Computational Problems of Quadratic Forms:

Complexity and Cryptographic Perspectives

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt beim Fachbereich Informatik und Mathematik
der Johann Wolfgang Goethe-Universität
in Frankfurt am Main

von

Rupert Hartung
aus Oberwesel

Frankfurt am Main, 2007

Contents

Introduction

I am the very model of a modern Major-General,
I've information vegetable, animal, and mineral,
[...]
I'm very well acquainted, too, with matters mathematical,
I understand equations, both the simple and quadratical
[...].

(W. S. Gilbert, from: *The Pirates of Penzance)*

At least at first sight, quadratic equations seem to be much more difficult than linear problems. This is what W. S. Gilbert may have had in mind when writing this verse for this self-assure officer; or so the juxtaposition between 'simple' and 'quadratical' equations suggests. Roughly speaking, the aim of this dissertation is to answer the question, *how* difficult the latter are.

Background. Quadratic forms play an important role in number theory as well as in several related areas. They already aroused the interest of Fermat, Euler, Lagrange, and Legendre (see [Wei84], [Dic34b, ch. VI–X, XII, XIII], [Dic34c, ch. I–XI]). Perhaps one of their main appeals is their seeming simplicity: Being merely a slight abstraction from quadratic equations, quadratic forms are easy to write down and ask questions about. More specifically, quadratic forms are just one step beyond linear ones, and the theory of linear forms (i. e. *linear algebra* by any other name) is thouroughly explored and easily understandible – at least from today's perspective. Curiously, this picture changes radically when turning to exponent two. Another reason to study quadratic forms lies in the fact that binary forms bear the structural information on quadratic number fields, but in an easier accessible way.

The mathematical literature produced by the mid of the 19th century has not only contributed singnificantly to the knowledge about quadratic forms, but also raised new questions. Especially Gauß' work *Disquisitiones Arithmeticae* [Gau89] enjoyed immense popularity among the mathematicians of his and the following generations: Not only did it mean a leap ahead in the theory, but it has also shaped much of the area today known as *algebraic number theory.*

Probably, the flourishing of this reasearch area inspired Hilbert to discuss it in his famous speech at the International Congress of Mathematicians in Paris in 1900. The eleventh item on his famous list of mathematical problems for the 20th century calls for a theory of quadratic forms over algebraic number fields.

The efforts initiated by Hilbert's speech have given rise to the *arithmetic theory* of quadratic forms, which explores quadratic forms over local and global fields and their respective rings of integers. Many question could be settled in this area. The honor of having accomplished Hilbert's task is usually granted to H. Hasse for the famous Hasse Principle (see [Has24]).

It is this branch of the theory which will prove the most significant here. Some central results are discussed in Sect. 1.2.5.

Despite these enormous advances, algorithmic questions have hardly ever come into focus. This is even more suprising in view of the fact that most of the theory starts with two classical decision problems:

(A) Given two quadratic forms, are they equivalent?
(B) Given a quadratic form and a scalar, can the scalar be represented by the form?

More than a century of research has provided us with comprehensive criteria for these questions. However, the algorithmic nature of such results is often barely a theoretical possibility. This point deserves a closer look. In the language of computer science, the arithmetic theory can only prove that questions (A), (B) are *decidable*, while making no statement about running times. Are these problems possibly *polynomial-time decidable*? In many important special cases, the answer is 'yes', still thanks to arithmetic results, see Sect. 3.1. However, these results do not extend to all forms. This gap is due to the complexity of the underlying *computational* problems:

(A') Given two equivalent forms, compute an equivalence transform.
(B') Given a form f and a scalar m, solve the equation $f(v) = m$ (if possible).

An efficent method which produces an equivalence transform or representation if it exists, obviously yields a procedure to decide their existence. But here it seems that the only algorithm considered in the literature to find a transformation is exhaustive search. This is most obvious when the attempt of Dickson and Ross to decide equivalence of a particular pair of forms is discussed (see [Cas78, p. 132], [CS93, p. 403]). The restriction to trivial algorithms also becomes explicit in Siegel's famous bound on equivalence transformations [Sie72], which reproves the decidability of (A) using analytic techniques. More precisely, he shows that for each pairs f, g of equivalent forms there is a constant $C > 0$ effectively computable from f, g such that there is a transformation from f to g whose coefficients are absolutely bounded by C. Siegel explicitly refers to the enumeration of all integral matrices up to some bound on the coefficients for testing equivalence.

In dimension three, Siegel's implicit bound has been made explicit and has been improved to polynomial size by Dietmann [Die03]. Still using trivial enumeration, this implies that problem (A) is in NP. Moreover, for problem (B), Grunewald and Segal [GS04] showed decidability by solving problem (B') if possible. Their algorithm is more sophisticated, yet still involves steps of exponential enumeration.

I am aware of exactly one literature reference asking explicitly for a complexity analysis of problems (A), (B), (A'), (B'). At the end of [CS93, ch. 15], Conway and Sloane formulate both decisional and computational equivalence and representation problems, along with a couple of additional questions, e. g. on class numbers. They mention several cases for which these problems are easy. For indefinite forms over $\mathbb{Z}$, they express their impression that "there do not seem to be good algorithms", and discuss the inefficiency of exhaustive enumeration.

Thus it may be stated that algorithms on quadratic forms have hardly been studied, and even less so complexity issues.

Out of fairness, we ought to mention the main exceptions to this rule. In [Gau89], Gauß solves all problems (A),(B),(A'),(B') for binary integral forms giving concrete non-trivial algorithms along with a correctness analysis (see also [Lag80], [BB97], [BV07] for improvements). His approach has been rediscovered by the founders of *computational algebraic number theory*, see [PZ89]. In particular, Gauß' algorithms and modifications thereof are employed for computations in quadratic and relative quadratic number fields, see [Coh93, ch. 5], [Coh00, sec. 2.6]. Still if forms are concerned these are mostly only binary ones.

Apart from the problems touched upon here, the old question how to solve the *Legendre equation*

$$ax^2 + by^2 + cz^2 = 0 \tag{1}$$

non-trivially, if possible, has fascinated generations of mathematicians. After Legendre had discovered the conditions under which (1) is solvable, Lagrange came up with a concrete solution method, see [Sma98, sec. 4.3.3], [Ser73, sec. 4.3]. A remarkable algorithm can also be found in [Gau89], recent improvents include [CM98], [CR03] [Sim05b], and [Sim05a].

Finally, for definite forms algorithmic and complexity theoretic investigations abound. Having a strong tradition in this particular field, computational aspects have gained considerable momentum on the advent of the LLL-algorithm [LLL82]. Algorithms for definite forms, often formulated in the language of lattices, constitute a vivid domain of research. This may be due to the requirements of the domains where lattices are applied, as discrete optimization, cryptanalysis, and lattice cryptography (see below).

Motivation. In this thesis, I will explore the *complexity* of problems (A), (B), (A'), (B'). This follows a twofold motivation:

At first, in the age of highly-efficient computing devices, decidability is a very weak notion. As computing capacities increased, the theory of computing has become more and more demanding of the efficiency a problem is solvable with: From decidability, requirements have shifted to polynomial-time decidability, and are even further shifting towards efficient parallelizability. Thus Hilbert's question adapted to the concerns of this day and age could read:

'Are equivalence and representability polynomial-time decidable over some ring?'

or more generally:

‘What is the complexity of deciding equivalence and computing transformations?’

We may arrive at similar questions if we apply similar reasoning to Hilbert’s Tenth Problem on solvability of general Diophantine equations.

This leads us to our second approach to these questions: The hardness of computational problems on indefinite quadratic forms allows to base cryptographic protocols on it, see Chapter 2. This follows the example of definite forms, or lattices, which have been employed in cryptography, e. g. in [AD97], [GGH97], [HPS98], [HPS01], [HHGP$^+$03]. The security of these crypto-schemes is based on the lattice problems SVP and CVP, whose hardness is illustrated by (partially randomized) NP-completeness results (see [MG02], [Kho05]). Moreover, this is taken as a hint that this type of primitives may still be secure and applicable in the (still hypothetical) age of quantum computers because quantum computers are considered unlikely to efficiently solve NP-complete problems (see [BBBV97]).

However, the hardness proofs use lattices of arbitrarily high dimension, which causes severe efficiency problems. In consequence, lattice cryptography plays a minor role in practice today. By contrast, for indefinite forms, we can prove hardness in fixed small dimension (Theorem 7.1.1), and we discover the NP-hardness of closely related problems (Chapter 9). In cryptography, this would allow for smaller key sizes, and thus also faster protocols. Adopting the vision for the future of lattice cryptography, we take this as an indication that quadratic form cryptography may be both suited for the post-quantum era as well as feasible for traditional computers.

It should be noted that there are two further families of cryptosystems related to quadratic forms or equations. At first, *multivariate cryptography* utilizes polynomial equation systems over finite fields, which are often quadratic. It was a scheme of Imai and Matsumoto [IM88] which became the igniter of this now fully developped and vivid branch of cryptography. Solving systems of quadratic equations over $\mathbb{F}_2$ is already NP-hard; this is understood as an indication that the concrete systems employed also are hard. However, NP-hardness only holds if the number of equations and variables is unbounded. This still requires relatively large keys, in contrast to our hardness results in small bounded dimension.

Furthermore, algorithmic problems of number fields have been employed in cryptography. Important protocols are proposed in [BW90], [BBT94], [BMM00]. This constitutes the branch of cryptography which is certainly closest to algorithmic algebraic number theory. As mentioned above, quadratic forms provide a data type highly suitable for computation in number fields. This refers mostly to binary forms, which are not useful in our context (see Sect. 7.2). Moreover, the underlying problems are often related to factoring, or discrete logarithms, while problems on higher-dimensional forms seem to be essentially harder.

Main results. The cryptosystems reviewed in Chapter 2 are proposed for indefinite anisotropic quadratic forms over $\mathbb{Z}$. This choice is the result of complexity investigations. The schemes themselves are quite flexible and could be implemented, after minor modifications, for various types of forms over various rings. The presentation in Chapter 2 emphasizes this flexibility. However, such variations usually have an impact on the security of the scheme.

Security relies on the hardness of problems (A') and (B'), which will be called **Trafo** and **Repr** (for formal definitions, see Sect. 1.3). These will be the main objects of study in this thesis.

The information that complexity theory can supply cryptography with is of two kinds: Efficient, or comparatively efficient algorithms rule out the instances in question, while hardness results encourage the use of the respective problem.

In the latter respect, we prove that variants of the problems **Trafo** and **Repr** over $\mathbb{Z}$ are NP-complete under randomized reductions (Chapter 9). More precisely, we ask for transformations and representations whose coefficients lie in given intervals. The hardness results refer to indefinite ternary quadratic forms (with several possible further restrictions). For isotropic forms, the results hold unconditionally, while for anisotropic forms it is subject to a number-theoretic assumption, which we call the *special Cohen-Lenstra Heuristic* (sCLH). This assumption claims class number one for 'sufficiently many' real quadratic fields with prime discriminant. It is inspired by and largely similar to the well-known Cohen-Lenstra Heuristic [CL84].

The proof of these theorems is based on a result of Adleman and Manders [MA78] who proved NP-completeness for solvability of the binary (inhomogeneous) equations

$$x^2 + by = a, \qquad |x| \leq c$$

in integers $x, y \in \mathbb{Z}$. We use a modification of this theorem with restrictions on a, b. The hardness of the representation proplem for isotropic forms follows directly. For anisotropic forms, we have to construct a small family of binary quadratic forms some of which represents the (unknown) integer y with high probability. The correctness of our construction is proved using the sCLH. Finally, the results on transformation problems are derived from those on representations. This step requires a bound on the number of orbits of representations under the automorphism group of the representing form.

A reductionist hardness result, this time for the original problem **Trafo**, is presented in Chapter 8. We show that computing transformations for equivalent indefinite forms over $\mathbb{Z}$ of any dimension $n \geq 3$ is no easier than extracting a square root of -1 modulo their determinant. The complexity of this task is closely related to that of factoring. We again emphasize that we can restrict to anistropic forms (if $n = 3$ or 4).

This estimate is useful because it gives an explicit lower complexity bound for the presumably hard problem **Trafo**. However, in the light of the NP-hardness results and for want of a subexponential algorithms for this problem, **Trafo** and factorization seem far from being polynomial-time equivalent. Therefore we include the factorization of the determinants into the input for most of our investigations. In particular, the NP-hardness results are still valid for the problems with the factorization given for free.

Perhaps our most suprising result reduces general **Trafo** instances over $\mathbb{Z}$ to such of small dimension. More precisely, the transformation problem (with factorization given) in any fixed dimension can be solved using an oracle for transformations in dimensions three and four. This holds for forms of odd squarefree determinant.

This result justifies the use of low-dimensional forms in cryptography. If the transformation problem is hard in any fixed dimension at all, then it is necessarily hard in dimensions ≤ 4.

The proof works by splitting off (a lattice on) a 'hyperbolic plane' from both forms in question, and reducing to the orthogonal complement.

Furthermore, we prove a result on the interrelationship between the transformation and representation problems. Here 'interrelationship' expresses a relaxed version of polynomial-time equivalence. More explicitly, we reduce **Repr** to **Trafo** instances—both times with free factorization—at the cost of restrictions on the determinants: The odd, squarefree determinant d of the form f of dimension n in the **Repr** instance is lifted to its $(n-1)$-th power under this reduction.

Conversely, **Trafo** instances are solved using an oracle capable of computing solutions to both **Repr** problems and **Trafo** problems of dimension $n-1$. Again the **Repr** instances refer to forms of determinant d^{n-1}. Most importantly, for $n=3$ the oracle access for lower-dimensional **Trafo** solutions can be dispensed with. Therefore we have some kind of mutual reductions of **Trafo** and **Repr** for ternary forms, though not exactly polynomial-time equivalence.

The importance of this result is due to the signature scheme of Sect. 2.3. Its security requires that, beside **Trafo**, also the problem **Repr** is infeasible (whereas identification as in Sect. 2.2 is based on **Trafo** only). Equivalence of **Trafo** and **Repr** would release us from the necessity to presuppose two unrelated cryptographic assumptions. Hence linking their complexity makes the conjunction of these two hardness assumptions more plausible.

The proof employs Minkowski duality between representations of scalars by f, and representations of $(n-1)$-dimensional forms by the adjoint of f. Passing from such a representation to a transformation for given instances means augmenting a $(n \times (n-1))$-matrix by a last column, subject to several linear and quadratic constraints. Very roughly, the reductions reflect the constructions of these missing coefficients.

Turning from lower to upper bounds on complexity, our first concern are binary forms over $\mathbb{Z}$. We learn that transformation problems can be solved in time polynomial in S, where S is any solution. This excludes the use of binary forms in protocols as in Chapter 2 because this would allow for key extraction in time polynomial in the size of the secret key. Together with the result of Sect. 7.1.3 this prompts us to concentrate on forms of dimensions three and four. Reference to solution size is necessary since in general, transformations between binary forms need not even be of polynomial size. The statement follows by analysis of an algorithm of Gauß.

An obvious variation of the computational problems consists in changing the base ring. We prove various upper complexity bounds for other rings than $\mathbb{Z}$. Over the rational number field $\mathbb{Q}$, the problem **Trafo** can be closely linked to factoring integers. For **Trafo** reduces to factoring the determinants of the forms involved, and conversely, **Trafo** is at least as hard as computing a modular square root of -1. As mentioned above, factoring and computing imaginary roots seem to be similarly hard and may even be polynomial-time equivalent. Ignoring this gap and using the result from Chapter 8, we may heuristically state that the transformation problem over $\mathbb{Z}$ is no easier than over $\mathbb{Q}$.

In Sects. 6.2 and 6.3, we explore rings of formal power series and polynomials in one variable, respectively. This setting is more general as we do not concentrate on a concrete base ring, but compare complexity of **Repr** over this ring with that over the ground field. For power series, it turns out that both decisional an computational problems are polynomial-time equivalent to the respective problem over the ground field.

In the case of polynomials, we reduce it to the problem of finding simultaneous representations over the ground field (which is much more general than single representations). Still, the solution over power series rings yields 'approximative' solutions to representation problems with polynomial coefficients. Hence hard instances may only arise if 'most' representations modulo powers of the indeterminate do not lift to polynomials.

We thus learn that the use of power series rings in our applications does not pay, as it features roughly the same level of security at the cost of larger keys. The use of polynomial rings at least is not encouraged. Besides, a more precise classification of complexity over polynomial rings seems to depend heavily on the ground field.

For finite fields, fields of p-adic numbers, and rings of p-adic integers, both **Trafo** and **Repr** are polynomial time. This follows almost immediately from classification theorems.

By localization, we can also solve the decisional equivalence and representability problems over $\mathbb{Z}$, for a large proportion of instances. For representations, indefiniteness of the forms is required. If the computational problems are hard, as we conjecture, this would establish an intriguing discrepancy phenomenon.

For definite forms in fixed dimension it is known that **Trafo** can be solved in polynomial time. Isotopic ternary forms allow for subexponential algorithms for both **Trafo** and **Repr**. These facts are collected in Chapter 5. There we also verify the decreasing effect of singularity and reducibility of forms on complexity.

Part of the results presented here have been published in [HS07b] and [Har07]. Another paper on this topic [HS07a] is in preparation.

Outline. In Chapter 1, we review the most important concepts from theoretical computer science that we are going to use. Then we introduce the basic notions of quadratic forms and cite important known facts about them. Finally we formulate and explain the computational problems **Trafo** and **Repr**, which we are going to analyze. This chapter contains prerequisites for the whole thesis. The other chapters are largely independent from each other.

In Chapter 2, we present an identification scheme by Schnorr which proves knowledge of an equivalence transformation. An enhanced scheme with long challenges is suitable for digital signature generations, even if at the cost of provable security. These applications serves as our main motivation to study the complexity of the underlying problem **Trafo**.

The impact of localization on complexity is studied in Chapter 3. We prove polynomial-time solvability of transformation and representation problems over all $\mathbb{F}_p$, $\mathbb{Z}_p$, and $\mathbb{Q}_p$. These insights are used to demonstrate polynomial-time decidability of equivalence and representability over $\mathbb{Z}$. There will be several references to these statements throughout the thesis, which explains why they precede its main parts.

Chapter 4 displays auxiliary algorithms. Section 4.1 is a survey on algorithmic prime selection. In Sect. 4.2.4 we show how to construct an integral form satisfying p-adic constraints. These methods will be needed in later chapters, particularly in Chapters 9 and 10.

In the second part of this thesis the reader may find results on several restrictions of **Trafo** and **Repr**. At first, we discuss properties of forms and their impact on complexity (Chapter 5).

Chapter 6 contains those results on base rings other than $\mathbb{Z}$ which are not yet discussed in Chapter 3, i. e. it is concerned with $\mathbb{Q}$, rings of formal power series, and of polynomials.

Finally, results with respect to the dimension can be found in Chapter 7; these are the reduction of the transformation problem to dimensions three and four, and the feasibility of it for binary forms.

The third and last part of this thesis comprises the remaining lower-bound results. It is opened by Chapter 8 which displays the imaginary root problem as a lower bound for **Trafo**. Then we present and prove NP-hardness results in Chapter 9, and finally in Chapter 10, we establish the 'near' equivalence of the problems **Trafo** and **Repr**.

Note that for the sake of convenience, the reader will find an overview of (non-standard) notation employed right after this introduction. Moreover, definitions and conventions explained in the text can be easily looked up by use of the index at the end of this document.

Literature. Of the extremely comprehensive literature on quadratic forms, we will primarily need the arithmetic theory. If possible, we have cited [Cas78], since this textbook is particularly focussed on forms over $\mathbb{Q}$ and $\mathbb{Z}$ and their localization. Other accounts of the arithmetic theory include [Bro06], [Jon50], [Eic52], [O'M63], [CS93, ch. 15], [Kne02], and [Kit93]. As far as local and global fields are concerned, one can as well refer to [Ser73] and [Lam05].

Acknowledgements. I would like to thank C.-P. Schnorr, my advisor, for all his support during my work on this thesis, and for calling my attention to the algorithmic aspects of quadratic forms in the first place.

I also wish to express my gratitude to all who have engaged in discussions with me on the subject. In particular, I have benefitted from interested questions and useful remarks on the part of A. Pfister, R. Scharlau, G. Schnitger, and D. Simon.

Thanks is also due to R. Dietmann, who introduced me to the the analytic theory of homogeneous forms. I also wish to thank J. Klüners and G. Malle for their comments on the sCLH (Sect. 9.3).

Table of Notation

$\lvert M\rvert$	cardinality of a set M
$\#M$	cardinality of a set M
$\uplus$	disjoint union
$\mathbb{Q}_p$	field of p-adic numbers
$\mathbb{Z}_p$	ring of p-adic integers (by convention, $\mathbb{Z}_\infty = \mathbb{Q}_\infty = \mathbb{R}$)
R^*	group of units of ring R
R^{*2}	group of squares of units of ring R
$\left(\frac{a}{p}\right)$	Legendre symbol modulo odd prime p
ρ_p	fixed element in $\mathbb{Z}_p^* \backslash \mathbb{Z}_p^*2$ *see Sect. 1.2.5*
$\omega(m)$	number of distinct prime factors of m
$a \mid b$	a divides b
$a \nmid b$	a does not divide b
$p \mid e\infty$	p is a prime dividing e, or $p = \infty$ *(see rem. after Theorem 1.2.8)*
$\nu_p(N)$	multiplicity of prime p in N, i. e. $\nu_p(N) = k \Leftrightarrow p^k \mid N$ and $p^{k+1} \nmid N$ for $N \in \mathbb{Z}$ or $\mathbb{Z}_p$, $\nu_p(N) = \nu_p(a) - \nu_p(b)$ for $N = \frac{a}{b} \in \mathbb{Q}$ or $\mathbb{Q}_p$. *By convention,* $\nu_p(0) = \infty$.
$\gcd(a_1, \ldots, a_n)$	greatest common divisor of $a_1, \ldots, a_n$ (in a UFD)
$\mathrm{GL}_n R$	group of regular n-ary square matrices over ring R, i. e. $\mathrm{GL}_n R = \{S \in R^{n\times n} \mid \det S \in R^*\}$
$\mathrm{SL}_n R$	special linear group over ring R, i. e. $\mathrm{SL}_n R = \{S \in R^{n\times n} \mid \det S = 1\}$
S_{ij}	(i,j)-entry of matrix S
S_{*j}	j-th column of matrix S (as a column vector)
S_{i*}	i-th row of matrix S (as a row vector)
S^t	transpose of matrix (or vector) S
$S^\#$	Lagrange adjoint of matrix S (with $S^\# S = SS^\# = (\det S)I$)
$\lvert S\rvert_\infty$	largest absolute value of the coefficients of matrix S
I_k	identity matrix of dimension k

e_i	i-th standard unit vector: $e_i = (0, \ldots, 0, \underbrace{1}_{i}, 0, \ldots, 0)^t$
$f(\cdot, \cdot)$	associated bilinear form of quadratic form f
$\lvert f \rvert_\infty$	largest absolute value of the coefficients of quadratic form f
$\langle a_1, \ldots, a_n \rangle$	diagonal quadratic form $\sum_i a_i x_i^2$ (Sect. 1.2)
$f \perp g$	orthogonal sum of quadratic forms f, g (Sect. 1.2)
$f\,T$	quadratic form f transformed by matrix T, *see Sect. 1.2.4*
$f \longrightarrow_R m$	form f represents m over R
$f \xrightarrow{*}_R m$	form f represents m primitively over R
$f \xrightarrow{*}_R m$	form f represents m primitively over $\mathbb{Z}$ (*see Sect. 1.2.2)*
$f \xrightarrow{*}_R g$	form f represents form g primitively over R
$f \xrightarrow{*} g$	form f represents form g primitively over $\mathbb{Z}$ (*see Sects. 10.1, 10.2)*
$\rho(f)$	reduction operator applied to form f *see Sects. 1.2.7, 2.1*
$f \sim g$	forms f and g are equivalent (over $\mathbb{Z}$)
$f \sim_R g$	forms f and g are equivalent over the ring R, *see Sect. 1.2.4*
$f \sim_g g$	f and g belong to the same genus, *see Sect. 1.2.6*
$\operatorname{sign} f$	signature of form f
$c_p(f)$	Hasse-Minkowski invariant of form f, *see Sect. 1.2.5*
$\mathcal{O}_R(f)$	group of automorphisms of form f, i. e. $\mathcal{O}(f) = \{T \in \mathrm{GL}_n R \mid f\,T = f\}$
$\mathcal{O}_R^+(f)$	group of *proper* automorphisms of f, i. e. $\mathcal{O}(f) = \{T \in \mathrm{SL}_n R \mid f\,T = f\}$ *R is suppressed in notation if $R = \mathbb{Z}$ should not cause any confusion with the asymptotic symbol $\mathcal{O}$*
`length` (X)	encoding length for object X *(see Sect. 1.1)*
$\mathrm{CRT}\big((a_1, m_1), \ldots, (a_k, m_k)\big)$	algorithmic call to the Chinese Remainder Theorem, *see p. 62*
$\preccurlyeq$	polynomial-time Turing reducible
$\preccurlyeq_{na}$	polynomial-time non-adaptively reducible
$\preccurlyeq_1$	polynomial-time Turing reducible with (at most) one oracle call
$\preccurlyeq_K$	Cook-Karp reducible *for these reducibility notions, see Sect. 1.1.4*

Part I

Quadratic Forms and their Applications

Chapter 1

Preliminaries on Quadratic Forms and Computational Problems

In this chapter we introduce definitions and theorems important for this whole thesis. We begin with collecting some concepts from theoretical computer science in Sect. 1.1. In Sect. 1.2, we review central aspects of the theory of quadratic forms. Finally, in Sect. 1.3, we discuss how to combine these topics, i. e. we define algorithmic problems of quadratic forms and begin with their analysis.

1.1 Computational Problems

1.1.1 Model of computation

We do not give many formal definitions in this section, but merely set up conventions. For more a detailed account, the interested reader is referred to [Pap94], [BDG88], [GJ79].

We generally use the computational model of a Turing machine. As we are only interested in complexity up to polynomial time, everthing done here carries over to any model of computation polynomial-time equivalent to Turing machines, e. g. polynomial-time k-string Turing machines.

For an algorithmic approach, the mathematical objects considered have to come in a machine-readable format, i. e. an encoding in strings over a fixed alphabet. In some cases such an encoding is essentially canonic; for instance, integers can be presented in binary, and integral quadratic forms may be given by its dimension and the array of its coefficients. Some more debatable cases are discussed in Sect. 1.3.1.

We will assume that some 'sensible' encoding has been chosen for each class of objects, and it will be kept fix. The *length* occupied by the encoding of an object ξ, i. e. the number of symbols (e. g. bits) used for it, will be denoted by

$$\texttt{length}\,(\xi)\,.$$

Analogously, we use probabilistic Turing machines as our model for randomized algorithms, see Sect. 1.1.3 for more details.[*]

We will write down algorithms as pseudocode programs, or merely sketch how to write down such a program in proofs, without explicitly referring to a Turing machine.

1.1.2 Problems

For the considerations made in this thesis, an intuitive notion of decisional and computational problems suffices. However, a few remarks will be useful.

It is important for us not to restrict to decisional problems only.

A computational problem consists of a set of inputs, and for each input, a set of admissible outputs (solutions).

Note that in contrast to the usual decision problems, the output need not be unique.

We can view decision problems as the special case of computational problems where all admissible answers are single bits. However, in this case we do require uniqueness of the answer.

The computational model in which we seek for solutions of a problem is formally either the Turing machine, or the probabilistic Turing maching. But for easier understanding we formulate algorithms either in pseudo-programming code, or we indicate in the proofs how an algorithm should be programmed.

As we are mostly interested in the complexity of computational problems up to (probabilistic) polynomial-time equivalence, we will often use the term "efficient" to mean 'in (possibly probabilistic) polynomial time'.

We introduce problems with parameters. Here it is important to note that thus define families of computational problems: For each value of the parameters, we obtain a new single problem to analyse.

Note that this definition also includes decision problems.

As for inputs and solutions, we will later restrict the set of potential parameters to (the encodings of) suitable mathematical objects.

1.1.1 ***Example.*** Let $\mathcal{M}$ range over polynomial-time decidable subsets of $\mathbb{N}$. Denote by $\mathbf{Fact}(\mathcal{M})$ the computational problem of factoring numbers from $\mathcal{M}$ into their prime divisors (see Sect. 5.2). Then **Fact** is a problem with parameters. Obviously, its complexity can differ widely for different $\mathcal{M}$: If $\mathcal{M}$ is the set

[*] Speaking in a nit-picking fashion, there are two more exceptions: Whenever we speak of reductions, we implicitly make use of oracle machines; and the cryptographic protocols of Chapter 2 formally require interactive Turing machines.

of powers of two, for instance, then $\mathbf{Fact}(\mathcal{M})$ can be solved efficiently. However, $\mathbf{Fact}(\mathbb{N})$ is not believed to be solvable in polynomial time.

Note that this definition of the factorization problem was for illustrational purposes only. A more general variant will be analyzed in Sections 9.2 and 6.1; see also Chapter 8.

Problem union. For computational problems $\mathbf{A}, \mathbf{B}$ define the problem union $\mathbf{A} \sqcup \mathbf{B}$ as follows: $\mathbf{A} \sqcup \mathbf{B}$ takes inputs in

$$(\{0\} \times I_{\mathbf{A}}) \cup (\{1\} \times I_{\mathbf{B}})$$

where $I_{\mathbf{A}}, I_{\mathbf{B}}$ is the set of inputs of $\mathbf{A}$, $\mathbf{B}$, respectively; and if the input was $(0, i)$, then the admissible outputs are the admissible outputs of problem $I_{\mathbf{A}}$ with respect to input i, and anagolously for $(1, i)$ and $I_{\mathbf{B}}$.

Heuristically, solving $\mathbf{A} \sqcup \mathbf{B}$ means being able to solve both $\mathbf{A}$ and $\mathbf{B}$.

1.1.3 Probabilistic Computation

As for general Turing machines, we will write down pseudo-code and identify such algorithms with probabilistic Turing machines. A probabilistic algorithm runs in polynomial-time if with probability $\geq \frac{2}{3}$, it outputs a correct solution in polynomial time; its behavior in other cases does not matter in the sense of this definition. This makes sense since we can always break the computation after polynomially many steps and output a nonsense string.

We will use the terms “random polynomial time” and “probabilistic polynomial time” synonymously. In slightly colloquial contexts (as the rough discussion of proof ideas) we will use the term “efficient” unspecifically for ‘random polynomial time’ and ‘deterministic polynomial time’.

1.1.4 Reductions

We write $\preccurlyeq$ for polynomial-time reductions of general computational problems, using polynomially many oracle calls. For decisional problems, this corresponds to a *Turing reduction* (see [Pap94, sec. 8.4]).

For decisional problems only, we denote by $\preccurlyeq_K$ a classical *Karp reduction.*

By $\preccurlyeq_{na}$ we denote non-adaptive reductions. It is the special case of a Turing reduction where first all questions have to be asked before the oracle gives its answers. For decisional problems, this type of reduction is often known as *truth table reduction.* We will often write down reductions with successive oracle queries for clarity, but mention non-adaptivety in statements if our reduction can be easily transformed into a non-adaptive one.

The symbol $\preccurlyeq_1$ denotes the special case of truth-table reduction where only one oracle call is permitted. For decision problems $\mathbf{A}, \mathbf{B}$, the reduction $\mathbf{A} \preccurlyeq_1 \mathbf{B}$ is equivalent to

$$\mathbf{A} \preccurlyeq \mathbf{B} \sqcup_K \bar{\mathbf{B}},$$

where $\bar{\mathbf{B}}$ denotes the complement of $\mathbf{B}$.

The symbol $\preccurlyeq_r$ will be used for random reductions, i. e. the executing oracle machine is probabilistic in the sense of Sect. 1.1.3. We combine it with the above notation with obvious meanings, i. e. $\preccurlyeq_{r,na}$, and $\preccurlyeq_{r,1}$.

Finally, the notation

$$\mathbf{A} \approx_* \mathbf{B}$$

abbreviates the reductions

$$\mathbf{A} \preccurlyeq_* \mathbf{B} \qquad \text{and} \qquad \mathbf{B} \preccurlyeq_* \mathbf{A},$$

for $*$ any legal combination of the discussed subscripts $_K$, $_{na}$, $_1$, $_r$, none.

1.2 Quadratic Forms

> *Every mathematician who is not indifferent to number theory has felt the charm of Fermat's theorem on the sum of two squares of natural numbers. A psychologist of the Jungian school would probably think that such diophantine problems are archetypal to a high degree.*
>
> *(Yu. I. Manin in [Man74])*

1.2.1 Quadratic Forms

Througout this thesis, let R be a commutative ring with unity in which 2 is not a zero divisor. A *quadratic form* f (often simply called *form*) over R is a homogeneous polynomial of degree two, i. e. a polynomial of the shape $f = \sum_{i,j=1}^{n} a_{ij}x_ix_j$ where $a_{ii} \in R$ and $a_{ij} = a_{ji} \in \frac{1}{2}R$. The number n of variables is called the *dimension* of f, denoted by $\dim f = n$, and f is called an n-ary form.

If $x = (x_1, \ldots, x_n)^t$ and $A = (a_{ij})_{ij}$, then we can also write $f = x^tAx$. Conversely, via this formula any symmetric $(n \times n)$-matrix A over $\frac{1}{2}R$ with diagonal entries in R gives rise to a unique quadratic form. In this situation, A is called the *associated matrix* of f.

If f is a form over $\mathbb{Z}$ (over $\mathbb{Z}_p$ for some prime p), then f is called *integral* (p-adically integral). If the associated matrix of f has coefficients in R rather than $\frac{1}{2}R$, then f is called *classically integral*; this distinction is of course only relevant if $2 \notin R^*$, thus, for $R = \mathbb{Z}$ and $R = \mathbb{Z}_2$[†]. If R is a unique factorization

[†] In the literature, the term 'integral' is sometimes used in the sense 'classically integral'.

domain (UFD), f is classically integral and $\gcd(a_{ii},\, 2a_{ij} \mid i \neq j) = 1$, then f is called *properly primitive.* It is called *improperly primitive* if it is not properly primitive, but if it is classically integral and $\gcd(a_{ii},\, a_{ij} \mid i \neq j) = 1$. Finally, f is called *primitive* if it is either properly or improperly primitive.

Most of the time it will be enough to consider properly primitive and improperly primitive forms. The reason is that for every classically integral form f, there is $\lambda \in R$ such that $\frac{1}{\lambda}f$ is still a form over R ('integral') and primitive. Moreover, if f is defined over R, but not classically integral, the $2f$ is (classically integral and) improperly primitive. Hence, up to multiplication with or division by a scalar each form falls into one of two families. The last distiction remaining, namely between properly and improperly primitive forms, cannot be easily removed; however, the phenomena observable within these families of forms do not differ too much.

We define $\det f := \det A$ as the *determinant* of the quadratic form f. If $\det f \neq 0$ then f is called *regular*, otherwise *singular.* From now on, we will tacitly assume that all occurring forms are regular unless otherwise stated.

To every quadratic form f, there is an associated bilinear form: If A is the associated matrix of f, then this bilinear form is given by $(x, y) \mapsto x^t Ay$. We will denote this by $f(x, y)$.

If $a_i \in R$, then the form $\sum_{j=1}^{n} a_i x_i^2$ is abbreviated as

$$\langle a_1, \dots, a_n \rangle,$$

and such a form is called *diagonal.* Moreover, if f, g are forms with associated matrices A, B, then we define the form $f \perp g$, the *orthogonal sum* of f and g, by taking

$$A \oplus B = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$$

as its associated matrix. Obviously, we have

$$\dim(f \perp g) = (\dim f) + (\dim g) \qquad \text{and} \qquad \det(f \perp g) = (\det f) \cdot (\det g).$$

For a symmetric matrix A consider its Laplace adjoint $A^{\#}$ consisting of signed maximal minors of A, which satisfies

$$AA^{\#} = A^{\#}A = (\det A)I.$$

Then $A^{\#}$ is obviously symmetric as well, and hence the associated matrix of a quadratic form. This form, if A was the associated matrix of the form f, will be called the *adjoint form* and denoted by $f^{\#}$.

1.2.2 Representations

Let $m \in \mathbb{Z}$. Then f is said to *represent* m if and only if there is $u \in R^n \backslash \{0\}$ such that $f(u) = m$. Write $f \longrightarrow_R m$, and call u a representation of m by f. In case R is a UFD and $\gcd(u_1, \ldots, u_n) = 1$, then this representation is called *primitive*. This fact is denoted by $f \overset{*}{\longrightarrow}_R m$. If $R = \mathbb{Z}$ we drop the subscript and write $f \overset{*}{\longrightarrow} R$.

In a general quadratic equation, one can get rid of linear terms, at the cost of a linear congruence condition on the solution of the homogeneous problem. This is expressed in the following proposition. It serves to illustrate the usefulness of studying representations.

For the case $n = 2$ over $\mathbb{Z}$, Proposition 1.2.1 is proven in [Gau89, art. 216]; a similar generalization (for the case $R = \mathbb{Z}$) can be found in [GS04]. For convenience, and according to Gauß, we use even linear terms without loss of generality.

Proposition 1.2.1 *Let R be a commutative ring. Let f be an n-ary quadratic form over R with associated matrix A, let $\det f =: d$ be not a zero divisor, and let $w \in R^n$, $h \in R$.*

Then the equation

$$f(x) + 2w^t x + h = 0 \tag{1.1}$$

is solvable for $x \in R^n$ if and only if the system

$$f(y) = -hd^2 + d\, f^{\#}(w) \qquad \textit{and} \qquad y \equiv A^{\#} w \bmod d \tag{1.2}$$

is solvable for $y \in R^n$.

Note that if R is a field, the linear congruences are trivially fulfilled.

