and idèles.

After stating the main theoretical results in Section 9.1, we will give an overview of the computation in theoretical terms in Section 9.2. Section 9.3 will describe how to perform the computation in practice using representations of adèles and idèles. A numerical example of such a computation is treated in Section 9.4. In section 9.5 we compare our method to the more classical method of [5]. Lastly in Section 9.6 we indicate how this computation can be generalized to CM-fields.

9.1 Theoretical background

In this section we will concisely state the main theoretical results needed for our Hilbert class field computation. Readers unfamiliar with these subjects are encouraged to read more elaborate treatments of the material. For class field theory, many good books exist. For the other topics, one could follow the references we give.

Class field theory. For an imaginary quadratic number field K, class field theory tells us that the sequence

$$1 \to K^* \to \widehat{K}^* \xrightarrow{A} \mathrm{Gal}(K^{\mathrm{ab}}/K) \to 1,$$

with A the Artin map, is exact. For $N \in \mathbb{Z}_{>0}$ the ray class field H_N of K modulo N satisfies

$$A^{-1}\,\mathrm{Gal}(K^{\mathrm{ab}}/H_N) = \mathcal{O}^* W_N,$$

where $W_N = \prod_{\mathfrak{p}} U_{\mathfrak{p}}^{\mathrm{ord}_{\mathfrak{p}}(N)}$ with $\mathfrak{p}$ ranging over the finite primes of K and $U_{\mathfrak{p}}^n$ the n-th multiplicative subgroup at $\mathfrak{p}$, i.e. $U_{\mathfrak{p}}^0 = \mathcal{O}_{\mathfrak{p}}^*$ and $U_{\mathfrak{p}}^n = 1 + \mathfrak{p}^n \mathcal{O}_{\mathfrak{p}}$ for $n \in \mathbb{Z}_{>0}$. In particular we have

$$A^{-1}\,\mathrm{Gal}(K^{\mathrm{ab}}/H) = \widehat{\mathcal{O}}^*$$

for the Hilbert class field H of K. This also means that we have

$$\mathrm{Gal}(H_N/H) \cong \widehat{\mathcal{O}}^*/\mathcal{O}^* W_N \cong (\mathcal{O}/N\mathcal{O})^*/\mathcal{O}^*,$$

where the last isomorphism is the natural one.

Complex multiplication. Let K be an imaginary quadratic number field with ring of integers $\mathcal{O}$ generated by θ. Embed K in $\mathbb{C}$ such that θ lies in the upper half plane. Let N be a positive integer and let $\mathcal{F}_N$ be the field of modular functions of level N over $\mathbb{Q}(\zeta_N)$, for ζ_N a primitive N-th root of unity. Then the ray class field H_N of K modulo N is given by

$$H_N = K(\{f(\theta) \mid f \in \mathcal{F}_N \text{ such that } f(\theta) \text{ is defined}\}).$$

That is, the field generated over K by the values $f(\theta)$, where f runs over all modular functions of level N over $\mathbb{Q}(\zeta_N)$ which do not have a pole at θ. This is for example proven in [11], see the corollary of Theorem 2 in Chapter 10.

The Shimura exact sequence. In [21], Chapter 6, Shimura constructs an action of $\mathrm{GL}_2(\widehat{\mathbb{Q}})$ on the automorphic function field $\mathcal{F} = \cup_{N=1}^{\infty} \mathcal{F}_N$, where $\mathcal{F}_N$ denotes the field of modular functions of level N over $\mathbb{Q}(\zeta_N)$. We will describe this action in an ad hoc manner here. Let N be a positive integer. The modular group $\mathrm{SL}_2(\mathbb{Z})$ acts on $\mathcal{F}_N$ by linear fractional transformations and this induces an action of $\mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$ on $\mathcal{F}_N$. Moreover the group $(\mathbb{Z}/N\mathbb{Z})^* \cong \mathrm{Gal}(\mathbb{Q}(\zeta_N)/\mathbb{Q})$ acts on the Fourier coefficients of functions in $\mathcal{F}_N$, which induces an action of $(\mathbb{Z}/N\mathbb{Z})^*$ on $\mathcal{F}_N$. Combining these two actions, we let $B \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ act on $f \in \mathcal{F}_N$ by setting $d = \det(B) \in (\mathbb{Z}/N\mathbb{Z})^*$, $U = \left(\begin{smallmatrix} 1 & 0 \\ 0 & d \end{smallmatrix}\right)^{-1} B \in \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$ and $f^B = (f^d)^U$. Now for each positive integer N, we described an action of $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ on $\mathcal{F}_N$. Passing to the projective limit this induces an action of $\mathrm{GL}_2(\widehat{\mathbb{Z}})$ on the automorphic function field $\mathcal{F}$. The group $\mathrm{GL}_2^+(\mathbb{Q})$ acts on $\mathcal{F}$ by linear fractional transformations. We have $\mathrm{GL}_2(\widehat{\mathbb{Q}}) = \mathrm{GL}_2(\widehat{\mathbb{Z}})\,\mathrm{GL}_2^+(\mathbb{Q})$ as discussed in Chapter 8. Writing an element $M \in \mathrm{GL}_2(\widehat{\mathbb{Q}})$ (non-uniquely) as $M = BA$ for $B \in \mathrm{GL}_2(\widehat{\mathbb{Z}})$ and $A \in \mathrm{GL}_2^+(\mathbb{Q})$ and setting $f^M = (f^B)^A$ for $f \in \mathcal{F}$ defines an action of $\mathrm{GL}_2(\widehat{\mathbb{Q}})$ on $\mathcal{F}$. Shimura proved that this action is such that the sequence

$$1 \to \mathbb{Q}^* \to \mathrm{GL}_2(\widehat{\mathbb{Q}}) \to \mathrm{Aut}(\mathcal{F}) \to 1$$

is exact (cf. [21], Theorem 6.23). It is referred to as *Shimura's exact sequence*.

