MATHEMATICS OF COMPUTATION
Volume 75, Number 254, Pages 941–981
S 0025-5718(05)01799-0
Article electronically published on December 8, 2005

TWO EFFICIENT ALGORITHMS FOR THE COMPUTATION OF IDEAL SUMS IN QUADRATIC ORDERS

ANDRÉ WEILERT

Abstract. This paper deals with two different asymptotically fast algorithms for the computation of ideal sums in quadratic orders. If the class number of the quadratic number field is equal to 1, these algorithms can be used to calculate the GCD in the quadratic order. We show that the calculation of an ideal sum in a fixed quadratic order can be done as fast as in $\mathbf{Z}$ up to a constant factor, i.e., in $O(\mu(n)\log n)$, where n bounds the size of the operands and $\mu(n)$ denotes an upper bound for the multiplication time of n-bit integers. Using Schönhage–Strassen's asymptotically fast multiplication for n-bit integers, we achieve $\mu(n) = O(n \log n \log\log n)$.

1. Introduction

In this paper we present two asymptotically fast algorithms for the greatest common divisor (GCD) computation in quadratic orders or, generally, for the computation of ideal sums if the class number is not equal to 1 (Algorithms 3.16 $\mathrm{SGCD}_{\mathcal{O}_D}$ and 4.2 $\mathrm{IDEALSUM}_{\mathcal{O}_D}$). We show that the calculation of an ideal sum in a chosen quadratic order can be performed as fast as in $\mathbf{Z}$ up to a constant factor (depending on the chosen order), i.e., in running time $O(\mu(n)\log n)$, where n is the size of the operands and $\mu(n)$ an upper bound for the running time of the multiplication of n-bit integers. It follows from these algorithms that the class number is a much more suitable algebraic invariant than the property of a quadratic order being euclidean. If the quadratic order is a principal domain, a euclidean algorithm with a suitable chosen euclidean function may fail to calculate the generator of an ideal sum (e.g., in the case of quadratic orders with discriminants $D = -19, -43, -67, -163$), however, there always exists such a generator.

At first we give a historical overview of GCD computations in $\mathbf{Z}$ and in quadratic number fields. Then we introduce some terminology regarding quadratic number fields and binary quadratic forms. In Section 2 we generalize Schönhage's technique of a controlled euclidean descent and corresponding algorithm to a newly introduced class of rings, the $\mathcal{S}$-euclidean domains. In Section 3 we apply this concept in order to calculate the sum of two principal ideals (for each of them we know one generator)

Received by the editor July 20, 2003 and, in revised form, January 7, 2005.
2000 *Mathematics Subject Classification.* Primary 54C40, 14E20; Secondary 46E25, 20C20.
Key words and phrases. Computational number theory, quadratic number fields, GCD computation, Euclidean algorithm.
This paper deals with the main results of my doctoral thesis [40]. I would like to thank my academic teachers Arnold Schönhage and Jens Franke (both at the University of Bonn, Germany).

in the ring of algebraic integers of an imaginary quadratic number field with any class number. Apart from the initial calculation of an $\mathcal{S}$-euclidean descent we compute the valuation of ideals at finitely many places, where the places are only depending on the chosen order, and not on the operands. Due to this valuation step our novel Algorithm 3.16 is not very closely related to the concept of a euclidean domain. In Section 4 we present a different approach for the calculation of ideal sums. Our Algorithm 4.2 computes the sum of two ideals (for which we know a suitable coding) in any quadratic order—independent of the class number or the discriminant D. This Algorithm 4.2 is uniformly fast because it does not need any precalculations. It uses the representation of ideals in the Hermite normal form and the correspondence between ideals in quadratic orders and binary quadratic forms. Its running time only depends on the size of the operands and the size of the discrimant.

We present both of these algorithms in this paper because Algorithm 3.16 also calculates a representation of the GCD g, $ax + by = g$, in the case of the five imaginary quadratic norm-euclidean maximal orders, while Algorithm 4.2 does not, but is uniformly fast for all quadratic orders.

1.1. **Historical overview.** The history of efficient GCD computations is based on Euclid's algorithm [14, Book VII, Propositions 1 and 2] (about 330 B.C.) that does not need any factorization of the integers. The algorithm calculates euclidean steps, i.e., divisions with remainders, as long as the remainder is not equal to zero. One can show that every step of the euclidean descent is reducing the size of the operands by at least a factor larger than 1. The last remainder which is not equal to zero is the greatest common divisor of the operands. One can calculate cofactors in the euclidean steps in order to represent the GCD as a linear combination of the operands.

Improvements to Euclid's algorithm were made only in the last 60 years due to the possibility of using computers for calculations. Lehmer [22] presented an improved version of Euclid's algorithm that calculates euclidean steps in single precision using the top-bits of the operands as long as possible. Another GCD algorithm is Stein's binary algorithm [36] (or, [19, Section 4.5.2, Algorithm B]) that uses only addition, subtraction and shifting (division by powers of two). This algorithm has a running time of $O(n^2)$ if the inputs are n-bit integers. There exist many further improvements to these algorithms, but none of these achieve a nearly linear running time. For a more detailed overview, we refer the reader to [40, Abschnitt 1.2].

In 1971 Schönhage and Strassen [32] presented an asymptotically fast algorithm using FFT methods for the multiplication of n-bit integers which achieves a running time of $\mu(n) = O(n \log n \log\log n)$. Knuth [18], who used this fast multiplication, found an efficient GCD algorithm with running time $O(\mu(n)\cdot(\log n)^4)$. Based on these algorithms, in 1971 Schönhage [28] used the correspondence between the euclidean descent and the continued fraction decomposition and developed an asymptotically fast GCD algorithm with running time $O(\mu(n) \log n)$. Until Schönhage implemented his GCD algorithm, it was doubted that this algorithm could be faster than the other known algorithm in practice; see [31, Sections 1.3.6, 6.1.3] and [29]. Schönhage used for his implementation the so-called technique of a controlled euclidean descent instead of the correspondence to the continued fraction decomposition. Later he was able to transfer this concept to the fast reduction of

binary quadratic forms [30]. Recently, Stehlé and Zimmermann [35] presented another asymptotically fast GCD algorithm for integers with the same asymptotical running time. It would be nice to also have a comparison of its computation time with Schönhage's implementation [31, Sections 1.3.6, 6.1.3].

Up to now we discussed GCD algorithms for integers, while the GCD can be defined in a more general context. A study, started by Gauß and finished in 1952, showed that only 21 rings of algebraic integers are euclidean domains w. r. t. the algebraic norm. Lemmermeyer [23] studied more generally for which of the number fields a euclidean algorithm is applicable. GCD computations in the ring of algebraic integers, which can be made in practice, were studied only in the last 30 years. Most of these algorithms are transferred from the ring of integers to the maximal order of a number field, especially to the ring of Gaussian integers $\mathbf{Z}[i]$. Caviness and Collins [6, 7] transferred Lehmer's GCD algorithm to $\mathbf{Z}[i]$.

We were able to specify an analogue to the binary algorithm for the Gaussian integers [38] that achieves a quadratic running time. Instead of the prime number 2 in case of the binary algorithm, we use powers of the prime element $1+i$ in order to reduce the operands. Based on the practical running time tests, Collins [11] presented a faster GCD algorithm for $\mathbf{Z}[i]$ with quadratic running time than the so-called $(1+i)$-ary algorithm. It calculates euclidean steps, but not necessarily least remainder euclidean steps, and in this sense, his algorithm is called "approximative". Furthermore, Schönhage's asymptotically fast GCD algorithm for integers can be transferred to the ring of Gaussian integers [39] such that there exists a GCD algorithm in $\mathbf{Z}[i]$ with running time $O(\mu(n)\log n)$, if the Gaussian numbers have length $O(n)$. This algorithm can also be used to calculate a quotient sequence and cofactors, from which we are able to compute the biquadratic residue symbol of the Gaussian numbers in linear time [41]. Moreover it is easy to see that this kind of GCD algorithm can be generalized to the ring of algebraic integers of the five norm-euclidean imaginary quadratic number fields (discriminants $-3, -7, -8, -11$, and -4, i.e., the ring of Gaussian integers).

Kaltofen and Rolletschek [16] studied GCD algorithms in the ring of algebraic integers of quadratic number fields with class number 1. In particular, they showed that there does not in general exist a sequence of euclidean steps (even if it is not always norm-decreasing) for discriminant $D \leq -19$ such that the GCD can be calculated in this way. Their algorithm requires quadratic running time in the size of the operands, but there are some expensive, but necessary, precalculations (e.g., the fundamental unit) in the case of real quadratic integers. For that reason their algorithm is not fast in a uniform manner for every quadratic maximal order with class number 1. The authors generalize their algorithm to all quadratic rings of algebraic integers, independent of the class number [17]. If the class number is not equal to one, then their algorithm calculates the "GCD" (in other words, the sum of the two principal ideals, generated by the two operands) as a canonical representative of the class group (all these representatives have to be precalculated) and a principal ideal. In addition to that, they presented another GCD algorithm with cubic running time for the four imaginary quadratic maximal orders with discriminant $D \leq -19$ and class number 1. This algorithm is based on lattice reduction and does not need any precalculations. Altogether, their algorithms are only able to calculate the sum of two principal ideals, and not of two ideals in general.

Another approach to the GCD computation is to consider the ideals of a quadratic order as $\mathbf{Z}$-modules of rank 2. Then one can use the concept of "Hermite normal form" (HNF) in order to compute ideal sums; see [13] and [8, Section 2.4.3]. It is possible to generalize the HNF concept to Dedekind domains [9] that can be used for a coding of ideals in relative extensions [10, Chapter 1].

1.2. **Notations.** Now we give a brief overview of quadratic number fields. More details and proofs for the following definitions and statements can be found, e.g., in [8, 15, 20, 24, 26].

We denote by $\mathbf{Z}, \mathbf{Q}, \mathbf{R}$ and $\mathbf{C}$ the ring of integers, the field of rational numbers, the field of real numbers, and the field of complex numbers, respectively.

Let K be a quadratic number field, i.e., $K \cong \mathbf{Q}(\sqrt{d})$ for a squarefree $d \in \mathbf{Z}$, $d \neq 0, 1$. Denote by σ the nontrivial field automorphism of K, and define the *norm* $\mathrm{Norm}: K \to \mathbf{Q}$ and *trace* $\mathrm{Tr}: K \to \mathbf{Q}$ by

$$\mathrm{Norm}(\alpha) = \alpha \cdot \sigma(\alpha), \quad \mathrm{Tr}(\alpha) = \alpha + \sigma(\alpha), \qquad \text{for } \alpha \in K.$$

The *ring of algebraic integers* of K, denoted by $\mathcal{O}_K$, consists of the $\alpha \in K$ such that α is a zero of a monic quadratic polynomial with coefficients in $\mathbf{Z}$. An *order* $\mathcal{O}$ in K is a subring of $\mathcal{O}_K$ with $1 \in \mathcal{O}$ and with field of fractions K. Every order $\mathcal{O}$ in K satisfies $\mathbf{Z} \subset \mathcal{O} \subset \mathcal{O}_K$ and is a free $\mathbf{Z}$-module of rank 2. Due to this inclusion, the order $\mathcal{O}_K$ is called the *maximal order* of K. The index of $\mathcal{O}$ in $\mathcal{O}_K$ is finite and called the *conductor*. Every positive integer f occurs as the conductor of an order $\mathcal{O}$ in K, namely $\mathcal{O} = \mathbf{Z} + f\mathcal{O}_K$. If $\mathcal{O} = e_1\mathbf{Z} + e_2\mathbf{Z}$, then the *discriminant* D of $\mathcal{O}$ is defined by $D = (e_1 \cdot \sigma(e_2) - e_2 \cdot \sigma(e_1))^2$; this is an integer which does not depend on the choice of the basis e_1, e_2. We have $D = f^2 \cdot D_0$, where f is the conductor of $\mathcal{O}$ and D_0 is the discriminant of $\mathcal{O}_K$; we also call D_0 the discriminant of K. The integer D is not a square, and $D \equiv 0$ or $1 \bmod 4$. Conversely, any nonsquare integer D that is 0 or $1 \bmod 4$ is the discriminant of a uniquely determined order in a quadratic field, namely $\mathcal{O} = \mathbf{Z}[(D + \sqrt{D})/2] \subset K$. We denote the unique order in K with discriminant D by $\mathcal{O}_D$. If D is a positive (negative) integer, then we call the order $\mathcal{O}_D$ *real quadratic* (*imaginary quadratic*).

We call the discriminant D_0 of a quadratic number field K a *fundamental discriminant.* This means that $D_0 \neq 1$ and either $D_0 \equiv 1 \bmod 4$ and is squarefree, or $D_0 \equiv 0 \bmod 4$, $D_0/4$ is squarefree and $D_0/4 \equiv 2$ or $3 \bmod 4$. Then we denote the unique quadratic number field with discriminant D_0 by $\mathbf{Q}(\sqrt{D_0})$. A $\mathbf{Z}$-basis of the maximal order $\mathcal{O}_{D_0} = \mathcal{O}_K$ is called an *integral basis* of K.

Let $\mathcal{O}_D$ be a quadratic order with a discriminant D in the quadratic number field K. An (integral) *ideal* H is an $\mathcal{O}_D$-submodule of $\mathcal{O}_D$. H can be considered as a $\mathbf{Z}$-module of rank 2 as well. We call I a *fractional ideal* if there exists a positive integer d such that dI is an integral ideal. The ring $\mathcal{O}_D/I$ is finite, and its cardinality is called the *norm* $\mathrm{Norm}(I)$ of the ideal I.[1] If there exists a fractional ideal I' in $\mathcal{O}_D$ with $I \cdot I' = \mathcal{O}_D$, then we call I an *invertible* ideal. We call two fractional ideals $I \neq 0$ and J equivalent if there exists an $\alpha \in K, \alpha \neq 0$ such that $J = \alpha I$. Then we define the *class group* $Cl(\mathcal{O}_D) = Cl(D)$ as the set of equivalence classes of invertible ideals in $\mathcal{O}_D$, and the *class number* $h(\mathcal{O}_D) = h(D)$ as its cardinality (the class group is always a finite group).

[1] In the case of a principal ideal $\alpha\mathcal{O}_D$ the two norm definitions may only differ in the sign: $\mathrm{Norm}(\alpha\mathcal{O}_D) = |\mathrm{Norm}(\alpha)|$.

If D is a fundamental discriminant, i.e., $\mathcal{O}_D$ is the maximal order of a quadratic number field, then we call $Cl(\mathcal{O}_K)$ the class group and $h(\mathcal{O}_K)$ the class number of K. It is known that there exist only nine imaginary quadratic number fields with $h(D) = 1$, namely $D = -3, -4, -7, -8, -11, -19, -43, -67, -163$. On the other hand, real quadratic number fields generally seem to have small class numbers, but it is not known whether there exist infinitely many real quadratic number fields with class number 1.

The maximal order $\mathcal{O}_K$ is always a Dedekind domain, which does not hold true for every quadratic order. In particular, every fractional ideal I can be written in a unique way as

$$I = \prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(I)},$$

the product being over a finite set of prime ideals, and the exponents $v_{\mathfrak{p}}(I)$ being in $\mathbf{Z}$ (nonnegative if I is an integral ideal). Let D be a fundamental discriminant. We can classify the prime ideals in $\mathcal{O}_D$ (which are all lying over a prime number p in $\mathbf{Z}$) based on the value of the Jacobi symbol (D/p): Let p be a prime number. If $(D/p) = -1$, then p is inert and $\mathfrak{p} = p\mathcal{O}_D$; if $(D/p) = 0$, then p is ramified and $p\mathcal{O}_D = \mathfrak{p}^2$; if $(D/p) = +1$, then p is split and $p\mathcal{O}_D = \mathfrak{p}_1\mathfrak{p}_2$.

As mentioned above, we can represent every integral ideal $\mathfrak{a}$ in a quadratic order as a $\mathbf{Z}$-module of rank 2. Furthermore, there exists a unique matrix in $\mathbf{Z}^{2\times 2}$ w. r. t. a chosen integral basis $(1, \omega)$, the Hermite normal form:

$$\begin{pmatrix} a & b \\ 0 & c \end{pmatrix}, \quad \text{where } a, b, c \in \mathbf{Z},\ c \text{ divides both } a \text{ and } b, \text{ and } 0 \le b < a. \tag{1.1}$$

The ideal $\mathfrak{a}$ has $\mathfrak{a} = a\mathbf{Z} + (b + c\omega)\mathbf{Z}$ as its $\mathbf{Z}$-module representation. Moreover, a is the smallest positive integer in $\mathfrak{a}$, and $\mathrm{Norm}(\mathfrak{a}) = ac$ [8, Proposition 5.2.1].

Now we would like to introduce some terminology related to binary quadratic forms which was already introduced by Gauß [12, Section 5]. A *binary quadratic form* f is a homogeneous quadratic polynomial in two variables

$$f(x, y) = ax^2 + bxy + cy^2 = (x, y) \begin{pmatrix} a & \frac{1}{2}b \\ \frac{1}{2}b & c \end{pmatrix} (x, y)^T, \quad a, b, c \in \mathbf{Z},$$

with discriminant $D = b^2 - 4ac$. Instead of f we write (a, b, c) or $(a, b, *)$, because c is determined by a, b and D. We call f *primitive* if the greatest common divisor of a, b, c is equal to 1.

Define the operation of a matrix $M = \left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right) \in \mathrm{PSL}_2(\mathbf{Z})$ for a quadratic form f (identifying a matrix $M \in \mathrm{SL}_2(\mathbf{Z})$ with $-M$ because the operation of $-M$ is the same as the operation of M) as

$$M{\cdot}f = M{\cdot}f(x, y) := f((x, y)M^T) = f(\alpha x + \beta y, \gamma x + \delta y).$$

Two quadratic forms $f = (a, b, c)$ and $f' = (a', b', c')$ are called *equivalent* ($f \sim f'$) if there exists an $M \in \mathrm{PSL}_2(\mathbf{Z})$ such that $M{\cdot}f = f'$. This can also be written as

$$M^T{\cdot} \begin{pmatrix} a & \frac{1}{2}b \\ \frac{1}{2}b & c \end{pmatrix} {\cdot} M = \begin{pmatrix} a' & \frac{1}{2}b' \\ \frac{1}{2}b' & c' \end{pmatrix}. \tag{1.2}$$

The *module group* $\Gamma := \mathrm{PSL}_2(\mathbf{Z})$ is generated by the matrices

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

Now we show the correspondence between quadratic forms of discriminant D and ideals in the quadratic order with discriminant D. In particular, the multiplication of ideals corresponds to the composition of binary quadratic forms. The $\mathbf{Z}$-module $a\mathbf{Z} + \frac{-b+\sqrt{D}}{2}\mathbf{Z}$ is an integral ideal in $\mathcal{O}_D$ if and only if $4a$ divides $b^2 - D$. Let $\Gamma_\infty := \{S^m : \ m \in \mathbf{Z}\} \subset \Gamma$ be a multiplicative subgroup of the module group that operates on quadratic forms. Then we define

$$\mathcal{F}(D) := \{(a,b,c) : \ D = b^2 - 4ac\}, \qquad F(D) := \mathcal{F}(D)/\Gamma_\infty$$

as sets of quadratic forms with discriminant D. Denote the set of fractional ideals in $\mathcal{O}_D$ by $\mathcal{I}(D)$. Define $I(D) := \mathcal{I}(D)/\mathbf{Q}^\times$, where $\mathbf{Q}^\times$ acts multiplicatively on fractional ideals. We can define mappings

$$\phi_{FI} : \mathcal{F}(D) \to \mathcal{I}(D) \times \mathbf{Z}/2\mathbf{Z}, \quad \phi_{FI}(a,b,c) = \left(a\mathbf{Z} + \frac{-b+\sqrt{D}}{2}\mathbf{Z}, \ \operatorname{sgn} a \right), \tag{1.3}$$

$$\phi_{IF} : \mathcal{I}(D) \times \mathbf{Z}/2\mathbf{Z} \to \mathcal{F}(D), \qquad \phi_{IF}(\mathfrak{a}, s) = s \cdot \frac{\operatorname{Norm}(x\omega_1 - sy\omega_2)}{\operatorname{Norm}(\mathfrak{a})}, \tag{1.4}$$

where (ω_1, ω_2) is a $\mathbf{Z}$-basis of $\mathfrak{a}$ with $\omega_1 \in \mathbf{Q}$ (always possible, e.g., using a HNF representation of $\mathfrak{a}$) and $(\omega_2\sigma(\omega_1) - \omega_1\sigma(\omega_2))/\sqrt{D} > 0$. These mappings are inverse isomorphisms and induce isomorphisms on the level of the equivalence classes:

$$\phi_{FI} : \ F(D) \to I(D) \times \mathbf{Z}/2\mathbf{Z}, \qquad \phi_{IF} : \ I(D) \times \mathbf{Z}/2\mathbf{Z} \to F(D).$$

We call a quadratic form $(a,b,c) \in \mathcal{F}(D)$ reduced if

$$\begin{cases} |\sqrt{D} - 2|a|| < b < \sqrt{D}, & \text{if } D > 0, \\ |b| \le a \le c \text{ and, additionally, } b \ge 0 \text{ if } |b| = a \text{ or } a = c, & \text{if } D < 0. \end{cases}$$

Denote the set of reduced forms by $\mathcal{R}(D)$. There exists an equivalent reduced form for every quadratic form. The mapping

$$\rho : \ \mathcal{F}(D) \to \mathcal{F}(D), \quad \rho(a,b,c) = (c, b', c'), \qquad b' \in J_{c'}, \ b' \equiv -b \mod 2c,$$

$$\text{where } J_w := \begin{cases} \{x \in \mathbf{R} : \ -|w| < x \le |w|\}, & \text{if either } D < 0 \\ & \quad \text{or } D > 0 \text{ and } |w| \ge \sqrt{D}, \\ \{x \in \mathbf{R} : \ \sqrt{D} - 2|w| < x \le \sqrt{D}\}, & \text{if } D > 0 \text{ and } |w| < \sqrt{D}, \end{cases}$$

is reducing, i.e., after finitely many steps we have calculated an equivalent reduced form. If the discriminant is negative, there exists exactly one equivalent reduced form for every quadratic form, which does not hold true for $D > 0$. In the case of a positive discriminant there exists more than one equivalent reduced form for a quadratic form. The restriction of ρ to $\mathcal{R}(D)$ is a permutation of $\mathcal{R}(D)$; the ρ-orbits of $\mathcal{R}(D)$ are called cycles. Every cycle contains an even number of elements because the sign of the first coefficient alternates. Two reduced forms are equivalent if and only if they belong to the same cycle.