Proof : First let x be a solution to 1.1. Then

$$y := dx + A^{\#} w$$

satisfies $y \equiv A^{\#} w \bmod d$ and

$$\begin{aligned} f(y) = y^t A y &= d^2 x^t A x + 2dx^t A A^{\#} w + w^t A^{\#} A A^{\#} w \\ &= d^2(x^t A x + 2x^t w) + dw^t A^{\#} w = -d^2 h + df^{\#}(w). \end{aligned}$$

Conversely, let y satisfy 1.2. First note that the second condition means that there is x with $y = dx + A^{\#} w$. Thus it holds that

$$-hd^2 + df^{\#}(w) = f(y) = f(dx + A^{\#} w) = d^2\big(f(x) + 2x^t w\big) + d\, f^{\#}(w),$$

which, as d is not a zero divisor, is equivalent to

$$f(x) + 2x^t w + h = 0,$$

which was to be shown. □

1.2.3 Properties of forms

Now we can define further properties of quadratic forms: A quadratic form is called *reducible* if it factors into two linear polynomials in $R[x]$. Reducible forms are studied in Sect. 5.2.

A form f is called *isotropic* if it represents zero, otherwise *anisotropic*. A vector $v \neq 0$ such that $f(v) = 0$ is then called an *isotropic vector*.

Now let there be a (canonical) embedding $R \hookrightarrow \mathbb{R}$. Typically, we think of the cases $R = \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ here. Then a form over R is called *indefinite* if it (its real image) represents both positive and negative values, and *definite* otherwise. Cleary every (regular) isotropic form is necessarily indefinite. Definite forms correspond to lattices in Euclidean space and are not considered here.

1.2.4 Transformations

Let f be a quadratic form of dimension n with associated matrix A. Let $S \in \mathrm{GL}_n R$, i. e. S is a $(n \times n)$-matrix over R with $|\det S| \in R^*$. Then $f\,S := f(Sx)$ is a quadratic form with associated matrix $S^t AS$. If there is an $S \in \mathrm{GL}_n R$ such that $g = fS$, then f, g are called *equivalent* over R, or R-equivalent, denoted by $f \sim_R g$. It is easy to see that this in fact constitutes an equivalence relation. If we talk about the R-class $\mathrm{cls}\,_R f$ of f, we always mean with respect to this relation. The equivalent forms f, g are called *properly equivalent* if the equivalence transformation S can be chosen with $\det S = 1$. The equivalence classes with respect to to proper R-equivalence are called *proper R-classes.*

In all these defintions and notations, we drop the mention of the ring R if we are working over the rational integers.

Note the associative law

$$f\,(S\,T) = (f\,S)\,T.$$

We fix for future reference the easy

Lemma 1.2.2 *Let f, g be equivalent quadratic forms over a ring R.*

(a) $\det g \in (\det f)\,R^{*2}$. *In particular, for $R = \mathbb{Z}$, the determinants of equivalent forms always coincide.*

(b) If R is an integral domain, f is regular, and $f\,T = g$ for $T \in R^{n \times n}$, then $T \in GL_n R$.

Proof : Let A (B) be the associated matrix of f (g, respectively).

(a) There is $S \in \mathrm{GL}_n R$ such that $f\,S = g$, hence $B = S^t A S$ and thus
$$\det g = \det B = (\det S)^2 \det A = (\det S)^2 \det f.$$

(b) From $f\,T = g$ we conclude that
$$(\det T)^2 (\det f) = \det g \in (\det f)\, R^{*2},$$
where the last equality is due to part a). As $\det f$ is not a zero divisor, it follows that $\det T \in R^*$ and thus $T \in \mathrm{GL}_n R$.

□

A key technique in classifying forms is the well-known

Lemma 1.2.3 (Completion of the square) *Let R be either a field with encoding of characteristic $\neq 2$, or $R = \mathbb{Z}_p$ for an odd prime p. Then every quadratic form over R is equivalent to a diagonal form.*

Moreover, if the dimension n of the forms is fixed, an equivalent diagonal form and a transformation can be computed in polynomial time.

Proof : See [Cas78, ch. 2, lm. 1.4 and ch. 8, thm. 3.1]. We briefly review the arguments to estimate the algorithmic complexity.

For R a field, start by finding a vector $v \in R^n$ with $f(v) \neq 0$. If none of the standard unit vectors satisfies this, choose a pair of standard unit vectors e_i, e_j ($i \neq j$) such that $f(e_i, e_j) \neq 0$, since then $v := e_i + e_j$ satisfies $f(v) = 2f(e_i, e_j) \neq 0$. Otherwise, the form f is identically zero, and the statement is trivial. Hence we have found a non-isotropic vector v, and if R is a field, then v can be extended to a basis of R^n. Applying the base change matrix to f, we may assume that $f(e_1) \neq 0$. Let $A = (a_{ij})_{ij}$ be the associated matrix of (this updated) f.

Now consider the matrix
$$T := \begin{pmatrix} 1 & -\frac{a_{12}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ & 1 & & \\ & & \ddots & 0 \\ 0 & & & 1 \end{pmatrix}.$$

Then $f\,T = \langle a_{11} \rangle \perp f_0$ for some $(n-1)$-ary form f_0. Now employ induction for the proof and perform a recursive self-call on f_0 for the algorithm.

To estimate the complexity of this procedure, first note that the number of arithmetic operations being constant as n is so. Note that finding a non-isotropic vector v (or detecting f as identically zero) requires only a complete scan and $\neq 0$-tests through the coefficients of f. To form a basis with v, we can also choose a subset of the $e_1, \ldots, e_n$.

Hence the most crucial part is controlling the growth of the coefficients. Let $A^{(k)}$ be the associated matrix of f after k iterations of the algorithm, i.e.

the first k rows and columns only have non-zero entries on the main diagonal. Moreover, for $k = 1, \ldots, n$ let

$$d_k := \det \begin{pmatrix} a_{11} & \ldots & a_{1k} \\ \vdots & & \vdots \\ a_{1k} & \ldots & a_{kk} \end{pmatrix} .$$

Note that this refers to the original (i. e. input) form f. By a straightforward induction one can see that the entries of $d_k A^{(k)}$ are determinants of $(k+1) \times (k+1)$ submatrices of $A^{(0)}$. This implies that the coefficients of all intermediate forms in the algorithm are quotients of sums of at most $n!$ products of at most n of the input coefficients. As R is a field with encoding then the coefficients are of polynomial size in the input length.

Consider the partial transformation T_k formed in one round of the algorithm. Its coefficients are, except zeros and ones, quotients of entries of $A^{(k)}$. Therefore, these are of polynomial size as well, and hence so is their product T, the output transformation.

If $R = \mathbb{Z}_p$, the algorithm follows roughly the same outline; however, instead of a non-isotropic vector, we need v such that $\nu_p(f(v))$ is minimal in $\mathbb{Z}_p^n$. But this can be accomplished by a coefficient scan as well, this time with keeping score of the current minimal $\nu_p(a_{ij})$ and the indices i, j. □

The size estimates on the coefficients in this proof are highly exponential in the dimension n. For instance, if we diagonalize a classically integral form over $R = \mathbb{Q}$, the resuling diagonal form will have entries whose enumerator and denominator are bounded by $(n!)\|f\|^n$ (Here $\|f\|$ stands for the absolute value of the absolutely largest coefficient of f). These bounds might not be sharp. However, the growth effect may incur severe efficiency problems in practice, see [Sim05b].

The same idea can be for slightly restricted problem over UFDs. Essentially the next lemma expresses that completion of the square is possible in the quotient field.

Lemma 1.2.4 *Let f a quadratic form of dimension $n \geq 2$ over the UFD with encoding R. If*

$$f \xrightarrow{*}_R t \neq 0,$$

then there are $b_2, \ldots, b_n \in R$ and a form f^* of dimension $n-1$ over R such that

$$tf = (tx_1 + b_2x_2 + \ldots + b_nx_n)^2 + f^*(x_2, \ldots, x_n);$$

in particular, then tf and $\langle 1 \rangle \perp f^*$ are equivalent over the quotient field of R. Moreover, $\det f^* = t^{n-2}d$.

Moreover, f^* and the b_i can be computed in polynomial time.

Proof : As in Lemma 1.2.3 we may assume without loss that $f(e_1) = t$. Then perform the first step of the proof of Lemma 1.2.3 to annihilate the a_{1i},

$i > 1$ in the associated matrix of f, over the quotient field K of R. If (a_{ij}) denotes the associated matrix of f, this yields

$$(tf)\begin{pmatrix} \frac{1}{t} & -\frac{a_{12}}{t} & \cdots & -\frac{a_{1n}}{t} \\ & 1 & & 0 \\ & & \ddots & \\ 0 & & & 1 \end{pmatrix} = \langle 1\rangle \perp f^*,$$

where the associated matrix (b_{ij}) of f^* satisfies

$$b_{i-1,j-1} = ta_{ij} - a_{1i}a_{1j}$$

for $i, j = 2, \ldots, n$, hence f^* is integral.

Multiplying with the inverse of the tranformation, we obtain

$$\begin{aligned} tf &= \big(\langle 1\rangle \perp f^*\big)\begin{pmatrix} t & a_{12} & \ldots & a_{1n} \\ & 1 & & 0 \\ & & \ldots & \\ 0 & & & 1 \end{pmatrix} \\ &= (tx_1 + a_{12}x_2 + \ldots a_{1n}x_n)^2 \ + \ f^*(x_2, \ldots, x_n), \end{aligned}$$

which had to be shown. Obviously, f^* can be computed in polynomial time, and the b_i are just coefficients of f. □

Finally, we mention a simple yet important special case. Consider *unary* quadratic forms over a field K, i. e. forms of dimension one, which are essentially the same as ring elements. Such a form $\langle a\rangle$, with $a \in K$, is regular if and only if $a \neq 0$. Let us determine the set of equivalence classes of such forms. The group of transformations $\mathrm{GL}_1 K$ coincides with the group of units K^* of K. The set of equivalence classes of unary forms is then described by the the factor group K^*/K^{*2}, the group of *square classes.* It has exponent 2. Square classes are useful in the context of general quadratic forms as well: If $aK^{*2} = bK^{*2}$, i. e. if a, b belong to the same square class, then by the above argument $\langle a\rangle \perp f \sim_K \langle b\rangle \perp f$ for any quadratic form f. Moreover, a is then (primitively) represented by a form f if and only if b is (primitively) represented by f.

1.2.5 Class structure over Important Rings

The following result is fairly useful for the classification of quadratic forms. For proofs, see [Cas78, thm. 2.4.1 and lm. 8.3.3].

Lemma 1.2.5 (Witt's Lemma)
Let R be either a field of characteristic $\neq 2$, or $R = \mathbb{Z}_p$ for an odd prime p. Let f be a regular quadratic form and h_1, h_2 arbitrary quadratic forms. If

$$f \perp h_1 \sim_R f \perp h_2,$$

then

$$h_1 \sim_R h_2.$$

This does not hold for $R = \mathbb{Z}$ or $R = \mathbb{Z}_2$.

In the following subsections, we apply the general discussion of quadratic forms to the specific rings which are of interest to us. These are primarily the local fields $\mathbb{Q}_p$ for p prime and $\mathbb{R}$, the field of rational numbers $\mathbb{Q}$, the rings of p-adic integers $\mathbb{Z}_p$, and the ring of rational integers $\mathbb{Z}$. For each ring, we give conditions for the equivalence of quadratic forms. For proofs and more details, see the following parts of [Cas78]: Chapter 2 for finite fields, Chapt. 4 for the $\mathbb{Q}_p$, Chapt. 6 for $\mathbb{Q}$, Chapt. 8 for the $\mathbb{Z}_p$, and Chapts. 9 and 11 for $\mathbb{Z}$.

Recall that $\mathbb{Z}_p$ and $\mathbb{Q}_p$ are frequently called *local* rings resp. fields, as opposed to the *global* rings and fields, in our case only $\mathbb{Z}$ and $\mathbb{Q}$, respectively. The field of reals $\mathbb{R}$ is also considered local, and by convention $\mathbb{Z}_\infty = \mathbb{Q}_\infty = \mathbb{R}$. We will therefore frequently refer to $\mathbb{Z}_p$ or $\mathbb{Q}_p$ where p is called a *symbol*, which means p is a prime or $p = \infty$.

Fields of p-adic numbers

Let p be an odd prime. Then the group of square residue classes of $\mathbb{Q}_p^{*2}/\mathbb{Q}_p^*$ is of order 4; it is generated by p and any non-square $\rho_p \in \mathbb{Q}_p^* \backslash \mathbb{Q}_p^{*2}$ satisfying $\nu_p(\rho_p) = 1$. This element can be chosen in $\mathbb{Z} \subseteq \mathbb{Q}_p$ coprime to p and satisfying

$$\left(\frac{\rho_p}{p}\right) = -1.$$

For $p = 2$, however, the group $\mathbb{Q}_2^{*2}/\mathbb{Q}_2^*$ is of order 8 with $\{2, 3, 5\}$ as a generating set. More precisely, we have the following characterization of square classes:

Lemma 1.2.6 *Let p be prime and let $a, b \in \mathbb{Z}_p^*$. If $p = 2$ and $a \equiv b \bmod 8$, or if p is odd and $a \equiv b \bmod p$, then $a\mathbb{Z}_p^{*2} = b\mathbb{Z}_p^{*2}$ and $a\mathbb{Q}_p^{*2} = b\mathbb{Q}_p^{*2}$.*

In particular, a quadratic form f represents a over $\mathbb{Q}_p$ if and only if it represents b over $\mathbb{Q}_p$, and f represents a primitively over $\mathbb{Z}_p$ if and only if it represents b primitively over $\mathbb{Z}_p$.

Lemma 1.2.6 follows easily from a general version of Hensel's Lemma, see [Eis95, thm. 7.3].

Lemma 1.2.7 *Let f be a form over $\mathbb{Q}_p$ with $p \nmid \det f$ and $\dim f \geq 2$. Then f is isotropic.*

Moreover, if f has coefficients in $\mathbb{Z}_p$, then f represents every $m \in \mathbb{Z}_p$ coprime to p primitively.

Next we completely classify quadratic forms over $\mathbb{Q}_p$. To this end we introduce two algebraic symbols: The *Hilbert norm residue symbol* of a pair of p-adic numbers, and the *Hasse-Minkowski invariant* of p-adic quadratic forms.

The *Hilbert norm residue symbol* is a mapping

$$\left(\frac{\cdot,\cdot}{p}\right) : \ \mathbb{Q}_p^* \times \mathbb{Q}_p^* \longrightarrow \{1, \ -1\}. \tag{1.3}$$

It is defined by

$$\left(\frac{a,b}{p}\right) = \begin{cases} 1 & \text{if } ax_1^2 + bx_2^2 - x_3^2 \text{ is isotropic over } \mathbb{Q}_p, \\ -1 & \text{otherwise} \end{cases}$$

for $a, b \in \mathbb{Q}_p^*$.

It can be shown (see [Cas78, sec. 3.2]) that $\left(\frac{\cdot,\cdot}{p}\right)$ has the following properties:

(i) It is symmetric, i. e.

$$\left(\frac{a,b}{p}\right) = \left(\frac{b,a}{p}\right)$$

for all $a, b \in \mathbb{Q}^*$.

(ii) It is bilinear with respect to multiplication, i. e.

$$\left(\frac{ab,c}{p}\right) = \left(\frac{a,c}{p}\right)\left(\frac{b,c}{p}\right)$$

for all $a, b, c \in \mathbb{Q}^*$.

(iii) It is trivial on squares, i. e.

$$\left(\frac{a,b^2}{p}\right) = 1$$

for all $a, b \in \mathbb{Q}_p^*$.

(iv) Let p be an odd prime and $\rho_p \in \mathbb{Q}_p^* \backslash \mathbb{Q}_p^{*2}$. Then $\left(\frac{\cdot,\cdot}{p}\right)$ takes the following values:

$$\left(\frac{\rho_p,\rho_p}{p}\right) = 1, \qquad \left(\frac{p,\rho_p}{p}\right) = -1, \qquad \text{and} \qquad \left(\frac{p,p}{p}\right) = \left(\frac{-1}{p}\right).$$

(Note that $\left(\frac{-1}{p}\right)$ is the Legendre symbol of -1 modulo p.)

(v) Consider $p = 2$. Then $\left(\frac{\cdot,\cdot}{p}\right)$ takes the following values:

$$\left(\frac{5,5}{2}\right) = \left(\frac{5,7}{2}\right) = \left(\frac{2,7}{2}\right) = \left(\frac{2,2}{2}\right) = 1 \quad \text{and}$$

$$\left(\frac{7,7}{2}\right) = \left(\frac{2,5}{2}\right) = -1.$$

(vi) For $p = \infty$, the norm residue symbol evaluates to

$$\left(\frac{a,b}{\infty}\right) = \begin{cases} -1 & \text{if } a, b < 0, \\ 1 & \text{otherwise} \end{cases}$$

for all $a, b \in \mathbb{R}^*$.

Note that these properties allow for efficient algorithmic evaluation of the symbol.

Now let f be a quadratic form over $\mathbb{Q}_p$. Then there are $a_1, \dots, a_n \in \mathbb{Q}_p^*$ such that

$$f \sim_{\mathbb{Q}_p} \langle a_1, \dots, a_n \rangle$$

by Lemma 1.2.3. We define the *Hasse-Minkowski invariant* of f by and

$$c_p(f) := \prod_{i<j} \left(\frac{a_i, a_j}{p} \right).$$

By convention, empty products equal 1, and hence $c_p(f) = 1$ for unary form.

It can be shown that this is well-defined (i. e. independent from the diagonalization), and that it does not change when f is replaced by an $\mathbb{Q}_p$-equivalent form (see [Cas78, sec. 4.1]).

We can now state the classification theorem of forms over $\mathbb{Q}_p$ (see [Ser73, thm. 7 of ch. 2]):

Theorem 1.2.8 *Let p be a prime. Let f, g be quadratic forms over $\mathbb{Q}_p$ of the same dimension n and determinant d.*

(a)

$$f \sim_{\mathbb{Q}_p} g$$

if and only if $c_p(f) = c_p(g)$. In this case, f and g are also properly $\mathbb{Q}_p$-equivalent.

(b) For each $n > 1$ and d, there are exactly two $\mathbb{Q}_p$-classes of forms.

(c) If f is anisotropic, then $c_p(f) = -1$ and $n \leq 4$.

As an abbreviation, we will often write $p|e\infty$ to mean "p is a prime dividing e, or $p = \infty$", and similarly we will write $p \not| e\infty$ to mean "p is a prime not diving e".

Note for Theorem 1.2.8 that if f, g are p-adically integral and $p \not| 2d\infty$, then trivially $c_p(f) = c_p(g) = 1$.

The field of real numbers

By convention, $\mathbb{Q}_\infty = \mathbb{Z}_\infty = \mathbb{R}$.

The squares in $\mathbb{R}^*$ are exactly the positive numbers, and hence the square class group of $\mathbb{R}$ is of order 2, generated by the number -1.

We have the following clasification of forms, which is also called 'Sylvester's law of inertia'.

Proposition 1.2.9 *Let f be an n-ary form over $\mathbb{R}$. Then there is a uniquely determined integer $0 \leq s \leq n$ such that*

$$f \sim_{\mathbb{R}} \langle \underbrace{-1, \ldots, -1}_{s}, 1, \ldots, 1 \rangle. \tag{1.4}$$

s is called the signature *of f and denoted by $s = \operatorname{sign} f$.*

Two n-ary forms f, g are $\mathbb{R}$-equivalent if and only if their signatures coincide.

A transformation for the equivalence (1.4) can be computed efficiently (to some desired precision).

Note that there are different definitions of the signature in the literature.

The rational number field

The square class group of $\mathbb{Q}$ is infinite. It is generated by the (positive) prime numbers and the number -1.

Perhaps the most prominent theorem in the theory of quadratic forms is the Hasse principle. In rough words, it states that the $\mathbb{Q}$-class of a rational quadratic form is uniquely determined by its properties over the collection of all local fields $\mathbb{Q}_p$, where p ranges over all primes and the symbol ∞ (or for short: "for all symbols p").

(These symbols represent the equivalence classes of non-trivial absolute values of the field $\mathbb{Q}$. They are also called "places" or "spots" of that field, employing the 'localization' metaphor. This concept is required when dealing with algebraic number fields in general. As we stay with the rationals, it seems appropriate to treat these places simply as symbols.)

Theorem 1.2.10 (Minkowski) *Let $d \in \mathbb{Q}\backslash\{0\}$ and $n \geq 2$. Let f_p be n-ary forms over $\mathbb{Q}_p$ for all symbols p such that*

$$\det f_p \in d\mathbb{Q}_p^{*2}.$$

Then there exists a rational quadratic form f satisfying

$$f \sim_{\mathbb{Q}_p} f_p$$

for all symbols p if and only if $c_p(f_p) = -1$ for only finitely many p, and

$$\prod_{\textit{all symbols } p} c_p(f_p) = 1. \tag{1.5}$$

Recall that by definition, $c_p(f) = 1$ for all unary forms. The infinitude of symbols p is not an obstacle because for all but finitely many symbols p, the form f_p satisfy $c_p(f_p) = 1$, and hence

$$f_p \sim_{\mathbb{Q}_p} \langle 1, \ldots, 1, d \rangle$$

for all but finitely many symbols p. Therefore there is only a finite amount of information about $\mathbb{Q}_p$-classes contained in the family $(f_p \,|\, p)$. We therefore do not lose generality when working with finitely many local forms In algorithms.

Theorem 1.2.11 (Rational Hasse Principle) *Let f, g be rational quadratic forms. Then the following are equivalent:*

1. *$f \sim_{\mathbb{Q}} g$,*
2. *$f \sim_{\mathbb{Q}_p} g$ for all symbols p,*
3. *$f \sim_{\mathbb{Q}_p} g$ for all symbols $p | 2d\infty$ except possibly one.*

From Theorems 1.2.11 and 1.2.8, one immediately obtains Meyer's Theorem (see [Mey91]):

Theorem 1.2.12 (Meyer) *Let f be an integral quadratic form of dimension $n \geq 5$. Then f is isotropic.*

Rings of p-adic integers

Over the rings $\mathbb{Z}_p$ of p-adic integers, there are many more classes of forms with the same dimension and determinant than over $\mathbb{Q}_p$. More precisely, while over $\mathbb{Q}_p$, $p \notin \{2, \infty\}$, there are exactly two classes of forms with determinant d and dimension n for each $d \in \mathbb{Q}_p$, $n \geq 2$, the number of $\mathbb{Z}_p$-classes for fixed d, n is unbounded, depending on the multiplicity $\nu_p(d)$ of p in d. Fortunately, however, there exists an easy normal form classifying the classes of forms completely.

Theorem 1.2.13 *Let f be a form over $\mathbb{Z}_p$, where p is an odd prime. Fix $\rho \in \mathbb{Z}_p^* \backslash \mathbb{Z}_p^{*2}$ (for example, $\rho \in \mathbb{Z}$ with $\left(\frac{\rho}{p}\right) = -1$).*

Then f is properly $\mathbb{Z}_p$-equivalent to a form of the shape

$$f_1 \perp \ldots \perp f_k,$$

where

$$f_i = p^{e_i} \langle \underbrace{1, \ldots, 1, r_i}_{\ell_i} \rangle$$

for $i = 1, \ldots, k$, such that $0 < e_1 < \ldots \leq e_k$, $\ell_i > 0$, and $r_i \in \{1, \rho_p\}$.

This normal form is uniquely determined by f and ρ. It can be computed in polynomial time, given f, p, and ρ.

Morever, a normal form with respect to some *$\rho \in \mathbb{Z}_p^* \backslash \mathbb{Z}_p^{*2}$ can be computed in polynomial-time given only f as input.*

The last modification is important for keeping algorithms deterministic, since there is no unconditionally provable method known to produce a nonsquare ρ without using randomness, see the remark after the proof of Theorem 3.2.1.

As a consequence of this classification, if $\det f = \det g = d$ is coprime to p, then $f \sim_{\mathbb{Z}_p} g$.

One simple case is useful to remember because it helps finding the normal form.

Lemma 1.2.14 *Let p be an odd prime and $u_i \in \mathbb{Z}_p^*$. Then*

$$p^e\langle u_1, \ldots, u_n\rangle \sim_{\mathbb{Z}_p} p^e\langle 1, \ldots, 1, u_1 \ldots u_n\rangle.$$

The implied transformation can be computed in polynomial time.

Theorem 1.2.13 and Lemma 1.2.14 are classical results; they can be found e. g. in [Cas78, thm. 3.1, lm. 3.4 of ch. 8]. The algorithmic statement we added is immediate from the proofs in the sources cited.

The case $p = 2$ is a bit more involved.

Theorem 1.2.15 *There is a set $\mathcal{S}$ of forms over $\mathbb{Z}_2$ such that each f is properly $\mathbb{Z}_2$-equivalent to one and only one form $f_0 \in \mathcal{S}$. This form f_0 can be computed from f in polynomial time. The forms in $\mathcal{S}$ can be chosen rationally integral.*

In particular, if f is properly primitive, $\dim f = 2$ and $\det f$ is odd, then f is $\mathbb{Z}_2$-equivalent to exactly one of the forms

$$\{\langle 1, d\rangle, \langle 3, 3\rangle, \langle 3, 7\rangle \mid d = 1, 3, 5, 7\}.$$

Moreover, if f is properly primitive, $n := \dim f \geq 3$ and $\det f$ is odd, then f is $\mathbb{Z}_2$-equivalent to exactly one of the n-ary forms

$$\begin{aligned}\{\langle 1, \ldots, 1, 1, 1, d\rangle, \quad &\langle 1, \ldots, 1, 1, 3, 3\rangle, \langle 1, \ldots, 1, 1, 3, 7\rangle, \\ &\langle 1, \ldots, 1, 3, 3, 3\rangle, \langle 1, \ldots, 1, 3, 3, 7\rangle \quad \mid d = 1, 3, 5, 7\}\end{aligned}$$

For details see [Wat76], [Jon44]. Perhaps easier a criterion of equivalence is given by congruence conditions rather than normal forms.

Proposition 1.2.16

(a) Let p an odd prime. If

$$f \equiv g \bmod p^{\nu_p(d)+1},$$

then $f \sim_{\mathbb{Z}_p} g$.

(b) If

$$f \equiv g \bmod 2^{\nu_2(d)+3},$$

then $f \sim_{\mathbb{Z}_2} g$.

1.2.6 Classes and Genera

Each class of forms is the union of one or two proper classes $\operatorname{cls}^+ f$. Integral forms f and g are said to be in the same *genus* (denoted by $\operatorname{gen} f$) if they are equivalent over all rings of p-adic integers $\mathbb{Z}_p$; here p ranges over all rational primes and the symbol ∞ ($\mathbb{Z}_\infty = \mathbb{R}$ by convention). Clearly every genus is a union of classes.

If a statement is said to hold for all p, or all symbols p, then we include the case $p = \infty$. It is excluded if p is called prime. All $p|e\infty$ means p ranges over the prime divisors of e and the symbol ∞.

We say that f and g lie in the same *genus* if

$$f \sim_{\mathbb{Z}_p} g$$

for all symbols p. We denote this by

$$f \sim_g g.$$

It is easy to see that

$$f \sim g \Longrightarrow f \sim_g g \Longrightarrow \det f = \det g$$

Lemma 1.2.17 *Let f, g be integral forms with the same odd determinant d. Let*

$$f \sim_{\mathbb{Z}_p} g$$

for all $p|d\infty$. Then $f \sim_g g$.

Proof : By hypothesis, $f \sim_{\mathbb{Z}_p} g$ for all $p|d\infty$. As noted after Theorem 1.2.15 it follows that $\Rightarrow \sim_{\mathbb{Z}_p} g$ for all p except possibly $p = 2$. Hence $\Rightarrow \sim_{\mathbb{Q}_p} g$ for all p except possibly $p = 2$. Then by Theorem 1.2.11, also $f \sim_{\mathbb{Q}_2} g$. Since d is odd, Theorem 1.2.15 yields that also $f \sim_{\mathbb{Z}_2} g$. □

One of the most important features of genera is that they coincide with the classes in many interesting cases. The following criterion is taken from [Cas78, p. 202f.]. We will frequently employ it to prove equivalence of integral quadratic forms in cases a transformation cannot be written down explicitly.

We call an integer m *k-power free* if $e \in \mathbb{Z}$, $e^k|m$ implies $e = \pm 1$. A rational number $\frac{a}{b}$, $a, b \in \mathbb{Z}$ coprime, is k-power free if a is k-power free. The next result can be found in [Cas78, thm. 1.3 and 1.5 of ch. 9].

Theorem 1.2.18 (Eichler) *Let f, g be integral quadratic forms of dimension $n \geq 3$ and determinant d which is $\frac{n(n-1)}{2}$-power free and satisfies*

$$2^{n(n-3)/2+\lfloor (n+1)/2 \rfloor} \not| d$$

if f is classically integral. Then

$$f \sim g \ \Leftrightarrow\ f \sim_g g.$$

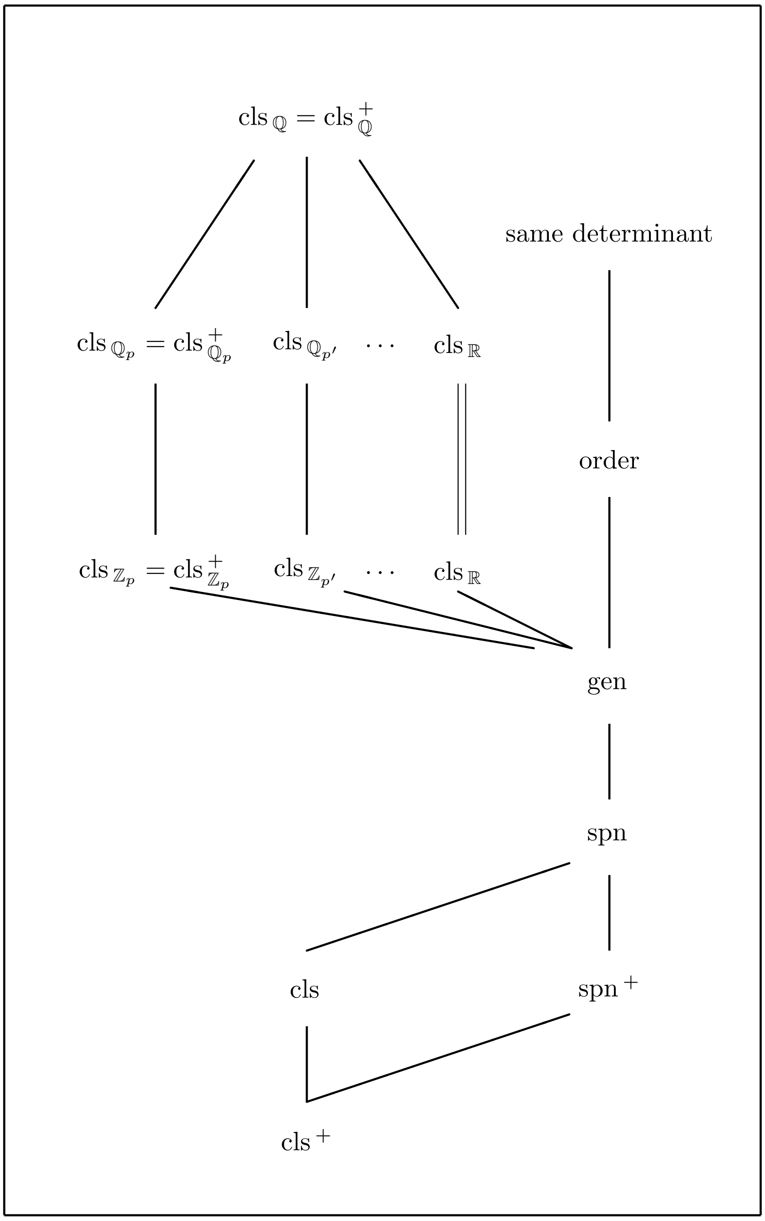
$\mathrm{cls}_{\mathbb{Q}} = \mathrm{cls}^{+}_{\mathbb{Q}}$
same determinant
$\mathrm{cls}_{\mathbb{Q}_p} = \mathrm{cls}^{+}_{\mathbb{Q}_p}$
$\mathrm{cls}_{\mathbb{Q}_{p'}}$
$\cdots$
$\mathrm{cls}_{\mathbb{R}}$
order
$\mathrm{cls}_{\mathbb{Z}_p} = \mathrm{cls}^{+}_{\mathbb{Z}_p}$
$\mathrm{cls}_{\mathbb{Z}_{p'}}$
$\cdots$
$\mathrm{cls}_{\mathbb{R}}$
gen
spn
cls
spn^{+}
cls^{+}

Figure 1.1: Class structure of quadratic forms over $\mathbb{Z}$

A line indicates that the upper set contains the lower. All items in the diagram refer to a fixed integral form f. We start from the proper class $\mathrm{cls}^{+} f$. Above it, we find the $\mathbb{Z}$-class (see Sect. 1.2.4). Spinor genus *and* proper spinor genus *do not occur explicitly in this thesis (see [O'M63, sec. 102] for definitions). The genus is discussed in Sect. 1.2.6.*

In the upper left of the picture we find equivalence classes over rings containing $\mathbb{Z}$, namely the rings of p-adic integers $\mathbb{Z}_p$ (for p a prime), the field of real numbers, the fields of p-adic number $\mathbb{Q}_p$, and the field of rational number $\mathbb{Q}$. In all these cases class and proper class coincide.

The notion order *in the right hand upper part was introduced by Minkowski in [Min11]. Two forms belong to the same order if the greatest common divisors of certain minors of their associated matrices coincide. The only orders we consider here are that of properly primitive and that of improperly primitive forms, see Proposition 1.3.2. Finally, forms of one order have the same determinant.*

1.2.7 Reduction

Following the case of definite forms, several concepts of *reduction* have been proposed. We try to capture the main aspects in the following definition. A *reduction theory* is a pair $(\Omega,\ \rho)$ with the following properties:

(i) Ω is a set of quadratic forms such that every equivalence class of quadratic forms contains at least one, but at most finitely many elements of Ω.

(ii) Membership in Ω decidable in deterministic polynomial time.

(iii) Moreover, ρ is a (deterministic or probabilistic) polynomial-time algorithm which takes as input a quadratic form f and outputs a pair $(f',\ T) = \rho(f)$, where $f' \in \Omega$, $T \in \mathrm{GL}_n R$, and $f\,T = f'$; in particular, f and f' are equivalent.

(iv) Restricted to Ω, ρ operates as the identity.

The elements of Ω are called *reduced* forms. The procedure ρ is called the *reduction algorithm* of the reduction theory.

For $R = \mathbb{Z}$, there are several concepts of reduction for indefinte quadratic forms. Most recently, variants of the LLL algorithm for definite forms (see [LLL82]) have been proposed by Simon [Sim05b], Ivanyos-Szánto [IAS96], and Schnorr [Sch07]. The main difficulty in generalizing LLL to indefinite forms is the fact that the (analogue to the) length of orthogonalized basis vectors of a lattice are simply values of the quadratic form in question at a rational non-zero vector. However, for indefinite forms this value may vanish, which thwarts the usual procedure of the algorithm, and in this event Simon's algorithm breaks, as he is primarily interested in solutions to $f(x) = 0$ rather than reduction, while Ivanyos and Szánto work with a randomized perturbation of the vector in question. However, these questions are irrelevant here because we are concerned mainly with anisotropic forms. So we may simply use the efficient, then coinciding LLL-algorithms of Simon-Ivanyos-Szánto-Schnorr as our concept of reduction. We denote the reduction operator by $\rho(\cdot)$; by $(g,\ T) = \rho(f)$ we mean that g is LLL-reduced, the algorithm returns form g on input f, and $f\,T = g$.

1.3 Problems of Quadratic Forms

We will now define the computational problems dealing with quadratic forms which will be of interest to us.

1.3.1 Rings

In the subsequent sections, we will define algorithmic problems on quadratic forms. This only makes sense if their coefficients can be specified in a (Turing-)machine readable format. In other words, we need an *encoding* of ring elements. We will use mainly the rings $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Z}_p$, and $\mathbb{Q}_p$, including $\mathbb{Z}_\infty = \mathbb{Q}_\infty = \mathbb{R}$. For $\mathbb{Z}$ and $\mathbb{Q}$, there are natural such encodings: Write the integers to some base b (e. g. $b = 2$)and represent rational numbers by pairs of integers $(a, b) \in \mathbb{Z}^2$ such that either $(a, b) = (1, 0)$, or a, b coprime; here a, b indicate enumerator and denominator of a fraction.

Using these representations, arithmetic operations can be evaluated efficiently, including the `div` and `mod` operations over $\mathbb{Z}$. Consequently we can efficiently perform many standard tasks over these rings: For instance, polynomials can be efficiently evaluated, and over $\mathbb{Q}$, systems of linear equations can be efficiently tested for solubility, and for solvable linear systems, the general solution can be efficiently constructed.

Furthermore, for $\mathbb{Z}$, the (extended) Euclidean Algorithm admits the computation of greatest common divisors, and their representation as linear combinations. This also enables us to solve the analogous linear algebra tasks mentioned for $\mathbb{Q}$, compute least common multiples, find common denominators of fractions, check systems of vectors for primitivity and form bases of $\mathbb{Z}^n$ out of them.

Moreover, we can as well perform all these arithmetic manipulations on the quotient rings $\mathbb{Z}/N\mathbb{Z}$.

Some of our results, especially in Sects. 5.1, 5.2, only rely on the efficiency of such basic operations. We will call a ring a *ring with encoding* if there is a representation of ring elements by (binary) strings such that

(i) addition, subtraction/negation, multiplication, and division—if defined—are polynomial-time computable; and

(ii) it is polynomial-time decidable whehter a given string represents a ring element, and whether it represents a unit.

As argued above, this also makes feasable many other basic tasks of arithmetic and linear algebra.

A *field (UFD, integral domain) with encoding* is a ring with encoding which is at the same time a field (UFD, integral domain). General fields with encoding, as opposed to the concrete fields $\mathbb{Q}$, $\mathbb{Q}_p$, $\mathbb{F}_p$ with a canoncial encoding, will be employed in Sects. 6.2 and 6.3. There we will study forms whose coefficients are power series and polynomials in one variable, respectively, over an arbitrary field with encoding of characteristic $\neq 2$.

Sometimes, this minimal arithmetic requirements do not suffice, and we need a closer similarity to $\mathbb{Z}$ above. In particular our results on singular and reducible forms in Sects. 5.1, 5.2 refer to rings with a generalization of a Euclidean Algorithm.