The Shimura reciprocity law. Let K be an imaginary quadratic number field with ring of integers $\mathcal{O}$ generated by θ. View $\widehat{K}$ as a $\widehat{\mathbb{Q}}$-vector space with basis $(\theta, 1)$. For $x \in \widehat{K}$, let $g_\theta(x)$ be the transpose of the matrix that represents the $\widehat{\mathbb{Q}}$-linear map $\widehat{K} \to \widehat{K}$ given by multiplication by x. This defines a homomorphism $g_\theta : \widehat{K}^* \to \mathrm{GL}_2(\widehat{\mathbb{Q}})$ called *Shimura's connecting homomorphism*.

We now have the following diagram with exact rows:

$$\begin{array}{ccccccccc} 1 & \longrightarrow & K^* & \longrightarrow & \widehat{K}^* & \longrightarrow & \mathrm{Gal}(K^{\mathrm{ab}}/K) & \longrightarrow & 1 \\ & & & & \downarrow{\scriptstyle g_\theta} & & & & \\ 1 & \longrightarrow & \mathbb{Q}^* & \longrightarrow & \mathrm{GL}_2(\widehat{\mathbb{Q}}) & \longrightarrow & \mathrm{Aut}(\mathcal{F}) & \longrightarrow & 1. \end{array}$$

The *Shimura reciprocity law* states the following. For any $f \in \mathcal{F}$ and any $x \in \widehat{K}^*$ the equality

$$f(\theta)^x = f^{g_\theta(x^{-1})}(\theta)$$

holds (cf. [21], Theorem 6.31). Note that on the left hand side x acts on the element $f(\theta) \in K^{\mathrm{ab}}$ via the Artin map. On the right hand side the matrix $g_\theta(x^{-1})$ acts on the modular function $f \in \mathcal{F}$ as explained above.

9.2 Overview of the theoretical computation

In this section we give an overview of the computation that we will perform. We state the whole computation in theoretical terms: we use (exact) idèles and adèles, not their representations. In Section 9.3 we explain how to perform the computation in practice.

Let K be an imaginary quadratic number field with ring of integers $\mathcal{O}$ generated by θ. Embed K in $\mathbb{C}$ such that θ lies in the upper half plane. Let $N \in \mathbb{Z}_{>0}$ and let $f_0 \in \mathcal{F}_N$ such that f_0 does not a have a pole at θ. We will compute the Hilbert class field H of K as follows.

Finding class invariants. By complex multiplication we know $f_0(\theta) \in H_N$, the ray class field of K modulo N. In order to find a class invariant $\alpha \in H$, we want to compute the action of $\mathrm{Gal}(H_N/H)$ on $f_0(\theta) \in H_N$. We start off by taking generators $b_1, ..., b_k$ of the finite group $(\mathcal{O}/N\mathcal{O})^*/\mathcal{O}^*$. Next we pick representatives $x_1, ..., x_k \in \widehat{\mathcal{O}}^*$ of the images of $b_1, ..., b_k$ under the natural isomorphism

$$(\mathcal{O}/N\mathcal{O})^*/\mathcal{O}^* \xrightarrow{\sim} \widehat{\mathcal{O}}^*/\mathcal{O}^* W_N.$$

Since $\widehat{\mathcal{O}}^*/\mathcal{O}^* W_N \cong \mathrm{Gal}(H_N/H)$ via the Artin map, in order to compute the action of $\mathrm{Gal}(H_N/H)$ on $f_0(\theta)$, it suffices to compute the action of $x_1, ..., x_k$ on $f_0(\theta)$. By Shimura's reciprocity law we have $f_0(\theta)^{x_i} = f_0^{g_\theta(x_i^{-1})}(\theta)$ for $1 \leq i \leq k$. Below we will explain how to explicitly compute the expression on the right. Based on the computed action of $\mathrm{Gal}(H_N/H)$ on $f_0(\theta)$, we can usually find an $f_1 \in \mathcal{F}_N$ such that $\alpha = f_1(\theta) \in H$ and α has small height. See Section 9.4 for an example, or Section 5 of [5] for a wide range of examples. In many cases, one can pick $f_1 = \zeta_N^a f_0^b$ for $a, b \in \mathbb{Z}$ and ζ_N a primitive N-th root of unity.

Computing minimal polynomials. What is left to check is that $\alpha = f_1(\theta)$ is a class invariant and to compute its minimal polynomial $h_K^\alpha \in K[X]$ over K. From the degree of h_K^α we can tell whether $K(\alpha)$ is a strict subfield of H or whether α is a class invariant. We will compute h_K^α by means of the formula

$$h_K^\alpha = \prod_{\overline{\alpha} \in \mathcal{A}} (X - \overline{\alpha}),$$

for $\mathcal{A} = \{\sigma(\alpha) \mid \sigma \in \mathrm{Gal}(H/K)\}$ the set of conjugates of α over K. Compute the class group $Cl_K = \{c_1, ..., c_m\}$ of K. Take representatives $y_1, ..., y_m \in \widehat{K}^*$ of the images of $c_1, ..., c_m$ under the natural isomorphism

$$\mathrm{Cl}_K \xrightarrow{\sim} \widehat{K}^*/(K^* \widehat{\mathcal{O}}^*).$$

As the Artin map induces an isomorphism $\widehat{K}^*/(K^*\widehat{\mathcal{O}}^*) \cong \mathrm{Gal}(H/K)$ we have $\mathcal{A} = \{\alpha^{y_1}, ..., \alpha^{y_m}\}$. For each $i \in \{1, ..., m\}$ we can compute α^{y_i} using Shimura's reciprocity law: $\alpha^{y_i} = f_1^{g_\theta(y_i^{-1})}(\theta)$.

Computing actions. Wat is left to do is computing $f^{g_\theta(x^{-1})}(\theta)$ for $f \in \mathcal{F}_N$ and $x \in \widehat{K}^*$. As $g_\theta(x^{-1}) \in \mathrm{GL}_2(\widehat{\mathbb{Q}})$, there exist $B \in \mathrm{GL}_2(\widehat{\mathbb{Z}})$ and $A \in \mathrm{GL}_2^+(\mathbb{Q})$

such that $g_\theta(x^{-1}) = BA$, which we compute using Algorithm 8.4. Let B_N denote the image of B in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$. Set $d = \det(B_N)$ and $U = \left(\begin{smallmatrix} 1 & 0 \\ 0 & d \end{smallmatrix}\right)^{-1} B_N \in \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$. Take a lift $V \in \mathrm{SL}_2(\mathbb{Z})$ whose image in $\mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$ equals U. Now $f^{g_\theta(x^{-1})} = (f^d)^{VA}$ and these actions can be computed explicitly: d acts on the Fourier coefficients of f and VA acts by fractional linear transformations.