Schönhage [30] showed that the reduction of a binary quadratic form with coefficients bounded by 2^n can be calculated in running time $O(\mu(n) \log n)$ by using his technique of a controlled descent. In particular, the sign of the discriminant D does not significantly affect the calculation steps in Schönhage's algorithm.

Definition 1.1. We call the quadratic form $(a,b,c) \in \mathcal{F}(D)$ *minimal above* s if

$$a, \tfrac{1}{2}b, c \ge s \quad \text{and} \quad \left(a - b + c < s \text{ or } \left(\tfrac{1}{2}b - a < s \text{ and } \tfrac{1}{2}b - c < s\right)\right).$$

This defines a measure for the size of the operands. Schönhage's algorithm for fast reductions of binary quadratic forms is based on the following theorem.

Theorem 1.2 (Schönhage)**.** *Let $s > 0$, and let $(a, b, c) \in \mathcal{F}(D)$ be a quadratic form with $a, \frac{1}{2}b, c \geq s$. Then there exists a uniquely defined matrix $M \in \Gamma$, $M \geq 0$, and a unique defined quadratic form $(\alpha, \beta, \gamma) \in \mathcal{F}(D)$ minimal above s such that $(a, b, c) = M \cdot (\alpha, \beta, \gamma)$. The coefficients of $M = \left(\begin{smallmatrix} u & u' \\ v & v' \end{smallmatrix}\right)$ are bounded by*

$$(u + u')^2 \leq a/s, \quad 2(u + u')(v + v') \leq b/s, \quad (v + v')^2 \leq c/s.$$

If the coefficients a, b, c are bounded by 2^n, then the running time of the reduction to α, β, γ can be bounded by $O(\mu(n) \log n)$, independently of s.

Remark 1.3. We refer to [30, Lemma 1, Lemma 2] for the proof. We can extend the fast reduction as above to any quadratic form $f \in \mathcal{F}(D)$ as we can calculate an equivalent quadratic form $\hat{f} \sim f$ with only positive (and not too large) coefficients using at most two actions of T and S^m (cf. [30, § 4]). Now we can apply an algorithm according to Theorem 1.2 (cf. [30, Algorithm "MR"] or [40, Algorithm 3.2 "REDUCTION"]) for the form $\hat{f}$ with parameter $s = \frac{1}{2}$. We obtain an equivalent quadratic form $\tilde{f} = (\tilde{a}, \tilde{b}, \tilde{c})$ minimal above $\frac{1}{2}$ as a result with $\tilde{a}, \tilde{b}, \tilde{c} \geq 1$. All these calculations are independent of the sign of the discriminant D. The calculation of the/an equivalent reduced quadratic form can be done with at most two actions of T and S^m. Altogether, we are able to compute a complete reduction of the quadratic form $f = (a, b, c)$ with coefficients $|a|, |b|, |c| < 2^n$ to an equivalent reduced form in running time $O(\mu(n) \log n)$.

Denote the canonical reduced form with discriminant D with $\phi_{FI}(\mathbf{1}_D) = (\mathcal{O}_D, 1)$ by $\mathbf{1}_D$. This form has the coefficients $(1, b, (D - b^2)/4)$, where b is the greatest integer with the same parity as D less than or equal to 1 if $D < 0$, and to $\lfloor \sqrt{D} \rfloor$ if $D > 0$.

2. Concept of $\mathcal{S}$-euclidean domains

Now we generalize the class of euclidean domains to the class of $\mathcal{S}$-euclidean domains which contains euclidean domains as a special case. This generalization is based on algorithmic aspects such that we are able to present the asymptotically fast Algorithm 2.19 (DESCENT$_R$) that computes a controlled $\mathcal{S}$-euclidean descent in an $\mathcal{S}$-euclidean domain where the triangle inequality is satisfied for the $\mathcal{S}$-euclidean function. After this general introduction to $\mathcal{S}$-euclidean domains we will focus only on quadratic orders and apply the novel concept.

2.1. Generalization of euclidean domains.

While we now introduce the concept of $\mathcal{S}$-euclidean domains, we show in the next section that all the rings of algebraic integers of imaginary quadratic number fields are $\mathcal{S}$-euclidean.

Definition 2.1. Let R be an integral domain, and let $\mathcal{S} \subset R \setminus \{0\}$ be a *finite* subset. We call R *$\mathcal{S}$-euclidean* w. r. t. a euclidean function $f : R \to \mathbf{R}_{\geq 0}$ if

(S1) $f(x) = 0 \Leftrightarrow x = 0$.
(S2) There exist $q \in R$ and $s \in \mathcal{S}$ for all $x, y \in R \setminus \{0\}$ with $f(s \cdot x - qy) < f(y)$.
(S3) The set $\{f(x) : x \in R,\ f(x) < \kappa\}$ is finite for every $\kappa > 0$.

Remark 2.2.

(1) The finiteness of $\mathcal{S}$ ensures a practical calculation of an $\mathcal{S}$-euclidean step as in (S2) if there exists a computable division with remainder in R, because one can calculate the division for every $s \in \mathcal{S}$.
(2) If R is $\{1\}$-euclidean, then R is euclidean.
(3) The conditions (S1), (S2) and (S3) do not necessarily imply that R is a unique factorization domain (which is true if R is euclidean).

Definition 2.3. Let R be an $\mathcal{S}$-euclidean domain w. r. t. f. Then there exist $s \in \mathcal{S}$ and $q, r \in R$ for all $x, y \in R \setminus \{0\}$ according to (S2) such that $s{\cdot}x = qy + r$ with $f(r) < f(y)$. We call such a division step an $\mathcal{S}$-*euclidean step* (not unique in general).

Definition 2.4. Let R be an integral domain, let $f\colon\ R \to \mathbf{R}_{\geq 0}$, and let $\mathcal{S} \subset R \setminus \{0\}$ be a finite subset. Assume that (S1) and (S3) are satisfied. Then we define the $\mathcal{S}$-*euclidean minimum* of R w. r. t. f as

$$E(R, \mathcal{S}, f) := \inf\{\kappa > 0 :\ \forall x, y \in R \setminus \{0\}\ \exists\, q \in R, s \in \mathcal{S} :\ f(s{\cdot}x - qy) < \kappa{\cdot}f(y)\}.$$

If $E(R, \mathcal{S}, f) < 1$, then R is $\mathcal{S}$-euclidean w. r. t. f.

We now show that the cofactors of an $\mathcal{S}$-euclidean descent are bounded w. r. t. the $\mathcal{S}$-euclidean function f. Assume that f is multiplicative, satisfies the triangle inequality, and $f(x) = 0$ or $f(x) \geq 1$ for $x \in R$. Denote the field of fractions with $K := \mathrm{Quot}(R)$. Let $x, y \in R$. Without loss of generality we assume $f(x) \geq f(y)$. Set $x_0 := x$, $x_1 := y$ as starting values. Then, for $1 \leq j \leq r$,

(2.1)
$$s_j{\cdot}x_{j-1} = q_j x_j + x_{j+1}, \quad \text{where } s_j \in \mathcal{S},\ q_j, x_{j+1} \in R \text{ and } f(x_{j+1}) \leq E_R{\cdot}f(x_j)$$

is an $\mathcal{S}$-euclidean step w. r. t. x_{j-1}, x_j. Thereby, let $E_R < 1$ be a good upper bound for the $\mathcal{S}$-euclidean minimum with $0 < E(R, \mathcal{S}, f) \leq E_R < 1$ such that one can calculate every single $\mathcal{S}$-euclidean step in an efficient manner. An $\mathcal{S}$-euclidean descent is a sequence of $\mathcal{S}$-euclidean steps, as long as the remainder x_{j+1} does not equal zero. It follows from (2.1) and (S3), that this sequence is always finite, i.e., $x_{r+1} = 0$ for a minimal $r \in \mathbf{N}$. We can rewrite an $\mathcal{S}$-euclidean step using a 2×2-matrix as

$$\begin{pmatrix} s_j & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{j-1} \\ x_j \end{pmatrix} = \begin{pmatrix} q_j & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_j \\ x_{j+1} \end{pmatrix}, \quad \text{where } q_j \in R, s_j \in \mathcal{S}.$$

The matrix on the left-hand side is not generally invertible in R because $s_j \in \mathcal{S}$ is not necessarily a unit. Another representation for this $\mathcal{S}$-euclidean step is

$$Q_j{\cdot}\begin{pmatrix} x_{j-1} \\ x_j \end{pmatrix} = \begin{pmatrix} x_j \\ x_{j+1} \end{pmatrix},\ Q_j := \begin{pmatrix} 0 & 1 \\ s_j & -q_j \end{pmatrix},\ \det Q_j = -s_j,\ Q_j \in R^{2\times 2}. \tag{2.2}$$

The matrix Q_j is not generally invertible in R, but always in K because the determinant does not vanish due to $0 \notin \mathcal{S}$.

Define the product of the matrices Q_κ and the product of the $\mathcal{S}$-factors s_κ as $M_j \in R^{2\times 2}$ and, respectively, $S_j \in R \setminus \{0\}$ for $1 \leq j \leq r$. Set M_0 as the identity matrix and $S_0 := 1$. Setting $M_j = Q_j{\cdot}M_{j-1}$ and $S_j = s_j{\cdot}S_{j-1}$ yield $\det M_j = (-1)^j{\cdot}S_j$. From a representation of M_j as

$$M_j = (-1)^j{\cdot}\begin{pmatrix} v_j & -u_j \\ -v_{j+1} & u_{j+1} \end{pmatrix}, \tag{2.3}$$

it follows that

$$u_0 = 0, \quad u_1 = 1, \quad u_{j+1} = q_j u_j + s_j u_{j-1}, \tag{2.4}$$

$$v_0 = 1, \quad v_1 = 0, \quad v_{j+1} = q_j v_j + s_j v_{j-1}. \tag{2.5}$$

Therefore we get the representation

$$M_j \cdot \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} x_j \\ x_{j+1} \end{pmatrix}, \tag{2.6}$$

and from this (using the K-inverse matrix M_j^{-1}) we get a representation of x_0, x_1 as

$$\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = M_j^{-1} \cdot \begin{pmatrix} x_j \\ x_{j+1} \end{pmatrix} = \frac{1}{S_j} \cdot \begin{pmatrix} u_{j+1} & u_j \\ v_{j+1} & v_j \end{pmatrix} \begin{pmatrix} x_j \\ x_{j+1} \end{pmatrix}. \tag{2.7}$$

Now we are going to show that M_j's coefficients are bounded w. r. t. the $\mathcal{S}$-euclidean function f. To prove that we use the notations as introduced above. Under these assumptions it is easy to show that $f(\varepsilon) = 1$ for every R-unit ε, and that f can be canonically extended to the field of fractions, $f : K \to \mathbf{R}_{\geq 0}$.

Lemma 2.5. *Let $1 \leq j \leq r$, and let q_j be a quotient of an $\mathcal{S}$-euclidean descent. Then $f(q_j) \geq 1$.*

Proof. We have $f(x_j) \leq f(x_{j-1})$, because in case of $j = 1$ we assumed $f(x_0) \geq f(x_1)$, and in case of $j \geq 2$ we calculated x_j in an $\mathcal{S}$-euclidean step. Considering the $\mathcal{S}$-euclidean step $s_j \cdot x_{j-1} = q_j x_j + x_{j+1}$ with $f(x_{j+1}) < f(x_j)$, it follows from the multiplicativity of f that $f(s_j \cdot x_{j-1}) \geq f(x_{j-1}) \geq f(x_j) > f(x_{j+1})$. Assume that $q_j = 0$. This yields $s_j \cdot x_{j-1} = x_{j+1}$, which is a contradiction to the inequality above. Therefore we have $q_j \neq 0$, thus $f(q_j) \geq 1$, because f avoids every value between 0 and 1. □

Lemma 2.6. *$u_j \neq 0$ for $1 \leq j \leq r$, and $v_j \neq 0$ for $2 \leq j \leq r$.*

Proof. We defined $u_1 = 1 \neq 0$. Thus we have to show the statements for u_j, v_j only for $2 \leq j \leq r$. Using (2.6) we get $x_j = (-1)^j (v_j x_0 - u_j x_1)$. Assume $v_j = 0$. Then $u_j x_1 (-1)^{j+1} = x_j \neq 0$. Thus $f(x_1) \leq f(x_j)$, because $f(u_j) \geq 1$ and $f(-1) = 1$. But we have $f(x_j) < f(x_1)$ for $2 \leq j \leq r$, because x_j is calculated by an $\mathcal{S}$-euclidean step. We conclude that $f(x_1) < f(x_1)$, which is a contradiction. The proof for $u_j \neq 0$ for $2 \leq j \leq r$ is almost the same. □

Definition 2.7. Set $U_j := \dfrac{u_{j+1}}{u_j}$ for $1 \leq j \leq r$, and set $V_j := \dfrac{v_{j+1}}{v_j}$ for $2 \leq j \leq r$.

Remark 2.8. U_j, V_j are well defined because the occurring denominators do not vanish (Lemma 2.6). We calculate $U_1 = q_1$, $U_2 = q_2 + s_2/q_1$, $V_2 = q_2$, and $U_j = q_j + s_j/U_{j-1}$ and $V_j = q_j + s_j/V_{j-1}$ for $3 \leq j \leq r$ (cf. (2.4), (2.5)).

Lemma 2.9. *Assume $f(V_j) \geq 1$ for $2 \leq j \leq r$. Then $f(v_{j+1}) \leq \frac{1}{1 - E_R} \cdot f(S_j \cdot x_1 / x_j)$.*

Proof. We have $S_j \cdot x_1 = v_{j+1} x_j + v_j x_{j+1}$ in R using (2.7). Dividing this equation by x_j (does not vanish for $2 \leq j \leq r$) and separating v_{j+1}, we get

$$\frac{S_j \cdot x_1}{x_j} = v_{j+1} \cdot \left(1 + \frac{v_j}{v_{j+1}} \cdot \frac{x_{j+1}}{x_j}\right) = v_{j+1} \cdot \left(1 + V_j^{-1} \cdot \frac{x_{j+1}}{x_j}\right).$$

Applying the $\mathcal{S}$-euclidean function f to this equation yields

$$f(v_{j+1}) = \frac{f\left(S_j \cdot \frac{x_1}{x_j}\right)}{f\left(1 + V_j^{-1} \cdot \frac{x_{j+1}}{x_j}\right)}. \tag{2.8}$$

Because of $f\left(\frac{x_{j+1}}{x_j}\right) \leq E_R$ and $f(V_j^{-1}) \leq 1$, we can bound the denominator as follows:

$$f\left(1 + V_j^{-1} \cdot \frac{x_{j+1}}{x_j}\right) \geq 1 - f(V_j^{-1}) \cdot f\left(\frac{x_{j+1}}{x_j}\right) \geq 1 - E_R > 0.$$

This estimate and (2.8) imply $f(v_{j+1}) \leq \frac{1}{1-E_R} \cdot f(S_j \cdot x_1/x_j)$. □

Now we introduce the f-maximum of the $\mathcal{S}$-set in order to eliminate the factor $f(S_j)$ in the proven estimate.

Definition 2.10. Let R be an $\mathcal{S}$-euclidean domain w. r. t. f. Then we define the *f-maximum of the euclidean set* $\mathcal{S}$ as $S := \max_{s \in \mathcal{S}} f(s)$.

Lemma 2.11. *There exists $c' \in \mathbf{R}_{\geq 0}$ such that $f(S_{j-1}) \leq f(x_1/x_j)^{c'}$ for $1 \leq j \leq r$.*

Proof. There is nothing to show in the case of $j = 1$. Thus assume $j \geq 2$. Then we have $f(S_{j-1}) \leq S^{j-1}$ and $f(x_j) \leq E_R^{j-1} f(x_1)$. If we choose any $c' \geq \log_{E_R^{-1}} S$, then we obtain

$$f(x_1/x_j)^{c'} \geq f(x_1/x_j)^{\log_{E_R^{-1}} S} \geq \left(E_R^{-(j-1)}\right)^{\log_{E_R^{-1}} S} = S^{j-1} \geq f(S_{j-1}).$$

□

Corollary 2.12. *Let $c \in \mathbf{R}_{>0}$ be a fixed chosen constant with $c \geq 1 + \log_{E_R^{-1}} S$. Let $2 \leq j \leq r$, and assume $f(V_j) \geq 1$. Then we have $f(v_{j+1}) \leq \frac{S}{1-E_R} \cdot f(x_1/x_j)^c$.*

Proof. The claim follows from the Lemmas 2.9 and 2.11 due to $f(S_j) \leq S \cdot f(S_{j-1})$. In particular, we have $c \geq 1$, because $E_R^{-1} > 1$ and $S \geq 1$. □

Proposition 2.13. *Let $c \geq 1 + \log_{E_R^{-1}} S$. Then, for $0 \leq j \leq r$, we have*

$$f(v_{j+1}) \leq \frac{S}{1-E_R} \cdot f(x_1/x_j)^c, \tag{2.9}$$

$$f(u_{j+1}) \leq \frac{S}{1-E_R} \cdot f(x_0/x_j)^c + 1. \tag{2.10}$$

Proof. The estimate (2.9) holds true in the case of $j = 0$ and $j = 1$, because $v_1 = 0$ and $v_2 = s_1$.

In the case of $j = 2$ we have $V_2 = q_2$. Lemma 2.5 implies $f(V_2) \geq 1$, such that the claim follows from Corollary 2.12.

Now let $3 \leq j \leq r$, and assume that the claim holds true for $j - 1$. Then either $f(V_j) \geq 1$ such that the claim (2.9) follows from Corollary 2.12, or we have $f(V_j) < 1$. The inductive assumption for $j - 1$ yields

$$f(v_j) \leq \frac{S}{1-E_R} \cdot f(x_1/x_{j-1})^c = \frac{S}{1-E_R} \cdot f(x_1/x_j)^c \cdot \underbrace{f(x_j/x_{j-1})^c}_{\leq E_R^c < 1} < \frac{S \cdot f(x_1/x_j)^c}{1-E_R}.$$

The claim (2.9) follows from this inequality using $f(v_{j+1}) < f(v_j)$. We use $x_{j+1} = (-1)^{j+1}(v_{j+1}x_0 - u_{j+1}x_1)$ (cf. (2.6)) in order to show the estimate (2.10). From this we obtain $f(u_{j+1}x_1) \le f(v_{j+1}x_0) + f(x_{j+1})$ using the triangle inequality. Furthermore, it follows from (2.9) that $f(u_{j+1}) \le \frac{S}{1-E_R} \cdot f(x_0/x_j)^c + f(x_{j+1}/x_1)$. From this inequality we can conclude that $f(u_{j+1}) \le \frac{S}{1-E_R} \cdot f(x_0/x_j)^c + 1$ because of $f(x_{j+1}/x_1) \le 1 \le f(x_0/x_j)$. □

2.2. **Controlled $\mathcal{S}$-euclidean descent in $\mathcal{S}$-euclidean domains.** Let R be an $\mathcal{S}$-euclidean ring, where the $\mathcal{S}$-euclidean function $\|\cdot\|$ is multiplicative, satisfies the triangle inequality, and avoids every value between 0 and 1. Under these assumptions we are able to transfer the concept of a controlled euclidean descent[2] and its fast computation to the $\mathcal{S}$-euclidean domains.

The GCD calculation for imaginary quadratic maximal orders consists of the computation of such an asymptotically fast $\mathcal{S}$-euclidean descent and the subsequent valuation at finitely many certain prime places. For this reason we need the imaginary quadratic order to be a Dedekind domain, i.e., we are able to calculate the ideal sum in this way in maximal orders only.

A controlled $\mathcal{S}$-euclidean descent in the ring R can be specified as follows. In particular, we do not need any information about the group of units, apart from the fact that $+1$ and -1 is contained.

Theorem 2.14. *Let $x, y \in R$ and let $\sigma \in \mathbf{N}_{>0}$ with $\|x\|, \|y\| \ge \sigma$. Then there exist $u, v \in R$ and a matrix $M \in R^{2\times 2}$ such that*

$$M \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix}, \quad M \text{ is invertible in } \mathrm{Quot}(R), \text{ i.e., } \det M \neq 0, \tag{2.11}$$

$$\text{and} \quad 2\max(\|x\|, \|y\|) > \|u\|, \|v\| \ge \sigma > \min(\|u+v\|, \|u-v\|). \tag{2.12}$$

Furthermore the coefficients of the matrix $M = (m_{ij})_{i,j}$ are bounded by

$$\begin{aligned} \|m_{ij}\| &\le S_R \cdot \frac{1 + E_R^{C_R}}{1 - E_R} \cdot \left(\frac{\max(\|x\|, \|y\|)}{\sigma} \right)^{C_R} + 2 \\ &\le \left(1 + \frac{S_R}{1 - E_R} \right) \cdot (1 + E_R^{C_R}) \cdot \left(\frac{\max(\|x\|, \|y\|)}{\sigma} \right)^{C_R}, \end{aligned} \tag{2.13}$$

where $C_R := 1 + \lceil \log_{1/E_R} S_R \rceil$ and $S_R := \max_{s \in \mathcal{S}} \|s\|$.