Recall that a ring R is called a *principal ideal domain (PID)* if every ideal of R is principal, i.e. generated by one element. Then a PID R is called a *computational principal ideal domain (cPID)* if

(i) it is a ring with encoding, and if

(ii) given elements $a, b \in R$, we can efficiently compute a generator g of the ideal $aR + bR = gR$, along with a $\lambda, \mu \in R$ such that

$$\lambda a + \mu b = g.$$

Note that item (ii) generalizes the Extended Euclidean Algorithm in $\mathbb{Z}$. More precisely, the existence of a polynomial-time computable Euclidean function is sufficient for (ii), but seems not to be necessary: For instance, the rings of integers in an algebraic number field of class number 1 are all cPIDs, the generalized gcd-algorithm being given by [Coh00, sec. 1.3]. In contrast, the vast majority of these rings are not known to admit a Euclidean function, let alone an efficiently computable one (see [Lem95]).

It is obvious that as for the integers, this algorithm allows effient solution of several related tasks, as the computation of least common multiples.

It remains to discuss the important local rings $\mathbb{Z}_p$, $\mathbb{Q}_p$, and $\mathbb{R}$. At first sight, they seem inaccessible to computational encoding for Turing machines. This is because a generic element of $\mathbb{Z}_p$ ($\mathbb{Q}_p$) involves an infinitude of coefficients from $\{0, \ldots, p-1\}$, and the exact representation of a real number requires infinitely many (binary) digits.

For $\mathbb{R}$ the solution to this dilemma is obvious: All compuations are required only up to a certain *precision.* For $\mathbb{Z}_p$, we mimick this approach: We say that a p-adic equation is satisfied to precision k if and only if it is satisfied modulo p^{k+1}.

Thus, talking about $\mathbb{Z}_p$ we actually do the arithmetic in some $\mathbb{Z}/p^{k+1}\mathbb{Z}$. For algorithmic purposes, we consider a problem over $\mathbb{Z}_p$ efficiently solved if we can solve it to arbitrary precision k in time polynomial in both the input length and k.

This convention naturally extends to $\mathbb{Q}_p$: An element $\sum_{i=-n}^{\infty} a_i p^i$ of $\mathbb{Q}_p$ is known to precision k if the coefficients

$$a_{-n}, a_{-k+1}, \ldots, a_0, \ldots, a_{k-1}, a_k$$

have been computed.

(A similar notion of precision will be required in the context of formal power series, see Sect. 6.2.)

1.3.2 Encoding of properties

At times, we will want to restrict some of these problems to suitable subsets of quadratic forms. To keep notation simple and short, we demand the specification of a set $\mathcal{P}$ of properties of forms as a parameter of the problem. As far as this work goes, we are primarily interested in restrictions on the dimension, the determinant, the class structure, the definiteness, the regularity, and the isotropy of the forms. Without agreeing on one concrete encoding of $\mathcal{P}$, we demand that the integers involved should be given in binary representation, and that the following bounds on the specification lengths hold:

property	encoding length
$\dim f = n$	
$\dim f \geq (>, \leq, <) n$	$\log n + \mathcal{O}(1)$
$M \not| \det f$	$\log M + \mathcal{O}(1)$
$(\det f, M) = 1$	$\log M + \mathcal{O}(1)$
$\det f$ squarefree	$\mathcal{O}(1)$
$\operatorname{gen} f = \operatorname{cls} f$	$\mathcal{O}(1)$
$\operatorname{gen} f = \operatorname{cls}^+ f$	$\mathcal{O}(1)$
f (in)definite	$\mathcal{O}(1)$
f regular	$\mathcal{O}(1)$
f (an)isotropic	$\mathcal{O}(1)$

The length of the description of the set $\mathcal{P}$ thus equals the sum of the description lengths of the single properties it contains according to the above tabular plus $\mathcal{O}(1)$.

Sets of properties will be defined when needed on an ad-hoc basis, usually directly before the statement where they appear.

Whether a given form f satisfies all properties from a given set $\mathcal{P}$ is efficiently decidable for most of the above properties; for squarefreeness of the determinant and for isotropy in dimensions 3 and 4, the factorization of $\det f$ has to be known as well to decide it, and for the class structure properties, it is not known how to decide it in general but there are strong sufficient criteria which will, in particular, always be applicable to the forms we consider here.

By convention, a problem with the parameter $\mathcal{P}$ omitted refers to the empty set of properties, i. e. to all quadratic forms (unless otherwise restricted).

1.3.3 Representation problem

The computational problem most naturally associated with a quadratic form certainly is the question to solve quadratic equations. By Proposition 1.2.1, we do not lose generality by resticting ourselves to equations without linear terms.

Repr$^R(\mathcal{P})$ Representation problem over R
PARAMETERS: Set $\mathcal{P}$ of properties of quadratic forms.
INPUT: a quadratic form f satisfying all properties from $\mathcal{P}$, and $m \in \mathbb{Z}$ such that $f \longrightarrow_{\mathbb{Z}} m$.
OUTPUT: a vector $u \in \mathbb{Z}^n$ such that $f(u) = m$ (where n is the dimension of f).

Now let R be a cPID. Then we also consider the following variant of the representation problem:

***Repr**$^R(\mathcal{P})$ **Primitive representation problem over cPID** R
PARAMETERS: Set $\mathcal{P}$ of properties of quadratic forms.
INPUT: a quadratic form f satisfying all properties from $\mathcal{P}$, and $m \in \mathbb{Z}$ such that $f \overset{*}{\longrightarrow} m$.
OUTPUT: a primitive vector $u \in \mathbb{Z}^n$ such that $f(u) = m$ (where n is the dimension of f).

This definition is motivated by the insight that for every representation $f(x) = m$, $x \neq 0$, there is an underlying primitive representation $x = \lambda x_0$, x_0 primitive, and

$$m = f(x) = f(x_0)\,\lambda^2.$$

As an algorithmic consequence, one can solve many general representation instances by using an oracle for coprime representations, even if a coprime representation for the original instance does not exist. This is made precise by the following proposition. Denote (for the next proposition and its proof) by $\Omega(m)$ the number of prime factors of the integers m (counted with multiplicities).

Proposition 1.3.1 *Let $c > 0$ be fixed. Consider* **Repr**-*instances (f, m) over $\mathbb{Z}$ where m is presented fully factored and satisfies*

$$\Omega(m) \leq c \log\log m.$$

Then there is a polynomial-time oracle algorithm which, given access to a ***Repr**-*oracle, computes a representation of m by f.*

Moreover, this reduction is non-adaptive, and all oracle inquiries refer to the form f.

Proof : Sort the factors of m as

$$m = \pm \left(\prod_{i=1}^{r} p_i \right)^2 \prod_{j=1}^{s} q_j \tag{1.6}$$

where the q_j, $j = 1, \ldots, s$ are pairwise distinct (positive) primes. For every $I \subseteq \{1, \ldots, r\}$, ask the ***Repr**^{R}-oracle for a representation of

$$m' := \left(\prod_{i \in I} p_i \right)^2 \prod_{j=1}^{s} q_j$$

by f. If the oracle finds v' with $f(v') = m'$, then

$$v := \left(\prod_{i \notin I} p_i \right) v'$$

satisfies $f(v) = m$.

The running time of this algorithm is dominated by the number of calls to the ***Repr**-oracle, which amounts to at most

$$2^r \leq 2^{\Omega(m)} \leq (\log m)^c,$$

with r as in (1.6). Thus the algorithm runs in polynomial time. □

Note that the set of integers m with $\omega(m) \leq c \log\log m$ has density 1 (for $c \geq 1$) (see [HW60, thm. 431]), so Proposition 1.3.1 covers a large subset of **Repr**-instances.

If $R = \mathbb{Z}$, then we simply drop the superscript R.

If we want the forms in the instances to have fixed dimension n, we often abbreviate this problem variant by $\mathbf{Repr}_n^R$ or, if additional properties are stipulated, by $\mathbf{Repr}_n^R(\mathcal{P})$. If we do not restrict the forms other than by dimension we simply drop the parameter $\mathcal{P}$, i. e. we write $\mathbf{Repr}_n^R$ in this case.

1.3.4 Transformation problem

We define the computational problem

$\mathbf{Trafo}^R(\mathcal{P})$ **Transformation problem over** R
PARAMETERS: Set $\mathcal{P}$ of properties of quadratic forms, ring R with encoding.
INPUT: quadratic forms f, g satifying all properties from $\mathcal{P}$ and $f \sim_R g$.
OUTPUT: $S \in \mathrm{GL}_n R$ such that $g = f\,S$ (where n is the dimension of f).

As above we frequently drop the superscript R if $R = \mathbb{Z}$, or use subscript n to constrain the dimension of forms.

If $\mathcal{P}(n, d)$ stands for the properties $\dim f = n$ and $\det f = d$ for a quadratic form f, then we abbreviate

$$\begin{array}{rcl} \mathbf{Trafo}(\mathcal{P}^R(n,d)) & \text{by} & \mathbf{Trafo}_n^R(d) \text{ and} \\ \mathbf{Repr}(\mathcal{P}^R(n,d)) & \text{by} & \mathbf{Repr}_n^R(d). \end{array}$$

1.3.5 Primitiveness

Note that we do not lose generality by restricting to primitive forms in the problems **Repr** and **Trafo** (and variants to be defined later), as the following proposition reduces these problems to the primitive case.

Proposition 1.3.2 *Let R be a cPID. Denote by $\mathcal{P}^*$ the property of being primitive for a quadratic form over R. Then*

$$\mathbf{Repr}_n^R \preccurlyeq_1 \mathbf{Repr}_n^R(\mathcal{P}^*) \qquad \textit{and} \qquad \mathbf{Trafo}_n^R \preccurlyeq_1 \mathbf{Trafo}_n^R(\mathcal{P}^*)$$

Proof :

(a) Given an instance ($f = \sum_{i \leq j} a_{ij} x_i x_j$, m) of **Repr**, we can efficiently compute $a := \gcd(a_{ij} \mid i \leq j)$ as R is a cPID. The form $f' := \frac{1}{a} f$ then is a well-defined and primitive quadratic form over R. Obviously, for a vector $u \in R^n$ it holds that $f(u) = m$ if and only if a divides m in R and $f'(u) = \frac{1}{a} m$.

(b) Let (f, g) be an instance of **Trafo**. Compute the greatest common divisors a, b of the respective coefficients as in part a). Then $f \sim_R g$ implies that a and b are associated, i. e. $a = b\varepsilon$ for some $\varepsilon \in R^*$. This unit ε can also be efficiently computed from a, b by the definition of a cPID. Now for $T \in \mathrm{GL}_n(R)$ the equation system $f\,T = g$ is equivalent to

$$\varepsilon f'\,T = g',$$

where $f' := \frac{1}{a} f$ and $g' := \frac{1}{b} g$ are well-defined and primitive over R.

□

Hence, there remain only two cases for our study: the properly primivitive and the improperly primitive forms. These two families do not differ much in the phenomena to be discovered, but the distinction sometimes makes some additional technical effort necessary.

1.3.6 Complexity assumption

We believe that the problems **Repr**($\mathcal{P}$) and **Trafo**($\mathcal{P}$) are hard, i. e. not solvable only in exponential time, if $\mathcal{P}$ consists of the properties indefiniteness, anisotropy, and $\dim f \in \{3, 4\}$ for a quadratic form f.

Therefore, we think that public-key cryptosystems can be build on this problem. Chapter 2 contains a suggestion on this behalf.

The hardness assumption is supported by the results of Chapters 7–10: We prove that the complexity of these problems is concentrated in dimenions three and four. We learn that **Trafo** cannot be much harder than factoring. We show the randomized NP-hardness of related problems, and then link the complexity of both problems to one another.

Another reason to conjecture the hardness of **Trafo** and **Repr**, it seems that the only algorithm for those problems considered in the literature is exhaustive search. This is most obvious when the attempt of Dickson and Ross to decide equivalence of a particular pair of forms is discussed (see [Cas78, p. 132], [CS93,

p. 403]). The restriction to trivial algorithms also becomes explicit in Siegel's famous bound on equivalence transformations [Sie72], which reproves the decidability of (A) using analytic techniques. More precisely, he shows that for each pairs f, g of equivalent forms there is a constant $C > 0$ effectively computable from f, g such that there is a transformation from f to g whose coefficients are absolutely bounded by C. Siegel explicitly refers to the enumeration of all integral matrices up to some bound on the coefficients for testing equivalence.

In dimension three, Siegel's implicit bound has been made explicit and has been improved to polynomial size by Dietmann [Die03]. Still using trivial enumeration, this implies that problem (A) is in NP. Moreover, for problem (B), Grunewald and Segal [GS04] showed decidability by solving problem (B') if possible. Their algorithm is more sophisticated, yet still involves steps of exponential enumeration.

Apparently, the complexity of **Trafo** and **Repr** is hitherto discussed in the literature only once. At the end of [CS93, ch. 15], Conway and Sloane formulate both decisional and computational equivalence and representation problems, along with a couple of additional questions, e. g. on class numbers. They mention several cases for which these problems are easy. For indefinite forms over $\mathbb{Z}$, they express their impression that "there do not seem to be good algorithms", and discuss the inefficiency of exhaustive enumeration.

All this is an indication that we are at a loss if efficient algorithms are required for these problems.

Chapter 2

Cryptography

In this chapter we present cryptographic applications of indefinite quadratic forms. All of them are based on the **Trafo** problem: The public key is an instance of **Trafo**, i. e. two equivalent forms $f \sim g$, and a transformation $S \in \mathrm{GL}_n\mathbb{Z}$ such that $f\,S = g$ plays the role of the private key.

In Sect. 2.2, we present an identification scheme whose security is closely related to the complexity of the underlying problem. However, its practical usefulness is limited because it requires the sequential repetition of many rounds. This drawback is overcome in the signature scheme in Sect. 2.3. However, the security against fraudulent provers cannot be rigorously proven. In both cases, security against fraudulent verifiers is heuristic. These two schemes have been proposed in [HS07b] and [HS07a]. Before we turn to the protocols, we discuss how reduction theory can be used to randomize quadratic forms and unimodular matrices. In particular, we explain how key generation is performed for all schemes.

2.1 Randomization and Key Generation

We define a probability distribution on the set of regular integer matrices and quadratic forms. We will use the following size function for matrices:

$$\|T\| := \max_{i,j} |T_{ij}|.$$

Moreover, we fix the LLL reduction theory (Ω, ρ) from Sect. 1.2.7.

K is a security parameter. The notation

$$T \leftarrow \mathcal{U}_K$$

means that $T \in \mathbb{Z}^{n\times n}$ is selected uniformly at random such that $|T|_\infty \leq K$.

Denote by

$$(g, T) \leftarrow \mathcal{D}_K(f)$$

the process that a form g and a matrix $T \in \mathrm{GL}_n\mathbb{Z}$ satisfying $g = f\,T$ are sampled at random according to the random distribution $\mathcal{D}_K(f)$. The distribution most interesting for us is defined by the following procedure:

```
$(g, T) \leftarrow \mathcal{D}_K(f)$:

$T \leftarrow \mathcal{U}_K$,
$(g, T') := \rho(f\,T)$.
```

So the distribution of g in $\mathcal{D}_K(f)$ takes as values (some of) the finitely many reduced forms equivalent to f. It should not be expected that this is the uniform distribution: For definite forms, it has been observed that LLL-reduced forms often have much smaller coefficients than theoretically guaranteed ([LO85], [VV07]). This indicates that forms closer to the boundary of the set of reduced forms (in $\mathbb{R}^{n(n+1)/2}$) may have lower probability for this distribution; moreover, this effect might even intensify (see [Sim05b]).

Key generation can be accomplished as follows: A reduced form f_{-1} may be fixed as a system parameter. Then using the distribution $\mathcal{D}_K(f_{-1})$, we can generate random reduced forms $f_i = f_{-1}\,S_i$ with $i = 0, 1$. We can use (f_0, f_1) as public and $S := S_0^{-1} S_1$ as private key.

We denote by $\mathbb{P}[A]$ the probability of an event A. The *statistical distance* between two (discrete) random variables $\mathfrak{T}_0, \mathfrak{T}_1$ is defined as

$$\Delta(\mathfrak{T}_0, \mathfrak{T}_1) = \frac{1}{2} \sum_T \Big| \mathbb{P}[\mathfrak{T}_0 = T] - \mathbb{P}[\mathfrak{T}_1 = T] \Big|,$$

where T ranges over all objects which at least one of the $\mathfrak{T}_j$ may attain. Note that $\Delta(\cdot, \cdot)$ takes values in $[0, 1]$.

Two families $\mathfrak{T}_0^{(K)}, \mathfrak{T}_1^{(K)}$ of random variables ($K \in \mathbb{N}$) are called *statistically close* for $K \to \infty$ if their statistical distance is *negligible*; i. e. if

$$\Delta(\mathfrak{T}_0^{(K)}, \mathfrak{T}_1^{(K)}) = \mathcal{O}\left(\frac{1}{P(K)}\right)$$

for every polynomial $P(\cdot)$ (see [Gol01, sec. 3.2.2; def. 1.3.5] for details)*.

Heuristically, it may seem convincing that if K is large enough, then the distribution $\mathcal{D}_K(f)$ does not depend too heavily on the reduced initial form f from a fixed equivalence class. Precisely, we state:

Heuristic 2.1.1 *Let f_{-1} be a fixed quadratic form. Let $\mathfrak{f}_0, \mathfrak{f}_1, S$ arise from the key generation procedure above, i. e. the two pairs $(\mathfrak{f}_i, S_i)$ are independent and distributed according to $\mathcal{D}_K(f_{-1}$, and $S = S_0^{-1} S_1$. Then there is $K' = K'(K)$, depending polynomially on K, such that for $K \to \infty$ the following holds:*

If $(\mathfrak{g}_o, \mathfrak{T}_0)$ is randomly distributed according to $\mathcal{D}_{K'}(f_0)$ and if $(\mathfrak{g}_1, \mathfrak{T}_1)$ is randomly distributed according to $\mathcal{D}_{K'}(f_1)$ then the distributions of $\mathfrak{T}_0$ and $S^{-1}\mathfrak{T}_1$ are statistically close.

[*] Note that narrower definitions of the term ‘negligible’ occur in the literature as well.

2.2 An identification scheme

2.2.1 Specification of the Protocol

Let f_0, f_1 be integral indefinite quadratic forms of dimension $n \geq 3$ and let $S \in \mathrm{GL}_n\mathbb{Z}$ satisfy $f_0\,S = f_1$ (in particular, f_0 and f_1 are equivalent). Denote by $(\mathcal{P}, \mathcal{V})$ the following protocol between prover $\mathcal{P}$ and verifier $\mathcal{V}$:

public key (known to $\mathcal{P}$ and $\mathcal{V}$): quadratic forms f_0, f_1
secret key (known to $\mathcal{P}$ only): $S \in \mathrm{GL}_n\mathbb{Z}$ such that $f_0\,S = f_1$

$\mathcal{P}$ Prover		$\mathcal{V}$ Verifier
$(g, T) \leftarrow \mathcal{D}_K(f_0)$	$\xrightarrow{g}$	
	$\xleftarrow{i}$	select $i \in \{0, 1\}$ uniformly at random
$R := S^{-i}\,T$	$\xrightarrow{R}$	
		check whether $f_i\,R \overset{?}{=} g$

In the next section, we will prove that this scheme is a proof of knowledge, i. e. it is secure against fraudulent provers. Thereafter, we show that $(\mathcal{P}, \mathcal{V})$ is zero-knowledge under Heuristic 2.1.1, which formalizes security against dishonest verifier. Combining these two concepts yields a strong notion of security. A zero-knowledge proof is secure as an identification scheme if and only if extracting a secret key from the public one is intractible.

In [HS07a] it is proposed that $\|T\| \approx 2^{100}\|S\|$ and that 100 rounds be performed. In contrast, we have here presented the scheme in more general form. The question arises whether modifications of it may render the scheme more efficient while keeping the same level of security, or vice versa. More precisely, parameters which can be varied include

(i) the base ring of the forms,

(ii) restriction to forms with certain properties, and

(iii) the distribution of (g, T) in step 1.

The protocol remains a proof of knowledge under all these modifications (see Sect. 2.2.2). Suppose that the new random distribution also preserves the zero-knowledge property. Then the modified protocol is secure if and only if **Trafo** is still hard over the new base ring for the chosen type of forms. Most of this thesis, in particular Chapters 5, 6, 7, is devoted to the question for which choices of rings and form properties **Trafo** is still hard.

The results of those chapters do not recommend any other ring than $\mathbb{Z}$. Moreover, by Sect. 7.1.3 we should take $n = 3$ or $n = 4$ for efficiency, and by Sect. 5.4, Sect. 5.3 we should take indefinite anisotropic forms in these dimensions.

However, it seems possible to employ forms over rings $\mathcal{O}_K$ of integers in algebraic number fields K. Then forms should obey analogous constraints to those over $\mathbb{Z}$, namely they should be at least ternary, for dimensions 3,4 they should be anisotropic, and they must not be totally definite in case K is totally real. The complexity of the transformation problem for these sort of forms is beyond the scope of this thesis. But many theorems formulated here for $\mathbb{Z}$ carry over to $\mathcal{O}_K$, because the the arithmetic theory reveals very similar behavior of forms over $\mathbb{Z}$ and $\mathcal{O}_K$ (see [O'M63]). It is yet to be analyzed whether using, for instance, forms over the Gaussian integers $\mathbb{Z}[i]$, is advantageous over the rational integers.

2.2.2 Proof of knowledge

We turn to proving that the seqential iteration of this identifiaction scheme is a *proof of knowledge*. Intuitively, this means that a fraudulent prover which is accepted by the verifier with non-negligible probability must already know some private key fitting to the current public key. For a more thorough discussion of the following definitions see [Gol01, sec. 4.7].

An identification scheme is called *complete* if an honest prover with the secret key at his disposal always makes the verifier accept.

Denote by x the public key and by w a witness, i. e. a potential private key fitting to x of the scheme. Then the scheme is called *proof of knowledge* with *knowledge error* $\kappa(x)$ if it is complete, and if there is a probabilistic polynomial-time algorithm $\mathcal{X}$, the *knowledge extractor*, with the following property:

Let $\widetilde{\mathcal{P}}$ be any probabilistic polynomial-time algorithm (a 'dishonest prover') which makes $\mathcal{V}$ accept on common input x with probability $\varepsilon(x)$. Then there is $c > 0$ such that whenever $\varepsilon(x) > \kappa(x)$, $\mathcal{X}$ outputs some correct w, on input x and given rewindable black-box access to $\widetilde{\mathcal{P}}$, in expected time

$$\mathcal{O}\left(\frac{|x|^c}{\varepsilon(x) - \kappa(x)}\right).$$

Denote by $(\mathcal{P}, \mathcal{V})^k$ the simultaneous execution of k independent copies of the protocol $(\mathcal{P}, \mathcal{V})$.

Theorem 2.2.1 *$(\mathcal{P}, \mathcal{V})^k$ is a proof of knowlegde with knowledge error 2^{-k}.*

<u>*Proof :*</u> Obviously, $(\mathcal{P}, \mathcal{V})$ and hence $(\mathcal{P}, \mathcal{V})^k$ is complete. To prove knowledge extractibility, consider first a single iteration of the protocol. The prover commits g and replies to challenge $i \in \{0, 1\}$. If $\mathcal{V}$ accepts, rewind $\widetilde{\mathcal{V}}$ and challenge it with the complementary bit $1 - i$, but keep the old commitment g. Let R_i be $\widetilde{\mathcal{P}}$'s reply to challenge i. If $\widetilde{\mathcal{P}}$ is successful again, then

$$f_i\, R_i = g \qquad \text{for } i = 1, 2 \tag{2.1}$$

by specification of the protocol. This implies

$$f_0\,(R_0 R_1^{-1}) = f_1. \tag{2.2}$$

This means that we can compute a solution to the transformation instance (f_0, f_1) as soon as $\widetilde{\mathcal{P}}$ is successful twice with respect to the same commitment.

Now suppose that $\widetilde{\mathcal{P}}$ is successful with probability $\varepsilon > \frac{1}{2}$ in $(\mathcal{P}, \mathcal{V})$. Then we perform the following algorithm $\mathcal{X}$:

Make $\widetilde{\mathcal{P}}$ commit a form g (first step of the protocol). Then challenge it with bit $b = 0$, rewind it, and challenge it with bit $b = 1$ while keeping to the same commitment g. If $\widetilde{\mathcal{P}}$ passes both times, compute a transformation by (2.2). Otherwise, repeat until $\widetilde{\mathcal{P}}$ succeeds on both challenges.

The expected number of iterations of $\mathcal{X}$ equals $\frac{1}{\pi}$, where π denotes the probability that $\widetilde{\mathcal{P}}$ passes on both challenges. We estimate π from below.

Let π_i be the probability that $\widetilde{\mathcal{P}}$ passes on challenge i, $i = 0, 1$. Then

$$\pi \geq \pi_0 \, \pi_1 \tag{2.3}$$

and

$$\pi_0 + \pi_1 = 2\varepsilon. \tag{2.4}$$

We can thus consider

$$\pi = \pi_0(2\varepsilon - \pi_0) \tag{2.5}$$

as a quadratic function of π_0. Since $0 \leq \pi_i \leq 1$ for $i = 0, 1$, Equation (2.4) implies that

$$2\varepsilon - 1 \leq \pi_0 \leq 1. \tag{2.6}$$

It is easy to check that π as a function of π_0 as in (2.5) is concave and therefore takes its minimum on the boundary of its range (2.6). For both boundary values $\pi_0 = 1$ and $\pi_0 = 2\varepsilon - 1$ we obtain $\pi = 2\varepsilon - 1$. Hence we have established that

$$\pi \geq 2\varepsilon - 1.$$

We infer that $\mathcal{X}$ needs at most

$$\frac{1}{2\varepsilon - 1} = \frac{1/2}{\varepsilon - \frac{1}{2}}$$

iterations in expectation, each of which takes polynomial time.

This shows that $(\mathcal{P}, \mathcal{V})$ is a proof of knowledge with knowledge error $\frac{1}{2}$. Then by [Gol01, prop. 4.7.5], the protocol $(\mathcal{P}, \mathcal{V})^k$ is a proof of knowledge with knowledge error 2^{-k}. □

Remark. Note that Theorem 2.2.1 and its proof remain valid if the random distributions $\mathcal{D}_K$, the base ring (here: $\mathbb{Z}$), and the properties of the forms used are replaced by other choices. This means that all these variations of $(\mathcal{P}, \mathcal{V})^k$ are secure against fraudulent provers if the public key is a hard **Trafo**-instance.

2.2.3 Zero-knowledge property

An interactive protocol is called *stastistical zero-knowledge* if for every (possibly dishonest) verifier (i. e. probabilistic polynomial-time algorithm) $\tilde{\mathcal{V}}$, there is a probabilistic polynomial-time algorithm $\mathcal{M}$, the *simulator*, which produces random strings whose distribution is statistically close to that of the interaction between $\mathcal{P}$ and $\tilde{\mathcal{V}}$. The simulator has no access to the secret key, but with rewindable black-box access to $\tilde{\mathcal{V}}$ (for details, see [Gol01]).

Theorem 2.2.2 *Under heuristics 2.1.1, $(\mathcal{P}, \mathcal{V})$ and $(\mathcal{P}, \mathcal{V})^k$ are statistical zero-knowledge protocols.*

Proof : For $(\mathcal{P}, \mathcal{V})$, we specify a simulator $\mathcal{S}$ as follows:

```
repeat
    select random $j \in \{0, 1\}$;
    $(g, T) \leftarrow \mathcal{D}_K(f_j)$
    send $g$ to $\tilde{\mathcal{V}}$ and wait for challenge $i$;
until $i = j$
$R := T$;
output $(g, j, R)$
```

Obviously, the triples output by $\mathcal{S}$ are accepting interactions for $(\mathcal{P}, \mathcal{V})$.

Moreover, the algorithm runs in expected polynomial-time because independently from the distribution according to which $\tilde{\mathcal{V}}$ samples the challenge i, the probability that $i = j$ is at least $\frac{1}{2}$ independently in each iteration. Therefore, $\mathcal{S}$ requires at most two iterations of the *repeat* loop in expectation, which clearly implies expected polynomial running time.

Next, we have to show that under Heuristic 2.1.1, the distribution of $\mathcal{S}$'s output is statistically close to that of an interaction between an honest prover $\mathcal{P}$ and $\tilde{\mathcal{V}}$. First, by construction of $\mathcal{S}$, the distribution of j is exactly that of the challenge i from the interaction of $\mathcal{P}$ and $\tilde{\mathcal{V}}$. Now conditionally on $j = 0$, the matrix R is distributed according to $\mathcal{D}_K(f_0)$, as is R sent by the prover in the execution of the protocol. On the other hand, if $j = 1$, then the simulator outputs R according to $\mathcal{D}_K(f_1)$, whereas $\mathcal{P}$ chooses R according to $S^{-1}\mathcal{D}_K(f_0)$. It is here where our heuristics 2.1.1 tells us that these two distributions are statistically close. Finally, as the triples (g, j, R) are accepting interactions, g is uniquely determined by j, R since $g = f_j\, R$. Therefore, the two distributions in question are statistically close. □

In constrast to the proof-of-knowlegde condition, the zero-knowledge property of the protocol depends heavily on the family of distributions $\mathcal{D}_K$.

The main disadvantage of $(\mathcal{P}, \mathcal{V})$ is, of course, its restriction to one-bit challenges. To guarantee a certain level of security, one would have to run many copies of it in parallel; to have the zero-knowledge property, we would even need the single applications of $(\mathcal{P}, \mathcal{V})$ to be run *sequentially*, which would make communication too costly.

2.3 Identification with Long Challenges and Signatures

2.3.1 Specification of the protocol

The major drawback of the scheme of Sect. 2.2 is its restriction to one-bit challenges. Therfore security requires many sequential iterations (see Theorem 2.2.1), which is too inefficient for practical purposes.

A potential solution of this issue is given here. We present an identification scheme which admits arbitrarily long challenges (depending on key size and parameters). Therefore it is also suitable for the generation of digital signatures, see Sect. 2.3.2.

These advantages come at the price of provable security. In particular, perhaps it fails to be a proof of knowledge. This makes it harder to find out its actual level of security. It may nevertheless be secure in practice, and we will give heuristic arguments to support that claim.

The protocol relies on the hardness of both **Trafo** and **Repr**.

We denote the following protocol by $(\mathcal{P},\mathcal{V})'$. It was introduced in [HS07b], a refinement of it being proposed in [HS07a].

public key (known to $\mathcal{P}$ and $\mathcal{V}$): quadratic forms f_0, f_1
secret key (known to $\mathcal{P}$ only): $S \in \mathrm{GL}_n\mathbb{Z}$ such that $f_0\, S = f_1$

$\mathcal{P}$ Prover		$\mathcal{V}$ Verifier
$(g,T) \leftarrow \mathcal{D}_K(f_0)$	$\xrightarrow{g}$	
	$\xleftarrow{T'}$	$(h,T') \leftarrow \mathcal{D}_K(g)$
$e_1' := TT'e_1$, $e_4' := S^{-1}TT'e_4$	$\xrightarrow{e_1',e_4'}$	check whether $f_0 \overset{?}{\sim} g$, $f_0(e_1') \overset{?}{=} h(e_1)$, and $f_1(e_4') \overset{?}{=} h(e_4)$

2.3.2 Application to digital signatures

We have now described an identification scheme where the number of possible challenges is an increasing function of the security parameter K. For this situation there is a well-established technique to turn this identification protocol into a digital signature scheme [BR93]: Namely, replace the verifier by a public hash function. Then the challenge in the scheme is a function of the message to be signed. Of couse, this means that the hash function H has to take values in $\mathrm{GL}_4\mathbb{Z}$. In the Bellare-Rogaway construction, H has to be *collision resistant*, i. e. it is hard to find messages m, m' such that $H(m) = H(m')$; this condition can be slightly relaxed for this scheme (see [HS07a]). Moreover, H has to be a one-way function, as otherwise a fraudulent prover can choose a challenge, compute its preimage m, and compute a valid signature for m.

Conversely, consider the *random oracle model (ROM)*, i. e. assume that values of the hash function are retrieved from a public oracle which on new queries computes a random output, but replies with the same answer to repeated queries. Then it has been shown in [BR93] that the signature scheme obtained by replacing the verifier by a hash function in a zero-knowledge proof, is a non-interactive zero-knowledge proof in the random oracle model. The same argument extends to statistical zero-knowledge.

The ROM was introduced into cryptography in [FS87]. It seems to be an appropriate model to analyse security of protocol which uses a collision-free one-way hash function, which is independent from the cryptographic primitive of the protocol; The ROM is motivated by the impression that no useful structure can be realized in the values of a well-designed hash-function, so that an attacker would not have worse chances of success if H was replaced by a random oracle. It should be noted that the ROM was massively critized lately: Most prominently, [CGH98], examples of security flaws have been given for any concrete hash function, in cryptographic schemes secure in the ROM. However, as is argued in [KM07], these constructions contradict reasonable cryptographic practice, and could even motivate to trust in the ROM more than before.

2.3.3 Security against Fraudulent Provers

There is the following heuristic argument for security against a fraudulent $\widetilde{\mathcal{P}}$: In the last step, $\widetilde{\mathcal{P}}$ knows a transformation $R = TT'$ satisfying $f_0\, R = h$. She has to solve the equation $f_1(e'_4) = h(e_4)$ for $e'_4 \in \mathbb{Z}^4$. Heuristically, there seem to be essentially only two ways to solve such a problem: Either using an algorithm which solves the representation problem for f_1 (the R is ignored), or finding a transformation S satisfying $f_0\, S = f_1$, and computing $e'_4 := S^{-1}Re_4$ as in the protocol.

In the latter case, this would imply that $\widetilde{\mathcal{P}}$ has knowledge of some transformation from f_0 to f_1, which is our desired result. In the former case, $\widetilde{\mathcal{P}}$ would still have to solve a presumably hard problem, namely **Repr**.

However, the scheme is not known to be a proof of knowledge, in the sense that passing as prover reduces to solving the **Trafo**-instance (f_0, f_1). One obstacle to such a proof is the following: We can construct a successful fraudulent prover if we have an algorithm which solves the representation problem for f_0, f_1. Namely, in this case it suffices to solve the verification equations for e'_1, e'_4 in the last step. Therefore, $(\mathcal{P}, \mathcal{V})'$ can only be a proof of knowledge if solving representations for f_0, f_1 reduces to solving for a transformation from f_0 to f_1.

It seems reasonable to expect that computing representations is not significantly easier than computing transformations, even if there might not be a complexity reduction in general. In later chapters, we find cases in which we can reduce from a representation to a transformation problem; however, these reductions do not refer to the same (equivalence class of) forms (see the proofs of Theorems 9.1.8, 9.1.9 and Chapter 10 for complexity connections between **Trafo** and **Repr**).

2.3.4 Security against Fraudulent Verifiers

As with the proof-of-knowledge property, there is a major obstacle to the zero-knowledge property of $(\mathcal{P}, \mathcal{V})'$. Namely a simulator $\mathcal{S}$ would have to produce quintuples (g, T', T, e_1', e_4') where (among other relations) $f_1(e_4') = f_0(TT'e_4)$ holds. As argued in Sect. 2.3.3, this implies that heuristically, $\mathcal{S}$ should be able to either compute a transformation from f_0 to f_1, or to solve representation problems for f_1. Both tasks are presumably hard.

However, consider what we can learn about S if listening to the interaction of an honest prover $\mathcal{P}$ with a possibly fraudulent verifier $\widetilde{\mathcal{V}}$. We cannot gain any useful information if the distribution of e_4' does not reveal anything about S. This hypothesis is a heuristic similar to 2.1.1. Moreover, any information obtained cannot be combined over several identifications as the matrices T are statistically independent.

Chapter 3

Localization and Decisional Complexity

In this chapter, we consider the transformation and representation problems over the local rings $\mathbb{Z}_p$. At first, in Sect. 3.1, we analyze the decisional variants of **Trafo** and **Repr**. It turns out that deciding equivalence and representability is feasible over the $\mathbb{Z}_p$, and in consequence of local-global priciples as Theorem 1.2.18, these results extend to $\mathbb{Z}$ for a large set of instances.

Then we turn back to the computational problems. We show that these are also tractable over finite fields and over the p-adic integers in Sections 3.2 and 3.3, respectively.

3.1 Decision Problems

In computational complexity, the decisional and computational versions of problems frequently turn out to be polynomial-time equivalent; in particular, this holds for several NP-complete problems as SAT (cf. [GJ79, §§2.1, 5.1]).

Presumably, this is not the case with our problems of quadratic forms. The arithmetic theory has produced easy-to-check sufficient criteria for representability and equivalence. This leads to Theorems 3.1.3 and 3.1.5. By contrast, **Repr** and ***Repr** seem to be computationally intractible, see Sect. 1.3.6.

Over $\mathbb{Z}$, we let the decision problems include the factorization of the determinant into the input. This follows the general concept motivated in Chapter 8. The letter **F** in the problem symbol reflects this modification.

DRepr^R **Decisional representation problem over** R
INPUT: a quadratic form f over R and $m \in R$. *DECIDE:* whether
f represents m over R.

DFRepr($\mathcal{P}$) Decisional representation problem with factorization
PARAMETERS: Set $\mathcal{P}$ of properties of quadratic forms.
INPUT: a quadratic form f over $\mathbb{Z}$ satisfying all properties from $\mathcal{P}$, $m \in R$, factorization of $\det f$.
DECIDE: whether f represents m over $\mathbb{Z}$.

To prove that $\mathbf{DRepr}^{\mathbb{Z}_p}$ is in P, we first have to explore the image of binary forms more closely.

Lemma 3.1.1
Let p be an odd prime and $\left(\frac{\rho}{p}\right) = -1$.

(a) If $p \equiv 1 \bmod 4$, then $\langle 1,1\rangle \xrightarrow{*}_{\mathbb{Z}_p} a$ for all $a \in \mathbb{Z}_p$, and $\langle 1,\rho\rangle \xrightarrow{*}_{\mathbb{Z}_p} a \in \mathbb{Z}_p$ if and only if $p \not| a$.