9.3 The computation in practice

In this section we explain the details of how to perform the exact theoretical computation from the previous section in practice. We keep all the notation introduced in Section 9.2.

9.3.1 Constructing representations

Computing generators $\{b_1, ..., b_k\}$ of $(\mathcal{O}/N\mathcal{O})^*/\mathcal{O}^*$ and enumerating the ideal classes $\{c_1, ..., c_m\}$ in Cl_K can be done in practice, see for example [2]. In our implementation we use the functionality provided by PARI [14] to compute (generators of) these groups.

In the theoretical computation we find $x_1, ..., x_k \in \widehat{\mathcal{O}}^*$ based on $b_1, .., b_k$ and $y_1, ..., y_m \in \widehat{K}^*$ based on $c_1, ..., c_m$. In practice we replace these by representations of K-idèles.

For $i \in \{1, ..., k\}$ take a $u_i \in \mathfrak{R}(J_K)$ such that $\mathcal{R}_0(u_i) \subseteq x_i \mathcal{O}^* W_N$, for example by taking a representative $b_i' \in (\mathcal{O}/N\mathcal{O})^*$ of b_i and letting u_i to be the image of b_i' under the conversion from $(\mathcal{O}/N\mathcal{O})^*$ to J_K. Similarly for $j \in \{1, ..., m\}$ take a $v_j \in \mathfrak{R}(J_K)$ such that $\mathcal{R}_0(v_j) \subseteq y_j K^* \widehat{\mathcal{O}}^*$, for example the image of c_j under the conversion from Cl_K to J_K (note that Cl_K equals the ray class group of K modulo 1).

For any $x \in \mathcal{R}_0(u_i)$ we now have $f(\theta)^x = f_0(\theta)^{x_i}$. So for our purposes it suffices to compute the action of *some* $x \in \mathcal{R}_0(u_i)$ on $f(\theta)$. Therefore in our computations we may replace u_i with any $u_i' \in \mathfrak{R}(J_K)$ such that $\mathcal{R}_0(u_i') \subseteq \mathcal{R}(u_i)$. Likewise once we know that $f_1(\theta)$ is a class invariant, for any $y \in \mathcal{R}_0(v_j)$ we have $f_1(\theta)^y = f_1(\theta)^{y_j}$ and so we may replace v_j by a $v_j' \in \mathfrak{R}(J_K)$ with $\mathcal{R}_0(v_j') \subseteq \mathcal{R}_0(v_j)$.

It turns out that the u_i's and v_j's obtained from our conversions usually do not have high enough precision for our purposes. To this end we define the following operation on $\mathfrak{R}(J_K)$. Let $w \in \mathfrak{R}(J_K)$ not have exact finite part. Let $\mathfrak{p}_0$ be a finite prime of K and let $n \in \mathbb{Z}_{>0}$. By saying *increase the precision of* w *at* $\mathfrak{p}_0$ *by* n we shall mean replace w with the $w' \in \mathfrak{R}(J_K)$ defined as follows. The infinite part of w' is equal to that of w. The finite part of w' is the map $\mathcal{P}(w) \cup \{\mathfrak{p}_0\} \to K \times \mathbb{Z}_{\geq 0}$ given by

$$\mathfrak{p} \mapsto \begin{cases} w_{\mathfrak{p}} & \text{if } \mathfrak{p} \neq \mathfrak{p}_0; \\ (c(w_{\mathfrak{p}}), p(w_{\mathfrak{p}}) + n) & \text{if } \mathfrak{p} = \mathfrak{p}_0 \text{ and } \mathfrak{p} \in \mathcal{P}(w); \\ (1, n) & \text{if } \mathfrak{p} = \mathfrak{p}_0 \text{ and } \mathfrak{p} \notin \mathcal{P}(w). \end{cases}$$

For p a prime number we say *increase the precision of* w *at* p *by* n if we mean to increase the precision of w at $\mathfrak{p}$ by $e_{\mathfrak{p}/p} n$ for each finite prime $\mathfrak{p}$ of K above p, with ramification index $e_{\mathfrak{p}/p}$.

9.3.2 Shimura's connecting homomorphism

For L a number field, let $\mathcal{I}_L$ denote the group of fractional $\mathcal{O}_L$-ideals in L and let $\varphi_L : \widehat{L}^* \to \mathcal{I}_L$ be the homomorphism defined by $\varphi_L(x) = \prod_{\mathfrak{p}} \mathfrak{p}^{\mathrm{ord}_{\mathfrak{p}}(x_{\mathfrak{p}})}$, with $\mathfrak{p}$ ranging over the finite primes of L. Now we have a commutative diagram

$$\begin{array}{ccc} \widehat{K}^* & \xrightarrow{\varphi_K} & \mathcal{I}_K \\ {\scriptstyle N}\downarrow & & \downarrow{\scriptstyle \mathcal{N}} \\ \widehat{\mathbb{Q}}^* & \xrightarrow[\varphi_{\mathbb{Q}}]{} & \mathcal{I}_{\mathbb{Q}} \end{array}$$

where N denotes the restriction of the norm of $\widehat{K}/\widehat{\mathbb{Q}}$ and $\mathcal{N}$ is the ideal norm.