Proof. We give a constructive proof for this theorem because we calculate $\mathcal{S}$-euclidean steps in the algorithm in an efficient manner in the same way as in this proof.

Without loss of generality assume $\|x\| \ge \|y\| \ge \sigma > 0$. Set $u := x$, $v := y$ and $M := I$. If we have

$$\min(\|u - v\|, \|u + v\|) < \sigma, \tag{2.14}$$

then we have a representation as in (2.11) where (2.12) holds true. Otherwise

$$\min(\|u - v\|, \|u + v\|) \ge \sigma, \tag{2.15}$$

and we calculate an $\mathcal{S}$-euclidean step

$$s \cdot u = qv + r \quad \text{with } \|r\| \le E_R \cdot \|v\| < \|v\| \le \|u\| \tag{2.16}$$

[2]See also [29, Theorem 2.1] for a controlled euclidean descent in $\mathbf{Z}$, [39] for it in $\mathbf{Z}[i]$, and [30] for binary quadratic forms.

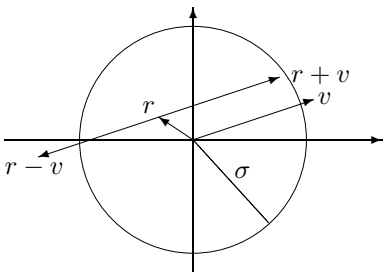

FIGURE 1. Size modification after an $\mathcal{S}$-euclidean step ($R \subset \mathbf{R}^2$)

for the operands u, v. If we now have $\|r\| \geq \sigma$, then we set $M := \left(\begin{smallmatrix} 0 & 1 \\ s & -q \end{smallmatrix}\right) \cdot M$ and change the names to $u_{\text{new}} := v,\ v_{\text{new}} := r$. Then we obtain

$$M \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u_{\text{new}} \\ v_{\text{new}} \end{pmatrix}, \quad \max(\|x\|, \|y\|) > \|u_{\text{new}}\|, \|v_{\text{new}}\| \geq \sigma.$$

If the newly calculated operands (called u, v again) still satisfy condition (2.15), we calculate further $\mathcal{S}$-euclidean steps as described. The set $\{\|z\| :\ z \in R,\ \|z\| < \|y\|\}$ is finite and contains 0 because R is $\mathcal{S}$-euclidean. Thus we are going to calculate a remainder $\|r\| < \sigma$ after finitely many steps in (2.16). If $\|r - v\| \geq \sigma$, then set $\varepsilon := -1$. Otherwise, i.e., $\|r - v\| < \sigma \leq \|v\|$, we have $\|r + v\| = \|2v - (v - r)\| \geq 2 \cdot \|v\| - \|v - r\| \geq \|v\| \geq \sigma$, where $\|v - r\| \leq \|v\|$. Set $\varepsilon := 1$ for this case. We can express (2.16) as $s \cdot u = (q - \varepsilon)v + (r + \varepsilon v)$, where $\varepsilon \in \{-1, +1\}$ was chosen in a manner that $\|r + \varepsilon v\| \geq \sigma$ (see Figure 1). Then we set $M := \left(\begin{smallmatrix} 0 & 1 \\ s & -(q-\varepsilon) \end{smallmatrix}\right) \cdot M$ and change the names to $u_{\text{new}} := v,\ v_{\text{new}} := r + \varepsilon v$. Then it holds true that $\|u_{\text{new}}\| = \|v_{\text{old}}\|,\ \|v_{\text{new}}\| = \|r + \varepsilon v_{\text{old}}\| \leq (1 + E_R) \cdot \|v_{\text{old}}\| < 2 \cdot \|v_{\text{old}}\|$. Furthermore, because of $\|u_{\text{old}}\|, \|v_{\text{old}}\| \leq \max(\|x\|, \|y\|)$, we obtain $\|u_{\text{new}}\|, \|v_{\text{new}}\| < 2 \max(\|x\|, \|y\|)$, and $M \cdot \left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right) = \left(\begin{smallmatrix} u_{\text{new}} \\ v_{\text{new}} \end{smallmatrix}\right)$.

Note that no further euclidean step for $u = u_{\text{new}},\ v = v_{\text{new}}$ has to be calculated because it holds true that $\|u - \varepsilon v\| = \|r + \varepsilon v_{\text{old}} - \varepsilon v_{\text{old}}\| = \|r\| < \sigma$. This satisfies condition (2.14) which means that we have found a suitable representation as in (2.11) which satisfies the conditions (2.12) as well.

We call such a modification of the remainder r to $r + \varepsilon v$ a *size modification* after an $\mathcal{S}$-euclidean step.

Altogether we calculate r $\mathcal{S}$-euclidean steps where a size modification is done at most by the last $\mathcal{S}$-euclidean step. We obtain a matrix $\hat{Q}_j = \left(\begin{smallmatrix} 0 & 1 \\ s_j & -\hat{q}_j \end{smallmatrix}\right)$ for every calculated quotient q_j $(1 \leq j \leq r)$, where $\hat{q}_j := q_j$ for $1 \leq j < r$, and $\hat{q}_r = q_r - \varepsilon$ with $\varepsilon \in \{-1, 0, +1\}$ as above. Note that $\varepsilon = 0$ means that no size modification has to be made in the r-th $\mathcal{S}$-euclidean step.

The matrix M is the product of the matrices $\hat{Q}_j$, $1 \leq j \leq r$. M is invertible in $\text{Quot}(R)$ because $s_j \neq 0$, thus $\det \hat{Q}_j \neq 0$.

It remains to show that the coefficients of M are bounded as stated. For $1 \leq j < r$, it holds true that $\hat{Q}_j = Q_j$, where Q_j is defined in (2.2). Thus we have the representation

$$\hat{Q}_r \cdot Q_{r-1} \cdot \ldots \cdot Q_2 \cdot Q_1 \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix}.$$

We can calculate the product of the matrices $Q_{r-1}\cdot\ldots\cdot Q_1$ as

$$M_{r-1} = (-1)^{r-1}\cdot\begin{pmatrix} v_{r-1} & -u_{r-1} \\ -v_r & u_r \end{pmatrix},$$

using (2.3). The coefficients of M_{r-1} are bounded as was shown in Proposition 2.13. In order to calculate M we can multiply the matrix M_{r-1} by $\hat{Q}_r$ after the r-th $\mathcal{S}$-euclidean step:

$$M = \hat{Q}_r\cdot M_{r-1} = (-1)^r\cdot\begin{pmatrix} v_r & -u_r \\ -v_{r+1}+\varepsilon v_r & u_{r+1}-\varepsilon u_r \end{pmatrix}.$$

Using the estimate of Proposition 2.13 we obtain an upper bound for the size of the coefficients of M:

$$\begin{aligned}\|u_{r+1}-\varepsilon u_r\| &\le \|u_{r+1}\| + \|u_r\| \\ &< \frac{S_R}{1-E_R}\cdot\left(\left\|\frac{x_0}{x_r}\right\|^{C_R} + \left\|\frac{x_0}{x_{r-1}}\right\|^{C_R}\right) + 2 \le S_R\cdot\frac{1+E_R^{C_R}}{1-E_R}\cdot\left\|\frac{x_0}{x_r}\right\|^{C_R} + 2\end{aligned}$$

because of $\|x_r/x_{r-1}\| \le E_R$ for $r \ge 2$. From $2 \le \|x_0/x_r\|^{C_R} + \|x_0/x_{r-1}\|^{C_R}$ it follows that

$$\|u_{r+1}-\varepsilon u_r\| \le \left(1+\frac{S_R}{1-E_R}\right)\cdot(1+E_R^{C_R})\cdot\left\|\frac{x_0}{x_r}\right\|^{C_R}.$$

The estimate for $\|-v_{r+1}+\varepsilon v_r\|$ can be shown in an analogous manner. The stated size bound (2.13) follows from $\|x_r\| \ge \sigma$. □

If R is a Dedekind domain, we can use the technique of a controlled $\mathcal{S}$-euclidean descent to calculate nearly the GCD of two elements apart from a finite set of places.

Definition 2.15. Let $\mathcal{S}$ be a finite subset of R, e.g., the set for which R is $\mathcal{S}$-euclidean. Then we define

$$V(\textstyle\prod\mathcal{S}) := V\left(\prod_{s\in\mathcal{S}} s\right) = \{\mathfrak{p} \text{ prime ideal in } R:\ \exists\, s\in\mathcal{S} : \mathfrak{p}\supset sR\}$$

which is a set of prime ideals of R that contains the ideal $(\prod\mathcal{S})$. In other words $V(\prod\mathcal{S})$ contains the prime ideals which are divisors of elements of $\mathcal{S}$.

In particular, $V(\prod\mathcal{S})$ is a finite set due to the finiteness of $\mathcal{S}$.

Lemma 2.16. *Let R be a Dedekind domain, and let $x_0, x_1, \ldots, x_r \in R$ be the elements of an $\mathcal{S}$-euclidean descent ($x_{r+1} = 0$). Then $v_\mathfrak{p}(x_0) \ge k$ and $v_\mathfrak{p}(x_1) \ge k$ if and only if $v_\mathfrak{p}(x_r) \ge k$ for every prime ideal $\mathfrak{p} \notin V(\prod\mathcal{S})$.*

Proof. Both implications are proven by inductions. Let $1 \le j < r$. The assumptions $v_\mathfrak{p}(x_{j-1}) \ge k$ and $v_\mathfrak{p}(x_j) \ge k$ imply that $v_\mathfrak{p}(x_{j+1}) = v_\mathfrak{p}(s_j\cdot x_{j-1} - q_j x_j) \ge \min(v_\mathfrak{p}(s_j\cdot x_{j-1}), v_\mathfrak{p}(q_j x_j)) \ge k$, which holds true for every prime ideal $\mathfrak{p}$ of R. It follows that $v_\mathfrak{p}(x_r) \ge k$ via induction, where the induction starts with the trivial case $j = 1$.

Now we are going to prove the other implication. Let $1 \le j \le r$. From $v_\mathfrak{p}(x_j) \ge k$ and $v_\mathfrak{p}(x_{j+1}) \ge k$ it follows that $v_\mathfrak{p}(x_{j-1}) = v_\mathfrak{p}(s_j\cdot x_{j-1}) = v_\mathfrak{p}(q_j x_j + x_{j+1}) \ge \min(v_\mathfrak{p}(q_j x_j), v_\mathfrak{p}(x_{j+1})) \ge k$. Thereby, the first equation sign holds true because we provided that $v_\mathfrak{p}(s_j) = 0$. The claim follows with $j = 1$ using decreasing induction in j, where the induction starts with $j = r$. □

Corollary 2.17. *Let the conditions be the same as in Lemma* 2.16. *Then* $\min(v_{\mathfrak{p}}(x_0), v_{\mathfrak{p}}(x_1)) = k$ *if and only if* $v_{\mathfrak{p}}(x_r) = k$ *for every prime ideal* $\mathfrak{p} \notin V(\prod \mathcal{S})$. □

Proposition 2.18. *Let* R *be an* $\mathcal{S}$*-euclidean Dedekind domain. If one calculates an* $\mathcal{S}$*-euclidean descent from* $x, y \in R$ *to* $u, v \in R$ *with* $\sigma = 1$ *according to Theorem* 2.14, *then* $\min(v_{\mathfrak{p}}(x), v_{\mathfrak{p}}(y)) = v_{\mathfrak{p}}(u) = v_{\mathfrak{p}}(v)$ *for every prime ideal* $\mathfrak{p} \notin V(\prod \mathcal{S})$.

Proof. We obtain $\min(\|u+v\|, \|u-v\|) < 1$ using Theorem 2.14, thus $\|u+v\| = 0$ or $\|u-v\| = 0$. Therefore we have $u = \varepsilon v$ with $\varepsilon \in \{-1, +1\}$, hence $v_{\mathfrak{q}}(u) = v_{\mathfrak{q}}(v)$ for all prime ideals $\mathfrak{q}$. The claim that for every prime ideal $\mathfrak{p} \notin V(\prod \mathcal{S})$ follows directly from Lemma 2.16 and Corollary 2.17. □

Now we are able to present our Algorithm 2.19 DESCENT$_R$ for the calculation of an asymptotically fast $\mathcal{S}$-euclidean descent. The algorithm is based on Lehmer's ideas that were improved by Knuth [18] and Schönhage [28, 29] for an asymptotically fast GCD calculation in $\mathbf{Z}$. This concept, which does not need special properties of $\mathbf{Z}$, was transferred to $\mathbf{Z}[i]$ [39], and is now generalized for $\mathcal{S}$-euclidean domains.

If the operands have large size, we split them into heads and tails w. r. t. a fixed chosen basis $b \in R \setminus (R^\times \cup \{0\})$ using a remainder-division for each of them, and we calculate an $\mathcal{S}$-euclidean descent only with the heads (i.e., the calculated quotient of the remainder-division). If we apply a remainder-division for one of the operands and for b^T, then we know that the tail w. r. t. $\|\cdot\|$ is bounded by $\|b^T\| \cdot E(R, \|\cdot\|)$, where $E(R, \|\cdot\|)$ denotes the euclidean minimum for the integral domain R. Set $B := \|b\|$. We have $B > 1$ because b is not a unit.

In order to give a general description of the Algorithm 2.19 DESCENT$_R$, we are going to introduce some constants which depend only on the $\mathcal{S}$-euclidean domain, and not on the operands:

$$E'_R := E_R + \frac{1 - E_R}{2} = \frac{1 + E_R}{2} < 1, \tag{2.17}$$

$$C_R := 1 + \lceil \log_{1/E'_R} S_R \rceil \geq 1, \quad \text{where } S_R := \max_{s \in \mathcal{S}} \|s\|, \tag{2.18}$$

$$m_R := \left(1 + \frac{S_R}{1 - E'_R}\right) \cdot \left(1 + {E'_R}^{C_R}\right), \tag{2.19}$$

$$\gamma_R := \max(0, \lceil \log_B(2m_R \cdot E(R, \|\cdot\|)) - \log_B(B-1) + \max(1 - C_R, \log_B S_R) \rceil), \tag{2.20}$$

$$\delta_R := \max\left(\left\lceil \log_B\left(2 \cdot \frac{B}{1 - E_R}\right) - 1 \right\rceil, \left\lceil \log_B(1 + B^{-(2+\gamma_R)} \cdot E(R, \|\cdot\|)) \right\rceil\right), \tag{2.21}$$

$$\ell_R := 1 + \max\left(\left\lceil \log_{1/E_R} B^{1+\delta_R} \right\rceil, \left\lceil \log_{1/E_R}(3 \cdot B - 2) \right\rceil\right). \tag{2.22}$$

These constants serve different purposes. Because we are going to calculate $\mathcal{S}$-euclidean steps with the head parts of the operands, we cannot guarantee a reduction w. r. t. $\|\cdot\|$ by E_R for the entire operands. Thus we introduce E'_R as the reduction factor for the entire operands, even if we calculate $\mathcal{S}$-euclidean steps with only the heads (with E_R as the reduction factor).

The constant C_R determines whether we use the divide-and-conquer technique in order to calculate an $\mathcal{S}$-euclidean descent. We choose C_R such that ${E'_R}^{-(C_R-1)}$ is an upper bound for S_R. This constant is also involved in an estimate for the

coefficients of the cofactor matrix according to Proposition 2.13. The factor of this estimate is bounded by m_R (cf. Theorem 2.14).

The constant γ_R is chosen such that the operands satisfy the minimum size requirement σ if they are transferred from the head parts using the cofactor matrix.

The constant δ_R determines that only operands larger than $B^{L+1+\delta_R}$ are split into heads and tails. Furthermore, it controls that we are able to guarantee E'_R as a reduction factor for the entire operands, even if we calculated the $\mathcal{S}$-euclidean steps only with the heads.

We are able to bound the number of iterations in the "while" loop (D9) of the Algorithm 2.19 by ℓ_R. We can distinguish between two cases in order to bound the number of iterations in (D9). Either no splitting was done so that we calculate $\mathcal{S}$-euclidean steps with the original operands, or a splitting was done so that we transferred the $\mathcal{S}$-euclidean descent to the entire operands (D8) satisfying some size bound.

Algorithm 2.19 (Fast computation of a controlled $\mathcal{S}$-euclidean descent in R)**.** *Given $x, y \in R$, $L \in \mathbf{N}$ with $\|x\|, \|y\| \geq B^L$, this algorithm calculates $u, v \in R$ and a matrix $M \in R^{2\times 2}$, invertible in* Quot(R), *according to Theorem* 2.14 *(with the parameter E'_R instead of E_R for the size bound of the matrix coefficients) such that*

$$M \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} \quad \text{and} \quad \|u\|, \|v\| \geq B^L > \min(\|u - v\|, \|u + v\|).$$

algorithm DESCENT$_R(x, y, L)$

(D1) ***if*** $\min(\|x\|, \|y\|) < B^{L+1+\delta_R}$ ***then*** ($0 \leq \delta_R < \infty$ *independent of* x, y)
  $u := x,\ v := y,\ M := I$; *(and go to D9)*
else
(D2) *Find a small* $N \in \mathbf{N}$ *with* $\|x\|, \|y\| < B^{L+N}$;
  (see the proof of Theorem 2.20 *for choosing such an* N*)*
  if $L \leq C_R N + \gamma_R$ ***then*** *(no splitting)*
    $T := 0,\ L_1 := L$;
  else *(split the operands; see Figure* 2*)*
    $L_1 := C_R N + \gamma_R,\ T := L + 1 - L_1,$
    split $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix} + b^T \cdot \begin{pmatrix} x'' \\ y'' \end{pmatrix}$, (*here* $\|x\|, \|y\| \geq B^{L+1+\delta_R}$)
    such that $B^{L_1} \leq \|x''\|, \|y''\| < B^{L_1+N-1}$,
      $\|x'\|, \|y'\| \leq B^T \cdot E(R, \|\cdot\|)$,
    and set $x := x'',\ y := y''$;
(D3) $H := L_1 + \lfloor N/2 \rfloor$;
  if $\min(\|x\|, \|y\|) < B^H$ ***then***
    $u' := x,\ v' := y,\ M := I$;
  else
(D4) $(u', v', M) :=$ ***DESCENT***$_R(x, y, H)$; $(\min(\|u' \pm v'\|) < B^H)$;
(D5) ***while*** $\min(\|u' - v'\|, \|u' + v'\|) \geq B^{L_1}$ ***and***
      $\max(\|u'\|, \|v'\|) \geq B^H$ ***do*** *(at most 2 times)*
    Perform one $\mathcal{S}$*-euclidean step on* u', v'
    preserving $\|u'\|, \|v'\| \geq B^{L_1}$,
    and with proper updating of M;
  if $\min(\|u' - v'\|, \|u' + v'\|) < B^{L_1}$ ***then***
    $u := u',\ v := v'$;

else

(D6) $(u, v, M') := \boldsymbol{DESCENT}_R(u', v', L_1)$; (*then* $\min(\|u \pm v\|) < B^{L_1}$);

(D7) $M := M' \cdot M$;

(D8) ***if*** $T > 0$ ***then***

$$\begin{pmatrix} u \\ v \end{pmatrix} := b^T \cdot \begin{pmatrix} u \\ v \end{pmatrix} + M \cdot \begin{pmatrix} x' \\ y' \end{pmatrix};$$

(D9) ***while*** $\min(\|u - v\|, \|u + v\|) \geq B^L$ ***do*** *(at most ℓ_R times)*
Perform one $\mathcal{S}$-euclidean step on u, v
preserving $\|u\|, \|v\| \geq B^L$,
and with proper updating of M;

(D10) ***return*** (u, v, M).

Theorem 2.20. *Given $x, y \in R$, Algorithm* 2.19 *$DESCENT_R$ calculates $u, v \in R$ and a matrix $M \in R^{2\times 2}$ according to Theorem* 2.14 *(E'_R instead of E_R). In addition to that, the number of $\mathcal{S}$-euclidean steps in the "while" loops* (D5) *and* (D9) *is bounded by* 2 *or, respectively, ℓ_R independent of the operands.*

Proof. We calculate every $\mathcal{S}$-euclidean step in the algorithm such that we can guarantee a reduction by the factor E_R w. r. t. $\|\cdot\|$. If we compute an $\mathcal{S}$-euclidean step only for the heads of the operands, we are not able to guarantee the factor E_R for the entire operands (calculated using the cofactor matrix). Anyway, we can always guarantee E'_R as a reduction factor for the entire operands as we will show in Lemma 2.22.

If a size modification is made after an $\mathcal{S}$-euclidean step in the "while" loop (D5) or (D9), then the minimum condition in (D5) or (D9) is not satisfied anymore such that the loop iteration terminates.

After these general remarks we will follow the algorithm step by step in order to prove its correctness. Without loss of generality we assume $\|x\| \geq \|y\|$. If step (D1) branches to (D9), we have $\|y\| < B^{L+1+\delta_R}$. In every $\mathcal{S}$-euclidean step in (D9) the intermediate operands are reduced w. r. t. $\|\cdot\|$ by the factor E_R at least. Thus we achieve a remainder less than B^L after finitely many steps such that a size modification has to be made. After this no further iteration is executed. Otherwise, if (D1) does not branch to (D9), we always have $N \geq 2$ in (D2).