(b) If $p \equiv 3 \bmod 4$, then $\langle 1,\rho\rangle \xrightarrow{*}_{\mathbb{Z}_p} a$ for all $a \in \mathbb{Z}_p$, and $\langle 1,1\rangle \xrightarrow{*}_{\mathbb{Z}_p} a \in \mathbb{Z}_p$ if and only if $p \not| a$.

Proof : Consider the form $f := \langle 1, r\rangle$, where $r \in \{1, \rho\}$, and let $a \in \mathbb{Z}_p$. We analyze the representation problem inductively mod p^n, for $n \in \mathbb{N}$.

At first, for $n = 1$, the equation $x^2 + ry^2 = a$ is always solvable modulo p because of a simple counting argument (see [Cas78, lemma 2.2.2]). If $x \equiv y \equiv 0$ in this reprensentation, then it can obviously not be lifted to a primitive representation over $\mathbb{Z}_p$. Otherwise, however, it guarantees the existence of a primitive representation over $\mathbb{Z}_p$ by Hensel's lemma.

Thus f represents primitively at least all p-adic integers not divisible by p. If $p|a$, then, by the above argument, there is a primitive representation of a mod p if and only if f is isotropic over $\mathbb{F}_p$. As f is binary, this occurs if and only if $-\det f = -r$ is a square in $\mathbb{F}_p$ by [Cas78, lm. 2.4 of ch. 4]. In the case $r = 1$ this means that $\left(\frac{-1}{p}\right) = 1$ or, equivalently, $p \equiv 1 \bmod 4$. In the case $r = \rho$ it implies $\left(\frac{-1}{p}\right) = -1$ and thus $p \equiv 3 \bmod 4$. □

Recall that the omission of the property set $\mathcal{P}$ stands for no restriction on the forms at all.

Theorem 3.1.2
Let p be a prime, or $p = \infty$. Then

$$\mathbf{DRepr}^{\mathbb{Z}_p}$$

is solvable in polynomial time.

Remark. By a slight variation of the argument below we will as well be able to decide primitive representability over $\mathbb{Z}_p$.

Proof : Let (f, m) be an instance of $\mathbf{DRepr}^{\mathbb{Z}_p}$. Write $n := \dim f$. Over $\mathbb{Z}_\infty = \mathbb{R}$, the form f represents m if and only if $m > 0$ and $\operatorname{sign} f < n$, or $m < 0$ and $\operatorname{sign} f > 0$, or $m = 0$ and $0 < \operatorname{sign} f < n$.

So let p be a prime and $d := \det f$. If $m = 0$ then (f, m) is a 'yes'-instance if and only if f is isotropic over $\mathbb{Q}_p$. By [Cas78, lmm. 2.4–2.7 of ch. 4], this is the case if and only if

(i) $n = 2$ and $-d \in \mathbb{Q}_p^{*2}$; or

(ii) $n = 3$ and $c_p(f) = \left(\frac{-1,-d}{p}\right)$; or

(iii) $n = 4$, $d \in \mathbb{Q}_p^{*2}$, and $c_p(f) = -\left(\frac{-1,-1}{p}\right)$; or

(iv) $n \geq 5$.

Hence we may restrict to $m \neq 0$.

Let p be odd. Let $m = p^k m_0$, $m_0 \in \mathbb{Z}_p^*$. To decide whether $f \longrightarrow_{\mathbb{Z}_p} m$, execute the following algorithm:

```
$i := 0$;
while $i \leq k$ do
    1.) ensure by Theorem 1.2.13 that $f$ is in normal form
    2.) split $f$ as $f = p^i f_0 \perp p^{i+1} f_1$ such that $\det f_0 \in \mathbb{Z}_p^*$
    3.) if $f_0 \longrightarrow_{\mathbb{Z}_p} p^{k-i} m_0$ then output 'yes'; fi
    4.) $f := p^{i+2} f_0 \perp p^{i+1} f_1$;
    $i := i + 1$;
od
output 'no'.
```

Note that step 1 only serves to prepare step 2. In step 2 it is understood that f_0, f_1 have to be p-adically integral.

We now detail how to perform step 3. Let $m' := p^{k-i} m_0$. If f_0 is an empty form (i. e. of dimension 0) then obviously f_0 does not represent m'. If $f_0 = \langle r \rangle$ is unary, then it represents m' if and only if $\left(\frac{m_0}{p}\right) = \left(\frac{r}{p}\right)$ and $k - i$ is even. If f_0 is at least ternary, then by Theorem 1.2.13

$$f_0 \sim_{\mathbb{Z}_p} x_1 x_2 - (\det f_0) x_3^2 + \sum_{i=4}^{n} x_i^2$$

and hence represents m'.

It remains to deal with $\dim f_0 = 2$. Let $d_0 := \det f_0$. Then Lemma 3.1.1 implies that f_0 represents m' if and only if

(i) $p \equiv 1 \bmod 4$ and $\left(\frac{d_0}{p}\right) = 1$; or

(ii) $p \equiv 1 \bmod 4$, $\left(\frac{d_0}{p}\right) = -1$, and $k - i$ is even; or

(iii) $p \equiv 3 \bmod 4$, $\left(\frac{d_0}{p}\right) = 1$, and $k - i$ is even; or

(iv) $p \equiv 4 \bmod 4$ and $\left(\frac{d_0}{p}\right) = -1$.

These conditions can all be checked in polynomial time.

We now prove correctness of the algorithm. Denote by (F, m) the original (static) problem instance and by f the form modified during runtime (note that the values of n_0, i, m' are also temporary). Use the following auxiliary

Claim: Every time in step 1, the current form f represents m over $\mathbb{Z}_p$ if and only if the original F does.

Proof of claim: We employ induction on the number of iterations. Consider one execution of the modification step 4. If we have arrived there then the algorithm did not halt at step 3 of the same iteration, i. e. there is no representation $v \in \mathbb{Z}_p^{n_0}$ of m' by f_0. We claim that then also no $\bar{v} \in \mathbb{Z}_p^n$ exists satisfying

$$f(\bar{v}) = m \qquad \text{and} \qquad \gcd_{\mathbb{Z}_p}(\bar{v}_1, \ldots, \bar{v}_{n_0}) = 1. \tag{3.1}$$

In fact, if (3.1) holds, then

$$p^i f_0(v) \equiv f(\bar{v}) = m \bmod p^{i+1}$$

and therefore

$$f_0(v) \equiv m' \bmod p,$$

where $v = (\bar{v}_1, \ldots, \bar{v}_{n_0})^t$. If $p \not| m'$ (i. e. $i = k$), then by Hensel's Lemma (or the above analysis of f_0) there is a representation of m' by f_0, and we should have halted in step 3, a contradiction.

So $p|m'$. In particular, $f_0(v) \equiv 0 \bmod p$. If f_0 is isotropic over $\mathbb{Z}_p$, then it represents m', with the same contradiction again. Therefore

$$v \equiv (0, \ldots, 0)^t \bmod p.$$

If there is still a representation of m by f, then the first n_0 coefficients of v are divisible by p. Hence m is also represented by

$$p^{i+2} f_0 \perp p^{i+1} f_1,$$

which is the modified f. This completes the induction step, and the claim is proven. ♢

Now we can prove that a representation exists if and only if the algorithm returns 'yes'. If the algorithm answers 'yes', then we have stopped in step 3, and there is $v \in \mathbb{Z}_p^{n_0}$ satisfying $f_0(v) = m'$. But then $f(\bar{v}) = m$, where $\bar{v}$ arises from v by adding zeros to the end. Hence (f, m) is a 'yes'-instance of $\mathbf{DRepr}^{\mathbb{Z}_p}$.

We now turn the converse implication. Let F represent m over $\mathbb{Z}_p$ and assume that the algorithm outputs 'no', i. e. the `while`-loop is left only on $i = k + 1$. We have already verified that if the algorithm does not halt during the i-th iteration, then f as of the $(i+1)$-th iteration still represents m. In particular, f represents m at the end of the last `while`-iteration.

It is easy to see that $p_i|f$ every time step 1 is executed. This implies that $p^{k+1}|f$ after the `while`-loop. Therefore, if $f(v) = m$,

$$0 \equiv f(v) = m \not\equiv 0 \bmod p^{k+1},$$

a contradiction. We have thus shown that if F represents m, then at some point the algorithm stops to return 'yes'.

Obviously, the algorithm runs in polynomial time. This concludes the proof for p odd.

To prove the theorem for $p = 2$, we follow the same strategy as for p odd. The algorithm needs only a slight modification: In step 2, f_0 has to be the 2-adically *classically* integral form of maximal dimension satisfying

$$f = 2^i f_0 \perp 2^{i+1} f_1$$

for some 2-adically classically integral f_1. Then the proof of correctness carries over. We only have to characterize the images of the possible f_0 in step 3.

As f is in normal form, f_0 is either an orthogonal sum of the diagonal forms listed in Theorem 1.2.15, or a form of the shape

$$h \perp \ldots \perp h \qquad \text{or} \qquad h \perp \ldots \perp h \perp h',$$

where h, h' are binary forms with associated matrices

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix},$$

respectively (see [Jon44]).

Note that if f contains a summand h, then it represents all 2-adic integers. Hence it suffices to determine the image of the properly primitive forms from Theorem 1.2.15. This can be done for $n_0 \leq 6$ by enumerating all primitive vectors

$$v \in (\mathbb{Z}_2/8\mathbb{Z}_2)^{n_0};$$

then f_0 represents m if and only if

$$f(v) \equiv 2^{-2i} m \bmod 8$$

for some $i \leq \lfloor \frac{k}{2} \rfloor$ and some such v. For $n \geq 7$, the normal form of f_0 represents the form $\langle 1,1,1,1 \rangle$ by Theorem 1.2.15, and this form represents all of $\mathbb{Z}_2$ (e. g. by Lagrange's Four Square Theorem, see [Cas78, p. 144]). □

In many intersting cases, the arithmetic theory admits carrying over this result to $\mathbb{Z}$.

Theorem 3.1.3
Let $\mathcal{P}$ denote following properties of an integral quadratic form f: It is indefinite, and either

(i) $\dim f \geq 4$, *or*

(ii) $\dim f = 3$, $d := \det f$ *is* $\frac{n(n-1)}{2}$*-power free, and if f is classically integral, then*

$$2^{n(n-3)/2+\lfloor (n+1)/2 \rfloor} \not| d.$$

Then **DFRepr**$(\mathcal{P})$ *is solvable in polynomial time.*

In constrast, we will see in Theorem 9.1.6 that for definite forms, the same problem is NP-complete under randomized reductions.

Proof : Let (f, m) be an instance of **DFRepr**$(\mathcal{P})$. The key technique consists in verifying that f represents m over all $\mathbb{Z}_p$. This is clearly necessary for m being represented by f over $\mathbb{Z}$. In case (i), it is also sufficient by [Kit93, thm. 6.6.1]. In case (ii) m is represented by some form $f' \in \operatorname{gen} f$ by [Cas78, thm. 1.3 of ch. 9]. Then the hypotheses on d and Theorem 1.2.18 imply that f and f' are equivalent, and hence the condition is also sufficient. It therefore remains to specify how to check local representability.

Note that it suffices to consider $\mathbb{Z}_p$ for the primes $p|2d$: Over $\mathbb{Z}_\infty = \mathbb{R}$, m is represented since f is indefinite. Moreover, if $p \not| 2d$ then f is $\mathbb{Z}_p$-equivalent to

$$f' = x_1x_2 - dx_3^2 + \sum_{i=4}^{n} x_i^2$$

by Theorem 1.2.13. In particular, $f'(v) = m$ for $v = (m, 1, 0, \ldots, 0)^t$. Hence f represents m over $\mathbb{Z}_p$.

For the $p|2d$, apply Theorem 3.1.2. □

Analogous statements hold for the decisional transformation problem. We keep to our convention on giving the factorization over $\mathbb{Z}$ for free.

DTrafo^R **Decisional transformation problem over** R
INPUT: quadratic forms f, g over R.
DECIDE: whether $f \sim_R g$.

DFTrafo$(\mathcal{P})$ **Decisional transformation problem with factorization**
PARAMETERS: Set $\mathcal{P}$ of properties of quadratic forms.
INPUT: quadratic forms f, g over $\mathbb{Z}$ satifying all properties from $\mathcal{P}$, factorization of $(\det f)(\det g)$.
DECIDE: whether $f \sim g$.

Theorem 3.1.4
Let p be any symbol (i. e. a prime or ∞). Then

$$\mathbf{DTrafo}^{\mathbb{Z}_p}$$

is solvable in polynomial time.

Proof : For $p = \infty$, forms f and g are $\mathbb{R}$-equivalent if and only if their signatures coincide (Proposition 1.2.9). For p a prime, compute the normal forms of f, g. By Theorems 1.2.13, 1.2.15, f and g are $\mathbb{Z}_p$-equivalent if and only if their normal forms coincide. This can be checked in polynomial time. □

Similarly, we have the following global result.

Theorem 3.1.5
Let $\mathcal{P}$ denote following properties of an integral quadratic form f: It is indefinite, $\dim f \geq 3$, and $d := \det f$ is $\frac{n(n-1)}{2}$-power free; and if f is classically integral, then

$$2^{n(n-3)/2+\lfloor (n+1)/2 \rfloor} \nmid d.$$

Then $\mathbf{DFTrafo}(\mathcal{P})$ *is solvable in polynomial time.*

Proof : Let (f, g) be an instance of $\mathbf{DFTrafo}(\mathcal{P})$. Without loss assume that $\det f = \det g =: d$, as otherwise $f \not\sim g$. Then the hypotheses on $\det f$ and Theorem 1.2.18 imply that f, g lie in a one-class genus. Hence (f, g) is a 'yes'-instance if and only if $f \sim_g g$; By Theorem 3.1.4, we can check whether $f \sim_{\mathbb{Z}_p} g$ for all $p|2d\infty$ in polynomial time. If this holds, $f \sim_g g$ by the remark after Theorem 1.2.13. □

An analogous result holds over the rationals: Over $\mathbb{Q}_p$, classifying the image of forms is even easier than over $\mathbb{Z}_p$; this is because there are at most two equivalence classes of forms to be considered per dimension and determinant, and because every $\mathbb{Q}_p$-form is isotropic by Meyer's theorem and hence represents all p-adic numbers (it is *universal*, see [Cas78, sec. 2.4]). The Strong Hasse Principle implies (see [Cas78, thm. 1.1, cor. 2 of ch. 6]) that m is represented by f over $\mathbb{Q}$ if and only if it is represented by f over all $\mathbb{Q}_p$, and again it suffices to consider all $p|2(\det f)\infty$ in case f is integral.

3.2 Forms over Finite Fields

In this section, we consider the transformation and representation problems over finite prime fields. The classification theorems of quadratic forms in Sect. 1.2.5 show that the class structure of quadratic forms over the fields $\mathbb{F}_p$ is relatively simple. We prove that the computational complexity of **Repr**, **Trafo** is also quite low, as they can be solved in random polynomial time.

Theorem 3.2.1 *Let p be an odd prime and let $\mathbb{F}$ be the finite field with p elements. Then* $\mathbf{Trafo}^{\mathbb{F}}$ *is solvable in random polynomial time.*

Proof : Let $\mathbb{F}$-equivalent forms f, g over $\mathbb{F}$ be given. Use Lemma 1.2.3 to transform either of them into diagonal shape, say

$$f\,T_1 = \langle a_1, \dots, a_n \rangle \qquad \text{and} \qquad g\,T_2 = \langle b_1, \dots, b_n \rangle$$

with $T_i \in \mathrm{SL}_n\mathbb{F}$. Then determine which of the a_i, b_i are squares in $\mathbb{F}$. This can be done in polynomial time by computing Jacobi symbols. Build a permutation matrix Π_i for either form such that $f\,T_1\Pi_1 = \langle s_1, \dots, s_k, q_{k+1}, \dots, q_n \rangle$ and $g\,T_2\Pi_2 = \langle s'_1, \dots, s'_\ell, q'_{\ell+1}, \dots, q'_n \rangle$ such that all q_i, q'_i are squares and all s_i, s'_i are non-squares. Without loss we may assume that $k \geq \ell$. Let $s := s_1$ (if $k \neq 0$). Then $s_i^{-1}s$, $(s'_i)^{-1}s$ are square for all i. By [CP01, sec. 2.3.2], we can compute square roots

$$\begin{aligned} t_i^2 &= s_i^{-1}s && \text{for } i = 1, \dots, k, \\ (t'_i)^2 &= (s'_i)^{-1}s && \text{for } i = 1, \dots, \ell, \\ r_i^2 &= q_i && \text{for } i = k+1, \dots, n, \text{ and} \\ (r'_i)^2 &= q'_i && \text{for } i = \ell+1, \dots, n. \end{aligned}$$

in random polynomial time for all i for which the respective right hand side is defined. Then

$$\begin{aligned} f' &:= f\,T_1\Pi_1 S_1 = \langle \underbrace{s, \dots, s}_{k \text{ times}}, \underbrace{1, \dots, 1}_{n-k \text{ times}} \rangle, \qquad \text{and} \\ g' &:= g\,T_2\Pi_2 S_2 = \langle \underbrace{s, \dots, s}_{\ell \text{ times}}, \underbrace{1, \dots, 1}_{n-\ell \text{ times}} \rangle, \end{aligned}$$

where

$$S_1 = \begin{pmatrix} t_1 & & & & & 0 \\ & \ddots & & & & \\ & & t_k & & & \\ & & & r_{k+1} & & \\ & & & & \ddots & \\ 0 & & & & & r_n \end{pmatrix}$$

and

$$S_2 = \begin{pmatrix} t'_1 & & & & & 0 \\ & \ddots & & & & \\ & & t'_\ell & & & \\ & & & r'_{\ell+1} & & \\ & & & & \ddots & \\ 0 & & & & & r'_n \end{pmatrix}.$$

As f' and g' are $\mathbb{F}$-equivalent, their determinant may only differ by a square in $\mathbb{F}$. Hence $k - \ell$ is even. Therefore it suffices for the completion of the algorithm to construct a transformation of the form $\langle s, s \rangle$ into the form $\langle 1, 1 \rangle$.

Note that there are $x, y \in \mathbb{F}$ satisfying $sx^2 + sy^2 = 1$ by [Cas78, ch. 2, lm. 2.2]. Such a solution can be found efficiently using [AEM87]. Then find x', y' such that

$$\det \begin{pmatrix} x & x' \\ y & y' \end{pmatrix} = s^{-1}$$

(e. g. by choosing x' uniform at random and solving for y', until success). Then the form

$$h := \langle s, s \rangle \begin{pmatrix} x & x' \\ y & y' \end{pmatrix}$$

has the right first coefficient and the right determinant. Therefore, another application of Lemma 1.2.3 transforms it into the form $\langle 1, 1 \rangle$. This completes the description of the algorithm. As argued in the single steps, it runs in random polynomial time. □

It is easy to modify this procedure to obtain a proper transformation (of determinant $+1$).

The reference [AEM87] used in this proof already shows that we can also efficiently find representations over prime fields. The algorithm of Adleman, Estes, and McCurley [AEM87] particularly makes sure that even probabilism, and the number-theoretic assumption as the Extended Riemann Hypothesis, can be avoided (cf. [PS87])*.

Theorem 3.2.2 *Let p be an odd prime and let $\mathbb{F}$ be the finite field with p elements. If $n \geq 2$, then $\mathbf{Repr}_n^{\mathbb{F}}$ is solvable in (deterministic) polynomial time.*

Proof : See [AEM87]. □

It may seem peculiar that we had to exclude the case $n = 1$. For unary forms, the representation problem is equivalent to computing square roots in $\mathbb{F}$. This can be done efficiently in practice. However, it is not known to be possible in *deterministic* polynomial-time unless the Extended Riemann Hypothesis holds (see [CP01, sec. 2.3.2]).

3.3 Forms over Local Rings

Lemma 3.3.1 *Let p be a prime, $m \in \mathbb{Z}_p$, and $n, k \in \mathbb{N}$. Let f be a binary quadratic form over $\mathbb{Z}_p$ such that $\det f \in \mathbb{Z}_p^*$.*

If f represents m primitively over $\mathbb{Z}_p$, then one can construct, to precision k, a primitive representation $v \in \mathbb{Z}_p^n$ satisfying $f(v) = m$, in polynomial time.

*Perhaps this result is even older. I am however not aware of a corresponding publication. The main objection of the authors of [AEM87] is to solve modular equations without having to factor the modulus.

Note that if f represents m and $p^2 \nmid m$, then f automatically represents m primitively.

Proof : First assume that $p \nmid m$. By Lemma 1.2.3 we may assume that f has been transformed into diagonal form, i. e.

$$f = \langle r_1, r_2 \rangle.$$

We also have $r_i \in \mathbb{Z}_p^*$, $i = 1, 2$ since $p \nmid m$ and f has $\mathbb{Z}_p$-coefficients.

The requirement of precision k means that the have to find a vector $\bar{v} \in (\mathbb{Z}/p^{k+1}\mathbb{Z})^2$ satisfying

$$f(v) \equiv m \bmod p^{k+1}, \tag{3.2}$$

such that $\bar{v}$ can be lifted to a global solution $v \in \mathbb{Z}_p^2$, i. e.

$$\Rightarrow\ (v) = m \qquad \text{and} \qquad v \equiv \bar{v} \bmod p^{k+1}.$$

A vector $\bar{v}$ satisfying (3.2) can be computed in (deterministic) polynomial-time by [AEM87].

We claim that any such solution can be lifted to a solution over $\mathbb{Z}_p$. First note that $p \nmid m$ implies

$$p \nmid r_i \bar{v}_i^2 \tag{3.3}$$

for at least one $i \in \{1, 2\}$. Then

$$\frac{\partial f(x)}{\partial x_i}\Big|_{x=\bar{v}} = 2 r_i \bar{v}_i$$

and thus

$$\nu_p \left(\frac{\partial f(x)}{\partial x_i}\Big|_{x=\bar{v}} \right) = 0$$

because of (3.3), and since p is odd. The solution can now be lifted by Hensel's Lemma [Eis95, thm. 7.3], since the solution $\bar{v}$ is correct modulo

$$\left(\frac{\partial f(x)}{\partial x_i}\Big|_{x=\bar{v}} \right)^2 p.$$

This proves the lemma for $p \nmid m$.

Next consider $p|m$. Then f is (regular) isotropic modulo p. In particular, it is $\mathbb{F}_p$-equivalent to the form $2x_1x_2$. By Theorem 3.2.1, we can compute $\bar{T} \in \mathbb{Z}_p^2$ such that

$$f\bar{T} \equiv 2x_1x_2 \bmod p \qquad \text{and} \qquad \det \bar{T} \not\equiv 0 \bmod p. \tag{3.4}$$

Let

$$\bar{T} = \begin{pmatrix} \bar{s} & \bar{t} \\ \bar{u} & q \end{pmatrix}.$$

Then (3.4) implies that $p \nmid \bar{s}$ and $p \nmid \bar{t}$, or $p \nmid \bar{u}$ and $p \nmid q$. So after multiplying $\bar{T}$ with the transposition matrix $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ we may assume that $p \nmid \bar{t}$.

We claim that f is $\mathbb{Z}_p$-equivalent to the form $2x_1x_2$; more prescisely, that $\bar{T}$ can be lifted to an equivalence transformation between these two. Let

$$T = \begin{pmatrix} s & t \\ u & q \end{pmatrix} \qquad \text{and} \qquad T^t \begin{pmatrix} r_1 & \\ & r_2 \end{pmatrix} T = \begin{pmatrix} a & b \\ b & c \end{pmatrix}.$$

Then the relation $f\,T = 2x_1x_2$ is equivalent to the three equations

$$\begin{aligned} r_1s^2 + r_2u^2 & && = 0, \\ r_1st + r_2uq & &-1 &= 0, \\ r_1t^2 + r_2q^2 & && = 0 \end{aligned}$$

in the three indeterminates s, t, u. The Jacobi determinant of this equation system amounts to

$$\det \begin{pmatrix} 2r_1s & r_1t & 0 \\ 0 & r_1s & 2r_1t \\ 2r_2u & r_2q & 0 \end{pmatrix} = -4r_1^2r_2t(\det T).$$

This value does not vanish modulo p if we sustitute s, t, u for $\bar{s}, \bar{t}, \bar{u}$. Hence the multidimensional variant of Hensel's Lemma (see [Eis95, p. 209]) states that there is a matrix $T \equiv \bar{T} \bmod p$ such that $f\,T = 2x_1x_2$. Moreover, the coefficients of T can inductively be computed to arbitrary precision.

Now that we know the transformation T to sufficient precision, we only need to find a representation of m by the form $2x_1x_2$. But such is obviously given by $x_1 = 1$, $x_2 = \frac{1}{2}m$. This concludes the proof for $p|m$. □

For the discussion of [AEM87] see also Sect. 6.1.

Lemma 3.3.2 *Let p be a prime, $m \in \mathbb{Z}_p$, and $n, k \in \mathbb{N}$. Let f be a quadratic form over $\mathbb{Z}_p$ such that*

$$\det f \in \mathbb{Z}_p^* \qquad \textit{and} \qquad \dim f \geq 3.$$

Then f represents m primitively over $\mathbb{Z}_p$, and a primitive representation can be efficiently computed (to precision k).

Proof : After application of Lemma 1.2.3, we have

$$f = \langle r_1, \ldots, r_n \rangle.$$

If $p \nmid m$, then m can be represented by $\langle r_1, r_2 \rangle$ according to Lemma 3.1.1. Hence we may choose $x_3 = \ldots = x_n = 0$, whereas x_1, x_2 can be efficiently computed by use of Lemma 3.3.1.

If, however, $p \nmid m$, then let $x_3 = 1$ and $x_4 = \ldots = x_n = 0$. Then $p \nmid r_3$ and therefore $p \nmid m - r_3$. Consequently, there is a representation of $m - r_3$ by $\langle r_1, r_2 \rangle$, which can be computed in polynomial time by Lemma 3.3.1. This representation (x_1, x_2) combines with $x_3, \ldots, x_n$ chosen so far to a representation of m by f, as desired. All computations are to precision k. □

Theorem 3.3.3 *Let p be a prime or $p = \infty$.*

Then

$$\mathbf{Repr}^{\mathbb{Z}_p} \qquad \textit{and} \qquad {}^*\mathbf{Repr}^{\mathbb{Z}_p}$$

are solvable in polynomial time.

Proof : Proceed according to the following algorithm. All computations are performed to the required precision.

```
$i := 0$;
$e := (0, \dots, 0)^t \in \mathbb{N}_0^n$; ### exponents of $p$ in the solution vector
compute $T \in \mathrm{GL}_n\mathbb{Z}_p$ such that $f\,T = \langle r_1, \dots, r_n \rangle$ is in diagonal form
while $i \le k$ do
    select the indices $j_1 \le \dots \le j_k$ satisfying $\nu_p(r_{j_u}) = i$;
    $f_0 := \langle p^{-i} r_{j_1}, \dots, p^{-i} r_{j_k} \rangle$;
    if $f_0 \longrightarrow_{\mathbb{Z}_p} p^{k-i} m_0$ then
        use Lemmata 3.1.1, 3.3.2 to compute $w$ s.t. $f_0(w) = p^{k-i} m_0$;
        define $v \in \mathbb{Z}_p^n$ by
            $v_j := \begin{cases} p^{e_{j_u}} w_{j_u} & \text{if } j = j_u, \\ 0 & \text{else;} \end{cases}$
        break ;
    fi
    for $u = 1, \dots, k$ do
        redefine $f$: replace the $j_u$-th diagonal coefficient, $r_{j_u}$, by $p^2 r_{j_u}$;
        adjust exponent vector $e_{j_u} := e_{j_u} + 1$;
    od
    $i := i + 1$;
od
output $v$.
```

This algorithm is similar in outline to the algorithm from the proof of Theorem 3.1.2. In particular, that proof also shows that the algorithm here will terminate with v defined if there is a representation of m by f. Moreover, we have detailed there how to evaluate the `if`-query.

Let

$$f = f_0 \perp pf_1 \perp \dots p^\ell f_\ell$$

with $\det f_i \in \mathbb{Z}_p^*$. The rationale for the above procedure is as follows: If m has a representation by f with some variable belonging to f_0 coprime to p, then it is represented by f_0 only; this follows from reduction modulo p and Lemmata 3.1.1, 3.3.2. If, however, all values of variables of f_0 need to be divisible by p to represent m, then we may replace f_0 by $p^2 f_0$ and its variables x_j by px_j. This is what happens in the modification step after the `while`-loop. The correctness of the algorithm then follows inductively. □

Theorem 3.3.4 *Let p be a prime or $p = \infty$.*
Then **Trafo**$^{\mathbb{Z}_p}$ *is solvable in polynomial time.*

Proof : Let an instance (f, g) of the transformation problem be given. By Theorems 1.2.13, 1.2.15, and Proposition 1.2.9, both f and g can be efficiently transformed into normal form, i. e. we can compute $S, T \in \mathrm{GL}_n\mathbb{Z}_p$ such that

$$f' := f\,S \qquad \text{and} \qquad g' := g\,T$$

are in normal form. Since normal forms are unique (see the results just cited), we must have $f' = g'$. Thus $T^{-1}S$ is a transformation from f to g. □

As for the decisional problems in Sect. 3.1, similar results hold over the fields of p-adic numbers $\mathbb{Q}_p$. This is implicitly contained in the proof of Theorem 6.1.1.

Chapter 4

Algorithms for Primes, Classes, and Genera

In this chapter we present some auxiliary algorithms needed in the later chapters, particularly in Chaps. 9 and 10.

In Sect. 4.1, we give an overview over methods to select primes with certain properties. This constitutes an elementary but important algorithmic task for many algorithms in number theory. The polynomial-time primality test [AKS04] has simplified this significantly. In this thesis, prime selection is used in Sect. 4.2.4 and in Chapter 9. Much of the results presented here may be considered folklore; however, an overview over selection techniques in the light of recent primality tests seems to be missing in the literature. Hence this summary may be as well of independent interest.

In Sect. 4.2.4, we show that the classical result on the existence of genera with locally prescribed behavior can be made algorithmically efficient: We present an algorithm which constructs a form f over $\mathbb{Z}$ which is p-adically integrally equivalent to a finite set of given f_p over $\mathbb{Z}_p$. The given forms f_p have to be 'compatible' in the sense that such a global form f exists. This will be applied in the algorithms in Chaps. 9 and 10.

4.1 Algorithmic Prime Selection

In this section, we give an overview over prime selection algorithms. First we consider the problem of finding the least prime, or any prime, exceeding a given integer; then we restrict to a coprime arithmetic progression. It turns out that both types of tasks can be accomplished in random polynomial time. For the unrestricted case, derandomization is possible under the Riemann hypothesis, whereas in the case of an arithmetic progression, we cannot give an analogous derandomization.

Prime selection has been a topic of profound investigation, in particular in connection with cryptographic key generation (e. g. see [Mih94]). Before polynomial-time primality tests have been known (see [AKS04]), the stumbling block was to guarantee that the output numbers were primes. “Probable prime generators” became popular, which only claimed to output primes with a certain high probability. The “provable prime generators” had to produce a primality certificate for each prime.

The primality test [AKS04] has simplified this task extremely. The consequences for prime selection collected below can certainly be considered well-known. But as there seems to be no comprehensive study of them in the literature, we decided to give an overview here.

We consider this question here because we will apply it in the next sextion as well as in later chapters (see Sect. 9.2). Prime selection, however, is a very fundamental task both in cryptography and in algorithmic number theory, and therefore it seemed to be worthwhile to ponder this topic a bit more closely.

Proposition 4.1.1 *(a) For every $\varepsilon > 0$, there is a probabilistic polynomial-time algorithm which given an integer N, outputs some prime $p > N$ such that*

$$p = \mathcal{O}(N \ln^{1+\varepsilon} N).$$

The implied constant only depends on ε.

(b) Assume the Riemann Hypothesis to be true. Then there is deterministic polynomial-time algorithm which given an integer N, outputs the smallest prime $p > N$. It then holds that for every $\varepsilon > 0$ that

$$p = N + \mathcal{O}((\ln N)^2 (\ln\ln\ln N)^{\varepsilon}).$$

The implied constant only depends on ε.

Remark: To the best of my knowledge, there is, up to now, no deterministic and method known to produce a prime larger than a given number which can be proven unconditionally to be polynomial time.

Proof : Recall that by [AKS04], a given integer can be proven or disproven to be prime in deterministic polynomial time. We shall use this primality test as a subroutine in the following algorithms.

(a) Whithout loss we may assume that $\ln N \geq 2$. Then perform the following random selection algorithm:

```
success := false;
while not success do
    select uniformly at random $p \in [N+1, N \ln^{1+\varepsilon} N] \cap \mathbb{Z}$
    if $p$ is prime then success := true;
od
output $p$.
```

The probability of success in one iteration of this algorithm amounts to

$$r = \frac{\#\text{ primes in } [N+1, N\ln^{1+\varepsilon} N]}{N\,(\ln^{1+\varepsilon} N - 1)}. \tag{4.1}$$

Let $\pi(x)$ denote the number of primes less or equal to x. Then (4.1) implies

$$r \geq \frac{\pi(N\ln^{1+\varepsilon} N) - N}{N\,(\ln^{1+\varepsilon} N - 1)}.$$

If we assume without loss that $\varepsilon < 1$, the Prime Number Theorem then tells us that

$$\begin{aligned} r &\geq \frac{\dfrac{N\ln^{1+\varepsilon} N}{\ln N + (1+\varepsilon)\ln\ln N} - N + \mathcal{O}\left(\dfrac{N\ln^{1+\varepsilon} N}{(\ln N + (1+\varepsilon)\ln\ln N)^2}\right)}{N\ \ln^{1+\varepsilon} N} \\ &= \frac{1}{\ln N + (1+\varepsilon)\ln\ln N} - \frac{1}{\ln^{1+\varepsilon} N} + \mathcal{O}\left(\frac{1}{(\ln N + (1+\varepsilon)\ln\ln N)^{1+\varepsilon}}\right) \\ &= \frac{1}{\ln N} + \mathcal{O}\left(\frac{1}{\ln^{1+\varepsilon} N}\right). \end{aligned} \tag{4.2}$$

The last equality is due to the fact that

$$\frac{1}{\ln N} - \frac{1}{\ln N + (1+\varepsilon)\ln\ln N} = \mathcal{O}\left(\frac{1}{\ln^{1+\varepsilon} N}\right),$$

which follows from

$$\begin{aligned} \frac{1}{\ln N} - \frac{1}{\ln N + (1+\varepsilon)\ln\ln N} &= (1+\varepsilon)\,\frac{\ln\ln N}{(\ln N)\,(\ln N + (1+\varepsilon)(\ln\ln N))} \\ &= \mathcal{O}\left(\frac{\ln\ln N}{\ln^2 N}\right). \end{aligned}$$

Now turning to the general case of several iterations, we face a random experiment with respect to the geometric distribution, whose success probability is bounded from below by (4.2). Therefore, the probability that it takes more than $t\ln N$ iterations to obtain a prime does not exceed

$$\begin{aligned} \left(\frac{1}{\ln N} + \mathcal{O}\left(\frac{1}{\ln^{1+\varepsilon}}\right)\right)^{t\ln N} &= \frac{1}{(\ln N)^{t\ln N}} + \mathcal{O}\left(\frac{2^{t\ln N}}{(\ln N)^{t\ln N+\varepsilon}}\right) \\ &= \exp\big(-(t\ln N)(\ln\ln N)\big) \\ &\quad + \mathcal{O}\Big(\exp\big(t\ln N - t(1+\varepsilon)(\ln N)(\ln\ln N)\big)\Big). \end{aligned} \tag{4.3}$$

Here the first equality is due to the fact that for every constant $C > 0$ and every $1 \leq k \leq \lceil t\ln N\rceil$, it holds that

$$\frac{C^k}{(\ln N)^{t\ln N + k\varepsilon}} = \mathcal{O}\left(\frac{C}{(\ln N)^{t\ln N+\varepsilon}}\right), \tag{4.4}$$

and rounding $t \ln N$ to the next integer and applying the binomial theorem, we thus obtain

$$\sum_{i=1}^{\lceil t \ln N \rceil} \binom{\lceil t \ln N \rceil}{i} = 2^{\lceil t \ln N \rceil} - 1$$

terms dominated by the right hand side of (4.4), apart from the main term.

But it easily seen that there are constants C_1, C_2 such that the rightmost side of (4.3) is stricly smaller than $\frac{1}{2}$ if

$$t \geq C_1 - C_2 \, \varepsilon.$$

Summing up, we find that after linearly many iterations, the algorithm returns a prime with probability larger than $\frac{1}{2}$.

(b) Perform the following exhaustive search algorithm:

```
success := false;
$p := N$;
while not success do
    $p := p + 1$;
    if $p$ is prime then success := true;
od
output $p$.
```

By a result of Selberg [MSC96, §VII.13 a)], for any $\varepsilon > 0$ this procedure terminates successfully after at most $(\ln N)^2 (\ln \ln \ln N)^\varepsilon$ iterations conditionally on the Riemann Hypothesis, except for possibly finitely many exceptions; hence certainly it succeeds after

$$\mathcal{O}\left((\ln N)^2 (\ln \ln \ln N)^\varepsilon\right)$$

iterations, and the returned prime p satisfies

$$p = N + \mathcal{O}\left((\ln N)^2 (\ln \ln \ln N)^\varepsilon\right).$$

Remark. The choice of the factor $(\ln \ln \ln N)^\varepsilon$ was somewhat arbitrary as Selberg's theorem holds for arbitrary functions tending to infinity as $N \to \infty$ instead.

Corollary 4.1.2 *(a) For every $\varepsilon > 0$, there is a probabilistic algorithm which given integers N and n (n fix), outputs some primes $p_1, \ldots, p_n$ satisfying*

$$N < p_1 < \ldots < p_n = \mathcal{O}(N \ln^{1+\varepsilon} N)$$

in expected running time polynomial in n and $\log N$.

(b) Assume the Riemann Hypothesis to be true. Then there is deterministic polynomial-time algorithm which given integers N and n, outputs primes $p_1, \ldots, p_n$ such that p_1 is the smallest prime exceeding N, and p_i is the smallest prime exceeding p_{i-1}, for all $i = 2, \ldots, n$.