Recall that Shimura's connecting homomorphism $g_\theta : \widehat{K}^* \to \mathrm{GL}_2(\widehat{\mathbb{Q}})$ sends $x \in \widehat{K}^*$ to the transpose of the multiplication by x map $\widehat{K} \to \widehat{K}$, where we view $\widehat{K}$ as a $\widehat{\mathbb{Q}}$-vector space with basis $(\theta, 1)$. Hence by definition of the norm, we have $N(x) = \det(g_\theta(x))$ for $x \in \widehat{K}^*$. Identify $\mathcal{I}_{\mathbb{Q}}$ in the obvious way with $\mathbb{Q}_{>0}$. Then for $x \in \widehat{K}^*$ the $\Delta \in \mathbb{Q}_{>0}$ such that $\Delta\widehat{\mathbb{Z}} = \det(g_\theta(x))\widehat{\mathbb{Z}}$ is given by

$$\Delta = \varphi_{\mathbb{Q}}(N(x)) = \mathcal{N}(\varphi_K(x)) = \prod_{\mathfrak{p}} \mathcal{N}(\mathfrak{p})^{\mathrm{ord}_{\mathfrak{p}}(x)}$$

with $\mathfrak{p}$ ranging over the finite primes of K. Writing the minimal polynomial of θ over $\mathbb{Q}$ as $X^2 + BX + C$ we have the explicit formula

$$g_\theta(s\theta + t) = \begin{pmatrix} t - Bs & -Cs \\ s & t \end{pmatrix}$$

for $s, t \in \widehat{\mathbb{Q}}$.

We define a map $\tilde{g}_\theta : \mathfrak{R}(J_K) \to \mathfrak{R}(\mathrm{GL}_2(\widehat{\mathbb{Q}}))$ based on g_θ as follows. Let $u \in \mathfrak{R}(J_K)$. First we will construct $(s, t) \in \mathfrak{R}(\widehat{\mathbb{Q}})^2$ such that $\mathcal{R}_0(u) \subseteq \mathcal{R}(s)\theta + \mathcal{R}(t)$, where we view J_K^0 inside $\widehat{K} = \widehat{\mathbb{Q}}\theta + \widehat{\mathbb{Q}}$. Denote the image of u under the conversion from J_K to $\mathbb{A}_K$ by a. Let $b = (t, s)$ be image of the finite part of a under the conversion from $\widehat{K}$ to $\widehat{\mathbb{Q}}^2$, where we choose θ as our distinguished generator of K over $\mathbb{Q}$. Define $E \in \mathfrak{R}(\widehat{\mathbb{Q}})^{2\times 2}$ by

$$E = \begin{pmatrix} t - Bs & -Cs \\ s & t \end{pmatrix}.$$

Define $\Delta = \prod_{\mathfrak{p}} \mathcal{N}(\mathfrak{p})^{\mathrm{ord}_{\mathfrak{p}}(u)}$ where $\mathfrak{p}$ ranges over the finite primes of K. Now we define $\tilde{g}_\theta(u) = (E, \Delta) \in \mathfrak{R}(\mathrm{GL}_2(\widehat{\mathbb{Q}}))$ and this ensures $g_\theta(\mathcal{R}_0(u)) \subseteq \mathcal{R}(\tilde{g}_\theta(u))$.

Note that for the computation of Δ only finitely many primes need to be considered: either u has exact finite part or we only need to consider primes in $\mathcal{P}(u)$. We implemented the map $\tilde{g}_\theta$ as `shimura_connecting_homomorphism()` in `shimura.py` [6].

From the continuity of arithmetic in $\mathfrak{R}(\widehat{K})$ and the definitions of our conversions, it follows that the following algorithm will terminate.

Algorithm 9.1. Evaluating Shimura's connecting homomorphism.
INPUT: $u \in \mathfrak{R}(J_K)$ and a precision $Q \in \mathbb{Z}_{>0}$.
OUTPUT: a pair $(v, \mathcal{M})$ with $v \in \mathfrak{R}(J_K)$ and $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_2(\widehat{\mathbb{Q}}))$ such that $\mathcal{R}(v) \subseteq \mathcal{R}(u)$ and $m(\mathcal{M}) \geq Q$ and $g_\theta(\mathcal{R}(v)) \subseteq \mathcal{R}(\mathcal{M})$.
ALGORITHM:

1. Set $v := u$.
2. Compute $\mathcal{M} := \tilde{g}_\theta(v)$.
3. If $\mathcal{M}$ does not have precision at least Q, then increase the precision of v at p with $\mathrm{ord}_p(Q) - \mathrm{ord}_p(m(\mathcal{M}))$ (as described in Section 9.3.1) for each prime number p such that $\mathrm{ord}_p(m(\mathcal{M})) < \mathrm{ord}_p(Q)$ and go to Step 2.
4. Output $(v, \mathcal{M})$.

Algorithm 9.1 is implemented in `shimura_connecting_homomorphism()` as well by means of the `output_prec` parameter (see `shimura.py`, [6]).

9.3.3 Factorizing the adèlic matrices

In Section 9.2 we needed to factor matrices $g_\theta(x^{-1}) \in \mathrm{GL}_2(\widehat{\mathbb{Q}})$ for $x \in \widehat{K}^*$ as $g_\theta(x^{-1}) = BA$ with $B \in \mathrm{GL}_2(\widehat{\mathbb{Z}})$ and $A \in \mathrm{GL}_2^+(\mathbb{Q})$. In practice $g_\theta(y)$ will be replaced by $\tilde{g}_\theta(u) \in \mathfrak{R}(\mathrm{GL}_2(\widehat{\mathbb{Q}}))$ for some $u \in \mathfrak{R}(J_K)$. We will use Algorithm 8.4 to compute a factorization in this case as described in Algorithm 9.2 below.

For $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_2(\widehat{\mathbb{Q}}))$ we define the *denominator* $\mathrm{den}(\mathcal{M})$ *of* $\mathcal{M}$ to be the smallest positive integer d such that $d\mathcal{M}$ is integral. For $r \in \mathbb{Q}$ we denote the usual numerator and denominator of r by $\mathrm{num}(r)$ en $\mathrm{den}(r)$.