In case the operands x, y have to be split, we calculate the splitting for x (y in an analogous manner) using a remainder-division w. r. t. to the euclidean minimum $E(R, \|\cdot\|) < \infty$ as $x = x'' \cdot b^T + x'$ with $\|x'\| \leq B^T \cdot E(R, \|\cdot\|)$; see Figure 2. Now we can estimate $\frac{\|x\|}{B^T} - E(R, \|\cdot\|) \leq \|x''\| \leq \frac{\|x\|}{B^T} + E(R, \|\cdot\|)$. Because the operands x, y are both greater than $B^{L+1+\delta_R}$ w. r. t. $\|\cdot\|$ (no branch in (D1)), we can conclude that $\|x''\| \geq \frac{\|x\|}{B^T} - E(R, \|\cdot\|) \geq B^{L+1+\delta_R-T} - E(R, \|\cdot\|) = B^{L_1} \cdot (B^{\delta_R} - B^{-L_1} \cdot E(R, \|\cdot\|)) \geq B^{L_1}$, because $\delta_R \geq \log_B(1 + B^{-(2+\gamma_R)} \cdot E(R, \|\cdot\|)) \geq \log_B(1 + B^{-L_1} \cdot E(R, \|\cdot\|))$ using the definition for δ_R and $L_1 \geq 2 + \gamma_R$. Furthermore, we have $\|x''\| \leq B^{L+N-T} + E(R, \|\cdot\|) = B^{L_1+N-1} + E(R, \|\cdot\|)$ as an upper bound. In addition to that,

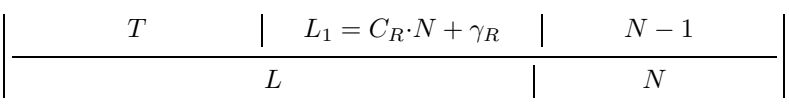

FIGURE 2. $\mathcal{S}$-euclidean descent in R: Splitting with tails

we assume that N was chosen minimal such that $\|x''\| < B^{L_1+N-1}$. Therefore N can be determined in the following manner. Let $N_{\min}$ be the smallest N which satisfies $\|x\| < B^{L+N_{\min}}$. Then we can choose N with $N_{\min} \le N \le N_{\min} + \Delta$, where $\Delta := \lfloor \max(0, \log_B E(R, \|\cdot\|) - (\gamma_R + 3)) + \log_B 2 \rfloor + 1$ independent of the operand x. In particular, we have $\Delta > \log_B E(R, \|\cdot\|) - (\gamma_R + 3) + \log_B 2$ which yields

$$\begin{aligned} B^{L_1+N+\Delta-1} &= \tfrac{1}{2}B^{L_1+N+\Delta-1} + \tfrac{1}{2}B^{L_1+N+\Delta-1} \\ &> B^{L_1+N-1} + \tfrac{1}{2}B^{(2+\gamma_R)+2+(\log_B E(R,\|\cdot\|)-(\gamma_R+3)+\log_B 2)-1} \\ &\ge B^{L_1+N-1} + E(R, \|\cdot\|) \ge \|x''\|. \end{aligned}$$

Thus we choose N as the smallest $N \ge N_{\min}$ which satisfies $\|x''\| < B^{L_1+N-1}$. The explanation above shows that there exists such an N which is at most Δ larger than $N_{\min}$.

Now we show that the number of iterations of the "while" loop (D5) is bounded by 2 at most. Denote the heads of the operands x, y with $\tilde{x}, \tilde{y}$ in case of $T > 0$, and assume $\|\tilde{x}\| \ge \|\tilde{y}\|$. Otherwise, if $T = 0$, we set $\tilde{x} := x$, $\tilde{y} := y$. We are calculating a descent of N "bits" (for example in case of $R = \mathbf{Z}$ and $B = 2$) from $L_1 + N$ to L_1 bits for the operands $\tilde{x}, \tilde{y}$. In order to achieve this, we first calculate a descent of $\lceil N/2 \rceil$ bits to $H := L_1 + \lfloor N/2 \rfloor$ bits in (D4) and (D5).

If $\|\tilde{y}\| < B^H$ in (D3), we calculate one $\mathcal{S}$-euclidean step and get a remainder r with $\|r\| \le E_R \cdot B^H < B^H$ instead of applying procedure DESCENT$_R$. If now $\|r\| \ge B^{L_1}$, then no further loop iteration is executed because $\max(\|\tilde{y}\|, \|r\|) < B^H$. Otherwise, if $\|r\| < B^{L_1}$, then a size modification with $\tilde{y}$ is made ($u' := \tilde{y}$ and $v' := r + \varepsilon\tilde{y}$ with $\varepsilon \in \{-1, +1\}$ such that $\|v'\| \ge \|\tilde{y}\|$). Then $\|u' - \varepsilon v'\| < B^{L_1}$ such that no further loop iteration is done. In the other case, i.e., if we called DESCENT$_R$ in (D4) recursively, there exists an $\varepsilon \in \{-1, +1\}$ such that $\|u' + \varepsilon v'\| < B^H$. After the calculation of an $\mathcal{S}$-euclidean step we get a remainder less than B^H. Either a size modification is necessary for the remainder which terminates the "while" loop, or a further $\mathcal{S}$-euclidean step is calculated. This leads to a remainder less than B^H for which a size modification can be done. In this case, the "while" loop terminates, and otherwise both operands are less than B^H such that no further iteration is done due to the maximum condition.

If the "while" loop (D5) terminates because $\min(\|u' - v'\|, \|u' + v'\|) < B^{L_1}$ is satisfied, then the condition in the following "if" clauses holds true such that no further DESCENT$_R$ (D6) is calculated. If $T = 0$, no splitting was done, the condition (D8) does not hold true, and $L = L_1$ such that no iteration in (D9) is executed.

From now on we assume $T > 0$ that implies $L > C_R N + \gamma_R$. The parameters T and L_1 are chosen such that $T + L_1 = L + 1$. We are calculating a descent from the heads x'', y'' (D3) to intermediate operands $\tilde{u}, \tilde{v}$ (D7). In addition to that there exists an $\varepsilon \in \{-1, +1\}$ such that

$$\|\tilde{u}\|, \|\tilde{v}\| \ge B^{L_1} > \|\tilde{u} + \varepsilon\tilde{v}\|. \tag{2.23}$$

We calculate u, v in (D8) as

$$\begin{pmatrix} u \\ v \end{pmatrix} := b^T \cdot \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} + M \cdot \begin{pmatrix} x' \\ y' \end{pmatrix} = b^T \cdot M \cdot \begin{pmatrix} x'' \\ y'' \end{pmatrix} + M \cdot \begin{pmatrix} x' \\ y' \end{pmatrix} = M \cdot \begin{pmatrix} x \\ y \end{pmatrix}. \tag{2.24}$$

Because of $B^{L_1} \le \|x''\|, \|y''\| < B^{L_1+N-1}$ and because of the minimum size of $\tilde{u}, \tilde{v}$, Lemma 2.23 implies that the coefficients m_{ij} $(i, j \in \{1, 2\})$ of the cofactor matrix

M (for the descent from x'', y'' to $\tilde{u}, \tilde{v}$) are bounded w. r. t. $\|\cdot\|$ by

$$\|m_{ij}\| \leq m_R \cdot \left(\frac{B^{L_1+N-1}}{B^{L_1}}\right)^{C_R} = m_R \cdot B^{C_R \cdot (N-1)}$$

(cf. Theorem 2.14 with E'_R instead of E_R). Furthermore, $\|x'\|, \|y'\| \leq B^T \cdot E(R, \|\cdot\|)$ such that the estimate

$$\begin{aligned}\|u\|, \|v\| &\geq B^{L_1+T} - 2m_R \cdot B^{C_R \cdot (N-1)} \cdot B^T \cdot E(R, \|\cdot\|) \\ &= B^{L_1+T} - 2m_R \cdot E(R, \|\cdot\|) \cdot B^{C_R N + \gamma_R + T - C_R - \gamma_R} \\ &= B^{L+1} - 2m_R \cdot E(R, \|\cdot\|) \cdot B^{L+1-(C_R+\gamma_R)} \\ &= B^L \cdot (B - 2m_R \cdot E(R, \|\cdot\|) \cdot B^{1-(C_R+\gamma_R)}) \geq B^L\end{aligned}$$

follows from (2.24) because $\gamma_R \geq \log_B(2m_R \cdot E(R, \|\cdot\|)) - \log_B(B-1) + 1 - C_R$. This means that $\|u\|, \|v\|$ cannot become too small. In addition to that it follows from (2.24) using the size bound (2.23) that

$$\begin{aligned}\|u + \varepsilon v\| &= \|(\tilde{u} + \varepsilon \tilde{v}) \cdot B^T + (m_{11} - \varepsilon m_{21}) x' + (m_{12} + \varepsilon m_{22}) y'\| \\ &< B^{L_1+T} + 4m_R \cdot B^{C_R(N-1)} \cdot B^T \cdot E(R, \|\cdot\|) \\ &= B^{L_1+T} + 4m_R \cdot E(R, \|\cdot\|) \cdot B^{C_R N + \gamma_R + T - (C_R+\gamma_R)} \\ &\leq B^{L+1} + 4m_R \cdot E(R, \|\cdot\|) \cdot B^{L+1-(C_R+\gamma_R)} \\ &= B^L \cdot (B + 4m_R \cdot E(R, \|\cdot\|) \cdot B^{1-(C_R+\gamma_R)}) \qquad (2.25) \\ &\leq B^L \cdot (3 \cdot B - 2) \qquad \text{(cf. (2.20))}.\end{aligned}$$

In the case where the operands were split ($T > 0$), one of the operands is less than $(3 \cdot B - 2) \cdot B^L$ w. r. t. $\|\cdot\|$ after at most one $\mathcal{S}$-euclidean step in (D9). Every further $\mathcal{S}$-euclidean step reduces the remainder w. r. t. $\|\cdot\|$ at least by the factor E_R (we can guarantee E_R instead of E'_R because we are calculating $\mathcal{S}$-euclidean steps with the entire operands, and not only with the heads). After $k \geq \log_{1/E_R}(3 \cdot B - 2)$ $\mathcal{S}$-euclidean steps we achieve at least a reduction by $E_R^k \leq \frac{1}{3 \cdot B - 2}$ which means that $\|u + \varepsilon v\| < B^L$. In the case $T = 0$ one of the operands is less than $B^{L+1+\delta_R}$ w. r. t. $\|\cdot\|$, and after one $\mathcal{S}$-euclidean step in (D9) this size bound holds true for both operands. One can bound the number of further $\mathcal{S}$-euclidean steps in (D9) in an analogous manner as above in order to achieve a remainder less than B^L. Altogether, the number of $\mathcal{S}$-euclidean steps in (D9) can be bounded by

$$\ell_R = 1 + \max\left(\left\lceil \log_{1/E_R} B^{1+\delta_R} \right\rceil, \left\lceil \log_{1/E_R}(3 \cdot B - 2) \right\rceil\right).$$

This statement holds true because we get a remainder less than B^L w. r. t. $\|\cdot\|$ after at most ℓ_R iterations (D9) such that a size modification is done which implies an afterwards termination of the "while" loop. □

Remark 2.21. The constants were chosen in such a way that the estimates hold true in the proof above. In addition to that one can be interested in discussing strategic aspects of the choice of the constants. For example, on the one hand it is an advantage to choose γ_R as small as possible in order to achieve a splitting in the recursive calls as often as possible. On the other hand, γ_R should be as large as possible in order to achieve a small factor $B + 4m_R \cdot E(R, \|\cdot\|) \cdot B^{1-(C_R+\gamma_R)}$ in

(2.25). Such a strategic discussion about the choice of the ring constants is not in the scope of this article.

We have omitted the proof of two statements in the proof of Theorem 2.20 which we now present as Lemmas 2.22 and 2.23.

Lemma 2.22. *Let $s{\cdot}x'' = q{\cdot}y'' + r''$ be an $\mathcal{S}$-euclidean step for the heads x'', y'' in any recursive call of the Algorithm* 2.19 *DESCENT$_R$, hence $\|r''\| \le E_R{\cdot}\|y''\|$. Then we have $\|r\| \le E'_R{\cdot}\|y\|$ for a corresponding $\mathcal{S}$-euclidean step $s{\cdot}x = qy + r$ for the entire operands x, y in the outermost recursion level.*

Proof. Let x, y be the operands in the outermost recursion level. During the recursive calls of DESCENT$_R$, these operands are split severals times in heads and tails, hence

$$\begin{aligned} x &= x_0 + b^{T_1}{\cdot}(x_1 + b^{T_2}{\cdot}(x_2 + \cdots + b^{T_k}{\cdot}x_k)\cdots), \\ y &= y_0 + b^{T_1}{\cdot}(y_1 + b^{T_2}{\cdot}(y_2 + \cdots + b^{T_k}{\cdot}y_k)\cdots). \end{aligned}$$

Thereby we have $T_\kappa > 0$ because a splitting was done in the recursion level κ. In the innermost recursion level we have split the operands in heads x'', y'' and tails $\hat{x}, \hat{y}$ as

$$x'' := x_k,\ \hat{x} := x - b^\tau x'',\ y'' := y_k,\ \hat{y} := y - b^\tau y'', \quad \text{where } \tau := \sum_{\kappa=1}^{k} T_\kappa.$$

Now we are able to bound the size of the tails by

$$\begin{aligned} \|\hat{x}\|, \|\hat{y}\| &\le E(R, \|\cdot\|){\cdot}B^{T_1} + E(R, \|\cdot\|){\cdot}B^{T_1+T_2} + \cdots + E(R, \|\cdot\|){\cdot}B^{\tau} \\ &\le B^\tau{\cdot}E(R, \|\cdot\|){\cdot}\sum_{\kappa=0}^{k-1} B^{-\kappa} \qquad (\text{as } T_\kappa > 0) \\ &\le B^\tau{\cdot}E(R, \|\cdot\|){\cdot}\frac{1}{1-B^{-1}} = B^\tau{\cdot}E(R, \|\cdot\|){\cdot}\frac{B}{B-1}, \end{aligned} \tag{2.26}$$

because the splitting was done by calculating a euclidean step w. r. t. the euclidean minimum $E(R, \|\cdot\|)$ for the ring R.

Now we return to the k-th recursion level, and let L_1, N be the parameter of the corresponding splitting. It follows from both the $\mathcal{S}$-euclidean steps $s{\cdot}x'' = q{\cdot}y'' + r''$ and $s{\cdot}x = qy + r$ (dividing them by y'' or, respectively, y, and subsequent subtraction in order to eliminate q) that

$$s{\cdot}\left(\frac{x''}{y''} - \frac{x}{y}\right) = \frac{r''}{y''} - \frac{r}{y} \quad \Rightarrow \quad \left\|\frac{r}{y}\right\| \le \left\|\frac{r''}{y''}\right\| + \|s\|{\cdot}\left\|\frac{x''}{y''} - \frac{x}{y}\right\|. \tag{2.27}$$

We know $\|r''/y''\| \le E_R$, thus it remains to show that the second term can be bounded by $\frac{1-E_R}{2}$:

$$\begin{aligned} \left\|\frac{x''}{y''} - \frac{x}{y}\right\| = \left\|\frac{x''y - xy''}{yy''}\right\| &= \left\|\frac{x''\hat{y} + x''b^T y'' - \hat{x}y'' - x''b^T y''}{yy''}\right\| \\ &\le \left\|\frac{x''\hat{y}}{yy''}\right\| + \left\|\frac{\hat{x}}{y}\right\| \le \frac{B}{B-1}{\cdot}E(R, \|\cdot\|){\cdot}B^{\tau-(L+1+\delta_R)}{\cdot}\left(\left\|\frac{x''}{y''}\right\| + 1\right) \\ &< \frac{B}{B-1}{\cdot}E(R, \|\cdot\|){\cdot}B^{-L_1-\delta_R}{\cdot}(B^{N-1} + 1). \end{aligned}$$

Because of $L_1 = C_R N + \gamma_R$ and $\gamma_R \geq \log_B(2{\cdot}E(R, \|\cdot\|)) - \log_B(B-1) + \log_B S_R$ using (2.20), it follows that

$$\begin{aligned}\|s\|{\cdot}\left\|\frac{x''}{y''} - \frac{x}{y}\right\| &< S_R{\cdot}2\frac{B}{B-1}{\cdot}E(R, \|\cdot\|){\cdot}B^{-C_R N - \gamma_R - \delta_R + N - 1} \\ &\leq B{\cdot}B^{-(C_R-1)N - \delta_R - 1} \leq B^{-\delta_R},\end{aligned}$$

because $B^{N-1} + 1 \leq 2{\cdot}B^{N-1}$, $C_R \geq 1$, and $N \geq 0$. It follows from (2.21) that $\delta_R \geq \log_B\left(2{\cdot}\frac{B}{1-E_R}\right) - 1$, which implies

$$\|s\|{\cdot}\left\|\frac{x''}{y''} - \frac{x}{y}\right\| \leq \frac{1-E_R}{2}.$$

Then the statement follows from (2.27) using the definition of E'_R (2.17). □

Lemma 2.23. *The coefficients (m_{ij}) of the cofactor matrix M are bounded by*

$$\|m_{ij}\| \leq \left(1 + \frac{S_R}{1-E'_R}\right){\cdot}(1 + {E'_R}^{C_R}){\cdot}\left(\frac{\max(\|x''\|, \|y''\|)}{B^{L_1}}\right)^{C_R}$$

in step (D8) *of Algorithm* 2.19 *DESCENT$_R$.*

Proof. We calculate an $\mathcal{S}$-euclidean descent with several $\mathcal{S}$-euclidean steps for the heads of the entire operands. Each of these steps reduces the remainder at least by a factor of E_R w. r. t. $\|\cdot\|$. If the remainder is smaller than a certain minimum size σ, we calculate a size modification of the remainder. Let $s{\cdot}x = qy + z$, $\|z\| \leq E_R{\cdot}\|y\|$, be an $\mathcal{S}$-euclidean step for x, y of an $\mathcal{S}$-euclidean descent (in particular, e.g., the last calculated $\mathcal{S}$-euclidean step in a recursive call), which is followed by a size modification $z + \varepsilon y$. Then the calculation of a further $\mathcal{S}$-euclidean step $1{\cdot}(z + \varepsilon y) = \varepsilon y + z$, $\|z\| \leq E_R{\cdot}\|y\|$ removes the previous size modification such that we can guarantee a reduction by the factor E_R. For that reason we must not consider the size modification in general, apart from the size modification after the last $\mathcal{S}$-euclidean step (if done) of an $\mathcal{S}$-euclidean descent in order to bound the matrix coefficients.

Lemma 2.22 says that all $\mathcal{S}$-euclidean steps w. r. t. the full operands (not only for the heads) are reducing the size at least by the factor E'_R such that we can bound M's coefficients in the same manner as in Theorem 2.14 (replacing E_R by E'_R). □

These two lemmas complete the proof of Theorem 2.20. Finally we prove an estimate for the running time of Algorithm 2.19 DESCENT$_R$.

Proposition 2.24. *Let $t(l, n)$ denote the maximum running time of algorithm DESCENT$_R(x, y, L)$ for every $L \leq l$ and any x, y with $\|x\|, \|y\| < B^{L+N}$, where $N \leq n$. Then we have $t(l, n) \leq O(\hat{\mu}_R(l+n){\cdot}\log(n+1))$, where $\hat{\mu}_R(l+n)$ denotes a smooth upper bound for the multiplication time or, respectively, for the calculation of an $\mathcal{S}$-euclidean step of two operands smaller than B^{l+n} w. r. t. $\|\cdot\|$.*

Proof. Set $C'_R := C_R + \gamma_R/2$. In the case of a splitting of the operands in heads and tails we have $C'_R N \geq C_R N + \gamma_R$, because $N \geq 2$.

Except for the recursive calls in (D4) and (D6) the algortihm calculates only a bounded number of operations with operands smaller than B^{l+n}. The case

$L > C_R N + \gamma_R$ is reduced to a problem with parameters $L_1 = C_R N + \gamma_R$, $N_1 = N - 1$, which allows the estimate

$$t(l,n) \le t(L_1, N_1) + O(\hat{\mu}_R(l+n)) \le t(C'_R{\cdot}n, n) + O(\hat{\mu}_R(l+n)). \tag{2.28}$$

The *divide-and-conquer* technique of the algorithm yields

$$t(C'_R{\cdot}2n, 2n) \le 2{\cdot}t(C'_R{\cdot}n, n) + O(\hat{\mu}_R(2n(1+C'_R))). \tag{2.29}$$

(In other words one calculates a descent of $2n$ bits by calculating recursively up to two descents of n bits each. In these recursive calls a splitting is done such that a running time of $t(C'_R{\cdot}n, n)$ is sufficent.) The estimated running time $t(l,n) \le O(\hat{\mu}_R(l+n){\cdot}\log(n+1))$ follows from (2.28) and (2.29). □

Remark 2.25. This running time with parameter $l = 0$ is an upper bound for the calculation of a complete $\mathcal{S}$-euclidean descent for operands $x, y \in R$ with $\|x\|, \|y\| < B^n$. It should be mentioned that some algorithmic properties of the domain R are encoded in the function $\hat{\mu}_R$ and in the constant of the O-notation, for example the cardinality of the set $\mathcal{S}$.

3. GCD computation in imaginary quadratic rings of algebraic integers

Now, after having introduced $\mathcal{S}$-euclidean domains, we are able to apply this concept and the corresponding fast Algorithm 2.19 DESCENT$_R$ to imaginary quadratic orders. We show that every imaginary quadratic order is $\mathcal{S}$-euclidean w. r. t. the absolute value and that, for every imaginary quadratic maximal order, we can compute the sum of two principal ideals using the concept of $\mathcal{S}$-euclidean domains and valuations at certain places (it is necessary that the order is a Dedekind domain).

If the reader is interested in more general computations of ideal sums in quadratic number fields, e.g., for the real quadratic case, we refer to Section 4.