It then holds that for every $\varepsilon > 0$ that

$$p_n = N + \mathcal{O}(n\,(\ln N)^2\,(\ln\ln\ln N)^{\varepsilon}).$$

In either case, the implied constant only depends on ε.

Proof :

(a) Apply the algorithm of Proposition 4.1.1 a), but halt only when n distinct primes have been found.

Suppose that $k < n$ primes have already been found. Then the probability bound (4.2) that another prime is found in one more iteration changes by a summand

$$-\frac{k}{N \ln^{1+\varepsilon} - N} \geq -\frac{n}{N \ln^{1+\varepsilon} - N}.$$

As n is fixed, the $\mathcal{O}$-term of (4.2) absorbs this summand, and the procedure succeeds in polynomial time.

(b) Apply Proposition 4.1.1 b) itereratively n times.

□

Proposition 4.1.3 *For every $\varepsilon > 0$, there is a probabilistic polynomial-time algorithm which given $C \in \mathbb{N}$ and coprime positive integers q, a, outputs some prime $p \equiv a \bmod q$ such that*

$$p = \mathcal{O}(q \ln^{1+\varepsilon} q).$$

Remark. Even if the Riemann hypothesis is true, an exhaustive search algorithm as in the proof of Proposition 4.1.1 will not succeed in polynomial time because it is known that the least prime in an arithmetic progression is at least linear in the modulus in the worst case, i.e. there are exponentially many numbers to test. More precisely, [Erd49], [Pom80] give superlinear estimates in $\varphi(q)$, and [Pra61], presents a superlinear estimate in q, each for a large family of examples (see also [Wag79] for a heuristic argument why such examples occur so frequently). Theoretically, however, there might be an alternative deterministic approach which succeeds in polynomial time.

Proof : Use the following algorithm (which is anagolous to the one from the proof of Proposition 4.1.1 a):

```
success := false;
while not success do
    select uniformly at random $p \in [N+1, N\ln^{1+\varepsilon} N] \cap (q\mathbb{Z}+a)$
    if $p$ is prime then success := true;
od
output $p$.
```

If r is the probability of success in one iteration and if $\pi(x;q,a)$ denotes the number of primes $p \leq x$ with $p \equiv a \bmod q$, then

$$\begin{aligned} r &= \frac{\#\left\{p \in [N+1, N\ln^{1+\varepsilon} N] \mid p \text{ prime}, p \equiv a \bmod q\right\}}{N(\ln^{1+\varepsilon} N - 1)} \\ &\geq \frac{\pi(N\ln^{1+\varepsilon} N; q, a) - N}{N(\ln^{1+\varepsilon} N - 1}. \end{aligned} \tag{4.5}$$

The Bombieri-Vinogradov Theorem (see [MSC96, §VIII.6]), now states that

$$\pi(N\ln^{1+\varepsilon} N; q, a) = \frac{N\ln^{1+\varepsilon} N}{\varphi(q)\, ln(N\ln^{1+\varepsilon} N)} + \mathcal{O}\left(\frac{N\ln^{1+\varepsilon} N}{\varphi(q)\, ln^2(N\ln^{1+\varepsilon} N)}\right)$$

once $q \leq \frac{\sqrt{N}}{\ln^B N}$, for some fixed $B > 0$; here $\varphi(q)$ denotes Euler's totient function. Using (4.5) and $\varphi(q) < q$, it now follows that

$$r \geq \frac{1}{q\,\ln N} + \mathcal{O}\left(\frac{1}{q\,\ln^{1+\varepsilon} N}\right)$$

by the same computations as in the proof of Proposition 4.1.1 a). As then there are constants C_1, C_2 such that we perform $t\ln N$ independent iterations, where $t \geq C_1 - C_2\,\varepsilon$, then we obtain, as above, a chance of at least $\frac{1}{2}$ that we the algorithm succeeds within this time. □

Moreover, for primes as well as for squarefree numbers, it is known that the least a in the arithmetic progression with the desired properties may be exponentially large in the length of the modulus (see [MSC96, §§VI.23, VIII.5], the search for upper bounds is known as 'Linnick's question'). Therefore, the naïve exhaustive search approach of enumerating the integers in the progression in ascending order, testing for primes (or squarefrees, if possible), and halting when a number of the desired type has been found, is doomed to run exponentially long. Moreover, to my knowledge there is no essentially different deterministic approach known to attack this problem. Hence, it is not obvious at all whether, and if so, prime selection from an arithmetic progression can be efficiently derandomized.

Obviously, an iterative version of this proposition could be established analogously to Corollary 4.1.2.

4.2 Construction of Genera

In Sect. 1.2.5, we asked the following question: Given forms f_p over $\mathbb{Q}_p$ (for some set of symbols $p \in P$), under which conditions does there exist a rational form f such that $f \sim_{\mathbb{Q}_p} f_p$? We learned in Theorem 1.2.10 that there is essentially only one condition, namely, that the product of the Hasse-Minkowski invariants be 1, see (1.5). Now we turn to an analogous question: Let the f_p be p-adically integral. Then we ask when a form f over $\mathbb{Z}$ exists such that $f \sim_{\mathbb{Z}_p} f_p$; in other words, we ask whether the given f_p define a *genus* over $\mathbb{Z}$ (see Sect. 1.2.6). Interestingly, the condition is no stronger than in the rational case (Theorem 1.2.10).

Theorem 4.2.1 (Existence of Genera.) *Let $d \in \mathbb{Z}\backslash\{0\}$. Let f_p be forms over $\mathbb{Z}_p$ such that*

$$\det f_p \in d\mathbb{Z}_p^{*2},$$

for all $p|2d\infty$, and

$$\prod_{p|2d\infty} c_p(f_p) = 1.$$

Then there exists an integral form f of determinant d such that

$$f \sim_{\mathbb{Z}_p} f_p$$

for all p.

We are now going to make this theorem algorithmic: Namely, we will proof that given forms f_p as in Theorem 4.2.1, one can efficiently *construct* the global form f. The classical proof is constructive, and most of its steps can be turned into an algorithm directly. This is done in the subsequent subsections. The final result is recorded in Theorem 4.2.7.

4.2.1 Local representations

We first need a method to make p-adical representations p-adically integral at 'uninvolved' primes p.

Proposition 4.2.2 *Let $n \geq 3$, $k \in \mathbb{N}_0$, $m, d \in \mathbb{Z}$, and let $p \not| d$ be a prime. Then Algorithm 1 constructs $\beta \in \mathbb{N}_0$ and a primitive representation w of $p^{2\beta}m$ over $\mathbb{Z}_p$ by the form*

$$f := \langle\underbrace{1, \dots, 1}_{n-1}, d\rangle$$

to precision k in polynomial time.

Proof : We first explain the rough outline of Algorithm 1. We always refer to the form $f := \langle 1, \dots, 1, d\rangle$ over $\mathbb{Q}_p$ or $\mathbb{Z}_p$. In the first step, we construct an

Algorithm 1:

```
input:   $n, k \in \mathbb{N}_0$, $m, d, p \in \mathbb{Z}$ with $n \geq 3$, $p$ prime, and $p \nmid d$.
output:  $\beta \in \mathbb{N}_0$ and $w \in \mathbb{Z}^n$ such that $\sum_{i=1}^{n-1} w_i^2 + dw_n^2 \equiv p^{2\beta} m \bmod p^{k+1}$.
Abbreviate $f := \langle 1, \ldots, 1, d \rangle$.

1. Find an isotropic vector $u$:
     use Lemma 3.3.1 to find $u_1, u_2$ to precision $k$ s. t. $u_1^2 + u_2^2 = -d$
         and $\gcd_{\mathbb{Z}_p}(u_1, u_2)$,
     $u := (u_1, u_2, 0, \ldots, 0, 1)^t$.
2. Find a vector not orthogonal to $u$:
     $i := 1$,
     while $f(u, e_i) = 0 \bmod p$ do $i := i + 1$ od
3. Find an isotropic vector not orthogonal to $u$:
     Put $v := \frac{f(e_i)}{2f(u, e_i)} u - e_i$,
4. Find a representation $x$ of $m$ over $\mathbb{Q}_p$:
     Put $w := u + \frac{m}{2f(u, v)} v$.
5. Determine $\beta$:
     Put $\beta := -\nu_p(w)$.
6. Modify the representation:
     Let $w := p^\beta w$.
```

isotropic vector $u \in \mathbb{Z}_p^n$ for f. In step 2, we find another vector $e \in \mathbb{Z}_p^n$ such that $f(u,e) \neq 0$. This vector is used in step 3 to construct $v \in \mathbb{Q}_p^n$ satisfying $f(v) = 0$ and $f(u,v) \neq 0$. Then we have represented a 'hyperbolic plane' by f over $\mathbb{Q}_p$, i. e. a binary form with associated matrix

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

see below. This subform allows us to find a representataion of m over $\mathbb{Q}_p$ in step 4. In the last step, this representation and m are modified so that the representation will be p-adically integral and primitive.

Now we show that the algorithm is correct. Observe that steps 1 through 3 construct a hyperbolic plane primitively represented by the form

$$f := \langle 1, \ldots, 1, d \rangle.$$

By Lemma 3.3.1, step 1 can be accomplished because the form $\langle 1, 1 \rangle$ represents all $p \not| d$ over $\mathbb{Z}_p$ (see [Cas78, ch. 2, lm. 2.2]). Obviously, then u is isotropic for f.

Step 2 will succeed unless all standard unit vectors are orthogonal to u. By linearity, this implies that $f(u, \cdot)$ is identically zero. As f is regular (over $\mathbb{Q}_p$), this can only happen for $u = 0$ [Cas78, ch. 2, lm. 1.2], which is excluded by step 1.

Then step 3 yields an isotropic vector because

$$f(v) = \frac{f(e_i)^2}{4\, f(u,e_i)^2}\, f(u) \;+\; f(e_i) \;-\; 2\, f(u,e_i)\, \frac{f(e_i)}{2\, f(u,e_i)} = 0 + f(e_i) - f(e_i) = 0,$$

as $f(u) = 0$. Moreover, v is f-non-orthogonal to u, since

$$f(u,v) = \frac{f(e_i)}{2\, f(u,e_i)}\, f(u) \;-\; f(u,e_i) = -f(u,e_i)$$

by step 2 and since u is isotropic.

The vector x constructed in step 4 satisfies $f(x) = m$ because

$$f(x) = f(u) + \frac{m^2}{4f(u,v)^2} f(v) + 2\frac{m}{2f(u,v)} f(u,v) = m$$

as $f(u) = f(v) = 0$.

For steps 5 and 6, it is obvious that the updated $x' = p^\beta x$ is in $\mathbb{Z}_p^n$ and primitive by the choice of β. Finally, $f(x') = f(p^\beta x) = p^{2\beta} m$.

It remains to bound the running time of the Algorithm 1. Step 1 terminates in polynomial-time by Lemma 3.3.1. The `while`-loop in step 2 is repeated at most n times. The remaining steps employ $\mathcal{O}(1)$ evaluations of the quadratic form f and its associated bilinear form to precision k, which amounts to $\mathcal{O}(n^2)$ arithmetic operations modulo p^{k+1}, and $\mathcal{O}(1)$ extra arithmetic operations modulo p^{k+1}. □

4.2.2 Composition of square classes

Our next subgoal is to find an integer t primitively represented by given local forms f_p which are 'compatible' in the sense that they define a genus (see Theorem 4.2.1).

Lemma 4.2.3 *Let P be a finite set, containing primes and possibly the symbol ∞. Let $t_p \in \mathbb{Z}$ be given for all $p \in P$.*

Then Algorithm 2 constructs a prime p_0 and an integer t such that

(i) p_0 is the only prime with $p_0|t$ and $p_0 \notin P$; for this prime, $p_0|t$ and $p_0^2 \not| t$,

(ii) $t\mathbb{Z}_p^{*2} = t_p\mathbb{Z}_p^{*2}$ for all $p \in P$

in random polynomial time.

Denote by

$$\mathrm{CRT}\big((a_1, m_1), \ldots, (a_k, m_k)\big)$$

a call to an efficient routine realizing the Chinese Remainder Theorem. It returns an integer a such that

$$a \equiv a_i \bmod m_i \qquad \forall\, i,$$

or `fail` if the system is unsolvable. Note that this task can be performed in polynomial time.

Proof : For each $p \in P\backslash\{\infty\}$, let $t_p = \varepsilon_p p^{s_p}$. Algorithm 2 computes the values s_p, ε_p and takes care that for the output value t, it holds that

$$\nu_p(t) = s_p \qquad \text{and} \qquad \frac{t}{p^{s_p}}\mathbb{Z}_p^{*2} = \varepsilon_p\mathbb{Z}_p^{*2}. \tag{4.6}$$

The number t is computed in the final step by

$$t = \pm p_0 \prod_{\substack{p\in P\\ p\neq\infty}} p^{s_p},$$

which already implies statement (i). Then (4.6) is accomplished by the choice of the prime p_0:

By construction,

$$\frac{t}{p^{s_p}} \equiv \varepsilon_p \bmod p \text{ for } p \neq 2, \quad \text{and}$$
$$\frac{t}{2^{s_2}} \equiv \varepsilon_2 \bmod 8.$$

Then (4.6) follows from Lemma 1.2.6. Analogously, we have $\operatorname{sign}(t) = \operatorname{sign}(t_\infty)$ and therefore $t\mathbb{R}^+ = t_\infty\mathbb{R}^+$ if $\infty \in P$ (recall that $\mathbb{Z}_\infty^{*2} = \mathbb{R}^{*2} = \mathbb{R}^+$). Statement (ii) follows.

By Proposition 4.1.3, prime selection can be accomplished in random polynomial time. All other steps of the algorithm are trivially polynomial time. □

Algorithm 2:

```
 input:    finite set $P$ of primes and possibly $\infty$, $t_p \in \mathbb{Z}$ for all $p \in P$.
output:    prime $p_0$, $t \in \mathbb{Z}$ satisfying (i), (ii) of Lemma 4.2.3.

if $\infty \notin P$ then $t_\infty := 1$;
    $t_\infty := \mathrm{sign}\,(t_\infty)$;
fi
if $2 \in P$ then
    $s_2 := \nu_2(t_2)$,
    $\varepsilon_2 := t_2/2^{s_2} \bmod 8$,
fi
for $p \in P \setminus \{2\}$ do
    $s_p := \nu_p(t_p)$,
    $\varepsilon_p := t_p/p^{s_p} \bmod p$,
od
for $p \in P \setminus \{2\}$ do
    $M_p := t_\infty \prod_{q \in P \setminus \{\infty, p\}} q^{s_q} \bmod p$,
    $\bar{M}_p := M_p^{-1} \bmod p$
od
if $2 \in P$ then
    $M_2 := t_\infty \prod_{q \in P \setminus \{\infty, 2\}} q^{s_q} \bmod 8$,
    $M_0 := \mathrm{CRT}\left((\varepsilon_2 \bar{M}_2, 8), (\varepsilon_p \bar{M}_p, p) \,|\, p \in P \setminus \{\infty\}\right)$,
else
    $M_0 := \mathrm{CRT}\left((\varepsilon_p \bar{M}_p, p) \,|\, p \in P \setminus \{\infty\}\right)$,
fi
$Q := \prod_{p \in P \setminus \{\infty\}} p$,
if $2 \in P$ then $Q := 8 \cdot Q$ fi
use Proposition 4.1.3 to select a prime $p_0 \equiv M_0 \bmod Q$;
$t := p_0 t_\infty \prod_{\substack{p \in P \\ p \neq \infty}} p^{s_p}$;
output $p_0$, $t$;
```

Algorithm 3:

```
 input:   integer $d \neq 0$, forms $f_p$ $(p|2d\infty)$, satisfying (4.7), (4.8).
output:   integer $t$ satisfying (4.9).

     if $n = 1$ then output $t := d$; fi
     for all $p|2d\infty$ do
1.)      find $t_p \in \mathbb{Z}$ with $p \not| t_p$ and $f_p \xrightarrow{*}_{\mathbb{Z}_p} t_p$
     od
2.)  $t :=$ Algorithm 2$\Big(t_p \Big| p|2d\infty\Big)$,
3.)  if $n = 2$ then
         $(\beta, w) :=$ Algorithm 1 $(2, 0, t, d, p_0)$,
         $t := p^{2\beta} t$,
     fi
     output $t$ with its prime factorization.
```

Lemma 4.2.4 *Let $d \in \mathbb{Z}\backslash\{0\}$ and primitive n-ary quadratic forms f_p (over $\mathbb{Z}_p$) be given for $p|2d\infty$ such that*

$$(\det f_p)\mathbb{Z}_p^{*2} = d\mathbb{Z}_p^{*2} \tag{4.7}$$

for all $p|2d\infty$, and

$$\prod_{p|2d\infty} c_p(f_p) = 1. \tag{4.8}$$

Then Algorithm 3 constructs $t \in \mathbb{Z}$ and the prime factorization of t in random polynomial time such that

$$\begin{array}{rcll} f_p \xrightarrow{*}_{\mathbb{Z}_p} & t & \forall\, p|2d\infty, & \textit{and} \\ \langle 1, \ldots, 1, d\rangle \xrightarrow{*}_{\mathbb{Z}_p} & t & \forall\, p \not| 2d\infty. & \end{array} \tag{4.9}$$

<u>*Proof :*</u> We first have to explain how to execute Step 1: Scan the diagonal entries of the associated matrix of f (i.e. $f(e_i)$, $i = 1, \ldots, n$) for a value t'_p not divisible by p, and if the coefficients are not rational integers, take some $t_p \in \mathbb{Z}$ with $t_p \equiv t'_p \bmod p$. If none of the diagonal coefficients is coprime to p, transform f to diagonal shape using Lemma 1.2.3. Then some entry of the diagonal is coprime to p because f_p is primitive.

Moreover, note that the prime factorization of t can be easily attained because by Lemma 4.2.3, all but one prime factor of t being found among the input primes.

Now we have to verify that the output t satisfies the claim. If $n = 1$ this is trivial, so suppose that t is computed in Step 2. Then t belongs to the same square class in $\mathbb{Z}_p$ as t_p by Lemma 4.2.3 (ii), and so

$$f_p \xrightarrow{*}_{\mathbb{Z}_p} t \tag{4.10}$$

holds for all $p|2d\infty$ after step 2. If step 3 is executed, i. e. if $n = 2$, then then t is only modified by a square coprime to $2d$, since $p_0 \not| 2d$. Hence (4.10) still holds at the end of the algorithm.

Now let $p \not| 2d\infty$. If additionally $n \geq 3$, then $\gcd(t, d) = 1$ because we have chose $t \equiv t_p \not\equiv 0 \bmod p$ for all $p|2d$. Hence

$$\langle 1, \ldots, 1, d\rangle \xrightarrow{*}_{\mathbb{Z}_p} t \tag{4.11}$$

is also satisfied for all $p \not| 2d\infty$ due to Lemma 1.2.7.

Hence it remains to cover the case in which $n = 2$ and $p \not| 2d\infty$. Then f_p represents t primitively over $\mathbb{Z}_p$ if and only if

$$f_p \sim_{\mathbb{Z}_p} tx^2 + bxy + cy^2 \tag{4.12}$$

for some $b, c \in \mathbb{Z}_p$, as every primitive vector can be extended to a regular matrix. If $p \not| t$, then $\frac{d}{t} \in \mathbb{Z}_p$, and (4.12) holds with $b = 0$, $c = \frac{d}{t}$ because there is only one class of binary quadratic forms of determinant d over $\mathbb{Z}_p$ (see [Cas78, ch. 9, thm. 3.1]).

Now we are left with the case where $p|t$, but $p \not| 2d\infty$, and $n = 2$. By Lemma 4.2.3, we then have $p = p_0$. Moreover, in step 3 of Algorithm 3 the integer t is modified in such a way that t is primitively represented by f, according to Proposition 4.2.2.

Finally, the algorithm terminates in polynomial-time because the subroutines called do. □

Remark. For $n \geq 3$, it would suffice in Algorithm 3 to combine the t_p via the Chinese Remainder Theorem and ignore all $p \not| 2d\infty$. The t would still be primitively represented locally everywhere. However, it might then be harder to obtain the prime factorization of t.

4.2.3 Approximation

We prove a result (Corollary 4.2.6) which enables us to find a form f 'close' to each of a given set of f_p. Here 'close' refers to the topology of $\mathbb{Z}_p$; i. e. it means congruent to a high power of p (for p prime). This will help us in Theorem 4.2.7 to guarantee the $\mathbb{Z}_p$-equivalence of forms obtained from modifying f_p, f, respectively.

Because of its topological interpretation, this method is also called 'approximation'. Very roughly, the key lemma 4.2.5 allows for Chinese remaindering with constraints.

Recall that a vector $u = (u_i)_i \in R^n$, where R is $\mathbb{Z}$ or some $\mathbb{Z}_p$, is called primitive if and only if $\gcd_R(u_i \,|\, i = 1, \ldots, n) = 1$.

As before, we denote by

$$\mathrm{CRT}\big((a_1, m_1), \ldots, (a_k, m_k)\big)$$

a call to an efficient routine realizing the Chinese Remainder Theorem. The output integer a satisfies

$$a \equiv a_i \bmod m_i \qquad \forall\, i$$

if this system is solvable.

Lemma 4.2.5

(a) Let $m_p = (m_{ip})_i \in \mathbb{Z}_p^n$ be primitive vectors $(i = 1, \ldots, n)$ for p from a finite set of primes P. Moreover, let $k_p \in \mathbb{N}_0$ for each $p \in P$. Then Algorithm 4 constructs an integral primitive vector $m = (m_i)_i \in \mathbb{Z}^n$ satisfying

$$m_i \equiv m_{ip} \bmod p^{k_p} \qquad \forall\, p \in P \tag{4.13}$$

in polynomial time.

(b) Let $C_p \in SL_n\mathbb{Z}_p$ and let $k_p \in \mathbb{N}_0$ be given, for p from a finite set of primes P. Then Algorithm 5 constructs $C \in SL_n\mathbb{Z}$ satisfying

$$C_{ij} \equiv (C_p)_{ij} \bmod p^{k_p} \qquad \forall\, p \in P \tag{4.14}$$

in polynomial time.

Remark. The only non-obvious part in these statements are primitiveness of m in part a), and the property $\det C = 1$ in part b). Without these additions the statement would just be a special case of the Chinese Remainder Theorem.

Proof : First consider correctness of algorithms 4 (a). For each i, the integer m_i arises from the m_{ip} by Chinese remaindering, hence (4.13) follows immediately.

It remains to prove primitiveness of m. So let p be a prime dividing $\gamma := \gcd(m_i \mid i = 1, \ldots, n)$. Then $p \notin P$ as the m_p are primitive (and (4.13) holds). Moreover, p divides m_1, and hence μ. It follows by construction that

$$m_2 \equiv 1 \bmod p^{\nu_p(\gamma)},$$

a contradiction to $p|\gamma$. Hence m is primitive.

Algorithm 4:

input: finite set P of primes, $(k_p)_p \in \mathbb{N}_0^P$, and $(m_p | p \in P) \in \prod_{p \in P} \mathbb{Z}_p^n$
output: $m \in \mathbb{Z}^n$ satisfying (4.13)

```
$m_1 := \mathrm{CRT}\big((m_{1p}, p^{k_p}) \,|\, p \in P\big)$
find maximal $\mu \in \mathbb{Z}$ with $\mu | m_1$ and $p \nmid \mu \quad \forall\, p \in P$
$m_2 := \mathrm{CRT}\big((m_{1p}, p^{k_p}), (1, \mu) \,|\, p \in P\big)$
for $i = 3, \ldots, n$ do
    $m_i := \mathrm{CRT}\big((m_{ip}, p^{k_p}) \,|\, p \in P\big)$
od
output $m := (m_1, \ldots, m_n)^t$.
```

Algorithm 5:

input: finite set P of primes, $(k_p)_p \in \mathbb{N}_0^P$, and matrices $C^{(p)} \in \mathrm{SL}_n\mathbb{Z}_p$ for $p \in P$
output: $C \in \mathrm{SL}_n\mathbb{Z}$ satisfying (4.14)
Notation: $c_i^{(p)}, c_i$ for the colums of $C^{(p)}, C$ respectively

```
$C := I_n$ ### identity matrix
for $j = 1, \ldots, n$ do
    for $p \in P$ do
        find $\ell_j^{(p)} \in \mathbb{Z}_p$ such that $c_j^{(p)} = \sum_{i=1}^n \ell_i^{(p)} c_i$ (to precision $k_p$)
    od
    for $i = 1, \ldots, j-1$ do
        $\ell_i := \mathrm{CRT}\Big((\ell_j^{(p)}, p^{k_p+1}) \,|\, p \in P\Big)$
    od
    if $j < n$ then
        for $i = j, \ldots, n$ do
            $\ell_i :=$ Algorithm 4 $(\ell_i^{(p)} \,|\, p \in P)$
        od
    else $\ell_n := \ell_n^{(p_1)}$ ### first $p$ in $P$
    fi
    $c_j := \sum_{i=1}^n \ell_i c_i$
od
output $C$.
```

Starting from the identity matrix, Algorithm 5 modifies one column in one iteration of the `for`-loop. After the k-th iteration,

$$\begin{pmatrix} C_{11} \\ C_{21} \\ \vdots \\ \vdots \\ C_{n1} \end{pmatrix}, \begin{pmatrix} C_{22} \\ \vdots \\ \vdots \\ C_{n2} \end{pmatrix}, \ldots, \begin{pmatrix} C_{kk} \\ \vdots \\ C_{nk} \end{pmatrix},$$

are primitive vectors. Each column c_{kp} is represented as an integral linear combination of the columns of the current matrix. Then coefficients of the c_i with $i \geq k$ form primivite vectors for each p, and thus can be combined by Lemma 4.2.5 a). In particular, for k the last coefficient ℓ_{np} is necessarily 1 or -1 and independent from p. The remaining coefficients are approximated by Chinese Remaindering. A detailed correctness proof can be found in [Cas78, thm. 2.1 of ch. 9].

The running time is certainly polynomial as it only involves Chinese remaindering and solving n-dimensional linear equation systems (see the computation of the $\ell_j^{(p)}$ in Algorithm 5), apart from few extra arithmetic operations. Perhaps it should be mentioned that to to obtain polynomial time in the compution of μ in Algorithm 4, it suffices to perform a trial division $v := v/p$ as long as possible, starting from $v := m_1$. (This can of course be sped up by a binary search on the exponent.) □

Corollary 4.2.6 *Let P be a finite set of primes. Let f be an n-ary integral quadratic form. For each $p \in P$, let $k_p \in \mathbb{N}_0$, an n-ary quadratic form f_p over $\mathbb{Z}_p$, and $T_p \in SL_n\mathbb{Z}_p$ be given such that*

$$fT_p = f_p.$$

Then in polynomial time we can construct a form f' and a matrix $T \in SL_n\mathbb{Z}$ such that

$$f' = f\,T$$

and

$$f' = f_p \bmod p^{k_p} \qquad \forall\, p \in P.$$

Proof : Apply Algorithm 5 to the T_p. □

4.2.4 Main algorithm

Now we can conclude with the constructive version of Theorem 4.2.1 on the existence of genera.

Theorem 4.2.7 *Let $d \in \mathbb{Z}\backslash\{0\}$. Let forms f_p over $\mathbb{Z}_p$ be given for $p|2d\infty$, such that*

$$\det f_p \in d\mathbb{Z}_p^{*2}, \tag{4.15}$$

such that

$$\prod_{p|2d\infty} c_p(f_p) = 1. \tag{4.16}$$

Then Algorithm 6 constructs a form f over $\mathbb{Z}$, along with matrices $T_p \in SL_n\mathbb{Z}_p$ such that

$$f\,T_p = f_p,$$

in random polynomial time. The T_p are computed to a given precision k. In particular, it holds that

$$f \sim_{\mathbb{Z}_p} f_p$$

for all $p|2d\infty$.

Proof : We first outline the main ideas of the algorithm. The algorithm is recursive, decreasing the dimension of the forms at every recursive self-call.

For $n = 1$, we have $f_p = \langle d_p \rangle$ such that d/d_p is a square in $\mathbb{Z}_p$ (and $p \not| d/d_p$ for $p \neq \infty$), so they are all $\mathbb{Z}_p$-equivalent to the global form $\langle d \rangle$. Step 1 of Algorithm 6 exactly outputs this solution in this case. Note that the computation of the $1{\times}1$-matrices $(\sqrt{\frac{d_p}{d}})$ can be accomplished in random polynomial-time by [CP01, sec. 2.3.2].

Otherwise, if $n > 1$, we construct the first coefficient t of the desired form f in step 1. We can choose any integer primitively represented by all the f_p. This is done by application of Lemma 4.2.4. For each p, we compute a local representation of t by f_p via Lemma 3.3.1 and transform f_p such that f_p has t as first coefficient as well.

Then in step 3, we reduce to lower dimension by completing the square for each f_p. Algorithm 6 is called recursively on the $(n-1)$-ary complement forms after the completion, yielding a form f^* over $\mathbb{Z}$. Finally, from f^* and t we compute the solution f. This requires composing the coefficients b_{ip} of the square completions via Chinese remaindering. The exponents at the primes p for this CRT application have been defined in step 2.

Finally, the transformations between the f_p and the global form f can be computed using normal forms. This is done in step 4.

Let us turn to correctness. If $n = 1$, then the algorithm already terminates in step 1. We have just argued that the algorithm works correct in this case. Hence we may inductively assume that it runs correctly and efficiently in dimension $n - 1$. In particular, the recursive self-call in step 3 yields a global form f^* which is $\mathbb{Z}_p$-equivalent to f_p^*, for each $p|2d\infty$, if the f_p^* satisfy the 'compatibility' conditions(4.15), (4.16). This is verified in [Cas78, ch. 9, proof of lm. 5.1].

We have to verify that f constructed in step 3 is integral. As the nominator is a divisor of t, it suffices to show that f is p-adically integral for every $p|t$. Indeed,

$$f_p^* \equiv f*, \qquad b_{ip} \equiv b_i \mod p^{k_p}$$

Algorithm 6:

```
 input:    $d \in \mathbb{Z}\backslash\{0\}$, $k \in \mathbb{N}$, quadratic forms $(f_p \,|\, p|2d\infty)$ of dimension $n$ satis-
           fying (4.15) and (4.16).
output:    integral quadratic form $f$, matrices $(T_p \,|\, p|2d\infty)$
           to precision $k$ such that $T_p \in \mathrm{SL}_n\mathbb{Z}_p$ and $f\,T_p = f_p$.
Notation: For $p \not| 2d\infty$, let $f_p := \langle 1, \ldots, 1, d\rangle$.

1. Find suitable first coefficient:
compute $t \in \mathbb{Z}\backslash\{0\}$ such that $f_p \xrightarrow{*}_{\mathbb{Z}_p} t$ for all symbols $p$ via Lemma 4.2.4;
if $n = 1$ then
    for $p|2d\infty$ do
        compute $\tau_p := \sqrt{\frac{d_p}{d}}$ in $\mathbb{Z}_p$ to precision $k$;
    od
    output $\Big(d, \big((\tau_p) \mid p|2d\infty\big)\Big)$;
fi ;
$Q := \{$ prime divisors of $t\}$
for $p \in Q$ do
    use Lemma 3.3.1 to compute a primitive representation $u_p \in \mathbb{Z}_p^n$
        of $t$ by $f_p$;
od
for $p|2d\infty$ do
    find representation $f_p(u_p) = t$;
od
$P := \big\{p \,\big|\, p|2d\big\} \cup Q$;
for $p \in P \cup \{\infty\}$ do
    construct a matrix $U_p \in \mathrm{SL}_n\mathbb{Z}_p$ with $u_p$ as first column
    $f_p := f_p\, U_p$;
od

2. Determine the precision needed:
for $p \in P\backslash\{2\}$ do $k_p := \nu_p(t) + \nu_p(d) + 1$ od
$k_2 := \nu_2(d) + 3$;

3. Reduce to dimension $n - 1$:
for $p \in P$ do
    use Lemma 1.2.4 to obtain $b_{ip}$, $f_p^*$ over $\mathbb{Z}_p$
        such that $t\, f_p = (tx_1 + b_{2p}x_2 + \ldots + b_{np}x_n)^2 + f_p^*(x_2, \ldots, x_n)$
od
compute $f_\infty^*$ with $f \sim_{\mathbb{R}} \langle t\rangle \perp f_\infty^*$;
$\big(f^*,\ (T_p^* \,|\, p \in P \cup \{\infty\})\big) :=$ Algorithm 6$(f_p^* \,|\, p \in P \cup \{\infty\})$
### recursive call to self
apply Corollary 4.2.6 to $\big(f^*,\ (k_p,\ f_p^*,\ T_p^* \,|\, p \in P)\big)$ to obtain $\big(f^{*\prime}, ((T_p^*)' \,|\, p \in P)\big)$
    satisfying $f^{*\prime}\,(T_p^*)' = f_p^*$ and $f^{*\prime} \equiv f_p^*$ mod $p^{k_p}$ (all $p \in P$);
for $i = 2, \ldots, n$ do
    $b_i :=$ CRT $\big((b_{ip}, p^{k_p}) \,|\, p \in P\big)$
od
$f := t^{-1}\,\big((tx_1 + b_2x_2 + \ldots b_nx_n)^2 + f^*(x_2, \ldots, x_n)\big)$

4. Compute transformations:
for $p \in P \cup \{\infty\}$ do
    compute $T_p \in \mathrm{SL}_n\mathbb{Z}_p$ such that $f\,T_p = f_p$, for all $p \in P$ (to precision $k$)
od
output $(f, (T_p \,|\, p \in P \cup \{\infty\}))$
```

(for all $i = 2, \ldots, n$), which implies

$$\begin{aligned} &(tx_1 + b_2x_2 + \ldots + b_nx_n)^2 + f^*(x_2, \ldots, x_n) \equiv \\ \equiv\ &(tx_1 + b_2x_{2p} + \ldots + b_{np}x_n)^2 + f^*(x_2, \ldots, x_n) \quad = \ tf_p \bmod p^{k_p}. \end{aligned}$$

Hence f is p-adically integral since $k_p \geq \nu_p(t)$.

Let us now verify that f is $\mathbb{Z}_p$-equivalent to all the given f_p. By construction, it holds on completion of step 3 that

$$\begin{aligned} tf &= (tx_1 + b_2x_2 + \ldots + b_nx_n)^2 & &+f^*(x_2, \ldots, x_n) \\ &\equiv (tx_1 + b_{2p}x_2 + \ldots + b_{np}x_n)^2 & &+f_p^*(x_2, \ldots, x_n) = tf_p \bmod p^{k_p}. \end{aligned}$$

By definition of the k_p it follows that

$$\begin{aligned} f &\equiv f_p \quad \bmod p^{\nu_p(d)+1} \qquad \text{for } p \text{ odd, and} \\ f &\equiv f_2 \quad \bmod 2^{\nu_2(d)+1}. \end{aligned} \tag{4.17}$$

Then by cite[ch. 8, lm. 5.1]cas, $f \equiv_{Zz_p} f_p$ for all $p|2d$. Furthermore, for $\mathbb{Z}_\infty = \mathbb{R}$ it holds that

$$\begin{aligned} tf = (tx_1 + b_2x_2 + \ldots + b_nx_n)^2 & \quad +f^*(x_2, \ldots, x_n) \\ \sim_{\mathbb{R}} \langle 1 \rangle \qquad\qquad & \qquad\qquad \perp tf_\infty^*, \end{aligned}$$

whence

$$f \sim_{\mathbb{R}} \langle t \rangle \perp f_\infty^* \sim_{\mathbb{R}} f_\infty.$$

(The last equivalence is due to the choice of f_∞^* in step 3.) Thus we have shown that the output form f does lie in the desired genus.

Now consider the running time of this algorithm. The factorization in step 1 can be retrieved from the output of Algorithm 3, see Lemma 4.2.4, hence needs only constant extra effort. Moreover, the representations u_p of t, for $p|2d\infty$, can be efficiently computed: We only have to modify the representation of t_p chosen in Algorithm 3 by $\sqrt{t/t_p}$ in $\mathbb{Z}_p$. Hence we only have compute (approximate) this square root, which (for $p \neq \infty$) is accomplished by arithmetic in $\mathbb{F}_p$ and Hensel's lemma.

The transformations T_p in step 4 can be computed (to precision k) in two alternative ways: First, thanks to Theorem 3.1.4 we can construct transformations over $\mathbb{Z}_p$ by use of normal forms. Otherwise, departing from (4.17), we can as well compute a transformation modulo p and apply Hensel's lemma as many times as necessary, see the proof of [Cas78, ch. 8, lm. 5.1]. In either way, the computation takes only polynomial time.

Moreover, in every recursive call of the algorithm the dimension decreases by one. After these remarks it is obvious that it Algorithm 6 runs in polynomial time. □

Part II

Classification with Respect to Complexity

Chapter 5

Cases of Low Complexity

In this chapter, we analyze various properties of quadratic forms and their impact on the complexity of the representation and transformation problems. In each case we will find that the properties in question either direclty reduce complexity, or at least admit for polynomial-time reduction to smaller problems.

These findings serve as a confirmation that hard problems should be searched for among indefinite anisotropic forms, which are exactly those not occurring in this chapter; see Sect. 1.3.6. Therefore, the cryptographic applications of Chapter 2 have been proposed to use indefinite anisotropic forms.

Together with Chapters 6 and 7, this chapter forms the part of this thesis where we classify complexity of **Repr** and **Trafo** according to different 'sorts' of forms: In Chapter 6, this is done with respect to the base ring, and in 7 with respect to the dimension of forms. This and the mentioned chapters complement each other as forms with the properties studied here occur in every dimension, with the obvious exception of isotropic ternaries. Moreover, singular and reducible forms occur in every dimension over every base ring.

At first, in Sect. 5.1, we prove that the representation and transformation problems for singular forms reduce to lower-dimensional problems. Subsequently, in Sect. 5.2, the problems for reducible forms are reduced to factorization in the respective ring. These two sections are the only part of this theses where these two properties are not excluded, and they form the reason for this restriction. In Sect. 5.3, we explore complexity for definite forms, and finally in Sect. 5.4 for isotropic (ternary) forms.