Algorithm 9.2. Factoring Shimura's connecting homomorphism.
INPUT: $Q \in \mathbb{Z}_{>0}$ and $u \in \mathfrak{R}(J_K)$.
OUTPUT: a pair $(\mathcal{B}, A)$ with $\mathcal{B} \in \mathfrak{R}(\mathrm{GL}_2(\widehat{\mathbb{Q}}))$ of precision at least Q and $A \in \mathrm{GL}_2^+(\mathbb{Q})$ such that for some $x \in \mathcal{R}_0(u)$ we have $g_\theta(x) \in \mathcal{R}(\mathcal{B})A$ and $\mathcal{R}(\mathcal{B}) \subseteq \mathrm{GL}_2(\widehat{\mathbb{Z}})$.
ALGORITHM:
1. Set $P_0 := 1$ and perform Algorithm 9.1 on u with precision P_0, obtaining u_1 and $\mathcal{M}_1$ such that $\mathcal{R}(u_1) \subseteq \mathcal{R}(u)$ and $\mathcal{M}_1 = \tilde{g}_\theta(u_1)$ has integral precision.
2. Set $P_1 := \mathrm{num}(\det(\mathcal{M}_1))\,\mathrm{den}(\mathcal{M}_1)$ and perform Algorithm 9.1 on u_1 with precision P_1, obtaining u_2 and $\mathcal{M}_2$ such that $\mathcal{R}(u_2) \subseteq \mathcal{R}(u_1)$ and $\mathcal{M}_2 = \tilde{g}_\theta(u_2)$ has precision at least P_1.
3. Perform Algorithm 8.4 on $\mathcal{M}_2$, obtaining $A_0 \in \mathrm{GL}_2^+(\mathbb{Q})$ which satisfies $\mathcal{R}(\mathcal{M}_2 A_0^{-1}) \subseteq \mathrm{GL}_2(\widehat{\mathbb{Z}})$.
4. Set $P_2 := \mathrm{lcm}(Q\,\mathrm{den}(A_0^{-1}), P_1)$ and perform Algorithm 9.1 on u_2 with precision P_2, obtaining u_3 and $\mathcal{M}_3$ such that $\mathcal{R}(u_3) \subseteq \mathcal{R}(u_2)$ and $\mathcal{M}_3 = \tilde{g}_\theta(u_3)$ has precision at least P_2.
5. Perform Algorithm 8.4 on $\mathcal{M}_3$, obtaining $A \in \mathrm{GL}_2^+(\mathbb{Q})$ which satisfies $\mathcal{R}(\mathcal{M}_3 A^{-1}) \subseteq \mathrm{GL}_2(\widehat{\mathbb{Z}})$.
6. Set $\mathcal{B} := \mathcal{M}_3 A^{-1}$ and output $(\mathcal{B}, A)$.

Correctness of Algorithm 9.2. Take an $x \in \mathcal{R}_0(u_3)$ and let $M = g_\theta(x)$. Then $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$ all represent M. Since P_0, P_1 and P_2 are all integral, $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$ have integral precision. Together this gives $\mathrm{den}(\mathcal{M}_k) = \mathrm{den}(M) = \mathrm{den}(\mathcal{M}_\ell)$ and $\det(\mathcal{M}_k) = \det(\mathcal{M}_\ell)$ for $k, \ell \in \{1, 2, 3\}$. It is easily verified that an $\mathcal{M} \in \mathfrak{R}(\mathrm{GL}_2(\widehat{\mathbb{Q}}))$ has good precision (cf. Section 8.2) if $m(\mathcal{M}) \geq \det(\mathcal{M})\,\mathrm{den}(\mathcal{M})$. Hence from the definitions of P_1 and P_2 it follows that $\mathcal{M}_2$ and $\mathcal{M}_3$ have good precision. Therefore we give correct input to Algorithm 8.4 in Steps 3 and 5 and so A_0 and A satisfy $MA_0^{-1}, MA^{-1} \in \mathrm{GL}_2(\widehat{\mathbb{Z}})$. Now Theorem 8.5 gives $m(\mathcal{B}) \geq Q$. From the definition of $\mathcal{B}$ and the fact that $M \in \mathcal{R}(\mathcal{M}_3)$ it is clear that $g_\theta(x)A^{-1} = MA^{-1} \in \mathcal{R}(\mathcal{B}) \subseteq \mathrm{GL}_2(\widehat{\mathbb{Z}})$. We conclude that Algorithm 9.2 is correct. □

We implemented Algorithm 9.2 in `shimura.py` at [6] in the function called `factored_shimura_connecting_homomorphism()`.

In practice we will choose Q to be the level N of our modular function $f \in \mathcal{F}_N$. If $\mathcal{B} \in \mathfrak{R}(\mathrm{GL}_2(\widehat{\mathbb{Q}}))$ has precision at least N and $\mathcal{R}(\mathcal{B}) \subseteq \mathrm{GL}_2(\widehat{\mathbb{Z}})$, then there exists a unique $\mathcal{B}_N \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ such that $\mathcal{R}(\mathcal{B})$ maps to $\{\mathcal{B}_N\}$ under the projection $\mathrm{GL}_2(\widehat{\mathbb{Z}}) \to \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$. We call $\mathcal{B}_N$ the *reduction of* $\mathcal{B}$ *modulo* N and it can be easily obtained from $\mathcal{B}$. For computing the action of $B \in \mathcal{R}(\mathcal{B})$ on f, it suffices to know $\mathcal{B}_N$. Hence Algorithm 9.2 enables us to compute the action of some $x \in \mathcal{R}_0(u)$ on f explicitly in terms of $u \in \mathfrak{R}(J_K)$.

9.3.4 The modular function

In practice a modular function $f \in \mathcal{F}_N$ is implemented as a numerical evaluation function using floating point arithmetic. SageMath already contains such an implementation for the Dedekind η function, which we used to implement our example modular functions, such as the Weber functions $\mathfrak{f}, \mathfrak{f}_1, \mathfrak{f}_2 \in \mathcal{F}_{48}$ (cf. [4], Section 4). See the file `modular.py` at [6].

```
sage: weber_f(1+I)
1.08117828783937 - 0.142339821931318*I
sage: weber_f2(-2+I/3)
1.28245645298445 - 0.740426578354544*I
```

We take the action of $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ on f as input for our computation. This is done by explicitly specifying the action of $(\mathbb{Z}/N\mathbb{Z})^*$ and the action of the matrices

$$S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

in $\mathrm{SL}_2(\mathbb{Z})$ on f. These two matrices generate $\mathrm{SL}_2(\mathbb{Z})$ and hence (their images) generate $\mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$. Given $U \in \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$ we write U in terms of S and T using the SageMath function `sl2z_word_problem()`.