3.1. Imaginary quadratic orders as examples of $\mathcal{S}$-euclidean domains.

Now we would like to show that every imaginary quadratic order (not only the maximal order) is an $\mathcal{S}$-euclidean domain. In particular, we need an applicable description for the "$\mathcal{S}$-euclidean" property in order to use it for an algorithm. This requirement is satisfied by

Theorem 3.1. *Let $\mathcal{O}_D$ be the imaginary quadratic order with discriminant $D < 0$. Then there exists a finite set $\mathcal{S} \subset \{1, \ldots, \lfloor\sqrt{|D|/3}\rfloor\}$ such that $\mathcal{O}_D$ is an $\mathcal{S}$-euclidean domain w. r. t. the absolute value $|\cdot|$ (considering $\mathcal{O}_D$ canonically embedded in $\mathbf{C}$).*

We postpone the proof of this theorem to the end of this section such that we can discuss some important conclusions for our following Algorithm 3.16 SGCD$_{\mathcal{O}_D}$. Even if this theorem holds true for every imaginary quadratic order, our Algorithm 3.16 is defined only for Dedekind domains, i.e., for rings of algebraic integers (maximal orders).

Remark 3.2. The ring of algebraic integers $\mathcal{O}_D$ with a fundamental discriminant $D < 0$ is a unique factorization domain (i.e., a principal ideal domain in case of algebraic number fields) if $D \in \{-3, -4, -7, -8, -11, -19, -43, -67, -163\}$. Moreover, $\mathcal{O}_D$ is norm-euclidean for $D \in \{-3, -4, -7, -8, -11\}$ such that one can choose $\mathcal{S} = \{1\}$.

Definition 3.3. Let $D < 0$ be a discriminant. Then we define a finite *euclidean set* $\mathcal{S}_D \subset \mathbf{N}_{>0}$ as $\mathcal{S}_D := \{1, \ldots, \lfloor\sqrt{|D|/3}\rfloor\}$.

In particular, we have $\mathcal{S}_D = \{1\}$ for $D \in \{-3, -4, -7, -8, -11\}$.

Corollary 3.4. *Every imaginary quadratic order $\mathcal{O}_D$ is $\mathcal{S}_D$-euclidean w. r. t. $|\cdot|$.*

We restrict our considerations on $\mathcal{S}$-euclidean sets for imaginary quadratic orders to subsets of positive integers because there exists an easy way to prove in a uniform manner that these orders are $\mathcal{S}$-euclidean. There may exist smaller $\mathcal{S}$-euclidean sets (not only consisting of integers, but also of "general" ring elements) such that an order is $\mathcal{S}$-euclidean w. r. t. such a set. A lower bound for the cardinality of such an $\mathcal{S}$-euclidean set of an imaginary quadratic order depends on the class number or, respectively, on the structure of the class group in a nontrivial manner.

Before we prove Theorem 3.1, we need some preparations.

Proposition 3.5. *Let $c \in \mathbf{R}_{>0}$ be a constant, let $J \subset \mathbf{R}_{>0}$ be a compact interval, and let U be an environment of 0 that contains the open interval $(-\eta, \eta)$ with $\eta \in \mathbf{R}_{>0}$. Then there exists a* finite *index set $M \subset \mathbf{N} \times \mathbf{N}_{>0}$ such that $J \subset \bigcup_{(k,s)\in M} \frac{c\cdot k+U}{s}$.*

Proof. Without loss of generality we assume $c = 1$. If not, we could change the scale by multiplying by $1/c$. Because of the inclusion $I_{k,s,\eta} := \left(\frac{k-\eta}{s}, \frac{k+\eta}{s}\right) \subset \frac{k+U}{s}$, it suffices to show that $\mathbf{R}_{>0} \subset \bigcup_{\substack{k\geq 0\\ s\geq 1}} \left(\frac{k-\eta}{s}, \frac{k+\eta}{s}\right)$. Define $S := \lfloor 1/\eta \rfloor$, thus $S + 1 > 1/\eta$. From Dirichlet's approximation theorem (cf. [27, Chapter V, A. Theorem] or [1, Chapter 7]) it follows that for every $\theta \in \mathbf{R}_{>0}$ there exist integers k and s with $0 < s \leq S$ such that

$$\left|\theta - \frac{k}{s}\right| \leq \frac{1}{s(S+1)} < \frac{\eta}{s} \quad \Rightarrow \quad \theta \in I_{k,s,\eta}.$$

Thus we have covered J with a countable infinite system of open sets. Because of the compactness of J there exists a finite subset of this system whose sets cover J already. □

Remark 3.6. For any $c \in \mathbf{R}_{>0}$ we can define an upper bound for the occuring denominators s by $S := \lfloor c/\eta \rfloor$.

Furthermore, it follows from the above proof that one is able to find such a finite set M. For every fixed denominator s with $0 < s \leq S$, there exist a finite number of $I_{k,s,\eta}$ which are intersecting with J. All these pairs (k, s) form a suitable finite set M.

Let $\mathcal{O}_D$ be an imaginary quadratic order with discriminant $D < 0$. $\mathcal{O}_D$ is a discrete subring in $\mathbf{C}$ such that we can define $c := \min\{\mathrm{Im}(x) > 0 : \ x \in \mathcal{O}_D\}$ which can be calculated as $c = \frac{1}{2}\sqrt{|D|}$. According to this ring constant we define

$$A_\eta := \{z \in \mathbf{C} : \ \eta \leq \mathrm{Im}(z) \leq c/2\} \text{ and } B_{\eta,k} := \{z \in \mathbf{C} : \ |\mathrm{Im}(z) - k\cdot c| \leq \eta\}$$

for $k \in \mathbf{Z}$ and $0 < \eta < c/2$. $B_{\eta,k}$ forms a 2η high, horizontal stripe in the complex plane which contains the $x \in \mathcal{O}_D$ with $\mathrm{Im}(x) = k\cdot c$ (see Figure 3 with $D \equiv 1 \bmod 4$).

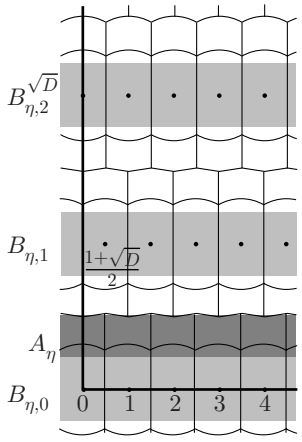

FIGURE 3. The lattice $\mathcal{O}_D$ ($D \equiv 1 \bmod 4$) and the sets $A_\eta, B_{\eta,k}$

Proposition 3.7. *Let the imaginary quadratic order $\mathcal{O}_D$ be embedded canonically in $\mathbf{C}$. Let $B_{\eta,k}$ and c be defined as above. Let $\eta > 0$. Then there exists a finite set $\mathcal{S} \subset \mathbf{N}_{>0}$ such that for every $z \in \mathbf{C}$ there exist an $s \in \mathcal{S}$ and a $k \in \mathbf{Z}$ with $s \cdot z \in B_{\eta,k}$.*

Proof. For $z \in \bigcup_{\kappa \in \mathbf{Z}} B_{\eta,\kappa}$ $s = 1$ is sufficient. Thus let us assume that no $B_{\eta,k}$ contains z. Then there exists $\hat{z} \in \mathcal{O}_D$ such that $z - \hat{z} \in A_\eta \cup (-A_\eta)$. Therefore A_η contains $z - \hat{z}$ or $\hat{z} - z$. (Multiplying z by -1 causes a transition from $B_{\eta,k}$ to $B_{\eta,-k}$ in the assumption.) For this reason we can assume $z \in A_\eta$ w. l. o. g.

Set $z' := \operatorname{Im}(z)$. Denote the projections of the sets A_η, $B_{\eta,0}$ on the imaginary part by $J := \operatorname{Im}(A_\eta)$ and $I := \operatorname{Im}(B_{\eta,0})$. Then $z' \in J$, and we have $\operatorname{Im}(B_{\eta,k}) = c \cdot k + I$. It follows from Proposition 3.5 that there exist $(k, s) \in \mathbf{N} \times \mathcal{S}$ such that $z' \in \frac{c \cdot k + I}{s}$. (In particular, $\mathcal{S}$ can be chosen as a finite set which follows from Remark 3.6.) Thus we have $s \cdot z' \in c \cdot k + I$, and this can be transformed into $s \cdot z \in B_{\eta,k}$ because of $\operatorname{Im}(s \cdot z) = s \cdot z'$. □

Remark 3.8.

(1) In the case of $0 < \eta < \sqrt{3}/2$, it follows that the distance between $z \in \mathbf{C} \cap B_{\eta,k}$ for any $k \in \mathbf{Z}$ to the nearest lattice point (consider $\mathcal{O}_D$ embedded canonically in $\mathbf{C}$) is less than 1. In other words, there exists a $\hat{z} \in R$ with $|\operatorname{Re}(z) - \operatorname{Re}(\hat{z})| \leq \frac{1}{2}$ and $|\operatorname{Im}(z) - \operatorname{Im}(\hat{z})| \leq \eta$, thus $|z - \hat{z}|^2 \leq (\frac{1}{2})^2 + \eta^2 < 1$.

(2) $\mathcal{S}$ can be chosen as $\mathcal{S} \subset \{1, \ldots, \lfloor c/\eta \rfloor\}$, which follows from Remark 3.6.

Now we have made all the preparations in order to prove Theorem 3.1.

Proof of Theorem 3.1. We have to show that the conditions (S1), (S2) and (S3) are all satisfied. Obviously, (S1) holds true for the absolute value $f = |\cdot|$. (S3) is satisfied because $\mathcal{O}_D$ is a discrete subring of $\mathbf{C}$. It follows from Proposition 3.7 that for every $z \in \operatorname{Quot} \mathcal{O}_D$ there exists an $s \in \mathcal{S} \subset \mathbf{N}_{>0}$ such that $s \cdot z \in B_{\eta,k}$. For a fixed chosen $\eta < \sqrt{3}/2$ the distance between $s \cdot z$ to a nearest lattice point in $\mathcal{O}_D$ w. r. t. the absolute value $|\cdot|$ is less than 1, which shows that (S2) holds true.

We have to show that $\mathcal{S}$ can be chosen as the stated finite set of the positive integers. We can bound the set $\mathcal{S} \subset \mathbf{N}_{>0}$ by $S = \lfloor c/\eta \rfloor$ with $c = \frac{1}{2}\sqrt{|D|}$ and

$0 < \eta < \frac{1}{2}\sqrt{3}$ due to Remark 3.8. Thus we have to study how we can specify the upper bound S in terms of the discriminant D and, according to this, what the minimal size for η is. Therefore define γ, $0 < \gamma \leq 1$, by $\sqrt{|D|/3} + \gamma = \lfloor \sqrt{|D|/3} \rfloor + 1 \in \mathbf{N}_{>0}$. It follows that

$$\frac{\sqrt{3}}{2 \cdot (1 + \gamma\sqrt{3/|D|})} < \eta < \sqrt{3}/2 \quad \Rightarrow \quad \sqrt{|D|/3} < \frac{c}{\eta} < \sqrt{|D|/3} + \gamma. \tag{3.1}$$

Thus we have $\lfloor \sqrt{|D|/3} \rfloor < c/\eta < \lfloor \sqrt{|D|/3} \rfloor + 1$ using the definition of γ such that we conclude that $S = \lfloor c/\eta \rfloor = \lfloor \sqrt{|D|/3} \rfloor$. □

3.2. **Fast GCD computation in rings of algebraic integers of imaginary quadratic number fields using the concept of $\mathcal{S}$-euclidean domains.** On the one hand we have shown how to compute asymptotically fast an $\mathcal{S}$-euclidean descent, on the other hand we have proved that the imaginary quadratic (maximal) orders are $\mathcal{S}$-euclidean domains. Now we combine these two facts in order to compute the sum of two principal ideals (quasi a GCD) in such rings of algebraic integers asymptotically fast.

Let K be an imaginary quadratic number field with fundamental discriminant $D < 0$, and let $\mathcal{O}_K = \mathcal{O}_D$ be its ring of algebraic integers. We know that the multiplication time is bounded by $\mu_{\mathcal{O}_D}(n) = O(\mu(n))$ for this maximal order and a fixed chosen integral basis. In addition to that we know the euclidean minimum

$$E_{\mathcal{O}_D} = E(\mathcal{O}_D, |\cdot|) = \begin{cases} \frac{\sqrt{|D|/4+1}}{2}, & \text{if } D \equiv 0 \bmod 4, \\ \frac{|D|+1}{4\sqrt{|D|}}, & \text{if } D \equiv 1 \bmod 4 \end{cases}$$

[23, Proposition 3.2] and know the behaviour of the prime ideals (inert, split, ramified) [8, Proposition 5.1.4] in order to efficiently calculate the valuation at the finitely many places $\mathfrak{p} \in V(\prod \mathcal{S}_D)$. Thus we can calculate the sum of two prinicipal ideals, each given by a generator, by calculating a controlled $\mathcal{S}_D$-euclidean descent with Algorithm 2.19 $\mathrm{DESCENT}_{\mathcal{O}_D}$ for these two generators, and then by calculating the valuations at the corresponding $\mathcal{S}_D$-places. The occuring ring constants of Algorithm 2.19 $\mathrm{DESCENT}_R$ are shown for the nine imaginary quadratic maximal orders with class number 1 and for a few maximal orders with class number 2 in Table 1. Thereby we chose $b = 2$ ($B = 2$) as the basis. For all of these maximal orders $R = \mathcal{O}_D$ we used the $\mathcal{S}_D$-euclidean set (cf. Definition 3.3). Note that in the case of $D \in \{-3, -4, -7, -8, -11\}$, the maximal orders are euclidean. Furthermore, we denote the maximum of $\mathcal{S}_D$ by S_R according to Theorem 3.1. This yields a lower bound for η as in (3.1), hence we chose $\eta < \sqrt{3}/2$ only a little above this bound (cf. (3.1)) in order to achieve as much progress as possible in every $\mathcal{S}_D$-euclidean step. Thus we have $E_R = \sqrt{\eta^2 + \frac{1}{4}}$. It should be mentioned that η can be close to $\sqrt{3}/2$ for a fixed chosen ring of algebraic integers $\mathcal{O}_D$, $D < 0$ such that E_R is close to 1. This means that—in the worst case—the progress of every $\mathcal{S}_D$-euclidean step can be small. The ring $\mathcal{O}_{-427}$ is an example for a large $E_R < 1$.

Fast computation of valuations in quadratic rings of algebraic integers. We can calculate the sum of two principal ideals (given by one generator each) in an imaginary quadratic ring of algebraic integers $\mathcal{O}_D$, using the Algorithm $\mathrm{DESCENT}_{\mathcal{O}_D}$ in order to compute an $\mathcal{S}_D$-euclidean descent, and then modifying the intermediate ideal at

TABLE 1. Examples for the ring-specific constants (or numerical approximation) in Algorithm 2.19 DESCENT$_R$ for $\mathbf{Z}$ and several imaginary quadratic maximal orders with class number 1 and 2.

R	$E(R,\lvert\cdot\rvert)$	E_R	E'_R	S_R	C_R	m_R	γ_R	δ_R	ℓ_R
$\mathbf{Z}$	$1/2$	$1/2$	0.7500	1	1	8.7500	4	1	3
$\mathcal{O}_{-3}$	$1/\sqrt{3}$	$1/\sqrt{3}$	0.7887	1	1	10.2528	4	2	5
$\mathcal{O}_{-4}$	$1/\sqrt{2}$	$1/\sqrt{2}$	0.8536	1	1	14.5105	5	2	7
$\mathcal{O}_{-7}$	$2/\sqrt{7}$	$2/\sqrt{7}$	0.8780	1	1	17.2667	5	3	11
$\mathcal{O}_{-8}$	$\sqrt{3}/2$	$\sqrt{3}/2$	0.9331	1	1	30.7895	6	3	21
$\mathcal{O}_{-11}$	$3/\sqrt{11}$	$3/\sqrt{11}$	0.9523	1	1	42.8521	7	4	36
$\mathcal{O}_{-19}$	$5/\sqrt{19}$	0.8820	0.9641	2	13	50.6843	8	4	29
$\mathcal{O}_{-43}$	$11/\sqrt{43}$	0.9602	0.9801	3	56	200.6255	11	5	104
$\mathcal{O}_{-67}$	$17/\sqrt{67}$	0.9592	0.9796	4	69	244.3554	12	5	101
$\mathcal{O}_{-163}$	$41/\sqrt{163}$	0.9417	0.9709	7	67	274.1137	14	5	71
$\mathcal{O}_{-15}$	$4/\sqrt{15}$	0.8165	0.9083	2	9	32.3862	8	3	15
$\mathcal{O}_{-20}$	$\sqrt{3/2}$	0.8976	0.9488	2	15	58.2239	9	4	34
$\mathcal{O}_{-24}$	$\sqrt{7}/2$	0.9575	0.9788	2	34	140.6459	10	5	97
$\mathcal{O}_{-403}$	$101/\sqrt{403}$	0.9746	0.9873	11	188	941.1931	17	6	189
$\mathcal{O}_{-427}$	$107/\sqrt{427}$	0.9957	0.9979	11	1103	5517.0133	20	8	1433

the $V(\prod \mathcal{S}_D)$ places using valuations. Thus, in this section, we now discuss how to quickly compute the valuations for an integral ideal.

Assume that we would like to calculate valuations at the place $\mathfrak{p}$. Our asymptotically fast computation of $\mathfrak{p}$-valuations will be based on a precalculated representation of $\mathfrak{p}^k$ as $\mathbf{Z}$-modules. For a fixed chosen quadratic ring of algebraic integers $\mathcal{O}_D$, we know the $\mathbf{Z}$-module representation for every prime ideal, and we only have to consider finitely many $\mathfrak{p} \in V(\prod \mathcal{S}_D)$. Starting with such a $\mathbf{Z}$-module representation of $\mathfrak{p}$, we use Hensel lifting in order to calculate the $\mathbf{Z}$-module representation for $\mathfrak{p}^k$. Thereby k depends on the size of the operand for which we are going to calculate the $\mathfrak{p}$-valuation.

Theorem 3.9 (Hensel lifting)**.** *Let $m \in \mathbf{N}_{\geq 1}$, and let $f, f_1, f_2, c_1, c_2 \in \mathbf{Z}[x]$ be polynomials with*

$$f \equiv f_1 f_2 \bmod m, \quad c_1 f_1 + c_2 f_2 \equiv 1 \bmod m.$$

Assume that the leading coefficient of f_2 is equal to 1, $n := \deg f = \deg f_1 + \deg f_2$, and $\deg c_1 < \deg f_2$, $\deg c_2 < \deg f_1$. Then there exists an algorithm that lifts the factorization of f to accuracy *m^l, $l \in \mathbf{N}_{\geq 1}$, i.e., there exist polynomials $f_1^*, f_2^*, c_1^*, c_2^* \in \mathbf{Z}[x]$ such that*

$$f \equiv f_1^* f_2^* \bmod m^l, \quad c_1^* f_1^* + c_2^* f_2^* \equiv 1 \bmod m^l,$$

the leading coefficient of f_2^* is equal to 1, and

$$f_1^* \equiv f_1 \bmod m,\ \deg f_1^* = \deg f_1, \qquad c_1^* \equiv c_1 \bmod m,\ \deg c_1^* < \deg f_2^*,$$
$$f_2^* \equiv f_2 \bmod m,\ \deg f_2^* = \deg f_2, \qquad c_2^* \equiv c_2 \bmod m,\ \deg c_2^* < \deg f_1^*.$$

The running time of this algorithm is bounded by $O(\mu(n)\mu(\text{size}(m^l)))$.

Proof. [37, Algorithm 15.10, Theorem 15.11, Theorem 15.12]. □

Lemma 3.10. *Let D be a fundamental discriminant, let $p \in \mathbf{N}_{>1}$ be a prime number, and let $(D/p) = +1$, hence $p\mathcal{O}_D = \mathfrak{p}\bar{\mathfrak{p}}$ is split. Let $k \in \mathbf{N}_{\geq 1}$, and let $b^2 \equiv D \bmod 4p^k$. Then*

$$\mathfrak{p} = p\mathbf{Z} + \frac{-b+\sqrt{D}}{2}\mathbf{Z}$$

is a $\mathbf{Z}$-module representation of $\mathfrak{p}$ (suitable choice of the sign of b), and we have for $1 \leq \kappa \leq k$

$$\mathfrak{p}^\kappa = p^\kappa\mathbf{Z} + \frac{-b+\sqrt{D}}{2}\mathbf{Z}.$$

Proof. It is well known that $\mathfrak{p} = p\mathcal{O}_D + \left(\omega - \frac{-b+D}{2}\right)\mathcal{O}_D$ [8, Proposition 5.1.4], because $b^2 \equiv D \bmod 4p^k$ implies that $b^2 \equiv D \bmod 4p$. In addition to that, we have a $\mathbf{Z}$-module representation of $\mathfrak{p}$ with the same two generators because $4p \,|\, (b^2 - D)$. This proves the claim for $\kappa = 1$.