The definitions of the properties studied here can be found in Sect. 1.2.

5.1 Singular Forms

Recall that a form f is called singular if $\det f = 0$. It is well-known (see [Gau89, art. 215] for forms over $\mathbb{Z}$, [Cas78, ch. 2, sec. 6], [O'M63, §42C] for forms over a field) that singular forms actually arise as a lower-dimensional form complemented by some linearly independent axes with value zero under the form (this is made precise in the first part of the proof of Proposition 5.1.1). This has the following algorithmic consequence. Recall the notion cPID from Sect. 1.3.1.

Proposition 5.1.1 *Let R be a cPID. Denote by $\mathcal{S}$ the property of being singular for quadratic form, $n \in \mathbb{N}_0$. Then for $n \geq 1$,*

(a) $\mathbf{Repr}_n(\mathcal{S}) \preccurlyeq \mathbf{Repr}_{n-1}$,

(b) $\mathbf{Trafo}_n(\mathcal{S}) \preccurlyeq \bigsqcup_{i=1}^{n-1} \mathbf{Trafo}_i$.

In both cases, one oracle call suffices.

Proof : We first have to construct efficiently the mentioned split of a singular form into a lower-dimensional form and zero. This can be done as follows: Let f be a singular quadratic form of dimension n over R with associated matrix A. Construct a vector $v \in R^n \backslash \{0\}$ with $A\,v = 0$ using linear algebra in the quotient field of R; this is possible because $\det A = 0$. Divide v by its content (the gcd of its coefficients) to obtain a primitive vector v'; i. e. we can efficiently construct a basis of R^n with v' as last vector. Denote the matrix of such a change of bases by T.

We have $Av = 0$, therefore $Av' = 0$ and thence $u^t Av' = 0$ for all $u \in R^n$. Thus the form $f' = f\,T$ has an associated matrix of the shape

$$A' = \begin{pmatrix} A_0 & 0 \\ 0^t & 0 \end{pmatrix}$$

with $A_0 \in R^{(n-1)\times(n-1)}$. Obviously, A_0 is symmetric, so it defines a quadratic form f_0 over R. This is the desired decomposition. We can now turn to the two statements of the proposition.

(a) Fix the above notation and let $m \in R$ such that $f \xrightarrow{*}_R m$. If $m = 0$, then return v', and we are done. Otherwise, consider an arbitrary $u \in R^n$ and compute

$$f'(u) = u^t \begin{pmatrix} A_0 & 0 \\ 0^t & 0 \end{pmatrix} u = u'^t\, A_0\, u',$$

where $u' = (u_1, \ldots, u_{n-1})^t$. Hence, to solve the representation problem, we proceed as follows:

```
input:    $n, f, m$ as above.

Compute the form $f_0$ as above;
ask $\mathbf{Repr}_{n-1}$-oracle for a primitive $u' \in R^{n-1}$ with $f_0(u') = m$;
output $T^{-1}\binom{u'}{0}$.
```

Obviously, this establishes a polynomial-time oracle algorithm with exactly one oracle call. Its correctness is seen from the above discussion.

(b) Without loss of generality we may assume that neither f nor g is identically zero. Let be an instance (f, g) of the transformation problem, we can efficiently find $T, U \in \mathrm{GL}_n R$ such that

$$\begin{aligned} f\,T \text{ has associated matrix } & \begin{pmatrix} A_0 & 0 \\ \underbrace{0}_{k} & \underbrace{0}_{n-k} \end{pmatrix}, \\ g\,U \text{ has associated matrix } & \begin{pmatrix} B_0 & 0 \\ \underbrace{0}_{k'} & \underbrace{0}_{n-k'} \end{pmatrix}, \end{aligned} \tag{5.1}$$

where $k, k' < n$, and A_0, B_0 are quadratic with $\det A_0, \det B_0 \neq 0$; we call the quadratic forms they are associated to f_0, g_0. Note that obviously, if f, g are equivalent, k and k' must coincide.

```
input:    equivalent singular $n$-ary forms $f$, $g$
compute $T, U, f_0, g_0$ as above
$S_0 :=$ oracle output on $(f_0, g_0)$
output $T^{-1} \begin{pmatrix} S_0 & 0 \\ \underbrace{0}_{k} & I_{n-k} \end{pmatrix} U$
```

It is evident that this oracle algorithm runs in polynomial-time and uses only one oracle call, the procedure for the first two steps being discussed above. Moreover, it is clear that the output matrix is contained in $\mathrm{GL}_n R$ if the oracle gives a valid answer, that is, if the oracle is presented a legitimate instance of $\mathbf{Trafo}_k^R$, for some $k < n$.

It remains to prove correctness. Let $S \in \mathrm{GL}_n R$ satisfy $f\,S = g$, then $S' := T^{-1} S U$ satisfies $f'\,S' = g'$, and thus

$$\begin{pmatrix} B_0 & 0 \\ 0^t & 0 \end{pmatrix} = S'^t \begin{pmatrix} A_0 & 0 \\ 0^t & 0 \end{pmatrix} S' = \begin{pmatrix} S_0^t A_0 S_0 & S_0^t A_0 u \\ u^t A_0 S_0 & u^t A_0 u \end{pmatrix}, \tag{5.2}$$

where

$$S' = \begin{pmatrix} S_0 & u \\ v^t & c \end{pmatrix} \tag{5.3}$$

with $S_0 \in R^{k\times k}$, $u, v \in R^{k\times(n-k)}$, and $c \in R^{k\times k}$. Hence (5.2) holds if and only if $S_0^t A_0 S_0 = B_0$ and $A_0 u = 0$. But since $\det A_0 \neq 0$, the only solution to the latter matrix equation is $u = 0 \in R^{k\times(n-k)}$. By (5.3), it follows that

$$(\det c)\,(\det S_0) = \det S' \in R^*,$$

hence $S_0 \in \mathrm{GL}_k R$.

Therefore, f_0 and g_0 are R-equivalent, and therefore the oracle call contains a valid $\mathbf{Trafo}_k^R$-instance. Moreover, if $f_0\, S_0 = g_0$, then the matrix

$$S' := \begin{pmatrix} S_0 & 0 \\ 0^t & I_{n-k} \end{pmatrix}$$

satisfies $f'\, S' = g'$. Therefore, the output matrix solves the original transformation problem.

□

5.2 Reducible Forms

Recall from Sect. 1.2.3 that a quadratic form over R is reducible if it factors into two linear polynomials in $R[x]$. Similar to singular forms, reducible forms constitute a somewhat degenerate case of quadratic forms. We will prove that reducibility results in diminished complexity of the computational problems: We show that the transformation problem can be solved in polynomial time, whereas to solve the representation problem efficiently, we have to know the factorization of the $m \in R$ to be represented into irreducible elements of R. For the most important case $R = \mathbb{Z}$, factorization is not known, and unlikely, to be feasible. Nevertheless this should be viewed as a case of low complexity because

(a) though no efficient algorithm is known for it, factorization can be accomplished in subexponential time for rings as $\mathbb{Z}$, while we are mainly interested in problems of exponential complexity (see Chapter 9)

(b) for other rings, factorization is trivial (e. g. fields), or solvable in (probabilistic) polynomial-time using existing algorithms (e. g. $\mathbb{F}_q[x]$, see [Coh93, sec. 3.4]; these algorithms either use probabilism or are conditional to the Extended Riemann Hypothesis),

(c) it is only the element to be represented which has to be factored, the form in question does not add significantly to the complexity of the problem. If we were to use this in a cryptographic primitive, we would have to ensure that m can be chosen hard to factor.

Lemma 5.2.1 *Let R be a cPID. If f is a reducible quadratic form over R, then linear forms ℓ_1, ℓ_2 such that*

$$f = \ell_1\, \ell_2 \tag{5.4}$$

can be constructed in polynomial time; in particular, the sizes of their coefficients are polynomial in the inputs.

Moreover, this decomposition is unique up to conversion of the indices and scalars; more precisely, if

$$\ell_1(x)\ell_2(x) = \ell_1'(x)\ell_2'(x) \tag{5.5}$$

with linear forms ℓ_i, ℓ_i' over R, then there are $a, b \in R\backslash\{0\}$ and a permutation $\pi \in \{id, (12)\} = \mathfrak{S}_2$ such that

$$a\,\ell_1(x) = b\,\ell_{1\pi}'(x) \qquad \textit{and} \qquad b\,\ell_2(x) = a\,\ell_{2\pi}'(x).$$

Proof : Let $(c_{ij})_{i,j=1}^n$ be the associated matrix of the reducible quadratic form f. Then f decomposes as in (5.4) into linear forms

$$\ell_1(x) = \sum_{i=1}^{n} p_i x_i \qquad \text{and} \qquad \ell_2(x) = \sum_{i=1}^{n} q_i x_i$$

if and only if

$$c_{ij} = a_i b_j \qquad \text{for all } i, j = 1, \ldots, n. \tag{5.6}$$

Denote by K the quotient field of R. Then (5.6) is solvable for p_i, q_j with $p_1 \neq 0$ over K if and only if it is solvable with $p_1 = 1$. But $p_1 = 1$ already enforces $q_j = c_{1j}$ for all $j = 1, \ldots, n$, and this yields $p_i = \frac{c_{i1}}{q_1}$ for all $i = 2, \ldots, n$, so that the linear forms are already determined. If, however, (5.6) is solvable for p_i, q_j over K only with $p_1 = 0$, then the equation system has become smaller, and the same consideration applies to p_2. Note that it can be efficiently tested whether a candidate for a solution in fact solves (5.6), and that there are at most $2n$ cases to be considered here.

Hence we can efficiently construct a decomposition (5.5) over K. Furthermore, as R is a cPID, we can efficiently compute the least common denominator p of the p_i, and the least common denominator q of the q_i, for $i = 1, \ldots, n$ in both cases. This leads to the equation

$$pq\, f = (p\ell_1)\,(q\ell_2),$$

where the brackets on the left hand side are linear forms with coefficients in R. By construction, $p\ell_1$ is not divisible by p in $R[x]$, and since R is a UFD, we necessarily have that

$$p \mid (q\ell_2)$$

in $R[x]$. Analogously, p must divide $p\ell_1$ in $R[x]$, hence

$$\frac{p}{q}\,\ell_1, \quad \frac{q}{p}\,\ell_2 \in R[x]$$

and thus we have a decomposition

$$f = \left(\frac{p}{q}\,\ell_1\right)\,\left(\frac{q}{p}\,\ell_2\right)$$

over R.

To prove uniqueness, again consider (5.5) as an equation in $K[x]$, which is a unique factorization domain. Hence up to reversion of the indices, the linear factors differ only by an element of K, i. e. there is $k \in K^*$ and a permutation π such that

$$\ell_1(x) = k\,\ell_1'(x) \qquad \text{and} \qquad \ell_2(x) = k^{-1}\,\ell_2'(x),$$

which in turn implies the statement. □

Next we define the factorization problem for arbitrary UFDs. Recall that $p \in R$ is *irreducible* if $p = ab$ with $a, b \in R$ implies $a \in R^*$ or $b \in R^*$.

> **Fact**$_R$ **Factorization problem over** R
> *PARAMETER:* UFD R with encoding.
> *INPUT:* $m \in R$.
> *OUTPUT:* If $m \neq 0$ and $m \notin R^*$, output $r \in \mathbb{N}$, irreducible elements $p_i \in R$, and $e_i \in \mathbb{N}$, such that $m = \prod_{i=1}^{r} p_i^{e_i}$.

In particular, the existence of a polynomial-time algorithm for **Fact**$_R$ implies that every m has a factorization where the sum of the encoding lengths of the p_i and e_i is polynomial in $\texttt{length}\,(m)$; in particular, we must have

$$\omega(m) = \mathcal{O}(\texttt{length}\,(m)^c) \tag{5.7}$$

for some $c > 0$ (for $R = \mathbb{Z}$, this condition holds with $k = 1$). By our convention on oracles (see Sect. 1.1.4), any statement referring to a **Fact**^{R}-oracle is vacuously true if R fails to satisfy (5.7).

Moreover, recall that elements θ, η of a commutative ring R are called *associated* if there is a unit $\varepsilon \in R^*$ such that $\theta = \varepsilon\eta$. Obviously, this establishes an equivalence relation.

We can now formulate our result.

Proposition 5.2.2 *Let R be a cPID. Denote by $\mathcal{R}$ the property of being reducible for a quadratic form. Then:*

(a) **Trafo**$^R(\mathcal{R})$ *is solvable in polynomial time.*

(b) If R allows a polynomial-time algorithm which given a positive integer C and a ring element $\theta \in R$, enumerates all elements $\eta \in R$ associated to θ with $\texttt{length}\,(\eta) \leq C$, then

$$\textbf{Repr}^R(\mathcal{R}) \preccurlyeq_1 \textbf{Fact}_R.$$

Before turning to the proof let us discuss the additional condition of part (b), which stipulates efficient enumeration of associates. In particular, this requires the number of units is polynomially bounded. Obviously, cPIDs with only finitely many units fulfill this property, as the rational integers $\mathbb{Z}$, the nine rings of integers of imaginary quadratic fields (see Sect. 1.3.1), or the polynomial ring in one variable over a finite field. Moreover, the criterion applies to rings whose group of units possesses a finite set of generators from which the elements g_i, $i = 1, \ldots, r$ of infinite order satisfy the condition

$$\texttt{length}\left(\prod_i g_i^{e_i}\right) \geq \sum_i e_i \texttt{length}\,(g_i)\,.$$

This criterion may be especially useful for the rings of integers of algebraic number fields.

It should also be noted that efficient enumeration of associates implies that there are only polynomially many for each $\theta \in R$. But I am not aware of any ring in natural encoding where this condition is violated.

<u>Proof :</u>

(a) Let f, g two equivalent reducible forms. By Lemma 5.2.1, we can efficiently factorize

$$f = \ell_1\,\ell_2 \qquad \text{and} \qquad g = \ell_1'\,\ell_2'.$$

Let $T \in \mathrm{GL}_n R$ satisfy $f\,T = g$, which here means that

$$(\ell_1\,T)\,(\ell_2\,T) = \ell_1'\,\ell_2',$$

where $(\ell_i\,T)(x) := \ell_i(Tx)$. The $\ell_i\,T$ are linear forms over R again; hence Lemma 5.2.1 implies that there are $a, b \in R\backslash\{0\}$ such that, without loss of generality,

$$a\,\ell_1\,T = b\ell_1' \qquad \text{and} \qquad b\,\ell_2\,T = a\ell_2'. \tag{5.8}$$

But for each a, b, this is just a system of $2n$ linear equations in n^2 unknowns. By hypothesis, it is solvable, and it can be efficiently solved over the quotient field K of R with a, b as symbolic parameters. Finally, it can be checked by computing least common denominators over R how the arising solutions can be lifted to R.

(b) For multiindices $d = (d_1, \ldots, d_r)$, $e = (e_1, \ldots, e_r) \in \mathbb{N}_0^r$ let us denote by $d \leq e$ componentwise majorization of d by e, $d_i \leq e_i$ for all i.

By Lemma 5.2.1, there is a polynomial P such that the lenghts of the coefficients of the factors ℓ_1, ℓ_2 of f are bounded by $P(\texttt{length}\,(f))$. Consequently, if $u \in R^n$ satisfies $u \leq K$, then the sizes of ℓ_1, ℓ_2 are bounded by $Q(\texttt{length}\,(f) + K)$ for some polynomial Q.

To hande the representation problem, perform the following algorithm:

```
input:    reducible quadratic form $f$ over $R$ and $m \in R$.

decompose $f = \ell_1\,\ell_2$ over $R$ as in Lemma 5.2.1;
factorize $m = \prod_{i=1}^r p_i^{e_i}$ with $e_i \in \mathbb{N}$, $p_i$ pairwise non-associated primes in $R$;
success := false; $J := 1$;
while not success do
    $K := \left\lceil J\,(\texttt{length}\,(f) + \texttt{length}\,(m))^J \right\rceil$;
    for $\alpha \in \mathbb{N}_0^r$, $\alpha \leq e$ do
        put $\eta := \prod_{i=1}^r q_i^{\alpha_i}$;
        for $\theta \in R$ associated to $\eta$ s. t. $\texttt{length}\,(\theta) \leq Q(\texttt{length}\,(f) + K)$ do
            if $\ell_1(u) = \theta$ is not solvable for $u$ in $R^n$ then next; fi ;
            compute $u^{(0)}, u^{(1)}, \ldots, u^{(n-1)} \in R^n$ s.t. $u = u^{(0)} + \sum_{i=1}^{n-1} \lambda_i u^{(i)}$,
                with $\lambda_i \in R$, is the general solution of $\ell_1(u) = \theta$;
            if $\frac{m}{\delta} = \ell_2(u^{(0)}) + \sum_{i=1}^{n-1} \lambda_i \ell_2(u^{(i)})$ unsolvable for $\lambda_i \in R$ then
                next;
            else
                fix a solution $\lambda_i$;
                success := true;
            fi ;
            if success then fi ;
        od ;
        **if** success **then fi** ;
    **od** ;
**if** success **then fi** ;
**od** ;
**output** $u := u^{(0)} + \sum_{i=1}^{n-1} \lambda_i u^{(i)}$.
```

Obviously this is a polynomial-time oracle algorithm with exactly one **Fact**-oracle call. To prove its correctness, note that in one iteration of the **while** loop, δ ranges exactly through all divisors of m of encoding length at most $Q(\texttt{length}\,(f) + K)$ (for the current value of K). If

$$m = f(u) = \ell_1(u)\,\ell_2(u),$$

then obviously $\delta := \ell_1(u)$ divides m in R, and

$$\ell_1(x) = \delta \qquad \text{and} \qquad \ell_2(x) = \frac{m}{\delta}. \tag{5.9}$$

Moreover, as was discussed before the algorithm, if there is a solution u to the representation problem with $\texttt{length}\,(u) \leq K$, then the encoding length of δ from (5.9) is bounded by $Q(\texttt{length}\,(f) + K)$ and thus occurs in the enumeration.

Finally note that via the **while** loop, K attains any polynomial bound, and if there is a polynomial sized solution u, then the algorithm computes one after a constant number of **while** loop iterations.

□

5.3 Definite Forms

In this section, we report on the fact that for definite forms in fixed dimension, the transformation problem can be solved in polynomial time. The representation problem, by contrast, stays hard, which can be seen from Theorem 9.1.6 and its discussion. The efficient solution of the transformation problem, however, already rules out the cryptographic applications from Chapter 2 for definite forms.

The complexity of both the algorithm presented here as well as more recent improvements depends heavily (i. e. more than simply exponentially) on n. It is a vivid line of research to further reduce the running time (whithin exponential bounds) in order to gradually increase the maximal dimension where such algorithms are still applicable in practice. But as we are primarily interested in very *low* fixed dimensions, mostly $n \in \{3, 4\}$, the exact dependence on n does not make a difference for our applications; in addition to be polynomial-time in theory, the algorithms cited here run quite fast in practice still for higher dimensions than 3,4 so that definite forms have to be ruled out for the schemes discussed in Chapter 2.

The first and essentially complete treatment of the problem to compute a transformation between given definite forms is given in [PP85] by Plesken and Pohst. A discussion of this algorithm is also given in [PS85], where a major improvement, the fingerprint, is introduced. These publications are mainly concerned with the construction of nontrivial automorphisms of lattices, but essentially the same algorithms also work for transformations between different equivalent lattices within the same time bounds.

These algorithms use as a subroutine the construction of a vector representing the minimum of the form (a shortest vector), of which the first procedure was given by Kannan [Kan87]; see [SH07] for current progressions in this field.

Together, these results show:

Theorem 5.3.1 (Plesken/Pohst/Souvignier) *Let $n \in \mathbb{N}$ be fixed and denote by $\mathcal{D}$ definiteness of a quadratic form. Then* $\mathbf{Trafo}_n(\mathcal{D})$ *is solvable in polynomial time.*

5.4 Isotropic Ternary Forms

Recall that a quadratic form is called isotropic if it represents zero. The Legendre equation

$$ax^2 + by^2 + cz^2 = 0, \qquad (x, y, z) \neq 0$$

i. e. the problem of finding an isotropic vector for a ternary diagonal form, and the question for a criterion for the isotropy of the form $\langle a, b, c\rangle$ have played an important role in the history of the theory of quadratic forms.

As the results cited in this section show, the problems on isotropic three-dimensional forms are closely related to integer factorization with respect to complexity. This seems to be in striking contrast to anisotropic ternary and quaternary forms; cf. Sect. 7.1.3 and Chapter 9.

It should be noted that basic techniques of this section extend to higher-dimensional isotropic forms; however, they do not lead to complexity-theoretic implications analogous to Theorem 5.4.3 beyond dimension three.

By convention, if a rational number u is written as $u = \prod_i p_i^{e_i}$ with p_i distinct primes and $e_i \in \mathbb{Z}$, we will call this decomposition the factorization of u.

In [Sim05a], Simon presents an algorithm which efficiently computes an isotropic vector of an isotropic rational quadratic form once the factorization of its determinant is given. For future reference, we write down a slight variant of his result adapted to our needs.

Theorem 5.4.1 (Simon) *There is a polynomial-time algorithm which given as input a rational isotropic quadratic form f of arbitrary dimension along with the factorization of* $\det f$*, constructs an integral isotropic vector for f.*

Proof : Let $k \in \mathbb{N}$ be such that $k\,f$ is an integral quadratic form. If k is chosen minimal, then obviously the knowledge of the factorization of $\det f$ also yields the factorization of $\det(kf)$. Moreover, a vector is isotropic for f if and only if it is isotropic for $k\,f$.

Hence, we may apply Algorithm 7 from [Sim05a] which constructs an isotropic vector as desired. □

Denote (temporarily) by **Iso** the problem of constructing an isotropic vector of an input form. The problem of constructing an isotropic vector of an input form is denoted **Iso** (see Sect. 1.3.3), where as usual, the missing ring subscript is interpreted as referring to $R = \mathbb{Z}$. Thus Simon's algorithm bounds the complexity of **Iso** from above by the complexity of factoring a rational number, or, which is equivalent, integer factorization. We now turn to the opposite argument, namely, a lower complexity bound for **Iso**, given by the problem of extracting modular square roots. But as there seems to be no essentially better way to extract square roots than to factor the module, and two essentially different square roots yield a non-trivial factor of the module, the upper and lower bounds we give on **Iso** can be considered close.

Sqrt Square root problem
INPUT: $N \in \mathbb{N}$ such that $-1 \in (\mathbb{Z}/N\mathbb{Z})^{*2}$.
OUTPUT: $s \in \mathbb{Z}$ such that $s^2 \equiv -1 \bmod N$.

Note that the condition on N can be rephrased as follows: Every odd prime factor p of N satisfies $p \equiv 1 \bmod 4$, and $4 \not| N$.

Theorem 5.4.2 (Schnorr) *There is a reduction* **Iso** $\succcurlyeq_1$ **Sqrt**.

More precisely, if $-1 \in (\mathbb{Z}/N\mathbb{Z})^{*2}$, then the following problems are polynomial-time equivalent:

1. *compute a (nontrivial) isotropic vector of $\langle 1, 1, -N\rangle$,*
2. *find a square root of -1 modulo N,*
3. *represent N by the definite binary form $\langle 1, 1\rangle$, i. e. as a sum of two squares.*

Proof : See [Sch04a]. □

In the same paper it is shown, that

Theorem 5.4.3 (Schnorr)

(a) *Let f be an isotropic ternary form representing $m \in \mathbb{Z}$. If an isotropic vector is given, then a representation $f(v) = m$ can be computed in polynomial time.*

(b) *Let f, g be equivalent isotropic ternary forms. If an isotropic vector is given for each of the two forms, then a transformation $f\,S = g$ can be computed in polynomial time.*

Thus, in effect, both representation and transformation problems are at most as hard as constructing an isotropic vector. Thus, the complexity of both problems is close to that of factoring.

Chapter 6

The Impact of the Base Ring on Complexity

In this chapter, we will study the transformation and representation problems over several rings different from the integers. We have already started whith these investigations in Sections 3.2 and 3.3, concerning finite prime fields and the rings of p-adic integers respectively.

We show that over the rational numbers, the complexity of the transformation problem is closely related to factoring (as detailed in Sect. 6.1), and therefore it can be solved in subexponential time.

Subsequently, we consider the representation problem over rings of formal power series and polynomials. We analyse the question how closely the complexity of our problems is related to the complexity over the base field. In particular, for power series the problems are

6.1 Forms over the Rational Numbers

In this section, we will show that the transformation problem over the rationals is essentially equivalent to factoring the determinants of the forms involved. More precisely, we show that $\mathbf{Trafo}^{\mathbb{Q}}$ is not harder than factoring, but at the same time not easier that extracting square roots of -1 modulo some divisor of the determinant. This problem can be considered 'almost equivalent' to factoring because two essentially different square roots yield a non-trivial factor of the modulus, and because there seems to be no essentially better way to extract square roots than to factor the modulus.

Recall the definition of **Fact**, the factorization problem, from Sect. 5.2.

Let p, $q \equiv 1 \bmod 4$ be distinct primes, and let $N := pq$. Let us call such N *counter-Blum integers* (in analogy to the Blum integers $N' = p'q'$ with $p', q' \equiv 3 \bmod 4$). Recall that then the equation

$$x^2 \equiv -1 \bmod N \tag{6.1}$$

is solvable for every counter-Blum integer N. The integer factorization problem is not likely to become significantly easier if restricted to such numbers; compare it to the well-known hardness hypothesis for Blum integers [BBS86].

We will consider the problem of computing a solution x of (6.1). Note that we have considered a slightly more general problem in Sect. 5.4.

Imag Imaginary root problem
INPUT: $N \in \mathbb{N}$ such that $N = pq$, with p, q primes $\equiv 1 \bmod 4$.
OUTPUT: $s \in \mathbb{Z}$ such that $s^2 \equiv -1 \bmod N$.

Theorem 6.1.1

(a) *For $n, s \in \mathbb{N}$, denote by $\mathcal{I}$ the properties f classically integral, binary, indefinite, and anisotropic for a quadratic form f. Then*

$$\textbf{Imag} \preceq_1 \textbf{Trafo}^{\mathbb{Q}}(\mathcal{I}).$$

(b)

$$\textbf{Trafo}^{\mathbb{Q}} \preccurlyeq_1 \textbf{Fact}_{\mathbb{Z}}.$$

For an instance (f, g) of $\textbf{Trafo}^{\mathbb{Q}}$*, it suffices to call the oracle to factor* $(\det f)(\det g)$.

Proof :

(a) Let N be an instance of **Imag**, i. e. $N = pq$, with prime factors $p, q \equiv 1 \bmod 4$. Define

$$f := \langle 1, -N\rangle, \qquad g := \langle -1, N\rangle.$$

Obviously, these are indefinite, and they are anisotropic since N is not a perfect square, see [Cas78, ch. 4, lm. 2.4]. We claim that f and g are $\mathbb{Q}$-equivalent. Obviously, f and g are equivalent over the reals, as both are $\mathbb{R}$-equivalent to the form $\langle 1, -1\rangle$. Moreover, we can compute Hasse-Minkowski invariants as follows. Recall the rules for the norm residue symbols from Sect. 1.2.5. Let $N = pq$. Then

$$c_p(f) = \left(\frac{1,\,-N}{p}\right) = 1,$$

and

$$c_p(g) = \left(\frac{-1,\,N}{p}\right) = \left(\frac{-1,\,p}{p}\right)\underbrace{\left(\frac{-1,\,q}{p}\right)}_{=1} = \left(\frac{-1}{p}\right) = 1.$$

An analogous computation works for q instead of p. Hence it follows from the Hasse Principle 1.2.11 that $f \sim_{\mathbb{Q}} g$.

Ask the oracle for $S = (s_{ij}) \in \mathrm{GL}_2\mathbb{Q}$ satisfying $f\,S = g$, i. e.

$$S^t \begin{pmatrix} 1 & \\ & -N \end{pmatrix} S = \begin{pmatrix} -1 & \\ & N \end{pmatrix},$$

whence

$$s_{11}^2 - s_{21}^2 N = -1$$

Compute $k \in \mathbb{N}$ minimal such that $\sigma_{11} := ks_{11}$, $\sigma_{21} := ks_{21}$ are integers (via the Euclidean Algorithm). Write $k = N^\ell k_0$ with $N \not| k_0$. Then the equation

$$\sigma_{11}^2 - \sigma_{21}^2 N = -k_0^2 N^{2\ell} \tag{6.2}$$

holds in $\mathbb{Z}$. We claim that we can assume $\ell = 0$ without loss: Indeed, otherwise (6.2) implies $\sigma_{11} = \sigma_{11}' N$ for some $\sigma_{11}' \in \mathbb{Z}$. Hence

$$(\sigma_{11}')^2 N - \sigma_{21}^2 = -k_0^2 N^{2\ell-1}.$$

But now it follows that $\sigma_{21} = \sigma_{21}' N$ with $\sigma_{21}' \in \mathbb{Z}$. Thus

$$(\sigma_{11}')^2 - (\sigma_{21}')^2 N = -k_0^2 N^{2(\ell-1)},$$

analogously to (6.2). Inductively we arrive at $\ell = 0$.

Now (6.2) with $\ell = 0$ implies

$$\sigma_{11}^2 \equiv -k_0^2 \bmod N. \tag{6.3}$$

If $\gamma := \gcd(k_0, N) \neq 1$, then γ is p or q, and the factorization of N allows to compute square roots of -1 modulo p and q, and combine them by means of the Chinese Remainder Theorem.

Otherwise, $\gcd(k_0, N) = 1$, and we can compute $\bar{k} \in \mathbb{Z}$ such that $\bar{k}\, k_0 \equiv 1 \bmod N$. Then (6.3) implies that $(\sigma_{11}\bar{k})^2 \equiv -1 \bmod N$.

(b) Let (f, g) be an instance of $\mathbf{Trafo}^{\mathbb{Q}}$, and let $n := \dim f$. Then $\varphi := f \perp (-g)$ is isotropic.

Retrieve the factorization of $\det\varphi = (-1)^n(\det f)(\det g)$ from the oracle. Then an algorithm by Simon ([Sim05a], see also [Sim05b]) constructs an isotropic vector $(0,0)^t \neq (v_1, v_2)^t \in \mathbb{Q}^{2n}$ for φ in deterministic polynomial time. If $f(v_1) = 0$ then also $g(v_2) = 0$. At least one of the $v_i \neq 0$, hence both f and g are isotropic as $f \sim g$. If $v_1 \neq 0 \neq v_2$, then these are isotropic vectors for f, g; if without loss $v_2 = 0$, then the factorization of $\det g$ can be employed again to construct a primitive isotropic vector v_2' for g. Now if v_1, v_2 are isotropic vectors for f, g, it is easy and well-known that one can find matrices $H_i \in \mathrm{GL}_n\mathbb{Q}$ with $f\, H_1 = h_0 \perp f_1$ and $g\, H_2 = h_0 \perp g_1$ for some $(n-2)$-ary forms f_1, g_1, where h_0 is the "hyperbolic plane" with associated matrix $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, see [Cas78, ch. 2, lm. 2.1 and its cor. 1]. By Witt's lemma (Lemma 1.2.5), $f_1 \sim_{\mathbb{Q}} g_1$ holds. We recursively call the procedure outlined here, yielding eventually a transformation $f_1 S = g_1$. Then return

$$T := H_1 \begin{pmatrix} 1 & & \\ & 1 & \\ & & S \end{pmatrix} H_2^{-1}$$

since then $f\, T = g$.

If, however, $a := f(v_1) \neq 0$, then also $g(v_2) = a$. Find two bases of $\mathbb{Q}^n$ with v_1, respectively v_2, as first vector. Thus we obtain matrices $U_i \in \mathrm{GL}_n\mathbb{Q}$ with

$$f\,U_1 = \langle a \rangle \perp f_2 \qquad \text{and} \qquad g\,U_2 = \langle a \rangle \perp g_2$$

for $(n-1)$-ary forms f_2,g_2, and if the recursion produces $f_2 S = g_2$, then output $T := U_1 \left(\begin{smallmatrix} 1 & 0 \\ 0 & S \end{smallmatrix}\right) U_2^{-1}$.

Finally, this recursion will be called at most n times so that we have established a polynomial-time algorithm.

□

We will reuse the technique from part a) in Chapter 8.

6.2 Rings of Formal Power Series

6.2.1 Introduction and summary of results

We have seen in Sections 3.2 and 6.1 that the transformation and representation problems are relatively easy over the fields $\mathbb{F}$ and $\mathbb{Q}$. It is natural to ask whether these problems are any harder over the polynomial rings $\mathbb{F}[x]$ or $\mathbb{Q}[x]$, or, more generally, over the polynomial ring of an arbitrary field K. We will come back to this question in Sect. 6.3; for now we show that over rings of formal power series, representations are no harder to compute than over the ground field.

Given a commutative ring R, the ring of formal power series over R, denoted $R[[x]]$, is the set of all

$$\sum_{i=0}^{\infty} a_i x^i,$$

where $a_i \in R$, and $x \notin R$ is a variable. We will not write out quadratic forms explicitly in this section, so there should be not confusion with the variables of the quadratic form.

The algebraic operations are defined in the following natural way:

$$\left(\sum_{i=0}^{\infty} a_i x^i\right) + \left(\sum_{i=0}^{\infty} b_i x^i\right) = \sum_{i=0}^{\infty} (a_i + b_i) x^i$$

and

$$\left(\sum_{i=0}^{\infty} a_i x^i\right) \left(\sum_{i=0}^{\infty} b_i x^i\right) = \sum_{i=0}^{\infty} \sum_{j=0}^{i} a_j b_{i-j} x^i.$$

Turning to algorithmic questions, power series are not directly suitable for finite encoding because a single series involves an infinitude of coefficients. But similar to real, or p-adic, numbers, we can approximate ring elements up to a certain desired precision. By convention, determining an element $\sum_i a_i x^i \in R[[x]]$ to precision $D \in \mathbb{N}_0$ means to find its coefficients $a_0, \dots, a_D$.

More generally, we can consider the question whether there is is an approximative solution to a problem up to precision D; this is equivalent to computing in the ring

$$R[[x]]/_{(x^{D+1})} \cong R[x]/_{(x^{D+1})}.$$

So if we talk about algorithms and computational problems over power series rings, we always assume that the input includes the specification of the desired precision D (e.g. by the unary string 1^D). An algorithm may then read a polynomial number of coefficients of the input power series, where polynomial means polynomial in D.

On the other hand, we assume that we can do exact arithmetic in the ground field. Therefore, the term 'precision' will have a unique meaning.

We first state our result on the ring of formal power series. Recall the definition of the computational problems from Sect. 1.3 and that of their decisional analogues from Sect. 3.1.

Theorem 6.2.1 *Let K be a field of characteristic $\neq 2$.*

(a) Let $D \in \mathbb{N}_0$ be given. Then the decisional representation problems over K, and over $K[[x]]$ to precision D are Karp-equivalent. Likewise, the computational representation problems are polynomial-time equivalent for one oracle call each. In other words:

$$\mathbf{DRepr}_n^{K[x]/(x^{D+1})} \approx_K \mathbf{DRepr}_n^K$$

and

$$\mathbf{Repr}_n^{K[x]/(x^{D+1})} \approx_1 \mathbf{Repr}_n^K.$$

(b) If K is infinite, then

$$\mathbf{DRepr}_n^{K[[x]]} \approx_K \mathbf{DRepr}_n^K \qquad \text{and} \qquad \mathbf{Repr}_n^{K[[x]]} \approx_1 \mathbf{Repr}_n^K.$$

6.2.2 Upper bounds

We start by proving the following arithmetic rules for matrices over rings of power series.

Lemma 6.2.2 *Let R be a ring with encoding. Let $X \in R[[x]]^{l_1 \times l_2}$, $Y \in R[[x]]^{l_2 \times l_3}$, $Z \in R[[x]]^{l_1 \times l_3}$ and let*

$$X = \sum_{i=0}^{\infty} X_i x^i, \qquad Y = \sum_{i=0}^{\infty} Y_i x^i, \qquad Z = \sum_{i=0}^{\infty} Z_i x^i$$

with $X_i \in R^{l_1 \times l_2}$, $Y_i \in R^{l_2 \times l_3}$, and $Z_i \in R^{l_1 \times l_3}$ for $i \in \mathbb{N}_0$.

(a) Let $X\,Y = Z$. Then

$$Z_q = \sum_{\nu=0}^{q} X_\nu Y_{q-\nu}.$$

In particular, $Z_0 = X_0\,Y_0$.

(b) Let $X^t = \sum_{i=0}^{\infty} X_i' x^i$. Then $X_i' = X_i^t$ for all $i \in \mathbb{N}_0$.

(c) Let $X^t Y X = Z$. Then

$$Z_q = \sum_{\substack{\lambda,\mu,\nu \in \mathbb{N}_0 \\ \lambda+\mu+\nu=q}} X_\lambda^t Y_\mu X_\nu.$$

Proof :

(a) Induction over q.

(b) Trivial.

(c) Follows from a) and b).

□

Note that it follows immediately from part b) that for associated matrices $A = \sum_{p=0}^{\infty} A_p x^p$, $A_p \in R^{n\times n}$ of quadratic forms all A_p are symmetric.

Proposition 6.2.3
Let K be a field with encoding, not of characteristic 2. Let $n \in \mathbb{N}$. Then

(a) $\mathbf{DRepr}_n^K \preccurlyeq_K \mathbf{DRepr}_n^{K[[x]]}$;

(b) $\mathbf{Repr}_n^K \preccurlyeq_1 \mathbf{Repr}_n^{K[[x]]}$;

(c) $\mathbf{DTrafo}_n^K \preccurlyeq_K \mathbf{DTrafo}_n^{K[[x]]}$;

(d) $\mathbf{Trafo}_n^K \preccurlyeq_1 \mathbf{Trafo}_n^{K[[x]]}$.

Proof :

a) and b) Let f be an n-ary form over R and let $M \in R$. Obviously, if $f \longrightarrow_R M$, then also $f \longrightarrow_{R[[x]]} M$ because $R \subseteq R[[x]]$. Conversely, let $u \in R[[x]]^n$ with $u = \sum_{p=0}^{\infty} u_p x^p$, $u_p \in R^n$, satisfy $f(u) = M$. If A is the associated matrix of f, then, by Lemma 6.2.2, we have $u_0^t A u_0 = M$ and therefore already $f(u_0) = M$.