For example, for any $d \in (\mathbb{Z}/48\mathbb{Z})^*$ we have

$$\mathfrak{f}_2^d = \begin{cases} -\mathfrak{f}_2 & \text{if } d \equiv 3,5 \mod 8; \\ \mathfrak{f}_2 & \text{if } d \equiv 1,7 \mod 8, \end{cases}$$

and the equalities

$$\mathfrak{f}_2^S = \mathfrak{f}_1 \quad \text{and} \quad \mathfrak{f}_2^T = \zeta_{48}^2 \mathfrak{f}_2$$

hold, for $\zeta_{48} = \exp(2\pi i/48) \in \mathbb{C}$. This information is hard-coded into our implementation in the function `print_action_on_weber_f2()` and enables us to compute $\mathfrak{f}_2^B$ for any $B \in \mathrm{GL}_2(\mathbb{Z}/48\mathbb{Z})$.

```
sage: B = matrix(Zmod(48), [[7, 10], [1, 39]])
sage: print_action_on_weber_f2(B)
  f2    ]--> zeta48^-5*f
```

The output should be read as: $\mathfrak{f}_2^B = \zeta_{48}^{-5}\mathfrak{f}$. These kind of expressions enable us to find class invariants, which we do by hand. See Section 9.4 below or [5] for examples.

By evaluating f we obtain numerical approximations to the conjugates of our class invariant α over K, from which in turn we obtain numerical approximations

to coefficients of h_K^α. If the approximations have high enough precision, we can recognize the exact coefficients in K from them using the LLL-algorithm; see for example the function `recognize_polynomial()` in `recip/polynomial.sage` at [24], which is based on Section 7 of [12].

The modular functions we implemented facilitate evaluation in arbitrary precision by means of the `prec` parameter:

```
sage: weber_f(1+I, prec=20) # use 20 bits precision
1.0812 - 0.14234*I
sage: weber_f(1+I, prec=75) # use 75 bits precision
1.081178287839374683366 - 0.142339821931318055124 0*I
```

We did not do a careful analysis of the precision we need to use. We simply increased the precision by hand if no polynomial in $K[X]$ could be recognized from the numerical approximations of the coefficients.

9.4 Numerical example

Let us demonstrate the computation by a numerical example. The whole computation can be followed on a computer as well: see `hilbert.py` at [6]. We redo Example 1 from Section 5 of [5] in our idèlic manner. This example concerns the field $K = \mathbb{Q}(\sqrt{-71})$, whose ring of integers is generated by $\theta = -\frac{1}{2} + \frac{1}{2}\sqrt{-71}$. The minimal polynomial of θ is $X^2 + X + 18$. We embed K in $\mathbb{C}$ with $\theta \in \mathbb{H}$. We take the cube root $\gamma_2 \in \mathcal{F}_3$ of the j-invariant satisfying $\gamma_2(i) = 12$ as our modular function (cf. [5]).

Using PARI [14] we compute that (the image of) $\theta - 1$ is a generator of $(\mathcal{O}/3\mathcal{O})^*/\mathcal{O}^*$. Let b denote the image of $\theta - 1$ in $(\mathcal{O}/3\mathcal{O})^*$. Then the image $u \in \mathfrak{R}(J_K)$ of b under the conversion from $(\mathcal{O}/3\mathcal{O})^*$ to J_K has represented subset

$$\mathcal{R}(u) = \mathbb{C}^* \times (\theta-1)U^1_{\mathfrak{p}_3} \times (\theta-1)U^1_{\mathfrak{q}_3} \times \prod_{\mathfrak{p}\nmid 3} \mathcal{O}^*_{\mathfrak{p}}$$

where $\mathfrak{p}_3$ and $\mathfrak{q}_3$ are the primes of K above 3. Applying Algorithm 9.2 on u, specifying $N = 3$ as our desired precision, results in the output $(\mathcal{B}, A)$ with

$$\mathcal{B} = \left(\begin{pmatrix} 1 \bmod 18 & 36 \bmod 324 \\ 16 \bmod 18 & 17 \bmod 18 \end{pmatrix}, 1 \right) \in \mathfrak{R}(\mathrm{GL}_2(\widehat{\mathbb{Q}})) \quad \text{and} \quad A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

As A is the identity matrix, its action is trivial. The entries of $\mathcal{B}$ are integral and the determinant is 1, so we see that $\mathcal{R}(\mathcal{B}) \subseteq \mathrm{GL}_2(\widehat{\mathbb{Z}})$, as expected. The reduction $\mathcal{B}_3$ of $\mathcal{B}$ to $\mathrm{GL}_2(\mathbb{Z}/3\mathbb{Z})$ satisfies

$$\mathcal{B}_3 = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} S^3 T^{-2} S,$$

with $S = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ and $T = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ the standard generators of $\mathrm{SL}_2(\mathbb{Z})$. We know that $(\mathbb{Z}/3\mathbb{Z})^*$ and S act trivially on γ_2 and $\gamma_2^T = \zeta_3^2\gamma_2$. It follows that we have $\gamma_2^{\mathcal{B}_3} = \zeta_3^2\gamma_2$. Since $\det(\mathcal{B}_3) = 2 \in (\mathbb{Z}/3\mathbb{Z})^*$ we also have $\zeta_3^{\mathcal{B}_3} = \zeta_3^2$ (viewing $\zeta_3 \in \mathcal{F}_3$). We conclude that $\zeta_3\gamma_2$ is left invariant under $\mathcal{B}_3$ and therefore $\alpha = \zeta_3\gamma_2(\theta)$ is left invariant under $\mathrm{Gal}(H_3/H)$. So we have $\alpha \in H$ as our candidate class invariant.