Due to this $\mathbf{Z}$-module representation of $\mathfrak{p}$ with one integer as a generator, we can use the correspondence between ideal multiplication and composition of binary quadratic forms in order to calculate powers of $\mathfrak{p}$. Assume that the claim holds true for every $\kappa \geq 1$ with $\kappa + 1 \leq k$. Then we calculate $\mathfrak{p}^\kappa \cdot \mathfrak{p}$ using the composition of the corresponding quadratic forms (mapping ϕ_{IF})

(3.2)
$$\left(p^\kappa, b, \frac{D-b^2}{4p^\kappa}\right) \cdot \left(p, b, \frac{D-b^2}{4p}\right) = \left(p^{\kappa+1}, b - 2p^\kappa w \cdot \frac{D-b^2}{4p}, *\right) \sim (p^{\kappa+1}, b, *),$$

because the assumption $(D/p) = +1$ implies that $D = b^2 - 4pc$ and p are coprime, hence b and p are coprime as well. There exist $u, w \in \mathbf{Z}$ such that $up + wb = 1$. It follows that $\frac{D-b^2}{4p}$ can be divided by p because of $k \geq \kappa + 1 \geq 2$. Thus

$$b - 2p^\kappa w \cdot \frac{D-b^2}{4p} = b - 2p^{\kappa+1}m, \quad m \in \mathbf{Z}.$$

For that reason the equivalence in (3.2) consists only of a Γ_∞-operation ($F(D)$), i.e., a change of the generators, but no change for the corresponding ideal. □

Remark 3.11. This lemma can be used in order to calculate powers of a prime ideal $\bar{\mathfrak{p}}$ in a $\mathbf{Z}$-module representation that lies above a split prime number p. We use this fact for the fast calculation of $\mathfrak{p}$-valuations.

Lemma 3.12. *Let I be an integral ideal in the quadratic ring of algebraic integers $\mathcal{O}_D$, and let $\mathfrak{p} \,|\, p$ be a prime ideal. Let $k \in \mathbf{N}$, and let $\varepsilon \in \{0, 1\}$ be such that $k + \varepsilon \equiv 0 \bmod 2$. Then $v_\mathfrak{p}(I) \geq k$ if and only if*

$$p^k \,|\, I \text{ if } \left(\frac{D}{p}\right) = -1; \quad p^{(k+\varepsilon)/2} \,|\, \mathfrak{p}^\varepsilon I \text{ if } \left(\frac{D}{p}\right) = 0; \quad p^k \,|\, \bar{\mathfrak{p}}^k I \text{ if } \left(\frac{D}{p}\right) = +1.$$

Proof. The claim follows from the decomposition of the prime numbers p in $\mathcal{O}_D$. □

Remark 3.13. Let $\mathfrak{p} \,|\, p$. In order to calculate $v_\mathfrak{p}(I)$ we can determine the maximum k such that $v_\mathfrak{p}(I) \geq k$. Lemma 3.12 shows how to reduce this problem to the question of whether an integral ideal I' is divisible by a suitable power of p. Thereby we have either I equal to I', or I' is the product of I and of an ideal for which a $\mathbf{Z}$-module representation is known (cf. Lemma 3.10).

Algorithm 3.14 (Fast calculation of the $\mathfrak{p}$-valuation of an integral ideal)**.** *Let $\mathfrak{p}$ be a prime ideal in the ring of algebraic integers $\mathcal{O}_D$ of the quadratic number field with fundamental discriminant D. Let I be an integral ideal, given in a representation with two $\mathbf{Z}$-generators. Then this algorithm calculates the $\mathfrak{p}$-valuation $v_{\mathfrak{p}}(I)$. Therefore $p \in \mathbf{N}$ denotes the unique prime number with $\mathfrak{p} \mid p$, and a representation with two generators of $\mathfrak{p}$ or $\bar{\mathfrak{p}}$ is known.*

algorithm $v_{\mathfrak{p}}(I)$

(V1) ***if*** $I = 0$ ***return*** ∞;
(V2) $k_{\max} := \lfloor \frac{1}{f} \log_p \mathrm{Norm}(I) \rfloor$; (*$f$ is the degree of $\mathfrak{p} \mid p$*)
(V3) *Calculate $p^{2^0}, p^{2^1}, \ldots, p^{2^\lambda}$ with $2^\lambda \le k_{\max} < 2^{\lambda+1}$ using successive squaring;*
(V4) ***if*** $(D/p) = +1$ ***then*** *(p split)*
  Compute $\bar{\mathfrak{p}}^{k_{\max}}$ in a representation as in Lemma 3.10;
  $I := \bar{\mathfrak{p}}^{k_{\max}} \cdot I$;
(V5) $v := 0$;
(V6) ***for*** $\kappa = \lambda, \ldots, 0$ ***do***
  if $p^{2^\kappa} \mid I$ ***then*** $v := v + 2^\kappa$, $I := I/p^{2^\kappa}$;
(V7) ***if*** $(D/p) = 0$ ***then*** *(p ramified)*
  $v := 2v$; ($\mathrm{Norm}(\mathfrak{p}) = p$)
  if $p \mid \mathfrak{p}I$ ***then*** $v := v + 1$;
(V8) ***return*** v.

Proposition 3.15. *Let $\mathfrak{p}$ be a prime ideal in the quadratic ring of algebraic integers $\mathcal{O}_D$ with the fundamental discriminant D. Let I be an integral ideal which is given by its HNF or by a 2-generator representation as $\mathbf{Z}$-module. Then Algorithm 3.14 calculates the valuation $v_{\mathfrak{p}}(I)$ in time $O(\mu(s) \log s)$, $s = O(\mathrm{size}(\mathrm{Norm}(I)))$.*

Proof. The case $I = 0$ is trivial, hence assume $I \neq 0$. Let $\mathfrak{p} \supset p\mathcal{O}_D$, i.e., $\mathfrak{p}$ lies above the prime number p. Assume $\mathfrak{p}^k \mid I$. It follows that $k \le \frac{1}{f} \log_p \mathrm{Norm}(I)$ due to $\mathrm{Norm}(\mathfrak{p}) = p^f \ge p$, hence we get an upper bound $k_{\max} \in \mathbf{N}$ for the maximum $\mathfrak{p}$-power that divides I.

The correctness of Algorithm 3.14 follows from Lemma 3.12. In (V3) we calculate 2^λ-powers of p such that we can obtain every power of p with an exponent less than $k_{\max}$ from these powers using multiplications. The running time for such computation can be bound by $O(\mu(\mathrm{size}(p^{k_{\max}}))) \le O(\mu(\mathrm{Norm}(I)))$ using a geometric series (cf. [25, 3.5.2]). Then, if $p\mathcal{O}_D$ is split, we calculate the $k_{\max}$-power of $\bar{\mathfrak{p}}$ according to Lemma 3.10. For that we lift a representation of $\bar{\mathfrak{p}}$ as in [8, Proposition 5.1.4] such that $b^2 \equiv D \bmod 4p^{k_{\max}}$.[3] This can be done in time $O(\mu(\mathrm{size}(p^{k_{\max}})))$. Afterwards we calculate the ideal $\bar{\mathfrak{p}}^{k_{\max}} \cdot I$ (like the composition of binary quadratic forms) and find out the largest p-power according to Lemma 3.12 which divides it. The running time for this step can be bounded by $O(\mu(s) \log s)$ because this step is as expensive as a GCD calculation in $\mathbf{Z}$.

In (V6) we calculate the maximum power of p which divides the ideal I (maybe changed in (V4)), independent of (D/p). This loop requires at most $O(\mu(s) \log s)$ bit operations.

[3] At first we know a solution for the equation $x^2 - D \equiv 0 \bmod 4p$, which we can transfer to a solution with *accuracy* $4p^{k_{\max}}$ using Hensel-lifting. In particular, we calculate a solution in case of $p = 2$ for $x^2 - D \equiv 0 \bmod p^{k+2}$, and in case of $p \neq 2$ for $x^2 - D \equiv 0 \bmod p^k$ using Hensel-lifting. After this we adjust x by adding p^k if $x \not\equiv D \bmod 2$.

In (V7) we treat the special case that p is ramified and double v because the ramification index is $e = 2$. Then it remains to test whether $\mathfrak{p}$ is contained in an odd power in the ideal I (all nonzero even powers are removed from I).

Altogether, the algorithm calculates the $\mathfrak{p}$-valuation correctly, and the stated running time holds true. □

Calculation of the sum of two principal ideals. Now we are prepared to present the fast Algorithm 3.16 $\mathrm{SGCD}_{\mathcal{O}_D}$ for the calculation of the sum of two principal ideals in an imaginary quadratic ring of algebraic integers $\mathcal{O}_D$. If this ring is a principal domain (i.e., $h(D) = 1$), then all the ideal operations can be calculated only with ring elements as generators of principal ideals. This can be done by the more simple Algorithm 3.17 $\mathrm{SGCD1}_{\mathcal{O}_D}$, which has the same asymptotical running time as Algorithm $\mathrm{SGCD}_{\mathcal{O}_D}$ with a smaller constant hidden in the O-notation.

Algorithm 3.16 (GCD calculation in $\mathcal{O}_D$ using an $\mathcal{S}_D$-euclidean descent)**.** *Algorithm* ***SGCD***$_{\mathcal{O}_D}$ *calculates the* "GCD ideal" $\mathfrak{g} = x\mathcal{O}_D + y\mathcal{O}_D$ *for* $x, y \in \mathcal{O}_D$. *Thereby* $\mathcal{O}_D$ *denotes the* $\mathcal{S}_D$*-euclidean ring of algebraic integers of the imaginary quadratic number field with fundamental discriminant* $D < 0$, *and the pair* $(1, \omega)$ *denotes an integral basis.*

algorithm SGCD$_{\mathcal{O}_D}(x, y)$

(S1) $x'' := \gcd_{\mathbf{Z}}(x_0, x_1)$, $y'' := \gcd_{\mathbf{Z}}(y_0, y_1)$; *(*$x = x_0 + \omega x_1$, $y = y_0 + \omega y_1$*)*

(S2) $x' := x/x''$, $y' := y/y''$; *(primitive part of the operands)*

(S3) *Calculate an asymptotically fast controlled* $\mathcal{S}_D$*-euclidean descent on* x' *and* y' *(Algorithm* 2.19 *DESCENT*$_{\mathcal{O}_D}(x', y', 0)$*)*
returning $g \in \mathcal{O}_D$ *(cofactor matrix is not required);*

(S4) $g' := g/\gcd_{\mathbf{Z}}(g_0, g_1)$; *(*$g = g_0 + \omega g_1$*)*

(S5) *Calculate representations of the primitive ideals* $x'\mathcal{O}_D, y'\mathcal{O}_D, \mathfrak{g}' := g'\mathcal{O}_D$ *as* $\mathbf{Z}$*-modules (two generators with one of them being an integer);*

(S6) ***for each*** $\mathfrak{p} \in V(\prod \mathcal{S}_D)$ ***do***

(a) *Calculate the* $\mathfrak{p}$*-valuations* $v_{\mathfrak{p}}(x'\mathcal{O}_D), v_{\mathfrak{p}}(y'\mathcal{O}_D), v_{\mathfrak{p}}(\mathfrak{g}')$ *using Algorithm* 3.14*;*

(b) *Calculate* $\mathfrak{g}' := \mathfrak{g}' \cdot \mathfrak{p}^{\min(v_{\mathfrak{p}}(x'\mathcal{O}_D), v_{\mathfrak{p}}(y'\mathcal{O}_D)) - v_{\mathfrak{p}}(\mathfrak{g}')}$
(using the corresponding composition of binary quadratic forms);

(S7) ***return*** $\mathfrak{g}' \cdot \gcd_{\mathbf{Z}}(x'', y'')$.

Algorithm 3.17 (GCD calculation in $\mathcal{O}_D$ with $h(D) = 1$ using an $\mathcal{S}_D$-euclidean descent)**.** *Algorithm* ***SGCD1***$_{\mathcal{O}_D}$ *calculates a generator* g *for the* "GCD ideal" $g\mathcal{O}_D = x\mathcal{O}_D + y\mathcal{O}_D$ *for* $x, y \in \mathcal{O}_D$. *Thereby* $\mathcal{O}_D$ *denotes the* $\mathcal{S}_D$*-euclidean ring of algebraic integers of the imaginary quadratic number field with fundamental discriminant* $D < 0$ *and class number* $h(D) = 1$, *and the pair* $(1, \omega)$ *denotes an integral basis.*

algorithm SGCD1$_{\mathcal{O}_D}(x, y)$

(S′1) *Calculate an asymptotically fast controlled* $\mathcal{S}_D$*-euclidean descent on* x *and* y *(Algorithm* 2.19 *DESCENT*$_{\mathcal{O}_D}(x, y, 0)$*) returning* $g \in \mathcal{O}_D$ *(cofactor matrix is not required);*

(S′2) ***for each*** $\mathfrak{p} \in V(\prod \mathcal{S}_D)$ ***do*** *(*p *is a generator of the principal ideal:* $\mathfrak{p} = p\mathcal{O}_D$*)*
$g := g \cdot p^{\min(v_{\mathfrak{p}}(x), v_{\mathfrak{p}}(y)) - v_{\mathfrak{p}}(g)}$;

(S′3) ***return*** g *(*$\mathfrak{g} = g\mathcal{O}_D$*).*

Theorem 3.18. *Let $\mathcal{O}_D$ be the imaginary quadratic ring of algebraic integers with fundamental discriminant $D < 0$, and let $(1, \omega)$ be an integral basis. For $x, y \in \mathcal{O}_D$ with $\mathrm{Norm}(x), \mathrm{Norm}(y) < 2^n$, Algorithm 3.16 $SGCD_{\mathcal{O}_D}$ calculates the ideal $\mathfrak{g} = x\mathcal{O}_D + y\mathcal{O}_D$ in running time $O(\mu_{\mathcal{O}_D}(n) \log n)$.*

If it is known that $h(D) = 1$, i.e., $\mathcal{O}_D$ is a principal domain, then Algorithm 3.17 $SGCD1_{\mathcal{O}_D}$ calculates a generator $g \in \mathcal{O}_D$ of the ideal $g\mathcal{O}_D = \mathfrak{g} = x\mathcal{O}_D + y\mathcal{O}_D$ in the same asymptotic running time.

Proof. In order to prove the correctness of the Algorithms 3.16 and 3.17, we have to show that for all prime ideals $\mathfrak{p}$ $v_\mathfrak{p}(\mathfrak{g}) = \min(v_\mathfrak{p}(x), v_\mathfrak{p}(y))$ holds true.

If $\mathfrak{p} \notin V(\prod \mathcal{S}_D)$, the claim follows from Lemma 2.16, because $\mathcal{O}_D$ is $\mathcal{S}_D$-euclidean w. r. t. $|\cdot|$ according to Theorem 3.1, and because the correct part of the GCD is calculated at these places in an $\mathcal{S}_D$-euclidean descent. The calculation of the part of the GCD at the places $\mathfrak{p} \in V(\prod \mathcal{S}_D)$ is done with valuations which removes an $\mathcal{S}_D$ part due to the $\mathcal{S}_D$-euclidean descent from the calculated ideal, if necessary.

Thus it remains to prove the stated running time. The norm is a positive definite quadratic form for the coefficients of an algebraic integer in $\mathcal{O}_D$ to a fixed chosen integral basis $(1, \omega)$. Set $x = x_0 + \omega x_1$. Because $\mathrm{Norm}(x)$ is bounded, it follows that $|x_0|, |x_1|$ are bounded. Without loss of generality let the absolute values of the coefficients x_0, x_1 be bounded by 2^n, hence the absolute values of the nonprimitive part of the operands $|x''|, |y''|$ are bounded by 2^n in step (S1). The multiplicativity of the norm implies that $\mathrm{Norm}(x'), \mathrm{Norm}(y'), \mathrm{Norm}(g), \mathrm{Norm}(g')$ are bounded by 2^n as well. We precalculate the $\mathbf{Z}$-part of the GCD ideal $\mathfrak{g}$ in order to discuss primitive elements or, respectively, primitive principal ideals. Thus we are allowed to reduce g to a primitive g' in (S4) because g cannot contain a nonprimitive relevant part of $\mathfrak{g}$.

The calculation of 2-generator representations of $x'\mathcal{O}_D, y'\mathcal{O}_D, g'\mathcal{O}_D$ can be done as follows using HNF-reduction: Let $z \in \mathcal{O}_D$ be primitive. The minimal polynomial of ω is $x^2 - \mathrm{Tr}(\omega)x + \mathrm{Norm}(\omega)$. Then the prinicipal ideal $z\mathcal{O}_D$ is generated by z and ωz as a $\mathbf{Z}$-module, hence

$$z\mathcal{O}_D = z\mathbf{Z} + z\omega\mathbf{Z} = (z_0 + \omega z_1)\mathbf{Z} + (-\mathrm{Norm}(\omega)z_1 + \omega(z_0 + \mathrm{Tr}(\omega)z_1))\mathbf{Z}.$$

Because z is primitive, there exist $\zeta_0, \zeta_1 \in \mathcal{O}_D$ such that $\zeta_0 z_0 + \zeta_1 z_1 = 1$.

$$\begin{aligned}
&\begin{pmatrix} z_0 & -\mathrm{Norm}(\omega)z_1 \\ z_1 & z_0 + \mathrm{Tr}(\omega)z_1 \end{pmatrix} \cdot \begin{pmatrix} z_0 + \mathrm{Tr}(\omega)z_1 & \zeta_1 - \mathrm{Tr}(\omega)\zeta_0 \\ -z_1 & \zeta_0 \end{pmatrix} \\
&\quad = \begin{pmatrix} z_0(z_0 + \mathrm{Tr}(\omega)z_1) + \mathrm{Norm}(\omega)z_1^2 & z_0(\zeta_1 - \mathrm{Tr}(\omega)\zeta_0) - \mathrm{Norm}(\omega)z_1\zeta_0 \\ 0 & 1 \end{pmatrix} \\
&\quad = \begin{pmatrix} \mathrm{Norm}(z) & b' \\ 0 & 1 \end{pmatrix}.
\end{aligned}$$

If we apply a column operation to this matrix which reduces b' by multiples of $\mathrm{Norm}(z)$ to b with $0 \le b < \mathrm{Norm}(z)$, we obtain the HNF of the primitive ideal $z\mathcal{O}_D$. It represents the ideal $z\mathcal{O}_D$, which has the following 2-generator representation as a $\mathbf{Z}$-module (one integer generator) according to (1.1):

$$z\mathcal{O}_D = \mathrm{Norm}(z)\mathbf{Z} + \left(b + \frac{D + \sqrt{D}}{2}\right)\mathbf{Z} = \mathrm{Norm}(z)\mathbf{Z} + \frac{-(-2b - D) + \sqrt{D}}{2}\mathbf{Z}.$$

The running time of the calculation of the $\mathbf{Z}$-module representation is $O(\mu(n)\log n)$, because the running time of the GCD computation dominates the whole computation.

The number of iterations in the loop (S6) for a fixed chosen ring of algebraic integers $\mathcal{O}_D$ is bounded by $O(1)$, because there are at most two prime ideals lying above a prime number, and the number of prime ideals in $\mathcal{S}_D$ is uniformly bounded, independent of the operands x, y. The calculation of the valuations in (S6a) requires $O(\mu(n)\log n)$ running time according to Proposition 3.15. The multiplication of the ideals in (S6b) is possible in the same time, because the calculation of powers of $\mathfrak{p}$ can be done in an analogous manner as the calculation of a prime power in Algorithm 3.14, and the subsequent ideal multiplication can be done using the composition of binary quadratic forms. Altogether the norms of the occuring ideals are bounded because of the multiplicativity of the norm, and because the norm of the GCD ideal divides the norm of the principal ideals $x\mathcal{O}_D, y\mathcal{O}_D$.

In the case of a principal ideal domain $\mathcal{O}_D$ we can simplify Algorithm 3.16 to Algorithm 3.17, because the whole ideal arithmetic can be done with generators of principal ideals. The calculation of the $\mathcal{S}_D$-euclidean descent is possible in time $O(\mu(n)\log n)$; the calculation of each step in the loop (S′2) can be done in time $O(\mu(n))$ such that the stated running time holds true for Algorithm 3.17 as well. It is only faster by a constant factor than Algorithm 3.16. □

Remark 3.19. Now we will discuss some possible improvements of the Algorithms 3.16 and 3.17.

(1) We can reduce the loop in (S6) or (S′2) to the possibly smaller set $V(\prod \hat{\mathcal{S}}_D)$, where $\hat{\mathcal{S}}_D$ contains only the $s \in \mathcal{S}_D$ which were used in at least one $\mathcal{S}_D$-euclidean step $s{\cdot}x = qy + r$ of the $\mathcal{S}_D$-euclidean descent.

It seems that this would not be any significant improvement for large operands because in most cases all the prime places of $V(\prod \mathcal{S}_D)$ will occur in an $\mathcal{S}_D$-euclidean descent. In addition to that, we had to extend Algorithm 2.19 ($\mathrm{DESCENT}_{\mathcal{O}_D}$) such that it returns the subset $\hat{\mathcal{S}}_D$ of the *used* $s \in \mathcal{S}_D$.

(2) Algorithm 3.17 $\mathrm{SGCD1}_{\mathcal{O}_D}$ only holds true for the nine fundamental discriminants $D \in -3, -4, -7, -8, -11, -19, -43, -67, -163$, for which $\mathcal{O}_D$ is a principal ideal domain [8, Section 7.2.4]. In general we can use Algorithm 3.16 $\mathrm{SGCD}_{\mathcal{O}_D}$ for the calculation of the ideal sum without knowledge of the class number $h(D)$, but this may not lead to a generator of the principal ideal, even in case of $h(D) = 1$.