Thus we have shown that $f \longrightarrow_R M$ if and only if $f \longrightarrow_{R[[x]]} M$. Therefore, to decide whether M is represented by f over R for part a), it suffices to pass the instance over to the $\mathbf{DRepr}_n^{R[x]}$-oracle, which can be done in time $\mathcal{O}(1)$.

In the same vein, on query (f, M), the $\mathbf{Repr}_n^{R[[x]]}$-oracle will respond with $u = \sum_{p=0}^{D} u_p x^p$ (for some desired precision $D \geq 0$), $u_p \in R^n$, with $f(u) = M$. By the above argument, it follows that $f(u_0) = M$.

c) and d) Let f, g be n-ary forms over R. Obviously, if $f \sim_R g$, then also $f \sim_{R[[x]]} g$. Conversely, let $T \in \mathrm{GL}_n R[[x]]$ with $T = \sum_{p=0}^{\infty} T_p x^p$, $T_p \in R^{n \times n}$, satisfy $f\,T = g$. So if A, B are the associated matrices of f, g respectively, then, by Lemma 6.2.2, we have $T_0^t A T_0 = B$ and therefore already $f\,T_0 = g$.

Thus we have shown that $f \sim_R g$ holds if and only if $f \sim_{R[[x]]} g$. Therefore, to decide equivalence over R for part c), it suffices to pass the instance over to the $\mathbf{DTrafo}_n^{R[[x]]}$-oracle, which can be done in time $\mathcal{O}(1)$.

Pass (f, g) to the $\mathbf{Trafo}_n^{R[[x]]}$-oracle. It will output $T = \sum_{p=0}^{D} T_p x^p$ for some D, $T_p \in R^{n \times n}$, with $f\,T = g$. By the above argument, it holds that $f\,T_0 = g$. Finally note that T_0 can be extracted from T in time linear in the encoding length of T.

□

Remark. It can be read off from the proof that analogous assertions hold for general representations of k-ary forms by n-ary forms, see [Min11]. Of these, we have covered the cases $k = 1$, i. e. representations in our (narrower) sense, and $k = n$, i. e. transformations.

6.2.3 Lower bounds for representations

We now turn to the more involved converse reductions.

Lemma 6.2.4 *Let K be a field with encoding, not of characteristic 2. Let f be an n-dimensional quadratic form over $K[[x]]$, let $M \in K[[x]]$, $q \in \mathbb{N}$, $q < |K|$ and let $u \in K[[x]]^n$ satisfy*

$$f(u) \equiv M \mod x^q.$$

Then there is $\tilde{u} \in K[x]^n$ such that

$$f(\tilde{u}) \equiv M \mod x^{q+1} \qquad \text{and} \qquad u \equiv \tilde{u} \bmod x^q.$$

The coefficient vectos at x^q of all possible vectors $\tilde{u}$ form an affine space over K. A parametrization of this affine space can be computed in polynomial-time (with respect to to the encodings of M, u, and f).

In particular, one such vector $\tilde{u}$ can be computed in polynomial time.

Proof : Denote by

$$A = \sum_{p=0}^{\infty} A_p x^p$$

the associated matrix of the form f, $A_p \in K^{n \times n}$ symmetric. Similarly, let

$$M = \sum_{p=0}^{\infty} M_p x^p \qquad \text{and} \qquad u = \sum_{p=0}^{\infty} u_p x^p$$

with $M_p \in K$ and $u_p \in K^n$ for $p \in \mathbb{N}_0$. Then by Lemma 6.2.2, we have

$$\sum_{\substack{\lambda,\mu,\nu\in\mathbb{N}_0\\ \lambda+\mu+\nu=q}} u_\lambda^t A_\mu u_\nu = M_p$$

for all $p = 0, \ldots, D$.

We may assume that $M_0 \neq 0$. Namely, the statement holds for M if and only if it holds for M replaced by the polynomial

$$M' := \sum_{i=0}^{q} M_i x^i.$$

So without loss M is a polynomial of degree at most q and not identically zero. Hence substituting $q+1$ arbitrary pairwise distinct values from K into M yields at most one $t \in K$ for which $M(t) \neq 0$. But replacing A, M, and u by $A(x-t)$, $M(x-t)$, and $u(x-t)$ leaves us with the case $M_0 \neq 0$ above. This is possible since $q < |K|$.

Now by hypothesis $f(u) \equiv M \bmod x^q$ with $q \geq 1$, which by Lemma 6.2.2 and equating coefficients yields

$$u_0^t A_0 u_0 = M_0 \neq 0.$$

This implies that $A_0 u_0 \neq 0$, and hence that the bilinear form $x \mapsto x^t A_0 u_0$ on K^n is nondegenerate.

Let

$$U := \frac{1}{2}\left(M_q - \sum_{\substack{\lambda+\mu+\nu=q\\ \lambda,\,\nu<q}} u_\lambda^t A_\mu u_\nu \right). \tag{6.4}$$

By the last paragraph, there is $v \in K^n$ such that

$$U = v^t A_0 u_0. \tag{6.5}$$

Obviously, the set of all such v forms an affine space which can be computed by standard efficient linear algebra.

For such a vector v, define $\tilde{u} \in (K[x])^n$ by

$$\tilde{u}_p := \begin{cases} 0 & \text{if } p > q, \\ v & \text{if } p = q, \\ u_p & \text{else.} \end{cases} \tag{6.6}$$

Then by construction u is a polynomial (rather than a series),

$$u \equiv \tilde{u} \bmod x^q,$$

and therefore

$$f(\tilde{u}) \equiv f(u) \equiv M \bmod x^q. \tag{6.7}$$

Hence it only remains to compute the q-th coefficient of $\tilde{u}^t A\tilde{u}$. By Lemma 6.2.2, this coefficient equals

$$\begin{aligned}\sum_{\lambda+\mu+\nu=q} \tilde{u}_\lambda A_\mu \tilde{u}_\nu &= v^t A_0 u_0 + (v^t A_0 u_0)^t \quad + \sum_{\substack{\lambda+\mu+\nu=q\\ \lambda,\,\nu<q}} u_\lambda^t A_\mu u_\nu \\ &= 2U \quad + \sum_{\substack{\lambda+\mu+\nu=q\\ \lambda,\,\nu<q}} u_\lambda^t A_\mu u_\nu \\ &= M_q,\end{aligned}$$

the second equality being due to (6.5) and the last one to (6.4). Therefore

$$f(\tilde{u}) \equiv M \bmod x^{q+1},$$

as was to be shown.

The same argument also shows that the set of q-th components of all possible $\tilde{u}$ satisfying the claim arise from the different solutions v of (6.5). As was argued there, this is is an affine space over K. □

Proposition 6.2.5 *Let K be a field not of characteristic 2.*

(a) Given $D \in \mathbb{N}_0$, a quadratic form f over $K[[x]]$, and $M \in K[[x]]\backslash\{0\}$, and access to a $\mathbf{DRepr}_n^K$-oracle, it can be decided in polynomial time whether M can be represented up to precision D, and such an approximative representation can be computed in polynomial time; in other words:

$$\mathbf{DRepr}_n^{K[x]/(x^{D+1})} \preccurlyeq_K \mathbf{DRepr}_n^K \quad \textit{and} \quad \mathbf{Repr}_n^{K[x]/(x^{D+1})} \preccurlyeq_1 \mathbf{Repr}_n^K.$$

(b) If K is infinite, then

$$\mathbf{DRepr}_n^{K[[x]]} \preccurlyeq_K \mathbf{DRepr}_n^K \qquad \textit{and} \qquad \mathbf{Repr}_n^{K[[x]]} \preccurlyeq_1 \mathbf{Repr}_n^K.$$

Proof :

(a) First consider the computational problem. If $D < |K|$, ask the oracle for the solution of the scalar problem modulo x. Then we can inductively apply Lemma 6.2.4 for D times to obtain a solution modulo x^{D+1}. If, however, $D \geq |K|$, then an exhaustive search on $K[x]/(x^{D+1})$ takes only polynomial time.

For the decision problem, it follows directly from the last paragraph that if $D < |K|$, then the problem modulo x^{D+1} is solvable if and only if the scalar problem modulo x is (see Proposition 6.2.3). In case $D \geq |K|$, perform an exhaustive search, and observe whether a solution can be found.

(b) If K is infinite, then a solution to the scalar problem modulo x can be lifted inductively by Lemma 6.2.4 to arbitrarily high degrees; hence a representation over $K[[x]]$ exists if and only if one exists for the scalar problem (again see Proposition 6.2.3). Moreover, up to a given precision D such a solution can be found by iterative application of Lemma 6.2.4.

□

Together with Proposition 6.2.3, this proves Theorem 6.2.1.

6.3 Polynomial Rings

In this section, we analyze the complexity of the representation problem when replacing the coefficient ring with its ring of polynomials (in some finite number of variables). It turns out that as a lower bound, we can only give the problem of solving simultaneous quadratic equations.

Simultaneous representation problem over R, $\mathbf{SRepr}_n^R$
INPUT: $r \in \mathbb{N}$, quadratic forms $(f_i \mid i = 1, \dots, r)$ of dimension n, and $m_i \in R^r$ such that there is $u \in R^n \backslash \{0\}$ satisfying $f_i(u) = m_i$.
OUTPUT: Vector $u \in R^n$ such that $f_i(u) = m_i$ for all i.

Analogously, we define the decision problem $\mathbf{DSRepr}_n^R$.

It should be noted that **DSRepr** is undecidable over $\mathbb{Z}$ [Mat93, sec. 1.2], and thus also **SRepr** may be much harder than **Repr** for many rings R. However, this is not the case in general: For K an algebraically closed field, for instance, $\mathbf{SRepr}_n^K$ can be solved using techniques from algebraic geometry (cf. Bézout's Theorem [Sha74, §IV.2.1]).

Theorem 6.3.1 *Let K be a field with encoding, not of characteristic 2. Let $n \in \mathbb{N}$. Then*

(a) $\mathbf{DSRepr}_n^K \succcurlyeq \mathbf{DRepr}_n^{K[x]} \succcurlyeq_K \mathbf{DRepr}_n^K$,

(b) $\mathbf{SRepr}_n^K \succcurlyeq \mathbf{Repr}_n^{K[x]} \succcurlyeq_1 \mathbf{Repr}_n^K$.

The remainder of this section is devoted to the proof of Theorem 6.3.1. As for power series, we can directly infer:

Proposition 6.3.2
Let K be a field with encoding, not of characteristic 2. Let $n \in \mathbb{N}$. Then

(a) $\mathbf{DRepr}_n^K \preccurlyeq_K \mathbf{DRepr}_n^{K[x]}$;

(b) $\mathbf{Repr}_n^K \preccurlyeq_1 \mathbf{Repr}_n^{K[x]}$, *and one oracle call suffices;*

(c) $\mathbf{DTrafo}_n^K \preccurlyeq_K \mathbf{DTrafo}_n^{K[x]}$*;*

(d) $\mathbf{Trafo}_n^K \preccurlyeq_1 \mathbf{Trafo}_n^{K[x]}$*, where one oracle call suffices.*

Proof : Analogous to Proposition 6.2.3. □

As $R[x]$ is a subring of $R[[x]]$, Lemma 6.2.2 can still be applied to matrices with polynomial coefficients. We can simply consider these polynomials as power series whose coefficients all vanish but finitely many.

Lemma 6.3.3 *Let K be a field with encoding, not of characteristic 2. Let f be an n-dimensional quadratic form over $K[x]$, let $M \in K[x]$, $q, k \in \mathbb{N}$ and let $u \in K[x]^n$ satisfy*

$$f(u) \equiv M \mod x^q.$$

Then there is $\tilde{u} \in K[x]^n$ such that

$$f(\tilde{u}) \equiv M \mod x^{q+k} \tag{6.8}$$

and

$$u \equiv \tilde{u} \bmod x^q. \tag{6.9}$$

The set

$$\left\{ \begin{pmatrix} \tilde{u}_q \\ \vdots \\ \tilde{u}_{q+k-1} \end{pmatrix} \in K^k \;\middle|\; \tilde{u} := \sum_{i=0}^{q-1} u_i x^i + \sum_{i=q}^{r+k-1} \tilde{u}_i x^i \right.$$

$$\left. \text{satisfies (6.8), (6.9)} \right\} =: \mathfrak{M} \tag{6.10}$$

forms an affine space over K. A parametrization of this affine space can be computed in polynomial time.

In particular, one such vector $\tilde{u}$ can be computed in polynomial time.

Proof : The first statement follows from Lemma 6.2.4 by an easy induction on k. We only have to show that $\mathfrak{M}$ of (6.10) is an affine space over K.

From Lemma 6.2.4, we know that when lifting the exponent of x by one, we obtain an affine space as set of q-th components, i. e. the set of solutions of a (possibly inhomogeneous) system of linear equations.

Now employ induction on k again. Suppose we already know that

$$\left\{ \begin{pmatrix} \tilde{u}_q \\ \vdots \\ \tilde{u}_{q+k-2} \end{pmatrix} \in K^{k-1} \;\middle|\; \tilde{u} := \sum_{i=0}^{q-1} u_i x^i + \sum_{i=q}^{r+k-2} \tilde{u}_i x^i \text{ satisfies} \right.$$

$$\left. f(\tilde{u}) \equiv M \bmod x^{q+k-1} \quad \text{and} \quad (6.9) \right\} =: \mathfrak{M}' \tag{6.11}$$

is an affine space over K, and that we know a parametrization

$$\mathfrak{M}' = w + C\,K^{k-1},$$

where $w \in K^{k-1}$ and $C \in K^{(k-1)\times(k-1)}$, not necessarily regular. Proceed according to the proof of Lemma 6.2.4. The newly added component in (6.10) depends affine linearly jointly on all the the $\tilde{u}_q, \ldots, \tilde{u}_{q+k-2}, v = \tilde{u}_{q+k-1}$. As the composition of affine functions is affine, the set $\mathfrak{M}$ from the statement is affine-linear as well. □

For the application of Lemma 6.3.3 to polynomials, we do not only have to lift to a correct solution modulo higher powers of the variable, but also to ensure that this lifting process will terminate after some finite number of steps.

For a vector $v \in K[x]^n$, define its degree by $\deg v := \max_{i=0}^n \deg(v_i)$. Analogously, we define the degree of a matrix over $K[x]$. By the degree $\deg f$ of a form f we mean the degree of its associated matrix; this should lead to no confusion since we only consider quadratic forms.

Lemma 6.3.4 *Let K be a field with encoding, not of characteristic 2. Let f be an n-dimensional quadratic form over $K[x]$, let $M \in K[x]$, $\deg M \leq q$, $k \in \mathbb{N}$ and let $u \in K[x]^n$ satisfy*

$$f(u) \equiv M \mod x^q.$$

Moreover, let $\tilde{u} \in K[x]^n$ with $\deg \tilde{u} \leq q-1$ satisfy $u \equiv \tilde{u} \bmod x^q$.

Then the equation

$$f(\tilde{u}) = M$$

over $K[x]$ is equivalent to a system of linear equations and $(\deg f) + q - 1$ quadratic equations over K. These quadratic equations correspond to representation problems for the forms $f_0, \ldots, f_{\deg f}$ via Proposition 1.2.1.

Proof : By the aid of Lemma 6.2.2, consider the (infinitely many) equations which arise by equating coefficients in $f(\tilde{u}) = M$ at like powers of x. By hypothesis, we have $M_i = 0$ and $\tilde{u}_i = 0$ for all $i \geq q$; hence it can be easily verified that the equations are automatically satisfied from degree $(\deg f)+2q-1$ onwards. The remaining equations are, of course, at most quadratic; more precisely, they are quadratic up to degree $(\deg f) + q$ and linear afterwards. The homogeneous part of the quadratic ones are exactly the scalar forms f_i, $i = 1, \ldots, \deg f$, where

$$f = \sum_{i=0}^{\deg f} f_i x^i.$$

By Proposition 1.2.1, these conditions correspond to a system of linear equations and representation problems with respect to the f_i. Since we are working over a field, the linear modular equations can be omitted. □

This concludes the proof of the the lower bound in Theorem 6.3.1: To solve $\mathbf{Repr}^{K[[x]]}$, we ask the oracle for the solution of the scalar problem (a single representation problem over K), and lift this solution via Lemma 6.3.4, which requires another oracle call.

Together with Proposition 6.3.2, this proves Theorem 6.3.1.

Chapter 7

Complexity for Varying Dimension

In this chapter, we turn back to quadratic forms over $\mathbb{Z}$, and analyze how complexity of the transformation problem changes with varying dimension. It will turn out that if **Trafo** is hard, then it is still hard if restricted to ternary and quaternary forms. This is particularly promising because this allows for small keys in cryptography, and because there are many anisotropic classes in these dimensions which do not allow for an attack as in Sect. 5.4.

7.1 Dimension Shift to Ternary and Quaternary Forms

7.1.1 Introduction and result

In this section, we prove that in fixed dimension $n \geq 3$, computing transformations is no harder than in small dimension, i. e. three or four. This shows that if computing transformations is hard for any bounded dimension $n \geq 3$ at all, then it is necessarily hard in dimensions 3, 4.

Instead of **Trafo**, we will consider the problem **FTrafo**. They differ only by an additional input in **FTrafo**: For an instance (f, g), the factorization of $\det f$ is given for free. This modification will be motivated in Chapter 8. We will, however, suppress this extra information in writing down instances. This should not lead to any confusion.

Theorem 7.1.1 *Let $n \geq 5$, and let $d \in \mathbb{Z}$ be odd and squarefree. Then*

$$\mathbf{FTrafo}_n(d) \preceq_1 \mathbf{FTrafo}_{n-2}(d).$$

Remark. Note that the converse of Theorem 7.1.1 is not obvious: Suppose we replace the forms f, g by adding some form, say h, orthogonally to each of them; i. e. compute $f' = f \perp h$ and $g' = g \perp h$. Assume further that an oracle supplies us with a matrix S' such that $f' S' = g'$. Then S' does not necessarily split into a direct sum of a transformation from f to g and an automorphism of h, so it is not clear how to obtain information on a matrix S satisfying $f\,S = g$.

Corollary 7.1.2 *Let $n \geq 5$, and let $d \in \mathbb{Z}$ be odd and squarefree. Then*

$$\begin{aligned} &\mathbf{FTrafo}_n(d) \preceq \mathbf{FTrafo}_3(d) && \textit{if } n \textit{ is odd, and} \\ &\mathbf{FTrafo}_n(d) \preceq \mathbf{FTrafo}_4(d) && \textit{if } n \textit{ is even.} \end{aligned}$$

□

For the proof we will need a decomposition algorithm which will be explained in the next section.

7.1.2 Direct sum decomposition of isotropic forms

It is an elementary and well-known fact that a regular isotropic form f over a field characteristic $\neq 2$ is equivalent to a form of the shape $H \perp f_0$, where f_0 is a regular $((\dim f) - 2)$-dimensional form, and H is the so-called *hyperbolic plane*, i. e. the form with associated matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{7.1}$$

Theorem 7.1.3, whose proof can be found in [Sch04b], shows that a similar decomposition exists over $\mathbb{Z}$; moreover, it can be computed efficiently in the case of at least ternary forms of squarefree determinant.

Theorem 7.1.3 (Schnorr) *There is a polynomial-time algorithm which, given an isotropic regular quadratic form f with $d := \det f$ squarefree and $n := \dim f \geq 3$, along with an isotropic vector $v \in \mathbb{Z}^n \backslash \{0\}$, constructs $U \in GL_n(\mathbb{Z})$ such that the form $f\,U$ has an associated matrix of the shape*

$$\begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & & & \\ \vdots & \vdots & & A_0 & \\ 0 & 0 & & & \end{pmatrix}, \tag{7.2}$$

where A_0 is a symmetric $(n \times n)$-matrix.

In particular, such a decomposition exists for f.

Remark. It is not usual to call the integral form H with associated matrix (7.1) a hyperbolic plane; this term is resticted to forms over fields. This distinction is motivated by the fact that over a field of characteristic $\neq 2$, the hyperbolic plane is, up to equivalence, the only isotropic binary form of non-zero determinant. This does not hold for H over $\mathbb{Z}$ as there are infinitely many such $\mathbb{Z}$-classes (e. g. for all $a \in \mathbb{N}$, the form with associated matrix $\begin{pmatrix} 0 & a \\ a & 0 \end{pmatrix}$). However, binary isotropic forms over $\mathbb{Z}$ can be called "lattices on the (rational) hyperbolic plane".

7.1.3 Conclusion of the proof

We can now finish the proof of Theorem 7.1.1.

Proof of Theorem 7.1.1: Let (f, g) be an instance of $\mathbf{FTrafo}_n(d)$. If they are definite the problem can be solved by use of Theorem 5.3.1 (note that n is fixed). So suppose they are indefinite. By Meyer's Theorem 1.2.12, the form f is isotropic since $n \geq 5$.

By Simon's algorithm ([Sim05a, Algorithm 7]), we can use our knowledge of the factorization of $d = \det f = \det g$ to construct an isotropic vector for each form. As d is squarefree, we can apply Theorem 7.1.3 to efficiently compute $S_1, S_2 \in \mathrm{GL}_n(\mathbb{Z})$ with

$$f\, S_1 = h \perp f_0 \qquad \text{and} \qquad g\, S_2 = h \perp g_0$$

where h is the form with associated matrix (7.1). Now by Witt's lemma for p-adic integers Lemma 1.2.5, $f_0 \sim_{\mathbb{Z}_p} g_0$ for all symbols $p \neq 2$. Since d is odd, it follows by Proposition 1.2.17 that $f_0 \sim_g g_0$. Now we deduce that $f \sim g$ from Theorem 1.2.18 because $\dim f_0 \geq 3$ and d has no multiple prime divisors.

Obviously, if the oracle outputs $T \in \mathrm{GL}_n\mathbb{Z}$ with $f_0\, T = g_0$, then a solution S of the original problem is readily computed via $S := S_1(I_2 \oplus T)S_2^{-1}$, where $I_2 \in \mathrm{GL}_2\mathbb{Z}$ is the identity matrix. □

7.2 Binary Forms

In [Gau89, art. 171–182], Gauß gave an algorithm for the transformation problem on indefinite binary quadratic forms. This algorithm has been extensively studied (see [Sha72], [Lag80], [Len82], [BB97]). It does not necessarily terminate in polynomial time; however, the running time is polynomial in the size of a minimal solution.

In a crytographic setting as in Chapter 2, a solution to the **Trafo**-instance (f, g) is used as the secret key. An upper bound B on the length of the keys is a public parameter. Hence Gauß' algorithm extracts the secret key from the public key (f, g) in time polynomial in B. This rules out the use of binary forms in the schemes of Chapter 2.

It should be noted that there is an alternative way of representing equivalence transformations, explored by Buchmann, Thiel, and Williams in [BTW95]. They show that given equivalent binary forms f, g, a matrix $S \in \mathrm{GL}_2\mathbb{Z}$ satisfying $f\,S = g$ can be represented as power products of small matrices, so that the length of this representation is polynomial in the sizes of f and g. If we ask for such a concise description of a transformation then the above argument does not apply any more. However, this would ask for a major adaptation of the crypto-schemes.

Theorem 7.2.1 *There exists an algorithm which, given equivalent binary integral quadratic forms f, g computes an equivalence transformation in time polynomial in*

$$\texttt{length}\,(f)\,, \qquad \texttt{length}\,(g)\,, \qquad \textit{and} \qquad \min\{\texttt{length}\,(S) \mid S \in GL_2\mathbb{Z}, f\,S = g\}.$$

Proof : At first, for indefinite binary forms, small representations can be constructed in polynomial time. This is because Gauß [Gau89, art. 183–204] has given a definition of reducedness for indefinite binary forms. Moreover, he presented an algorithm which given any indefinite binary form f, computes an equivalent reduced form f' and a matrix $T \in \mathrm{GL}_2\mathbb{Z}$ such that $f\,T = f'$. There are only finitely many reduced forms in each class. Moreover, given any reduced form f_0 say, an algorithm closely related to the reduction procedure enumerates the 'cycle' of all reduced forms

$$f_0, f_1, \ldots, f_{e-1}, f_e = f_0, f_{e+1} = f_1, \ldots$$

together with matrices

$$R_0 = I, R_1 \ldots, R_{e-1}, R_e, R_{e+1} \ldots$$

such that $f_0\,R_i = f_{i \bmod e}$, for all i. This cycle consists exactly of the reduced forms in the class of f_0. There is also a converse enumeration, producing the forms in order f_0, f_{e-1}, f_{e-2}, and so on. Reduction has been shown to be polynomial-time (in [BB97], see also [Lag80]); in particular, there is always a transformation into a reduced form of size polynomial relative to the size of the original form. Moreover, each step $f_i \mapsto f_{i+1}$ of the cycle enumeration process requires only time polynomial in the size of f_i. Enumeration of reduced forms starting from $f = ax^2 + bxy + cy^2$ of determinant $-D < 0$, is known to correspond to the standard continued fraction expansion of $\frac{\sqrt{D}-b}{2a}$, see [Coh93, sec. 5.2 and 5.6]: If $\frac{p_k}{q_k}$ is the k-th convergent in this expansion, then

$$R_k = \begin{pmatrix} p_{k-1} & p_k \\ q_{k-1} & q_k \end{pmatrix}$$

in the above notation. It is known that denominators of the convergents grow exponentially, more precisely

$$q_k \geq 2^{(k-1)/2} \tag{7.3}$$

see [Per54]. Hence, the size of the R_k grows exponentially in k.

Now let f, g be two equivalent indefinite binary forms linked by an (unknown) transformation S. The reduction algorithm yields another two equivalent forms f', g'. By the above argument, there is a matrix S' satisfying $f'\,S' = g'$ with the size of S' polynomial in that of S. Then by (7.3), this occurs if and only if f' and g' are only logarithmically many steps apart in the cycle, relative to their size. Hence S', and therefore S, can be computed by executing linearly many enumeration steps in the size of S.

As this argument holds for all S, it certainly also holds for that of minimal encoding length. □

Part III

Hardness and Interrelationship

Chapter 8

Comparison with Factorization

8.1 Introduction and Result

This part of this thesis (Chapters 8–10) is devoted to results which support our assumption that **Trafo**, **Repr** are computationally infeasible (see Sect. 1.3.6). In this short chapter, we show that finding transformations is not easier than computing $\sqrt{-1}$ modulo a composite number of unknown factorization. As argued in Sect. 6.1, this suggests that **Trafo** is at least as hard as factoring.

This is an important result because it establishes a concrete lower bound for the complexity of **Trafo** (and by Theorem 10.1.1, thus also of **Repr**). However, this result does not determine the exact complexity of these problems: By contrast, we conjectured that solving **Trafo** requires exponential time, whereas factoring can be accomplished in subexponential time (see the discussion on p. 127). Moreover, factorization is closely related to $\mathbf{Trafo}^{\mathbb{Q}}$ by Theorem 6.1.1. Computing transformations over the integers seems a much more complicate task, as may be illustrated by the much more complex structure of equivalence classes (see Sect. 1.2.5).

Recall from there that **Imag** stands for the problem of computing $\sqrt{-1}$ modulo a counter-Blum number.

Theorem 8.1.1

(a) For $n, s \in \mathbb{N}$, denote by $\mathcal{I}_{n,s}$ the properties

$$f \text{ classically integral}, \qquad \dim f = n, \qquad \operatorname{sign} f = s, \qquad \text{and}$$

$$\operatorname{gen} f = \operatorname{cls}^{+} f$$

for a quadratic form f.

Then for every $n \geq 3$ and $1 \leq s \leq n-1$, it holds that

$$\mathbf{Imag} \preceq_1 \mathbf{Trafo}(\mathcal{I}_{n,s}).$$

(b) For $n, s \in \mathbb{N}$, denote by $\mathcal{A}_{n,s}$ the set of all properties from $\mathcal{I}_{n,s}$, with anisotropy added.

Then for $s \in \{1,2\}$ and $s' \in \{1,2,3\}$ it holds that

$$\textbf{Imag} \preceq_1 \textbf{Trafo}(\mathcal{A}_{3,s}), \qquad \textbf{Imag} \preceq_{1,r} \textbf{Trafo}(\mathcal{A}_{4,s'}).$$

Remark.

(a) The theorem refers to indefinite forms only: This is guaranteed by the condition that $1 \le s, s' \le \dim f - 1$.

(b) All indefinite anistropic forms of dimension ≥ 3 occurr in part b) of the theorem, as there are no indefinite anisotropic forms of dimension ≥ 5 by Meyer's Theorem 1.2.12.

To prove our result, we employ a reduction similar to that of Sect. 6.1. A major point of the proof will be the verification that the forms constructed will be integrally equivalent. To this end, we will need a more sophisticated criterion of equivalence, related to Theorem 1.2.18. This criterion is discussed in Sect. 8.2.

8.2 Spinor Norm and an Equivalence Criterion

In the proof of Theorem 8.1.1, we have to reduce to an instance (f, g) of **Trafo**. To show integral equivalence of these forms f, g, we need another notion, the *spinor norm*.

The elements of

$$\mathcal{O}_R(f) := \{T \in \mathrm{GL}_{\dim f} R \mid f\,T = f\}$$

are called *R-automorphims* of f (where R a ring containing the coefficients of f). Moreover, the group of *proper automorphisms* of f over R is defined as

$$\mathcal{O}^+_R(f) := \{T \in \mathrm{SL}_{\dim f} R \mid f\,T = f\}.$$

Let f be an n-ary form and let K be a field containing the coefficients of f. Recall that $f(\cdot,\cdot)$ denotes the bilinear form associated to f, see Sect. 1.2.1. Let $v \in K^n$ satisfy $f(v) \neq 0$. Then we can define an element $\tau_v \in \mathcal{O}_K(f)$, the *symmetry* with respect to v, by

$$\tau_v(u) = u - \frac{2f(u,v)}{f(v)}\, v$$

for all $v \in K^n$.

The *spinor norm* is a homomorphism

$$\begin{aligned} \theta : \mathcal{O}_K(f) &\longrightarrow K^*/K^{*2} \\ \tau_v &\longmapsto f(v) \end{aligned}$$

For details, in particular for well-definedness, see [O'M63, sec. 55]. We will employ the spinor norm over the p-adic fields $\mathbb{Q}_p$.

With the help of this homomorphism, we can state a general sufficient criterion for the uniqueness of proper equivalence classes in a genus, whose proof can be found in [O'M63, 102:9 and 102:10].

Criterion 8.2.1 Let f be a quadratic form over $\mathbb{Z}$ of dimension $n \geq 3$ which is indefinite. If

$$\theta(\mathcal{O}^+_{\mathbb{Z}_p}(f)) \supseteq (\mathbb{Z}_p)^*$$

for all $p|2\det f$, then

$$\operatorname{gen} f = \operatorname{cls}^+ f.$$

The crucial point is that there are sufficient criteria which are easier to check. The following results can be found (in much wider generality) in [O'M63, sec. 102,104].

Fact 8.2.2 Let p be a prime and f be an integral n-ary quadratic form. If

(a) p is odd, $p \not| \det f$ and $n \geq 2$, or

(b) p is odd, $p^3 \not| \det f$ and $n \geq 3$, or

(c) $p = 2$, $p^2 \not| \det f$ and $n \geq 3$,

then it holds that

$$\theta(\mathcal{O}^+_{\mathbb{Z}_p}(f)) \supseteq (\mathbb{Z}_p)^*.$$

Lemma 8.2.3 *Let f be a quadratic form over the commutative ring R and $\lambda \in R$ not a zero divisor in R. Then*

$$\mathcal{O}_R(f) = \mathcal{O}_R(\lambda f) \qquad \textit{and} \qquad \mathcal{O}^+_R(f) = \mathcal{O}^+_R(\lambda f).$$

Proof : Indeed, for $T \in \mathrm{GL}_R(f)$, it holds that $(\lambda f)T = \lambda(fT)$, so $fT = f$ if and only if $(\lambda f)T = \lambda f$. □

8.3 Conclusion of the Proof

Proof of Theorem 8.1.1: Let N be an instance of **Imag**. Let $n \geq 3$ and $1 \leq s \leq n-1$ be given. Choose $a_1, \ldots, a_{n-2} \in \mathbb{Z}$, each of them coprime to $2N$, with exactly $s-1$ of them negative. Define

$$\begin{aligned} f &:= \langle \quad 1, \quad -N, \quad a_1N^2, a_2N^2, \ldots, a_{n-2}N\rangle \\ g &:= \langle \quad -1, \quad N, \quad a_1N^2, a_2N^2, \ldots, a_{n-2}N\rangle. \end{aligned} \tag{8.1}$$

Obviously, these are forms of dimension n and signature s.

We claim that $f \sim_g g$. As $\det f = \det g$ is odd, it suffices if

$$f \sim_{\mathbb{Z}_p} g \qquad \text{for } p|a_1 \ldots a_{n-2}N\infty$$

by Lemma 1.2.17. Therefore it suffices to prove that

$$f' := \langle 1, -N\rangle \sim_{\mathbb{Z}_p} g' := \langle -1, N\rangle. \tag{8.2}$$

Moreover, as $\det f' = \det g' = N$, it even suffices to verify (8.2) for all $p|N\infty$.

Obviously $f' \sim_{\mathbb{R}} g'$ holds. So let $p|N$ be a prime. Then p is odd and -1 is a square modulo p since N is counter-Blum (see Sect. 6.1). This implies that

$$\langle -1, N\rangle \sim_{\mathbb{Z}_p} \langle 1, -N\rangle.$$

This proves that $f \sim_g g$. We will apply Criterion 8.2.1 to f, g. Let $p|N$. Then $\langle N, a_1N\rangle$ is an orthogonal component of f since $n \geq 3$. Hence

$$\begin{aligned} \theta(\mathcal{O}^+_{\mathbb{Z}_p}(f)) &\supseteq \theta(\mathcal{O}^+_{\mathbb{Z}_p}\langle N, a_1N\rangle) \\ &= \theta(\mathcal{O}^+_{\mathbb{Z}_p}\langle 1, a_1\rangle) && \text{by 8.2.3,} \\ &\supseteq (\mathbb{Z}_q)^* && \text{by 8.2.2.} \end{aligned}$$

Therefore, Criterion 8.2.1 yields $\operatorname{gen} f = \operatorname{cls}^+ f$. We already know that $f \sim_g g$, whence $f \sim g$.

Now ask the oracle for $S = (s_{ij}) \in \mathrm{GL}_n\mathbb{Z}$ satisfying $f\,S = g$. Then

$$S^t \begin{pmatrix} 1 & & & & \\ & -N & & & \\ & & a_1N & & \\ & & & \ddots & \\ & & & & a_{n-2}N \end{pmatrix} S = \begin{pmatrix} -1 & & & & \\ & N & & & \\ & & a_1N & & \\ & & & \ddots & \\ & & & & a_{n-2}N \end{pmatrix},$$

so that

$$s_{11}^2 - s_{21}^2N + \sum_{i=1}^{n-2} s_{2+i,1}^2 a_i N^2 = -1$$

follows by evaluation the first matrix entry. In particular,

$$s_{11}^2 \equiv -1 \bmod N,$$

and s_{11} is a solution of the original **Imag**-instance. This completes the proof of part a).

To verify part b), we follow the above reduction with a more specific choice of the a_i. First consider the ternary case. Choose an arbitrary integer a_1' coprime to N, and let

$$a_1 \equiv -(a_1')^2 \bmod N.$$

The sign of a_1 has to be determined as above, depending on s. Then we claim that f in (8.1) is anisotropic.

Let $N = pq$ with p, q prime. We compute the Hasse-Minkowski invariant of f with respect to p:

$$c_p(f) = \left(\frac{p, -a_1 p}{p}\right) = \left(\frac{p, p}{p}\right)\left(\frac{p, -a_1}{p}\right) = \left(\frac{-1}{p}\right)\left(\frac{-a_1}{p}\right) = 1 \cdot 1 = 1. \qquad (8.3)$$

Here we have used that -1 is a square modulo p by hypothesis, and that $-a_1 \equiv (a_1')^2 \bmod N$ is a square modulo p by construction. Moreover, we can compute

$$\left(\frac{-1, -\det f}{p}\right) = \left(\frac{-1, -a_1 p^2}{p}\right) = \left(\frac{-1, -a_1}{p}\right) = 1$$

as $p \not| a_1$. Together with (8.3) we have $c_p(f) = \left(\frac{-1, -\det f}{p}\right)$, which implies that f is anisotropic by [Cas78, lm. 3.5 of ch. 4].

Finally consider the quaternary case. Find a prime r satisfying

$$\left(\frac{N}{r}\right) = -1. \qquad (8.4)$$

As N is not a perfect square in $\mathbb{Z}$, we can find such a prime in random polynomial time; this follows from quadratic reciprocity and Proposition 4.1.3. Subsequently, choose $a_1', a_2' \in \mathbb{Z}$ coprime to N such that

$$a_1' a_2' \equiv -N \bmod r, \qquad (8.5)$$

and such that exactly $s' - 1$ of these two are negative (e. g. choose $a_1' = \pm 1$ and $a_2' = \pm(r - \lambda N)$, where $\lambda \in \mathbb{Z}$ is such that $r < \lambda N$). Then set $a_1 := r a_1'$, $a_2 := r a_2'$.

We claim that f from (8.1) is anisotropic. Note that

$$\det f = -a_1 a_2 N^3 = (-N)(a_1' a_2') r^2 N^2 \in \mathbb{Q}_r^{*2} \qquad (8.6)$$

because of (8.5). Moreover, we have

$$f \sim_{\mathbb{Z}_r} \langle 1, -N, r, -Nr \rangle$$

by Lemma 1.2.14, and therefore

$$c_r(f) = \left(\frac{-N, r}{r}\right)\left(\frac{-N, -Nr}{r}\right)\left(\frac{r, -Nr}{r}\right) = \\ \left(\frac{r, r}{r}\right)\left(\frac{r, -N}{r}\right) = \left(\frac{-1}{r}\right)\left(\frac{-N}{r}\right) = 1$$

by (8.4). Together with (8.6), this shows that f is anisotropic, using [Cas78, lm. 2.6 of ch. 4]. □

8.4 Problem Modification

By Theorem 6.1.1 and Theorem 5.4.3, the complexity of the transformation problems both over $\mathbb{Q}$ on the one hand and over $\mathbb{Z}$ for isotropic ternary forms on the other, are both closely related to the factoring the determinants. For anisotropic forms over $\mathbb{Z}$, however, the problems seem to be much harder: We conjecture that the complexity of the transformation and representation problems over $\mathbb{Z}$ is exponential, see Sect. 1.3.6. In contrast, factorization can be accomplished in subexponential time (see, for instance, [CP01, sec. 6.3]).