Using PARI we compute that the class group Cl_K is cyclic of order 7, generated by the class $[\mathfrak{p}_2]$ of $\mathfrak{p}_2 = 2\mathcal{O} + \theta\mathcal{O}$. For $i \in \{1, ..., 7\}$ we compute the image $v_i \in \mathfrak{R}(J_K)$ of $[\mathfrak{p}_2]^i$ under the conversion from Cl_K to J_K. We have for example

$$\mathcal{R}(v_1) = \mathbb{C}^* \times \theta\mathcal{O}_{\mathfrak{p}_2}^* \times \prod_{\mathfrak{p} \neq \mathfrak{p}_2} \mathcal{O}_{\mathfrak{p}}^*.$$

We apply Algorithm 9.2 to each v_i, specifying output precision 3. For v_1 this gives $(\mathcal{B}_1, A_1)$ given by

$$\mathcal{B}_1 = \left(\begin{pmatrix} 1 \bmod 18 & 81 \bmod 162 \\ 9 \bmod 18 & 5 \bmod 9 \end{pmatrix}, 1 \right) \quad \text{and} \quad A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}.$$

Based on the obtained factorizations we can compute approximations to $\alpha^{[\mathfrak{p}_2]^i}$ in $\mathbb{C}$. For example, denoting the reduction of $\mathcal{B}_1$ modulo 3 by $\mathcal{B}_{1,3}$ we have

$$\alpha^{[\mathfrak{p}_2]} = (\zeta_3\gamma_2)^{\mathcal{B}_{1,3}}(A_1\theta) = \zeta_3^2\gamma_2(\theta/2) \approx -0.036501034995 - 82.427712003i.$$

From the approximations of the $\alpha^{[\mathfrak{p}_2]^i}$ we obtain an approximation to h_K^α, in which we recognize:

$$\begin{aligned} h_K^\alpha = {} & X^7 + 6745X^6 - 327467X^5 + 51857115X^4 - 2319299751X^3 \\ & + 41264582513X^2 - 307873876442X + 903568991567. \end{aligned}$$

This is the same polynomial found in [5] (modulo a $\pm$-typo in that paper which our computation revealed). For more details on this example and for two bigger examples involving the Weber functions $\mathfrak{f}, \mathfrak{f}_2 \in \mathcal{F}_{48}$, see `hilbert.py` at [6].

9.5 Comparison to Gee and Stevenhagen

Our method of computing Hilbert class fields using Shimura's reciprocity law is very similar to the method explained by Gee and Stevenhagen [5], but we differ in the way actions of idèles on modular functions are computed.

Gee and Stevenhagen translate the idèlic result of Shimura's reciprocity law in terms of ideals for their specific needs. Namely, for finding class invariants they construct a reduction $g_{\theta,N} : (\mathcal{O}/N\mathcal{O})^* \to \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ of Shimura's connection homomorphism modulo N. And for computing minimal polynomials of the class invariants they use the correspondence between ideal classes in Cl_K and primitive reduced quadratic forms of discriminant equal to the discriminant of K. This gives them an explicit formula for an ideal in each ideal class c, from which they derive explicit formulas for $f(\theta)^c$ (see Section 9 of [4] for the explicit formulas and their derivation).

In contrast our computation is based on the direct application of Shimura's reciprocity law. Our computation deals with idèles and adèles directly. Our adèlic matrix factorization algorithm from Chapter 8 enables us to compute the action of an arbitrary K-idèle (given as a sufficiently precise representation of K-idèles) on a modular function.

Which approach one likes better is a matter of taste. One may view the explicit formulas of Gee and Stevenhagen for the specific situation as the easiest. But one may also view our unified way of computing the action of any idèle on a modular function as a cleaner approach.

Concerning the performance, in terms of run times of the algorithms, we did not do any experimentation to compare the two methods. At the core, our computations are very similar: computing generators of $(\mathcal{O}/N\mathcal{O})^*$, enumerating the class group and evaluating the modular functions is done in the same manner. So only the computation of the explicit actions is done differently. We expect Gee and Stevenhagen to be faster in this phase as they use explicit formulas, whereas we perform a general algorithm. Both our implementation of the Shimura connecting homomorphism as the adèlic matrix factorization algorithm are not expensive though: we expect them be fast compared to other stages of our computation. Streng claims in [23] that the performance bottleneck in these kinds of Hilbert class field computations is the evaluation of the modular functions. Therefore, we expect our method to have very similar performance to the method of Gee and Stevenhagen.

At any rate, we think our method provides a viable alternative to the method of Gee and Stevenhagen. Hence this shows the usefulness of our representations of adèles and idèles: they can indeed be used to perform non-trivial computations in number theory.

9.6 Generalizing to CM-fields

The method described above can be generalized by replacing the imaginary quadratic field K with a *CM-field*: a totally imaginary quadratic extension of a totally real number field. For a thorough exposition of computing class fields of CM-fields using Shimura's reciprocity law we refer to [23]. Below we merely sketch the situation for CM-fields and indicate how our method could be generalized to this situation.

A CM-field K has even degree, say $2g$. The role of the upper half plane $\mathbb{H}$ will be taken over by the *Siegel upper half space* $\mathbb{H}_g$ *of genus* g, consisting of symmetric matrices in $\mathbb{C}^{g\times g}$ with positive definite imaginary part. A certain $\tau \in \mathbb{H}_g$, called a *primitive CM-point*, will take the role of θ. The modular function f will be replaced by a *Siegel modular function*, which is a certain type of meromorphic function $\mathbb{H}_g \to \mathbb{C}$.

The group $\mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$ will take the role of the group $\mathrm{GL}_2(\widehat{\mathbb{Q}})$. Their is an action of $\mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$ on the field of Siegel modular functions and this action can be given explicitly for the subgroups $\mathrm{GSp}_{2g}^+(\mathbb{Q})$ and $\mathrm{GSp}_{2g}(\widehat{\mathbb{Z}})$ similarly to what we saw for $\mathrm{GL}_2(\widehat{\mathbb{Q}})$.

The CM-field K has an associated number field K^r called the *reflex field of* K and an *idèlic type norm* $N : \widehat{K^r} \to \widehat{K}$. Also there is a map $\epsilon : \widehat{K}^* \to \mathrm{GL}_{2g}(\widehat{\mathbb{Q}})$ which is very similar to Shimura's connecting homomorphism. The composition $\epsilon \circ N$ maps $\widehat{K^r}^*$ to $\mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$. Now Shimura's reciprocity law for Siegel modular functions gives a relation of the form

$$f(\tau)^x = f^{\epsilon(N(x))^{-1}}(\tau)$$

for $x \in \widehat{K^r}^*$, with x acting via the Artin map on the left hand side.