We conclude this section by discussing how the running times of these algorithms depend on the fundamental discriminant. We assume that a fundamental discriminant $D < 0$ and the minimal polynomial for ω, where $(1, \omega)$ is an integral basis, are inputs for the Algorithm $\mathrm{SGCD}_{\mathcal{O}_D}$ or $\mathrm{SGCD1}_{\mathcal{O}_D}$. Then we can bound the running time by $O(c(D){\cdot}\mu_{\mathcal{O}_D}(N)\log N)$ with $N = O(n + \mathrm{size}(\omega))$, and $\mathrm{size}(\omega)$ denotes the size of the coefficients of the minimal polynomial of ω. $c(D)$ does not depend on the operands x, y, but it depends in many ways on properties of the imaginary quadratic ring of algebraic integers $\mathcal{O}_D$ (cf. Table 1). The calculation of an $\mathcal{S}_D$-euclidean step can be bounded by the running time for the calculation of at most $O(\sqrt{|D|})$ euclidean steps for corresponding operands because the $\mathcal{S}_D$-euclidean set has cardinality of $\lfloor\sqrt{|D|/3}\rfloor$. In addition to that $c(D)$ is influenced

in a nontrivial manner by further ring constants, e.g., $E'_{\mathcal{O}_D}$ and $\ell_{\mathcal{O}_D}$. In particular, we do not have estimates for $c(D)$ which are monotone in $|D|$, because some of the constants depend on suitable values for η (3.1), and η depends on the distance of $\sqrt{|D|/3}$ to the next integer. It seems that $c(D)$ grows asymptotically as fast as $\sqrt{D}/3$. If this holds true, then the running time of the Algorithms 3.16 and 3.17 is not polynomially bounded in the size of D, because a binary coding of D has size $O(\log|D|)$.

4. Ideal sums in quadratic orders

Now we present a different approach for the calculation of the sum of two ideals in any quadratic order. The following Algorithm 4.2 IDEALSUM$_{\mathcal{O}_D}$ generalizes the previous algorithm in three ways. We are able to compute ideal sums

- with ideals, not necessarily principal ideals,
- in real quadratic fields,
- in orders that are not necessarily maximal orders.

Let D be a squarefree number congruent to 0 or 1 modulo 4. Let $\mathcal{O}_D$ be the unique order with discriminant D of the quadratic number field $\mathbf{Q}(\sqrt{D})$. Let the pair $(1,\omega)$ form an integral basis of $\mathcal{O}_D$, where ω is specified by its minimal polynomial $x^2 - \mathrm{Tr}_{\mathbf{Q}(\sqrt{D})/\mathbf{Q}}(\omega)x + \mathrm{Norm}_{\mathbf{Q}(\sqrt{D})/\mathbf{Q}}(\omega)$ over $\mathbf{Q}$. For that reason we are not able to distinguish between ω and its conjugate $\sigma(\omega)$, where σ is the unique nontrivial automorphism of $\mathbf{Q}(\sqrt{D})$ which is defined by $\sigma:\ \sqrt{D} \mapsto -\sqrt{D}$.

4.1. **Coding of ideals.** Our Algorithm 4.2 requires as inputs two integral ideals in $\mathcal{O}_D$, coded as $\mathbf{Z}$-modules with the basis $(1,\omega)$. These inputs can be specified as two $2\times k$ matrices (e.g., the matrix in HNF for the two ideals). Now we show how to convert different codings of ideals in $\mathcal{O}_D$ into the preferred coding for the algorithm.

(1) $I = \sum_{\kappa=1}^{k} x_\kappa \mathcal{O}_D,\ x_\kappa \in \mathcal{O}_D$.

If the integral ideal I is given by k $\mathcal{O}_D$-generators (if I is a principal ideal, then $k=1$ is possible), then we can represent I with $2k$ $\mathbf{Z}$-generators because of $\mathcal{O}_D = \mathbf{Z} + \omega\mathbf{Z}$.

$$I = \sum_{\kappa=1}^{k} x_\kappa(\mathbf{Z}+\omega\mathbf{Z}) = \sum_{\kappa=1}^{k} x_\kappa\mathbf{Z} + \sum_{\kappa=1}^{k} \omega x_\kappa \mathbf{Z}.$$

Let

$$x_\kappa = x_\kappa^{(0)} + \omega x_\kappa^{(1)} \in \mathcal{O}_D, \qquad x_\kappa^{(0)}, x_\kappa^{(1)} \in \mathbf{Z},$$

be the unique representation w. r. t. the integral basis $(1,\omega)$. Then we have

$$\omega x_\kappa = -\mathrm{Norm}(\omega)x_\kappa^{(1)} + \omega(\mathrm{Tr}(\omega)x_\kappa^{(1)} + x_\kappa^{(0)})$$

such that the $\mathbf{Z}$-module I w. r. t. the integral basis $(1,\omega)$ can be represented by the $2\times 2k$ matrix

$$\begin{pmatrix} \cdots & x_\kappa^{(0)} & -\mathrm{Norm}(\omega)x_\kappa^{(1)} & \cdots \\ \cdots & x_\kappa^{(1)} & x_\kappa^{(0)} + \mathrm{Tr}(\omega)x_\kappa^{(1)} & \cdots \end{pmatrix}.$$

We see that I has a representation with finitely many $\mathbf{Z}$-generators (see case (2)). Furthermore we can assume that k is bounded by 2 because

every integral ideal in a quadratic order has a representation with no more than two $\mathcal{O}_D$-generators [8, Propositions 4.7.7 and 5.2.1].

(2) $I = \sum_{\kappa=1}^{k} y_\kappa \mathbf{Z},\ y_\kappa \in \mathcal{O}_D$.

The ideal I is given by k $\mathbf{Z}$-generators. Every y_κ has a unique representation $y_\kappa = y_\kappa^{(0)} + \omega y_\kappa^{(1)}$ w. r. t. the integral basis $(1, \omega)$. For that reason the $2 \times k$ matrix

$$\begin{pmatrix} \cdots & y_\kappa^{(0)} & \cdots \\ \cdots & y_\kappa^{(1)} & \cdots \end{pmatrix}$$

represents the ideal I as a $\mathbf{Z}$-module. Furthermore, one can represent I with a unique upper triangular matrix ($k = 2$), the HNF of I (see case (3)).

(3) Let I be given as a $\mathbf{Z}$-module by a $2 \times k$ matrix A.

We can assume that k is bounded by 8. Let the coefficients be bounded by 2^n in absolute value. Then we are able to compute the HNF of A in time $O(\mu(n)\log(n))$ [13, Theorem 2.1].

Corollary 4.1. *Let I, J be two integral ideals in $\mathcal{O}_D$, given in one of the codings* (1), (2) *or* (3) *w. r. t. the integral basis $(1, \omega)$. Let the generators for this basis be bounded by 2^n. Then we can compute the HNF of the ideal $I + J$ in time $O(\mu(N)\log N)$, where $N = O(n + \operatorname{size}(\omega))$.*

Proof. If one of the integral ideals I, J is given by two $\mathcal{O}_D$-generators (special case of 1), then we can calculate a representation of the ideal with four $\mathbf{Z}$-generators in time $O(\mu(n + \operatorname{size}(\omega)))$ using the minimal polynomial of ω, where

$$\operatorname{size}(\omega) = \max(\operatorname{size}(\operatorname{Tr}(\omega)), \operatorname{size}(\operatorname{Norm}(\omega)))$$

is an upper bound for the size of the coefficients of the minimal polynomial of ω. The size of the occuring coefficients in a representation with $\mathbf{Z}$-generators is bounded by $O(n + \operatorname{size}(\omega))$.

Altogether we can assume that the two ideals I, J are given by (at most) four $\mathbf{Z}$-generators with size $O(n + \operatorname{size}(\omega))$. Let A_I, A_J denote the corresponding matrices to I, J, each having two rows. Concatenating A_I and A_J yields a $2 \times k$ matrix with $k \leq 8$, which represents the ideal $I + J$ as a $\mathbf{Z}$-module. Using [13, Theorem 2.1] we can calculate the HNF of $I + J$ in time $O(\mu(N)\log N)$ with $N = O(n + \operatorname{size}(\omega))$.

For a fixed chosen order $\mathcal{O}_D$ and a fixed chosen integral basis $(1, \omega)$, we get a running time of $O(\mu(n)\log n)$ for the HNF computation. □

4.2. **The algorithm IDEALSUM.** The following Algorithm 4.2 IDEALSUM$_{\mathcal{O}_D}$ first calculates the HNF of the sum $I + J$, where I and J are the integral ideals in $\mathcal{O}_D$ and the input parameters for the algorithm. The HNF of the ideal $I + J$ w. r. t. the integral basis $(1, \omega)$ is unique [8, Proposition 5.2.1] such that we have a unique representation of $I + J$. But we are not able to determine whether the Ideal $I + J$ is a principal ideal, and, if it is a prinicipal ideal, which element in $\mathcal{O}_D$ is a generator of $I + J$. To put it in general words, we could be interested in a representation for $I + J$ as the product of a canonical representative of the class group $Cl(D)$ and an algebraic number in $\operatorname{Quot}\mathcal{O}_D = \mathbf{Q}(\sqrt{D})$. For that reason we change the representation of $I + J$ from a view as a $\mathbf{Z}$-module (HNF) to a representation as a binary quadratic form. Calculating a reduction of binary quadratic forms, we can find an answer to the question of whether $I + J$ is a principal ideal. If this is true and $\mathcal{O}_D$ is an imaginary quadratic order, we can calculate a generator for the

principal ideal $I+J$ without any knowledge of the class number $h(D)$ (particularly, this is always possible in the case of $h(D)=1$).

Algorithm 4.2 (Fast computation of the sum of two integral ideals in $\mathcal{O}_D$)**.** *Let I and J be two integral ideals in $\mathcal{O}_D$ in a suitable coding as discussed above. Then the algorithm* ***IDEALSUM***$_{\mathcal{O}_D}$ *calculates the HNF of $I+J$ and the representation of $I+J$ as a product of a canonical element of the class group $Cl(D)$ and of an element of* $\operatorname{Quot}\mathcal{O}_D$.

algorithm IDEALSUM$_{\mathcal{O}_D}(I,J)$

(I0) *Calculate (if necessary) the coding of I and J as $\mathbf{Z}$-matrices with two rows and a bounded number of columns;*

(I1) *Calculate the HNF $H=\left(\begin{smallmatrix}A' & B'\\ 0 & C'\end{smallmatrix}\right)$ of $I+J$ w. r. t. the integral basis $(1,\omega)$;*

(I2) *Set $A:=A'/C'$, $B:=-2\cdot B'/C'-\operatorname{Tr}(\omega)$.*
Then we have $I+J=C'\cdot\left(A\mathbf{Z}+\frac{-B+\sqrt{D}}{2}\mathbf{Z}\right)$;

(I3) *$g:=(A,B,*)$ (g is a quadratic form with discriminant D);*

(I4) *Calculate a reduced quadratic form $f=(a,b,c)$ which is equivalent to g and a matrix $M\in\Gamma$ using Schönhage's fast algorithm for the reduction of a binary quadratic form* [30] *and at most four further reduction steps ($M\cdot g=f$);*

(I5) *$s:=C'\cdot\left(\delta-\gamma\frac{b+\sqrt{D}}{2a}\right)$, where $M=\left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right)$;*

(I6) ***return*** *H, f, s.*

Now we have to show that this algorithm is correct, i.e., f and s are a unique representation for the ideal $I+J$, and that the running time of the algorithm is bounded.

Theorem 4.3. *Algorithm* 4.2 *IDEALSUM$_{\mathcal{O}_D}$ calculates for the inputs I and J a coding of the ideal $I+J$ in time $O(\mu(N)\log N)$, i.e., the Hermite normal form H and a quadratic form f as representative of the class group (using the mapping ϕ_{FI})*[4] *and a factor $s\in K=\operatorname{Quot}\mathcal{O}_D$. Let 2^n be an upper bound for the generators/coefficients of the coding of the ideals I,J, and set $N:=n+\operatorname{size}(\omega)$ for a fixed chosen integral basis $(1,\omega)$ of $\mathcal{O}_D$.*

Proof. We have to show that all steps of the Algorithm 4.2 IDEALSUM$_{\mathcal{O}_D}$ can be calculated in the stated running time, and that $I+J=s\cdot\phi_{FI}(f)$.

The integral ideals I and J are both given with finitely many generators such that we can calculate a $\mathbf{Z}$ matrix with two rows in line (I0) in time $O(\mu(N))$ which is a coding for the $\mathbf{Z}$-module $I+J$. From this matrix we can calculate the Hermite normal form $H\in\mathbf{Z}^{2\times 2}$ of $I+J$ in line (I1) in time $O(\mu(N)\log N)$ using Hafner and McCurley's theorem [13, Theorem 2.1]. We know from (1.1) that $H=\left(\begin{smallmatrix}A' & B'\\ 0 & C'\end{smallmatrix}\right)$ is a coding for the ideal

$$I+J=A'\mathbf{Z}+(B'+C'\omega)\mathbf{Z}=C'\cdot(A\mathbf{Z}+(B'/C'+\omega)\mathbf{Z}). \tag{4.1}$$

Denote by σ the nontrivial automorphism of the number field K. Then we calculate the discriminant D as

$$D=d(1,\omega)=\det\begin{pmatrix}1 & \omega\\ 1 & \sigma(\omega)\end{pmatrix}^2=(\omega-\sigma(\omega))^2.$$

[4] In contrast to (1.3) we are going to use ϕ_{FI} as a mapping from the set of quadratic forms into the set of ideals—without any coding of signs.

This implies that $\omega - \sigma(\omega) = \sqrt{D}$, and this defines which of the two conjugates is chosen for ω. One can calculate that

$$\omega - \frac{\sqrt{D}}{2} = \frac{2\omega - (\omega - \sigma(\omega))}{2} = \frac{\mathrm{Tr}(\omega)}{2}$$

such that (4.1) implies that

$$\begin{aligned} I + J &= C' \cdot \left(A\mathbf{Z} + \left(B'/C' + \frac{\mathrm{Tr}(\omega)}{2} + \frac{\sqrt{D}}{2} \right) \mathbf{Z} \right) \\ &= C' \cdot \left(A\mathbf{Z} + \frac{-(-2 \cdot B'/C' - \mathrm{Tr}(\omega)) + \sqrt{D}}{2} \mathbf{Z} \right), \end{aligned}$$

i.e., we calculate the corresponding quadratic form $g = \phi_{IF}(I + J)$ to f in (I2) and (I3). The reduction of this form g to an equivalent form $f \in \mathcal{R}(D)$ can be done in time $O(\mu(N) \log N)$ using Schönhage's fast reduction algorithm [30] (cf. Theorem 1.2), because g's coefficients are bounded by $O(2^N)$ in absolute value. Let $M \in \Gamma$ be the reduction matrix with $M \cdot g = f$.

It is known that the matrices

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

generate the entire module group Γ [5, Theorem 1.2] and that M has a representation as

$$M = S^{m_0} T S^{m_1} T S^{m_2} \cdots S^{m_{r-1}} T S^{m_r} \tag{4.2}$$

with minimal r (avoiding redundant factors like $(TS)^3 = -\mathrm{Id}$). Note that $M \cdot g = f$ is equivalent with

$$\begin{aligned} &(S^{m_0} T S^{m_1} \cdots S^{m_{r-1}} T S^{m_r})^T \cdot \hat{g} \cdot (S^{m_0} T S^{m_1} \cdots S^{m_{r-1}} T S^{m_r}) = \hat{f} \\ \Leftrightarrow\ &(S^{m_r T} (T^T (\cdots (T^T (S^{m_0 T} \cdot \hat{g} \cdot S^{m_0}) T) \cdots) T) S^{m_r}) = \hat{f} \end{aligned}$$

using (1.2), where $\hat{f}, \hat{g}$ are the corresponding matrices to the quadratic forms f, g.

Now we have to consider the behaviour of the ideals with the S- and T-operations applied to the corresponding quadratic forms. Let $g = (A, B, C) \in \mathcal{F}(D)$ be a quadratic form. An S-operation applied to g, $S^m \cdot g$, does not change the corresponding ideals, i.e.,

$$\phi_{FI}(g) = A\mathbf{Z} + \frac{-B + \sqrt{D}}{2}\mathbf{Z} = A\mathbf{Z} + \frac{-(B + 2mA) + \sqrt{D}}{2}\mathbf{Z} = \phi_{FI}(S^m \cdot g). \tag{4.3}$$

In contrast to this, the corresponding ideals of g and $T \cdot g$ differ by an element $t \in K$,

$$\begin{aligned} \phi_{FI}(g) &= A\mathbf{Z} + \frac{-B + \sqrt{D}}{2}\mathbf{Z} = -\frac{-B + \sqrt{D}}{2C} \cdot \left(C\mathbf{Z} + \frac{B + \sqrt{D}}{2}\mathbf{Z} \right) \\ &= -\frac{-B + \sqrt{D}}{2C} \cdot \phi_{FI}(T \cdot g). \end{aligned} \tag{4.4}$$

Applying (4.3) and (4.4) w. r. t. a representation of M yields

$$\begin{aligned} \phi_{FI}(g) = \phi_{FI}(S^{m_0} \cdot g) &= s_1 \cdot \phi_{FI}(T \cdot (S^{m_0} \cdot g)) = s_1 \cdot \phi_{FI}((S^{m_0} T) \cdot g) \\ &= \cdots = s_1 \cdots s_r \cdot \phi_{FI}(M \cdot g) = s_1 \cdots s_r \cdot \phi_{FI}(f) \end{aligned}$$

by the definition of the operation (cf. (1.2)). Let $g, f \in \mathcal{F}(D)$ be equivalent quadratic forms with

$$M \cdot g = f, \quad f = (a, b, c), \quad M = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \in \Gamma.$$

Then we have to show that the following relation for the corresponding ideals holds true:

$$\phi_{FI}(g) = \left(\delta - \gamma \frac{b + \sqrt{D}}{2a} \right) \cdot \phi_{FI}(f).$$

In order to prove this we use induction in the length r of a minimal representation of M (4.2).

In the case of $r = 0$ we have $M = S^{m_0}$ such that the S^{m_0}-operation yields the trivial factor $t = 1$ ($\gamma = 0,\ \delta = 1$ in M).

Now assume $r \geq 1$, and assume that the statement is valid for $r - 1$. We decompose M into the factors M', T and S^{m_r} with $M = M'TS^{m_r}$.

(1) Let $(a', b', c') = M' \cdot g$. Using the induction assumption we get the factor

$$t_1 = \delta' - \gamma' \frac{b' + \sqrt{D}}{2a'}, \qquad M' = \begin{pmatrix} \alpha' & \beta' \\ \gamma' & \delta' \end{pmatrix},$$

with the transition from $\phi_{FI}(g)$ to $\phi_{FI}(a', b', c')$, i.e.,

$$\phi_{FI}(g) = t_1 \cdot \phi_{FI}(a', b', c'). \tag{4.5}$$

(2) Now we are able to apply an operation of T to (a', b', c'). Then we get $t_2 = -\frac{-b' + \sqrt{D}}{2c'}$ for the transition from $\phi_{FI}(a', b', c')$ to $\phi_{FI}(T \cdot (a', b', c')) = \phi_{FI}(c', -b', a')$, i.e.,

$$\phi_{FI}(a', b', c') = t_2 \cdot \phi_{FI}(c', -b', a'). \tag{4.6}$$

(3) The operation of S^{m_r} on $(c', -b', a')$ does not yield a further factor. We calculate the corresponding quadratic form $S^{m_r} \cdot (c', -b', a') = (c', -b' + 2c'm_r, *) = (a, b, c)$ with

$$\phi_{FI}(c', -b', a') = \phi_{FI}(a, b, c). \tag{4.7}$$

Using equations (4.5), (4.6) and (4.7) we get

$$\phi_{FI}(g) = t_1 \cdot \phi_{FI}(a', b', c') = t_1 \cdot t_2 \cdot \phi_{FI}(a, b, c) = t_1 \cdot t_2 \cdot \phi_{FI}(f).$$

The transformation from g to the reduced equivalent quadratic form f yields the factor $t_1 \cdot t_2 \in K$ with

$$\begin{aligned} t = t_1 \cdot t_2 &= \left(\delta' - \gamma' \frac{b' + \sqrt{D}}{2a'} \right) \cdot \left(-\frac{-b' + \sqrt{D}}{2c'} \right) \\ &= -\delta' \frac{-b' + \sqrt{D}}{2c'} + \gamma' \frac{D - b'^2}{4a'c'} = -\gamma' - \delta' \frac{-b' + 2c'm_r - 2c'm_r + \sqrt{D}}{2c'} \\ &= -\gamma' + \delta' m_r - \delta' \frac{-b' + 2c'm_r + \sqrt{D}}{2c'} = \delta - \gamma \frac{b + \sqrt{D}}{2a}, \end{aligned}$$

where $M = M'TS^{m_r} = \begin{pmatrix} \beta' & -\alpha' + \beta' m_r \\ \delta' & -\gamma' + \delta' m_r \end{pmatrix} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$.

We have shown that we calculate the factor $s = C' \cdot t \in K$ from the coefficients of M and f in (I5). These coefficients are bounded by $O(2^N)$ such that we can bound the running time of this algorithm by $O(\mu(N) \log N)$. □

Now we would like to discuss two different cases for the application of this algorithm, one for an imaginary quadratic order, and one for a real quadratic order.

4.3. **Imaginary quadratic orders.** Let the notations be as in Algorithm 4.2 IDEALSUM$_{\mathcal{O}_D}$. In the case of an imaginary quadratic order $\mathcal{O}_D$, i.e., $D < 0$, every quadratic form $g \in \mathcal{F}(D)$ is equivalent to exactly one reduced form $f \in \mathcal{R}(D)$. Furthermore, the equivalence classes of positive definite quadratic forms are corresponding one-to-one to the elements of the class group $Cl(D)$ such that we can use the reduced positive definite forms as canonical representatives for the elements of the class group. For that reason we have $f = \mathbf{1}_D$ if and only if the ideal $I + J$ is principal. In this case we know that $s \in \mathcal{O}_D \subset K$ is a generator for the principal ideal $I + J$. In the case of $h(D) = 1$, g is always equivalent to the form $\mathbf{1}_D$, i.e., $I + J = s\mathcal{O}_D$ is principal. In general, we have a representation of the integral ideal $I + J$ consisting of a (fractional) ideal $\phi_{FI}(f)$ as a canonical representative of an element of the class group and of a factor $s \in K$.

Example 4.4. Let $D = -20$ be the discriminant. The field of fractions of $\mathcal{O}_D$ is the number field $K := \mathbf{Q}(\sqrt{-5})$ with discriminant $D = -20$. We choose $(1, \omega)$ as integral basis with $\omega = \sqrt{-5}$ and minimal polynomial $x^2 + 5$. We would like to calculate the ideal sum $\mathfrak{a}$ of the principal ideals generated by 7 and $1+2\sqrt{-5}$. Both of these generators are divisors of 21 in $\mathcal{O}_D$. We have the following representation of $\mathfrak{a}$ as $\mathbf{Z}$-module:

$$\begin{pmatrix} 7 & 0 & 1 & -10 \\ 0 & 7 & 2 & 1 \end{pmatrix}, \quad \text{and in HNF} \quad \begin{pmatrix} 7 & 4 \\ 0 & 1 \end{pmatrix}.$$

In terms of quadratic forms we get $g = (7, -8, 3)$ for the ideal $\mathfrak{a}$ which is equivalent to the reduced positive definite quadratic form $f = (2, 2, 3)$:

$$M \cdot g = f, \quad M := \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}.$$

We get a representation of $\mathfrak{a}$ consisting of the canonical element of the class group $\phi_{FI}(f) = 2\mathbf{Z} + \frac{-2+\sqrt{-20}}{2}\mathbf{Z} = 2\mathbf{Z} + (-1 + \sqrt{-5})\mathbf{Z}$ and an algebraic number $s = 2 - 1 \cdot \frac{2+\sqrt{-20}}{4} = \frac{3}{2} - \frac{\sqrt{-5}}{2} \in K$ (which can be decoded from the coefficients of matrix M). Altogether, we have $\mathfrak{a} = \phi_{FI}(f) \cdot s$, and, in particular, $\mathfrak{a}$ is not a principal ideal.

4.4. **Real quadratic orders.** We have to deal with a more difficult situation in the case of a real quadratic order $\mathcal{O}_D$ with discriminant $D > 0$, because for every quadratic form there can exist more than one equivalent reduced form [5, Theorem 3.5]. The mapping ρ, restricted to reduced forms, is certainly a permutation of equivalent reduced forms (Figure 4).

Even in this case we get a representation of the ideal sum applying Algorithm 4.2 as a product of an element of the number field $K = \operatorname{Quot} \mathcal{O}_D$ and a representative of an element of the class group. But such a representative of the class group is not unique because we can apply ρ at least once to get another equivalent reduced form (every cycle consists of at least two elements). It is known that the number of reduced quadratic forms with discriminant $D > 0$ is bounded by $O(\sqrt{D} \ln D)$ [8, Section 5.6.1]. For that reason it is quite expensive to apply ρ again and again in order to determine whether $f \sim \mathbf{1}_D$. In addition to that the product of the K-factors (based on the ρ-application) is growing in a very fast manner such that this further calculation is more expensive than the calculation of the reduced quadratic

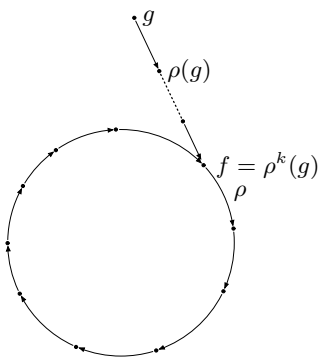

FIGURE 4. Reduction of quadratic forms with positive discriminant

form f. An explanation for this is that the repeated application of ρ, starting with the reduced form $f = (a, b, c)$ to the equivalent form (a, b, c) or $(-a, b, -c)$ in the same cycle as f, yields the fundamental unit[5] ε of the number field K (up to the sign and conjugation) as the product of the occuring K-factors. It is conjectured that the regulator $R(D) = \ln \varepsilon$ is "usually" of the order $\sqrt{D}$ using Brauer-Siegel's Theorem [34, 2, 3], and [21, Chapter XVI] if we assume that the class number of real quadratic number fields is usually very small (verified using heuristic observations by Cohen-Lenstra [8, Conjecture 5.10.2]). In general, the fundamental unit has coefficients with length $O(\sqrt{D})$ in a representation with a fixed integral basis which is not polynomial in the size of D. Therefore, the calculation of a generator of a principal ideal is in general quite expensive as the following example shows.

Example 4.5. Let $\mathcal{O}_D$ be the real quadratic order with discriminant $D = 102001$ and integral basis $(1, \omega)$, $\omega = \frac{1+\sqrt{D}}{2}$. Note that D is prime and $h(D) = 1$. We have $\mathbf{1}_D = (1, \lfloor\sqrt{D}\rfloor, *) = (1, 319, -60)$. Starting from the form $\mathbf{1}_D$ we reach the equivalent form $(-1, 319, 60)$, which can be identified with $\mathbf{1}_D$, after applying 649 cyclic ρ-reductions. The thereby calculated fundamental unit is

$$\begin{aligned}\varepsilon =&131562044923908472490340515184839643915855421101956234344958410577535366708110547076124433321885866721820587159944874253438016675776698975533800918061209612501634494021997475623199062688503046155657482339023207918903316604812940047905424960925615123327945093761601581445042290725659283849941921935301980656690663459231926824879\\ +&826457169687163437265796835212302463275895367640536256180037337646971398892147252262269455835714040839258874603860483179311600450492878335845906041629779965103299066775002473510320222583639990827404694971684963103602062362176265785732661202541989202570056716206101639556729609649823653974770698307419220754323558102029984402 \cdot \omega,\end{aligned}$$

where $\mathrm{Norm}(\varepsilon) = -1$ and $R(D) \approx 751.6133$. This example shows us that generators of prinicipal ideals can be very large because it can be necessary to calculate many ρ-reductions until one arrives at the form $\mathbf{1}_D$.

Now we would like to calculate the ideal sum $2288\mathcal{O}_D + (771 + \omega)\mathcal{O}_D$ which has as HNF obviously

$$\begin{pmatrix} 2288 & 771 \\ 0 & 1 \end{pmatrix}.$$

[5]The fundamental unit of a real quadratic number field (canonically embedded in $\mathbf{R}$) is an algebraic integer ε with norm equals to $+1$ or -1 and smallest absolute value $|\varepsilon| > 1$.

We convert this HNF into the corresponding quadratic form $g = (2288, -1543, 249)$, which is equivalent to the reduced form $f = (-100, 151, 198)$ with

$$M \cdot g = f, \quad M = \begin{pmatrix} -1 & 1 \\ -3 & 2 \end{pmatrix}.$$

Using M and Theorem 4.3 we decode the K-factor as $s' = -\frac{1}{4} - \frac{3}{100}\omega$. (Because there does not exist a unique reduced form f which is equivalent to g, we recommend using the HNF representation of the ideal sum for tests of equality or inclusion.) Calculations in the cycle of f with the ρ-mapping help us to find out whether the ideal sum is principal or not. In our example (class number 1, i.e., every ideal is principal) we arrive at the form $(-1, 319, 60)$ which can be identified with $\mathbf{1}_D$ after 324 ρ-steps:

$$\begin{aligned}\rho^{324}(-100, 151, 198) &= \rho^{323}(198, 245, -53) = \rho^{322}(-53, 285, 98) \\ &= \rho^{321}(98, 303, -26) = \cdots = \rho^{2}(-96, 281, 60) = \rho^{1}(60, 319, -1) = (-1, 319, 60).\end{aligned}$$

Thus we calculated a factor $s'' \in K$ which is approximately half as long as the fundamental unit because f is lying roughly halfway between $(1, 319, *)$ and $(-1, 319, *)$ in the cycle. Altogether we achieve as a factor $s = s' \cdot s''$,

$$\begin{aligned}s = &218759072317098327222092274640799438266218182292413685300557075742560267110820407 64 \\ &24558340501450114655820756504273990096894472398067208255817316271015052384898854 39 \\ -&13656397715170209193388544179957494494847032635857341364503584604229297423839852 386 \\ &66535810890521263222727897483912288469995367549982594219883013690467568135293131 \cdot \omega,\end{aligned}$$

and because of $f \sim \mathbf{1}_D$ (this follows from $h(D) = 1$ as well) we have $I + J = s\mathcal{O}_D$. One can calculate that $\mathrm{Norm}(2288\mathcal{O}_D) = 5234944$, $\mathrm{Norm}((771 + \omega)\mathcal{O}_D) = 569712$, and $\mathrm{Norm}(s\mathcal{O}_D) = -2288$. Even a multiplication of s with a power of the fundamental unit ε does not yield another generator for the principal ideal $I + J = s\mathcal{O}_D$ with a significantly smaller representation for the chosen integral basis because the form $f \in \mathcal{R}(D)$ has a large *distance* to the $\mathbf{1}_D$ in the cycle using ρ or ρ^{-1} (ρ is invertible for every cycle of equivalent reduced forms). It seems that the generator of a principal ideal could not be calculated in a different way in polynomial time of $\mathrm{size}(D) = O(\log |D|)$ because the coding of a generator can have a length of $O(\sqrt{D})$.

Remark 4.6. There exists at least one different kind of coding for the fundamental unit and for generators of principal ideals which has only a length of $O(\ln D)$ [4]. We do not discuss such a coding because it has no main focus on generators of principal ideals. Even for the representation of the ideal sum (despite the question of whether it is a principal ideal) we can use the HNF which can be calculated in an asymptotically fast manner.

Remark 4.7. The decision of whether an ideal is principal can be made asymptotically fast if we allow expensive precalculations for the real quadratic order. At first one calculates all the $O(\sqrt{D} \ln D)$ reduced quadratic forms, whose coefficients are bounded by $\sqrt{D}$, and identifies (a, b, c) with $(-a, b, -c)$. Each of these forms can be found in one of the $h(D)$ cycles which are numbered with $1, \ldots, h(D)$. Assume that cycle 1 contains the form $\mathbf{1}_D$. The coding of a form and of the containing cycle is possible with $O(\log \sqrt{D} + \log h(D)) = O(\log D)$ space. A coding for all the reduced forms requires $O(\sqrt{D} \log^2 D)$ space. Using suitable data structures (e.g., a balanced tree) it is possible to determine the number of the cycle which contains the given form in time $O(\log(\sqrt{D} \ln D) + \log D) = O(\log D)$. Therefore, after building

up such data structure, one can use Algorithm 4.2 and then determine whether the $\mathbf{1}_D$-cycle contains the reduced form f, i.e., whether $I + J$ is a principal ideal.

Note that the precalculation is at least as expensive as the calculation of the class number of an order for which no polynomial-time algorithm is known. The fastest known algorithm for the calculation of a class number is Buchmann's Algorithm [8, Section 5.9] and has a sub-exponential running-time.

5. Conclusion

We presented two different approaches and corresponding Algorithms 3.16 $\mathrm{SGCD}_{\mathcal{O}_D}$ and 4.2 $\mathrm{IDEALSUM}_{\mathcal{O}_D}$ for the calculation of an ideal sum in a quadratic order. In order to calculate the sum of two principal ideals in a fixed chosen imaginary quadratic ring of algebraic integers with fundamental discriminant $D < 0$, we can use both algorithms and achieve the same asymptotically running time $O(\mu(N) \log N)$.

If we consider D as input for the algorithms, then the running time of Algorithm 3.16 $\mathrm{SGCD}_{\mathcal{O}_D}$ contains additional factors which correspond to the ring-specific constants for the imaginary quadratic ring of algebraic integers. The running time for a single $\mathcal{S}_D$-euclidean step depends on the size of the euclidean set $\mathcal{S}_D$ and can be bounded by $O(\sqrt{|D|/3} \cdot \mu(N))$ for a fixed chosen integral basis. In addition to that, further factors, not monotone in $|D|$, influence the running time, e.g., the $\mathcal{S}_D$ euclidean minimum $E_{\mathcal{O}_D}$ depends on the distance of $\sqrt{|D|/3}$ to the next integer (cf. Table 1). It remains to discuss whether other $\mathcal{S}$-euclidean sets can be found for this algorithm which would lead to a smaller running time in D. Assuming the GRH one can conjecture that the size of the $\mathcal{S}$-euclidean set can be bounded by $6 \ln^2 |D|$, because the prime numbers $p \leq 6 \ln^2 |D|$ generates the class group [40, Section 4.4].

However, the running time of Algorithm 4.2 $\mathrm{IDEALSUM}_{\mathcal{O}_D}$ is not changed by a factor if we consider D as an input. Furthermore the discriminant D is coded in the chosen integral basis, and a suitable integral basis can be coded in $O(\log |D|)$ bits. For that reason this algorithm is uniformly fast for any discriminant and class number such that it is well suited for large discriminants.

Both different approaches are not suitable in an obvious manner for the computation of ideal sums in number fields of higher degrees because we used the correspondence between ideals and binary quadratic forms and their fast reduction. It is not possible to transfer the concept of the $\mathcal{S}$-euclidean domains to number fields of higher degrees because their rings of algebraic integers are not discrete subrings of $\mathbf{C}$ [33, Theorem 30]. Certainly the theory of $\mathcal{S}$-euclidean domains can be applied to number fields of higher degrees whose rings of algebraic integers are euclidean w. r. t. $|\cdot|$.

References

1. T. M. Apostol, *Modular Functions and Dirichlet Series in Number Theory*, second ed., Grad. Texts in Math., vol. 41, Springer-Verlag, Berlin, 1997. MR1027834 (90j:11001)
2. R. Brauer, *On the Zeta-Function of Algebraic Number Fields*, Amer. J. Math. **69** (1947), 243–250. MR0020597 (8:567h)
3. ———, *On the Zeta-Function of Algebraic Number Fields II*, Amer. J. Math. **72** (1950), 739–746. MR0039009 (12:482g)

4. J. Buchmann, C. Thiel, and H. Williams, *Short Representation of Quadratic Integers*, Computational Algebra and Number Theory (Sydney University, November 1992) (W. Bosma and A. van der Poorten, eds.), Math. Appl., vol. 325, Kluwer Academic Publishers, Dordrecht, Netherlands, 1995, pp. 159–185. MR1344929 (96c:11144)
5. D. A. Buell, *Binary Quadratic Forms, Classical Theory and Modern Computations*, Springer-Verlag, Berlin, 1989. MR1012948 (92b:11021)
6. B. F. Caviness, *A Lehmer-Type Greatest Common Divisor Algorithm for Gaussian Integers*, SIAM Rev. **15** (1973), no. 2, 414.
7. B. F. Caviness and G. E. Collins, *Algorithms for Gaussian Integer Arithmetic*, Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation SYMSAC'76 (Yorktown Heights) (R. D. Jenks, ed.), 1976, pp. 36–45.
8. H. Cohen, *A Course in Computational Algebraic Number Theory*, Grad. Texts in Math., vol. 138, Springer-Verlag, Berlin, 1996, Third, Corrected Printing. MR1228206 (94i:11105)
9. ———, *Hermite and Smith Normal Form Algorithms over Dedekind Domains*, Math. Comp. **65** (1996), no. 216, 1681–1699. MR1361805 (97e:11159)
10. ———, *Advanced Topics in Computational Number Theory*, Grad. Texts in Math., vol. 193, Springer-Verlag, Berlin, 2000. MR1728313 (2000k:11144)
11. G. E. Collins, *A Fast Euclidean Algorithm for Gaussian Integers*, J. Symbolic Comput. **33** (2002), 385–392. MR1890576 (2003a:11159)
12. C. F. Gauß, *Untersuchungen über höhere Arithmetik (Disquisitiones Arithmeticae)*, Chelsea Publishing Company, Bronx, New York, 1889, Neudruck 1965, Übersetzung ins Deutsche von H. Maser (ed.). MR0188045 (32:5488)
13. J. L. Hafner and K. S. McCurley, *Asymptotically Fast Triangulation of Matrices over Rings*, SIAM J. Comput. **20** (1991), no. 6, 1068–1083. MR1135749 (93d:15021)
14. T. L. Heath, *The Thirteen Books of Euclid's Elements*, second ed., vol. 2, Cambridge University Press, New York, 1956, Books III–IX. MR0075873 (17:814b)
15. K. Ireland and M. Rosen, *A Classical Introduction to Modern Number Theory*, second ed., Grad. Texts in Math., vol. 84, Springer-Verlag, Berlin, 1990. MR1070716 (92e:11001)
16. E. Kaltofen and H. Rolletschek, *Arithmetic in Quadratic Fields with Unique Factorization*, Proceedings of the EUROCAL'85 Conference on Computer Algebra (Linz, Austria, April 1–3, 1985) B. F. Caviness, ed., Lecture Notes in Comput. Sci., vol. 204, Springer-Verlag, Berlin, 1985, pp. 279–288. MR0826569 (87c:11099)
17. ———, *Computing Greatest Common Divisors and Factorizations in Quadratic Number Fields*, Math. Comp. **53** (1989), no. 188, 697–720. MR0982367 (90a:11154)
18. D. E. Knuth, *The Analysis of Algorithms*, Actes du Congrès International des Mathématiciens (1/10 septembre 1970, Nice, France) (Paris) (Comité d'Organisation du Congrès, ed.), vol. 3, Gauthier-Villars, 1971, pp. 269–274. MR0423865 (54:11839)
19. ———, *Seminumerical Algorithms*, third ed., The Art of Computer Programming, vol. 2, Addison-Wesley, Reading, MA, 1998. MR0633878 (83i:68003)
20. S. Lang, *Algebra*, third ed., Addison-Wesley, Reading, MA, 1993. MR1878556 (2003e:00003)
21. ———, *Algebraic Number Theory*, second ed., Grad. Texts in Math., vol. 110, Springer-Verlag, Berlin, 1994. MR1282723 (95f:11085)
22. D. H. Lehmer, *Euclid's Algorithm for Large Numbers*, Amer. Math. Monthly **45** (1938), 227–233.
23. F. Lemmermeyer, *The Euclidean Algorithm in Algebraic Number Fields*, Exposition. Math. **13** (1995), 385–416. MR1362867 (96i:11115)
24. H. W. Lenstra, Jr, *On the Computation of Regulators and Class Numbers of Quadratic Fields*, London Math. Soc. Lecture Note Ser. **56** (1982), 123–150. MR0697260 (86g:11080)
25. U. Manber, *Introduction to Algorithms. A Creative Approach*, Addison-Wesley, Reading, MA, 1989. MR1091251 (93a:68002)
26. J. Neukirch, *Algebraic Number Theory*, Grundlehren Math. Wiss., vol. 322, Springer-Verlag, Berlin, 1999. MR1697859 (2000m:11104)
27. D. J. Newman, *Analytic Number Theory*, Grad. Texts in Math., vol. 177, Springer-Verlag, Berlin, 1998. MR1488421 (98m:11001)
28. A. Schönhage, *Schnelle Berechnung von Kettenbruchentwicklungen*, Acta Inform. **1** (1971), 139–144.
29. ———, *IGCDOC, Computation of Integer GCD's*, Unpublished Manuscript, 1987.

30. ———, *Fast Reduction and Composition of Binary Quadratic Forms*, Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation ISSAC'91 (Bonn, Germany, July 15–17, 1991) S. M. Watt, ed., ACM Press, New York, 1991, pp. 128–133.
31. A. Schönhage, A. F. W. Grotefeld, and E. Vetter, *Fast Algorithms – A Multitape Turing Machine Implementation*, BI Wissenschaftsverlag, Mannheim, Germany, 1994. MR1290996 (96c:68043)
32. A. Schönhage and V. Strassen, *Schnelle Multiplikation großer Zahlen*, Computing **7** (1971), 281–292. MR0292344 (45:1431)
33. M. A. Shokrollahi and V. Stemann, *Approximation of Complex Numbers by Cyclotomic Integers*, Technical Report TR-96-033, International Computer Science Institute, Berkeley, September 1996.
34. C. L. Siegel, *Über die Classenzahl quadratischer Zahlkörper*, Acta Arith. **1** (1935), 83–86.
35. D. Stehle and P. Zimmermann, *A Binary Recursive GCD Algorithm*, Proceedings of the Sixth International Algorithmic Number Theory Symposium ANTS VI (Burlington, VT, June 13–18, 2004) D. Buell, ed., Lecture Notes in Comput. Sci., vol. 3076, Springer-Verlag, Berlin, 2004, pp. 411–425.
36. J. Stein, *Computational Problems Associated with Racah Algebra*, J. Comput. Phys. **1** (1967), 397–405.
37. J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, New York, 1999. MR1689167 (2000j:68205)
38. A. Weilert, $(1+i)$*-ary GCD Computation in* $\mathbf{Z}[i]$ *as an Analogue to the Binary GCD Algorithm*, J. Symbolic Comput. **30** (2000), 605–617. MR1797272 (2001k:11265)
39. ———, *Asymptotically Fast GCD Computation in* $\mathbf{Z}[i]$, Proceedings of the Fourth International Algorithmic Number Theory Symposium ANTS IV (Leiden, The Netherlands, July 2–7, 2000) W. Bosma, ed., Lecture Notes in Comput. Sci., vol. 1838, Springer-Verlag, Berlin, 2000, pp. 595–613. MR1850636 (2002k:11226)
40. ———, *Effiziente Algorithmen zur Berechnung von Idealsummen in quadratischen Ordnungen*, Dissertation, Mathematisch-Naturwissenschaftliche Fakultät der Rheinischen-Friedrich-Wilhelms Universität Bonn, Juli 2000.
41. ———, *Fast Computation of the Biquadratic Residue Symbol*, J. Number Theory **96** (2002), 133–151. MR1931197 (2003j:11006)

Department of Computer Science II, University of Bonn, Römerstrasse 164, 53117 Bonn, Germany

Current address: Liliencronstr. 8, 12167 Berlin, Germany

E-mail address: andre@weilert.de