The values $f(\tau)$ will generate abelian extensions of K^r and therefore the explicit computation of $f(\tau)^x$ for $x \in \widehat{K^r}^*$ will enable one to find class invariants

and their minimal polynomials.

The biggest challenge for using the above reciprocity law directly in computations was the representation of adèles and idèles and the explicit factorization of a matrix $M \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$ into $M = BA$ with $B \in \mathrm{GSp}_{2g}(\widehat{\mathbb{Z}})$ and $A \in \mathrm{GSp}_{2g}^{+}(\mathbb{Q})$. This thesis has provided such representations of adèles and idèles and an appropriate adèlic matrix factorization algorithm. Note that Algorithm 8.9 indeed handles this more general case of $\mathrm{GSp}_{2g}(\widehat{\mathbb{Q}})$ and not just $\mathrm{GL}_2(\widehat{\mathbb{Q}})$. Therefore, we think it should be possible to generalize the method described in this chapter to CM-fields.

The main ingredient that is still missing, since it played no role in the case $g = 1$, is the idèlic type norm $N : \widehat{K^r} \to \widehat{K}$. An implementation of this type norm will be required for generalizing our method to the case $g > 1$. We did not take any efforts to implement such a type norm, nor did we thoroughly check all details required for actually realizing a generalization sketched above. This could be a fruitful future project.

10 References

[1] Ana Cannas da Silva. *Lectures on symplectic geometry*, volume 1764 of *Lecture Notes in Mathematics*. Springer Verlag, Berlin, 2001.

[2] Henri Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics*. Springer Verlag, third corrected edition, 1996.

[3] Henri Cohen. *Advanced Topics in Computational Number Theory*, volume 193 of *Graduate Texts in Mathematics*. Springer Verlag, 2000.

[4] Alice Gee. Class invariants by Shimura's reciprocity law. *Journal de théorie des nombres de Bordeaux*, 11(1):45–72, 1999. `http://eudml.org/doc/248345`.

[5] Alice Gee and Peter Stevenhagen. Generating class fields using Shimura reciprocity. *Lecture Notes in Computer Science*, 1423, 1998. `https://doi.org/10.1007/BFb0054883`.

[6] Mathé Hertogh. Computing with adèles and idèles, a SageMath package, 2021. `https://github.com/mathehertogh/adeles`.

[7] Joris van der Hoeven. Ball arithmetic. `https://hal.archives-ouvertes.fr/hal-00432152`, 2009.

[8] David Hokken. Profinite number theory. Bachelor's thesis, Utrecht University, 2018. `https://dspace.library.uu.nl/bitstream/handle/1874/366790/ProfiniteNumberTheory.pdf?sequence=2&isAllowed=y`.

[9] Jolien Kamphuis. Fibonacci-dekpunten en contracties. Bachelor's thesis, Leiden University, 2019. `https://www.universiteitleiden.nl/binaries/content/assets/science/mi/scripties/bachelor/2018-2019/bsc-scriptie-jolien-kamphuis.pdf`.

[10] Greg Kuperberg. Kasteleyn cokernels. *Electronic Journal of Combinatorics*, 9(1):Research Paper R29, 30 p., 2002. https://arxiv.org/abs/math/0108150.

[11] Serge Lang. *Elliptic Functions*, volume 112 of *Graduate Texts in Mathematics*. Springer Verlag, New York, second edition, 1987. https://doi.org/10.1007/978-1-4612-4752-4.

[12] Hendrik Lenstra. Lattices. In *Algorithmic number theory: lattices, number fields, curves and cryptography*, volume 44 of *MSRI Publications*, pages 127–181, editors Joe Buhler and Peter Stevenhagen. Camebrigde University Press, 2008. http://library.msri.org/books/Book44/files/06hwl.pdf.

[13] Hendrik Lenstra. Profinite Fibonacci numbers. *Nieuw Archief voor Wiskunde*, 5/6(4):297–300, december 2005. http://www.nieuwarchief.nl/serie5/pdf/naw5-2005-06-4-297.pdf.

[14] The PARI Group. *PARI/GP version 2.11.4*. Univ. Bordeaux, 2019. http://pari.math.u-bordeaux.fr/.

[15] Vladimir Platonov and Andrei Rapinchuk. *Algebraic groups and number theory*, volume 139 of *Pure and Applied Mathematics*. Academic Press, Inc., Boston, MA, 1994. Translated from the 1991 Russian original by Rachel Rowen.

[16] Nathalie Revol and Fabrice Rouillier. MPFI, a multiple precision interval arithmetic library, 2001. http://perso.ens-lyon.fr/nathalie.revol/software.html.

[17] The Sage Developers. Elements, parents and categories in sage: a (draft of) primer, 2021. https://doc.sagemath.org/html/en/reference/categories/sage/categories/primer.html.

[18] The Sage Developers. How to implement new algebraic structures in sage, 2021. https://doc.sagemath.org/html/en/thematic_tutorials/coercion_and_categories.html.

[19] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.3.beta6)*, 2021. https://www.sagemath.org.

[20] The Sage Developers. Writing code for sage: General conventions, 2021. https://doc.sagemath.org/html/en/developer/coding_basics.html.

[21] Goro Shimura. *Introduction to the Arithmetic Theory of Automorphic Functions*. Princeton University Press, 1971.

[22] Jana Sotáková. Eta quotients and class fields of imaginary quadratic fields. Master's thesis, Universities of Leiden and Regensburg, 2017. https://www.universiteitleiden.nl/binaries/content/assets/science/mi/scripties/master/algant/2016-2017/thesis_sotakova.pdf.

[23] Marco Streng. An explicit version of Shimura's reciprocity law for Siegel modular functions. https://arxiv.org/abs/1201.0020.

[24] Marco Streng. RECIP, repository of complex multiplication SageMath code, formerly package for using Shimura's reciprocity law, 2011–2020. http://www.math.leidenuniv.nl/~streng/recip/.

[25] Nicolas Thiéry and Jason Bandlow. Tutorial: implementing algebraic structures, 2021. https://doc.sagemath.org/html/en/thematic_tutorials/tutorial-implementing-algebraic-structures.html.