This suggests that the complexity of factoring does not really add to the hardness of **Repr** and **Trafo**; in other words: The knowledge of the prime factors of the determinant does not make these problems significantly easier. Consequently, we will henceforth consider the problems

$$\mathbf{FTrafo}(\mathcal{P}), \qquad \mathbf{FRepr}(\mathcal{P}), \qquad {}^*\mathbf{FRepr}(\mathcal{P})$$

which are defined analogously to **Trafo**($\mathcal{P}$), **Repr**($\mathcal{P}$), and ***Repr**($\mathcal{P}$) with the sole modification that the factorization of d is included in the input (see Sections 1.3.3 and 1.3.4).

Note that these problems have already shown up in Theorems 3.1.3, 3.1.5, and 7.1.1.

For convenience, we will suppress the factorization in the notation of problem instances. This should not cause any confusion as we always specify the exact problem in question.

Chapter 9

NP-Hardness Results

9.1 Introduction and Summary of Results

In this chapter, we prove randomized NP-hardness of decisional variants of the problems **Trafo** and **Repr**. More precisely, these problems ask whether solutions (i. e. transformations or representations) exist in a cuboid included in the problem instance. We will see that hardness can already be achieved in fixed dimension. Moreover, we prove that the decisional representation problem for definite forms is NP-complete in dimensions $n \geq 5$.

For the case of anisotropic forms, we will use a number-theoretic assumption, the special Cohen-Lenstra Heuristics (sCLH), which is discussed in Sect. 9.3.

We begin by introducing the decision problems we are going to examine.

IRepr Interval representation problem
PARAMETERS: Set $\mathcal{P}$ of properties of quadratic forms.
INPUT: $n \in \mathbb{N}$, n-ary quadratic form f satifying all properties from $\mathcal{P}$, integer m, vectors $v, w \in (\mathbb{Z} \cup \{\pm\infty\})^n$, factorization of $\det f$.
OUTPUT: $x \in \mathbb{Z}^n$, $v_i \leq x_i \leq w_i$ for all i such that $f(x) = m$.

DIRepr Decisional interval representation problem
PARAMETERS: Set $\mathcal{P}$ of properties of quadratic forms.
INPUT: $n \in \mathbb{N}$, n-ary quadratic form f satifying all properties from $\mathcal{P}$, integer m, vectors $v, w \in (\mathbb{Z} \cup \{\pm\infty\})^n$, factorization of $\det f$.
DECIDE: Whether there is $x \in \mathbb{Z}^n$ such that $v_i \leq x_i \leq w_i$ for all i and $f(x) = m$.

Note that the factorization of the determinant is always included in the input, as is justified in Chapter 8.

To motivate why we consider these constraint problems, note that for the problems without interval restrictions the decisional variant seems to be significantly easier than the computational version, since **DFRepr** is efficiently solvable for important special cases, namely for forms belonging to one-class genera, see Theorem 3.1.3.

The next proposition sets up the equivalence between **IRepr** and **DIRepr**. This suggests that we may as well study **IRepr** if we are interested in the complexity of the computational problems. When proving NP-hardness, it is often easier (and more natural) to deal with a decisional problem as opposed to a computational problem. Again, the unrestricted problems **Repr** and **IRepr** are very unlikely to be equivalent.

Proposition 9.1.1
Let $\mathcal{P}$ be a set of properties. Then **IRepr**$(\mathcal{P})$ *and* **DIRepr**$(\mathcal{P})$ *are polynomial-time equivalent.*

Proof : Obviously **DIRepr**$(\mathcal{P}) \preccurlyeq$ **IRepr**$(\mathcal{P})$.

To reduce **IRepr**$(\mathcal{P})$ to **DIRepr**$(\mathcal{P})$, use the following divide-and-conquer algorithm:

```
input:   instance $(n, f, v, w, m)$ of IRepr$(\mathcal{P})$
output:  vector $x \in \mathbb{Z}^n$: $f(x) = m$, $v \le x \le w$
for $i = 1, \dots, n$ do
    while $w_i - v_i > 1$ do
        if DIRepr-oracle accepts on input $(n, f, (v_1, \dots, v_{i-1}, \lceil \frac{1}{2}(v_i + w_i) \rceil,$
            $v_{i+1}, \dots, v_n), w, m$ then
            $v_i := \lceil \frac{1}{2}(v_i + w_i) \rceil$
        else $w_i := \lfloor \frac{1}{2}(v_i + w_i) \rfloor$
        fi
    od
    $x_i := w_i$
od
output $x = (x_1, \dots, x_n)$.
```

Obviously, this algorithm constructs a solution, if one exists, in time $\mathcal{O}(n \cdot \log(\|w - v\|_\infty))$, which is polynomial. Moreover, as the same quadratic form f as in the input occurs in all the oracle calls, the properties $\mathcal{P}$ still hold. □

We now give the list of our hardness theorems. The proofs will be given in the subsequent sections.

Theorem 9.1.2 *Let $M \in \mathbb{N}$ odd,$n \in \mathbb{N}$, $n \ge 3$ be fixed. Let $\mathcal{P}$ consist of the properties*

$$\dim f = n, \qquad f \textit{ indefinite isotropic}, \qquad \operatorname{gen} f = \operatorname{cls}^+ f,$$

$$f \textit{ improperly primitive}, \qquad \textit{and} \qquad (\det f, M) = 1$$

for a quadratic form f. Then **DIRepr**$(\mathcal{P})$ *is NP-hard under randomized reductions with one-sided error; precisely:*

$$NP \subseteq RP^{\mathbf{DIRepr}(\mathcal{P})}.$$

The proof of Theorem 9.1.2 is relatively easy, in comparison with Theorems 9.1.2 and 9.1.8. Without the conditions $(\det f, M) = 1$ and $\operatorname{gen} f = \operatorname{cls}^+ f$ it would follow directly from [MA78] (see the proof of Theorem 9.1.2 in Sect. 9.7, Proposition 9.2.1, and the discussion thereafter). More precisely, this would even prove NP-hardness, without randomization.

To obtain an analogous result valid for primitive and anisotropic forms, we have to use a number-theoretic assumption which we call the *special Cohen-Lenstra Heuristics* (sCLH). It will be defined and thouroughly discussed in Sect. 9.3.

Recall that the complexity class RP (r*andom* p*olynomial time*) consists of all decicion problems for which there is a probabilistic worst-case polynomial-time algorithm which accepts any 'yes'-instances with probability $\geq \frac{1}{2}$, and rejects every 'no'-instance. Note that the success probability can be enlarged to $\geq 1-\varepsilon$, for $\varepsilon > 0$, by iterating such a test $\lceil |\log \varepsilon| \rceil$ times independently, and accepting if one of the executions accepts (see [Pap94] for details).

Theorem 9.1.3 *Let $M \in \mathbb{N}$ be fixed. Let $\mathcal{P}'_M$ consist of the properties*

$$\dim f = 3, \qquad f \textit{ indefinite anisotropic,} \qquad \operatorname{gen} f = \operatorname{cls}^+ f,$$

$$f \textit{ properly primitive,} \qquad \textit{and} \qquad (\det f, M) = 1$$

for a quadratic form f. If the sCLH holds true, then $\mathbf{DIRepr}(\mathcal{P}'_M)$ *is NP-hard under randomized reductions with one-sided error; more precisely:*

$$NP \subseteq RP^{\mathbf{DIRepr}(\mathcal{P}'_M)}.$$

The sCLH is employed to guarantee that we can, with high probability, represent every integer by some element of a small set of anisotropic quadratic forms to be constructed in the proof. This is much easier to ensure for isotropic forms.

Let us take a closer look at the properties constituting $\mathcal{P}$ of Theorem 9.1.2 and $\mathcal{P}'_M$ of Theorem 9.1.3. At first, we have seen in Sect. 7.2 that for indefinite binary forms, small representations can be constructed in polynomial time. Therefore, dimension 3 is in this sense minimal for a hardness result. Beside that, it is crucial that we actually have NP-hard problems in fixed dimension at all, in contrast to well-known lattice problems (see the beginning of this section). This also explains why we restrict ourselves to indefinite forms.

The fact that we can include the one-class condition as a hypothesis of this theorems makes it clear that hardness of the problems on forms does by no means depend on the number of classes in the genera of the input forms. Moreover, one-class genera abound, so we also back our statement that hardness is not an exceptional phenomenon.

A remarkable share of the effort necessary to prove these Theorems 9.1.3 and 9.1.9 is owed to including this property in the statements. It would follow directly from classical results [Cas78, p. 202f.] if we could restrict to forms with squarefree determinant in the reduction. However, our proof does not carry over to forms with squarefree determinant. See the discussion after Proposition 9.2.1 for details.

Finally, coprimeness of the determinant to a given number M allows, for instance, the restriction to determiants without small prime factors, and, most importantly, to odd determinant. In the arithmetic theory, a major difficulty in the classification of forms lies in the investigation of their behavior locally at the prime 2 (see [O'M63, §§ 63, 93], [Cas78, sec. 8.4], [Wat76], [Jon44]). Our theorems 9.1.3, 9.1.2 imply that the computational complexity of our problems is independent from the perfidies of dyadic arithmetic.

Under our reductions, 'yes'-instances are mapped to representations of integers *coprime* to the form determinant. This implies that Theorems 9.1.2, 9.1.3 still hold if we restrict **DIRepr** to coprime representations (as in Sect. 1.3.3).

Corollary 9.1.4
Suppose that the sCLH holds true. If **DIRepr**$(\mathcal{P}'_M)$ *is solvable in probabilistic polynomial time, then so is every NP-problem.*

Proof : This is the standard interpretation of NP-hardness. □

The conclusion of Corollary 9.1.4, although potentially weaker than P=NP, seems unrealistic, which is a good hint to the intractibility of **DIRepr**. For the sake of completeness, we mention one of the most important cirteria under which deterministic and probabilistic reducibility coincide, and under which tractibility of **DIRepr**$(\mathcal{P}'_M)$ and validity of the sCLH would indeed imply that P=NP. Recall that E denotes the set of decision problems solvable in deterministic linear-exponential time, i. e. in time

$$\mathcal{O}\left(2^{c\,\ell}\right)$$

in the input length ℓ, for some $c > 0$ (see [Pap94, ch. 20]). By the Time Hierarchy Theorem [Pap94, p. 145], there are problems in this set which require homogeneous circuits families of exponential size. Each of them may or may not have a subexponential *inhomogeneous* circuit family.

Corollary 9.1.5 *If there is a decision problem in E which has no subexponentional oracle ciruit family with access to a SAT-oracle, then* **DIRepr**$(\mathcal{P}')$ *is NP-hard.*

Proof : This follows directly from the derandomization results in [IW97]. □

In a bit more colloquial terms, Corollary 9.1.5 can be rephrased as follows: **DIRepr**$(\mathcal{P}')$ is NP-hard unless non-uniformity admits remarkable savings in complexity for *every* problem in E.

Let us note yet another interpretation of this problem: Theorem 9.1.3 refers to forms from one-class genera of dimension 3. We know that for such forms f, it can be decided in polynomial-time whether $f \xrightarrow{*} m$ once the factorization of $\det f$ is given, see Theorem 3.1.2 and the remark thereafter. This allows us to reformulate Theorems 9.1.2 and 9.1.3 in the following way: Let a ternary indefinite form f and an integer m be given. Then deciding whether all representations $x \in \mathbb{Z}^3$, $f(x) = m$ lie outside the cuboid given by $v, w \in (\mathbb{Z} \cup \{\pm\infty\})^3$ is co-NP hard under randomized reductions, and hence also presumably hard. This still holds if we restrict to instances (f, m) for which a representation of m by f exists.

Moreover, almost as a by-result of the proof, we obtain the following result for the decisional representation problem on definite forms: The next theorem states, in rough terms, that deciding whether a given lattice has a point on a origin-centered sphere of given radius, is hard. Note that for this result we do not require interval constraints to the problems, unlike for all other theorems in this section.

Theorem 9.1.6 *Let $M, n \in \mathbb{N}$, $n \geq 5$ be fixed. Let $\mathcal{D}$ describe the following properties of a quadratic form f:*

$$f \textit{ positive definite,} \qquad \dim f = n, \qquad \textit{and} \qquad (M, \det f) = 1.$$

Then **DRepr**$(\mathcal{D})$ *is NP-complete.*

Remember from Sect. 5.3 that for definite forms in constant dimension, transformations can efficiently be computed. Hence Theorem 9.1.6 highlights a major difference between representations and transformations in the case of definite forms. As we shall see now, this difference does not occur for a large class of indefinite forms.

Even more than in representations, we are interested in the hardness of transformation problems. Modifying **Trafo** in the same spirit as **Repr** above, we obtain hardness results similar to Theorems 9.1.2 and 9.1.3.

ITrafo Interval transformation problem
PARAMETERS: Set $\mathcal{P}$ of properties of quadratic forms.
INPUT: $n \in \mathbb{N}$, n-ary quadratic forms f, g satifying all properties from $\mathcal{P}$, matrices $A, B \in (\mathbb{Z} \cup \{\pm\infty\})^{n \times n}$, factorization of $\det f$.
OUTPUT: $T \in \mathrm{GL}_n(\mathbb{Z})$, $A_{ij} \leq T_{ij} \leq B_{ij}$ for all i, j such that $fT = g$.

DITrafo Decisional interval transformation problem
PARAMETERS: Set $\mathcal{P}$ of properties of quadratic forms.
INPUT: $n \in \mathbb{N}$, n-ary quadratic forms f, g satifying all properties from $\mathcal{P}$, matrices $A, B \in (\mathbb{Z} \cup \{\pm\infty\})^{n \times n}$, factorization of $\det f$.
DECIDE: Whether there exists $T \in \mathrm{GL}_n(\mathbb{Z})$, $A_{ij} \leq T_{ij} \leq B_{ij}$ for all i, j such that $fT = g$.

Again, these problems turn out to be equivalent.

Proposition 9.1.7
Let $\mathcal{P}$ be a set of properties. Then **ITrafo**$(\mathcal{P})$ *and* **DITrafo**$(\mathcal{P})$ *are polynomial-time equivalent.*

Proof : Analogous to the proof of Proposition 9.1.1. □

Theorem 9.1.8 *Let an odd $M \in \mathbb{N}$ be fixed. Let $\mathcal{P}_M$ consist of the properties*

$$\dim f = n, \qquad f \text{ indefinite isotropic,} \qquad \operatorname{gen} f = \operatorname{cls}^+ f,$$

$$f \text{ improperly primitive,} \qquad \text{and} \qquad (\det f, M) = 1$$

for a quadratic form f. Then **DITrafo**$(\mathcal{P}_M)$ *is NP-hard under randomized reductions with one-sided error; precisely:*

$$NP \subseteq RP^{\mathbf{DITrafo}(\mathcal{P}_M)}.$$

Finally, we obtain a result on transformations of anisotropic indefinite ternary forms.

Theorem 9.1.9 *Let $M \in \mathbb{N}$ be fixed. Let $\mathcal{P}'_M$ consist of the properties*

$$\dim f = 3, \qquad f \text{ indefinite anisotropic,} \qquad \operatorname{gen} f = \operatorname{cls}^+ f,$$

$$f \text{ properly primitive,} \qquad \text{and} \qquad (\det f, M) = 1$$

for a quadratic form f. If the sCLH holds true, then **DITrafo**$(\mathcal{P}'_M)$ *is NP-hard under randomized reductions with one-sided error; precisely:*

$$NP \subseteq RP^{\mathbf{DITrafo}(\mathcal{P}'_M)}.$$

This last theorem encourages us to use forms with properties $\mathcal{P}'_2$ in our applications, see Chapter 2.

For all Theorems 9.1.2–9.1.5, 9.1.8, and 9.1.9, the interval restriction can be chosen to be origin-symmetric, and to restrict only one component; this will turn out directly from the proof in Sect. 9.7. But as the proof of Proposition 9.1.1 does not carry over directly to this special case we have chosen to define the problems **IRepr** and **DIRepr** in greater generality.

The remainder of this chapter is organized as follows: The reductions establishing the core of the proofs will be explicated in Sect. 9.2. Section 9.3 contains statement and discussion of the special Cohen-Lenstra heuristics, the number theoretic assumption we are going to use. In Sect. 9.4 we digress slightly to study the set of numbers represented by certain binary forms, and in Sect. 9.5, it is confirmed that the one-class-genus property holds for the forms constructed in the proofs of Theorems 9.1.2, 9.1.3, and 9.1.8. To derive hardness of the transformation problems from the theorems on **DIRepr**, we have to count and classify orbits of representations under the automorphism group of a form. This is the content of Sect. 9.6. The final proofs are summarized in Sect. 9.7.

9.2 From SAT to Squares

As an intermediate step between the classical NP-hard problems and quadratic forms, we use the following problem on binary Diophantine equations.

MS Modular square problem
PARAMETER: $M \in \mathbb{N}$.
INPUT: Integers $a, b, c \in \mathbb{Z}$ with $c > 0$, a odd, $(a, b) = 1 = (ab, M)$, and there is an odd prime p such that if $u^2|b$, then u is a power of p.
DECIDE: Whether there is $x \in \mathbb{Z}$, $|x| \leq c$ such that $x^2 \equiv a \bmod b$.

We also use the subproblem without the condition that a, b be coprime:

MS′ Narrow Modular Square Problem
PARAMETER: $M \in \mathbb{N}$.
INPUT: Integers $a, b, c \in \mathbb{Z}$ with $c > 0$, a odd, squarefree, such that $(ab, M) = 1$ and there is an odd prime p such that if $u^2|b$, then u is a power of p.
DECIDE: Whether there is $x \in \mathbb{Z}$, $|x| \leq c$ such that $x^2 \equiv a \bmod b$.

Recall from Sect. 1.1.4 that $\preccurlyeq$ denotes a deterministic Karp reduction, and $\preccurlyeq_r$ a probabilistic Karp reduction with one-sided error.

Proposition 9.2.1 *Let $M \in \mathbb{N}$ be arbitrary. Then*

*(a) If the Riemann Hypthesis is true, then $3SAT \preccurlyeq_1$ **MS**(M).*

*(b) $3SAT \preccurlyeq_{r,1}$ **MS**$'(M) \preccurlyeq_1$ **MS**(M).*

Proposition 9.2.1 is similar to a result by Adleman and Manders [MA78], and we follow in essence the outline of their proof. They proved NP-hardness (unconditionally and without randomness) of the analogue of **MS**(1) without the constraints imposed on a, b. In fact, their proof does not imply our proposition: They work with integers a, b (as in **MS**) such that b is divisible by unboundedly many unboundedly high powers of odd primes, as well as unboundedly high powers of 2. We will make essential use of the restrictions on a, b. Most importantly, we will construct quadratic forms whose determinant is divisible by b, and the properties of b help ensuring the the one-class condition in Theorems 9.1.3, 9.1.2, 9.1.9, and 9.1.8.

Proof of Proposition 9.2.1: Let Φ be a Boolean formula in 3-CNF. Without loss of generality Φ contains each possible clause at most once, and no clause of Φ contains any variable both complemented and uncomplemented. Let ℓ be the number of variables in Φ. Choose an enumeration $\sigma_1, \ldots, \sigma_m$ of all clauses in the variables $x_1, \ldots, x_\ell$ with exactly three literals containing no variable both complemented and uncomplemented, such that both the bijection $i \mapsto \sigma_i$ and its inverse are polynomial-time (e. g. a suitable lexicographic enumeration). Denote by $\sigma \in \Phi$ the assertion that clause σ occurs in Φ, and by $x_j \in \sigma$ ($\bar{x}_j \in \sigma$) that the j-th variable occurs uncomplemented (complemented) in clause σ. Let $n = 2m + \ell$.

For a fixed assignment to the boolean variables x_i, we define

$$r_i = \begin{cases} 1 & \text{if } x_i = \textbf{true,} \\ 0 & \text{if } x_i = \textbf{false,} \end{cases} \qquad i = 1, \ldots, \ell.$$

Moreover, for a clause σ, define

$$W(\sigma, r) = \sum_{i: x_i \in \sigma} r_i + \sum_{i: \bar{x}_i \in \sigma} (1 - r_i). \tag{9.1}$$

For $k = 1, \ldots, m$, let furthermore

$$R_k := \begin{cases} y_k - W(\sigma_k, r) + 1 & \text{if } \sigma_k \in \Phi, \\ y_k - W(\sigma_k, r) & \text{if } \sigma_k \notin \Phi, \end{cases} \tag{9.2}$$

where y_k are new variables, for $k = 1, \ldots, m$. Since Φ is in 3-CNF, we have $W(\sigma_k, r) = 0$ if assignment r does not render clause σ true, and $1 \leq W(\sigma_k, r) \leq 3$ otherwise.

Hence the equation system

$$R_k = 0, \qquad k = 1, \ldots, m \tag{9.3}$$

has a solution

$$r \in \{0, 1\}^\ell, \qquad y \in \{0, 1, 2, 3\}^m \tag{9.4}$$

if and only if Φ is satisfiable.

Now choose a prime $p \geq 5$ not dividing the M from the statement of the theorem. As $-3 \leq R_k \leq 4$ for all choices (9.4) of the variables, (9.3) is equivalent to

$$\sum_{k=1}^{m} R_k p^k = 0. \tag{9.5}$$

We may estimate

$$\left| \sum_{k=1}^{m} R_k p^k \right| \leq 4 \sum_{k=1}^{m} p^k < p^{m+1} - 2$$

as $p \geq 5$; hence (9.5) is equivalent to

$$\sum_{k=1}^{m} R_k p^k \equiv 0 \bmod p^{m+1},$$

or, equivalently, as p is odd, with

$$\sum_{k=1}^{m} (2\,R_k) p^k \equiv 0 \bmod p^{m+1}. \tag{9.6}$$

Now replace the y_k, $k = 1, \ldots, m$ and the r_i, $i = 1, \ldots, \ell$, from (9.2) by new variables α_i, $i = 1, \ldots, n$, each ranging independently over $\{1, -1\}$, by the formula

$$\begin{aligned} y_k &= \frac{1}{2}\big((1-\alpha_{2k-1}) + 2((1-\alpha_{2k}))\big), \\ r_i &= \frac{1}{2}(1-\alpha_{2m+i}), \end{aligned} \tag{9.7}$$

which obviously induces a bijection between the sets over which the two sequences of variables range.

After this change of variables the left hand side of (9.6) is still integral, and thus the congruence notation makes sense. Using (9.7) and collecting terms, (9.6) can be rephrased as

$$\sum_{j=1}^{n} c_j \alpha_j \equiv \tau' \bmod p^{m+1} \tag{9.8}$$

for some $c_j, \tau' \in \mathbb{Z}$; explicitly, we have

$$\begin{aligned} -\tau' &= \sum_{k=1}^{m} (5 - \sum_{i:x_i \in \sigma_k} 1 + \mathbf{1}_{\sigma_k \in \Phi}) p^k, \\ c_{2k-1} &= -p^k, \\ c_{2k} &= -4p^k, \\ c_{2m+i} &= \sum_{k=1}^{m} (\mathbf{1}_{x_i \in \sigma_k} - \mathbf{1}_{\bar{x}_i \in \sigma_k}) p^k, \end{aligned} \tag{9.9}$$

where $k = 1, \ldots, m$, $i = 1, \ldots, \ell$, and

$$\mathbf{1}_\Psi = \begin{cases} 1 & \text{if } \Psi \text{ is true,} \\ 0 & \text{if } \Psi \text{ is false.} \end{cases}$$

Without affecting solvability or the number of solutions, we may as well introduce an extra variable α_0, define $c_0 := 1$ and $\tau := \tau' + 1$, and write

$$\sum_{j=0}^{n} c_j \alpha_j \equiv \tau \bmod p^{m+1}. \tag{9.10}$$

Thus we have learnt that Φ was satisfiable if and only if (9.10) is solvable for $\alpha \in \{-1, 1\}^{n+1}$.

For later use, we verify that $p \not| \tau$: Indeed, τ' is divisible by p by (9.9); and thus

$$\tau = \tau' + 1 \equiv 1 \bmod p. \tag{9.11}$$

Now define p_0 to be some prime exceeding $4 \cdot p^{m+1}(n+1)$, and let p_j be some prime exceeding p_{j-1}, for $j = 1, \ldots, n$, such that all of them are of polynomial size in p^m; they can be found using Proposition 4.1.1 and Corollary 4.1.2. For part (a), we employ the Riemann Hypothesis, and randomness for part (b).

Choose θ_j, for $j = 1, \ldots, n$, as the smallest positive integer satisfying

$$\theta_j \left\{ \begin{array}{lll} \equiv c_j & \bmod & p^{m+1}, \\ \equiv 0 & \bmod & \prod_{i \neq j} p_i, \\ \not\equiv 0 & \bmod & p_j. \end{array} \right. ; \tag{9.12}$$

which can easily produced with the aid of the Chinese Remainder Theorem. Now we can reformulate (9.10) as follows: Φ is satisfiable if and only if there is $\alpha \in \{1, -1\}^{n+1}$ such that

$$\sum_{j=0}^{n} \theta_j \alpha_j \equiv \tau \bmod p^{m+1}. \tag{9.13}$$

Finally, set $K := \prod_{j=0}^{n} p_j$ and $c := \sum_{j=0}^{n} \theta_j$. Now we claim:

Claim: For $x \in \mathbb{Z}$, the conditions $|x| \leq c$ and $c^2 \equiv x^2 \bmod K$ hold if and only if

$$x = \sum_{j=0}^{n} \theta_j \alpha_j \tag{9.14}$$

for some $\alpha \in \{1, -1\}^{n+1}$.

Proof of claim: By choice of the θ_j, every x of the shape (9.14) satisfies $c^2 \equiv x^2 \bmod K$. Conversely, if $c^2 \equiv x^2 \bmod K$, or equivalently, $0 \equiv c^2 - x^2 = (c-x)(c+x) \bmod K$, then for every $j = 0, \ldots, n$, it holds that $p_j | c-x$ or $p_j | c+x$. In fact, this disjunction is exclusive: Assume that $p_j | c-x$ and $p_j | c+x$ for some j. Then $p_j | 2c$, and thus $p_j | c$ as p_j is an odd prime. But $c \equiv \theta_j \not\equiv 0 \bmod p_j$ by construction, which establishes the desired contradiction.

Thus for every $j = 0, \ldots, n$, the prime p_j divides either $c+x$ or $c-x$. Hence

$$\alpha_j := \left\{ \begin{array}{rl} 1 & \text{if } p | c - x, \\ -1 & \text{if } p | c + x. \end{array} \right.$$

is well-defined. Now set $x' := \sum_{j=0}^{n} \alpha_j \theta_j$. Then $x \equiv x' \bmod p_j$ for all j, whence

$$x \equiv x' \bmod K. \tag{9.15}$$

Moreover, as $|x'| \leq \sum_{j=0}^{n} \theta_j = c$ and as $|x| \leq c$ by hypothesis, we may conclude that

$$|x - x'| \leq 2c. \tag{9.16}$$

Now p_j was chosen so that

$$\frac{2 \cdot p^{m+1}}{p_j} \leq \frac{1}{2(n+1)}. \tag{9.17}$$

By construction of the θ_j,

$$\frac{\theta_j}{\prod\limits_{\substack{i=0\\i\neq j}}^{n} p_i} = \min\left\{\theta \in \mathbb{N} \,|\, \theta \not\equiv 0 \bmod p_j,\ \theta \equiv c_j \bmod p^{m+1}\right\} < 2 \cdot p^{m+1},$$

and therefore

$$\theta_j < 2 \cdot p^{m+1} \prod_{\substack{i=0\\i\neq j}}^{n} p_i = \frac{2 \cdot p^{m+1} K}{p_j} \leq \frac{K}{2(n+1)}$$

by (9.17). Summing over all j, this yields

$$c < \frac{K}{2}, \tag{9.18}$$

and with (9.16), this tells us that $|x - x'| < K$. But together with (9.15) this implies that x and x' coincide. *(end of proof of claim)* $\diamondsuit$

Combining (9.14) and (9.13), we obtain that the $3SAT$ formula Φ has $D \in \mathbb{N}_0$ satisfying truth assignments if and only if there are D numbers $x \in \mathbb{Z}$, $|x| \leq c$ such that

$$\begin{aligned} c^2 - x^2 &\equiv 0 \bmod K, \\ x &\equiv \tau \bmod p^{m+1}. \end{aligned} \tag{9.19}$$

Now we take a closer look at the second condition of (9.19). Consider the equation

$$(\tau - \xi)\,(\tau + \xi) = \tau^2 - \xi^2 \equiv 0 \bmod p^{m+1}. \tag{9.20}$$

If $\xi \in \mathbb{Z}$ solves it, then $p^{k_1} | \tau - \xi$ and $p^{k_2} | \tau + \xi$ for some $k_1, k_2 \in \mathbb{N}_0$ such that $k_1 + k_2 \geq m + 1$. Suppose at first that $k_1 \leq k_2$. Then

$$p^{k_1} | (\tau - \xi) + (\tau + \xi) = 2\tau \equiv 2 \bmod p \tag{9.21}$$

by (9.11). Thus (9.21) implies $k_1 = 0$ and therefore $k_2 \geq m + 1$, which means that $\tau \equiv -\xi \bmod p^{m+1}$. Taking into account the analogous case $k_2 \leq k_1$, we conclude that (9.20) is equivalent to

$$\xi \equiv \tau \bmod p^{m+2} \qquad \text{or} \qquad \xi \equiv -\tau \bmod p^{m+2}. \tag{9.22}$$

Now if $\xi \equiv -\tau \bmod p^{m+1}$, $|\xi| \leq c$, and $c^2 - \xi^2 \equiv 0 \bmod K$, then clearly $x := -\xi$ obeys (9.19). All these considerations can be subsumed under the statement that formula Φ is satisfiable if and only if there is an integer x with $|x| \leq c$ such that

$$\begin{aligned} c^2 - x^2 \equiv{}& \ 0 \quad \bmod K, \\ \tau^2 - x^2 \equiv{}& \ 0 \quad \bmod p^{m+1}. \end{aligned} \tag{9.23}$$

By the Chinese Remainder Theorem, the equations (9.23) are jointly equivalent to

$$p^{m+1}(c^2 - x^2) + K(\tau^2 - x^2) \equiv 0 \bmod p^{m+1}K$$

which in turn is equivalent to

$$(p^{m+1} + K)\, x^2 \equiv K\tau^2 + p^{m+1}c^2 \bmod p^{m+1}K.$$

But as K is prime to p by the construction of the p_j, and $p^{m+1} + K$ is prime to K, we finally reach the equation

$$x^2 \equiv a \bmod b \tag{9.24}$$

where

$$a \equiv (p^{m+1} + K)^{-1}(K\tau^2 + p^{m+1}c^2) \bmod p^{m+1}K \tag{9.25}$$

and $b = p^{m+1}K$. Then (9.24) is solvable for $x \in \mathbb{Z}$ with $|x| \leq c$ if and only if Φ is satisfiable. Now by construction, K is odd and squarefree, and a is odd and coprime to b, so (a, b, c) is an instance of **MS**, and we have proven part (a) of the proposition.

Let us now justify part (b). By Proposition 4.1.3, we can select a prime p from the arithmetic progression (9.25) in random polynomial time. Of course, then a is squarefree. The second reduction is trivial. □

Remark: In the final step of the proof of part b) we have selected a prime in an arithmetic progression. It would have sufficed to select any squarefree number rather than a prime. One might hope that this could be done more efficiently as sqarefree numbers are much more frequent than primes: By a theorem of Landau (see [MSC96, §VI.37 1a)]), the fraction of squarefree numbers $a \leq x$ satisfying (9.25) is bounded from below by

$$\frac{6}{\pi^2\, p^{m+1}K} x + \mathcal{O}(x).$$

Hence choosing $x \geq C\,(p^{m+2}K)^2$ for some appropriate $C > 0$ and selecting a random number $a \leq x$ from the arithmetic progression (9.25) yields a squarefree integer with probability not below $\frac{6}{\pi^2}$. Unfortunately, we are not aware of any efficient way to test whether a given integer is squarefree, the obvious strategy to accomplish this task resorting to its factorization. Therefore, it is not obvious whether the use of primes can be discarded.

We may ask whether and how Proposition 9.2.1 can be further generalized. This interests us because the determinants of forms for which we can prove Theorems 9.1.3, 9.1.2, 9.1.8, and 9.1.8 depend on b. In the actual version, the number of distinct prime factors of b still grows arbitrarily large, as does the maximal multiplicity of a single prime factor. These two properties of b seem to play completely different roles in the hardness proofs:

On the one hand, if we impose a fixed upper bound B on the number of distinct prime factors of b, the problems analogous to **MS** and **MS**' are unlikely to stay NP-hard. We can decide the existence of small square roots by computing all of them; this requires only the factorization of b, which can be achieved in subexponential time*. Square roots modulo b are then computed by combining square roots modulo prime powers dividing b. The number of different roots we have to consider is bounded by 2^B. Now if B is constant B (or $B = \mathcal{O}(\log\log(b))$), then the analogue of **MS** can be solved in subexponential time.

This means that the starting point of our reductions potentially becomes easier. However, **DITrafo** and **DIRepr** for forms with at most B distict prime divisors of the determinant do not seem to be necessarily easier in general than the cases we consider. Therefore, it is an interesting question whether these problems still are NP-hard; but if so, an alternative proof outline is required.

On the other hand, we do not know how restricting the maximal multiplicity of single primes should affect the hardness of the **MS** problem. If we were able to prove Proposition 9.2.1 for squarefree b, say, the reductions would carry over, and we would have generalized the theorems.

9.3 The Special Cohen-Lenstra Heuristics

In [CL84], Cohen and Lenstra suggested a very general heuristic framework for the prediction of the average behavior of the class group of a number field K. The CLH strives to explain striking observations on class groups, as for instance the abundance of class number one and the rareness of non-cyclic class groups for real-quadratic fields, and intends to give a convincing link between these seemingly independent phenomena. Though still unproven, it has enjoyed a vivid reception, and is thought of as a realistic way of extrapolating the long-run behavior of class numbers and groups. In particular, for real quadratic fields numerical results suggest that large class numbers, even class numbers with odd part larger than one are really rare.

We are going to exploit this empirical feature of class numbers. Namely, in the proof of Theorem 9.1.3 we have to ensure that every integer can be represented by one of few binary indefinite forms constructed in the reduction. This is easy to verify locally over the $\mathbb{Z}_p$; to infer the same for representations over $\mathbb{Z}$, however, we need the condition that these forms belong to one-class genera. If the negative determinant of these forms is prime, it suffices if the number field $\mathbb{Q}[\sqrt{-d}]$ has class number one.

To argue like that, we need a variant of the CLH, which we will call *special Cohen-Lenstra Heuristic (sCLH)*, as specified in 9.3.1. This assumption is restricted to real-quadratic fields only (as opposed to general number fields in the

*Factoring can be achieved in random subexponential time by the algorithms of Dixon [Dix81] (see also [Pom87], [Val91]), or the class-group relation method of H. Lenstra and Pomerance [LP92]. The running time of these algorithms has the shape

$$\mathcal{O}\left(\exp\left((c+o(1))\sqrt{\log N \log\log N}\right)\right)$$

for some $c > 0$, where N is the number we want to factor. Heuristically, the Elliptic Curve Method [Len87] and the Quadratic Sieve [Pom82] achieve similar running times and the Number Field Sieve [LL83], [LLMP90] even performs better, but up to now no useful bounds on their running times have been proven rigorously.

CLH) and makes a statement only for class number one (as opposed to many general properties of number fields); however, it also contains a claim about the rate of convergence to long-run behavior so that it is not a consequence of the original CLH. Note that according to [CL84], the unexplained abundance of class number one was one of the key oberservations which led Cohen and Lenstra to enunciate their heurisitic.

We cannot draw upon proven statements as the conjecture is still wide open; even our variant would imply that there are infinitely many real quadratic fields with class number 1, which is still unknown for number fields in general.

The remainder of this section is organized as follows: First we explain the general philosophy the original CLH is based upon. Then we review some known facts and empirical observations on class groups of real quadratic fields. We argue how these observations are explained by the CLH, and after that we discuss our own modifications. Finally, we formulate our new assumption as 9.3.1.

The method. The CLH is based on the thought experiment that, roughly speaking, all properties of class groups which are not determined a priori (e. g. by the factorization of the discriminant of the field), develop according to a certain random model, they obtain a corresponding very comprehensive conjecture on the distribution of such properties on large sets of discriminants.

More precisely, consider any ('reasonable') complex-valued functions f on the set of isomorphic classes of finite abelian groups (to Cohen and Lenstra, being non-negative suffices to be reasonable for a function). This function formalizes observable 'properties' of the class groups; typical examples of such functions being the group order, different sorts of ranks, the number of elements with a given order, or the 1–0 indicator function of being cyclic, of having a certain isomophic class, or of having a certain order. Then

$$\lim_{x\to\infty} \frac{\sum\limits_{\text{disc } K\leq x} f(C(K))}{\sum\limits_{\text{disc } K\leq x} 1}$$

can be considered as an average of f on the groups $C(K)$. Here K ranges over the set of number fields with a fixed Galois group and signature. $C(K)$ denotes the prime-to-n part of the class group of K where n equals the degree of the Galois closure of K. The n-part of the class group is partially determined a priori. For the case of our interest, namely real quadratic fields, we will explain this fact in detail in the subsequent paragraphs.

Class numbers of real quadratic forms. Recall that the class number of a number field is defined as the order of its ideal class group (see [Neu92, sec. I.6]). To find all real quadratic fields $\mathbb{Q}[\sqrt{d}]$, it suffices to let d run through all squarefree positive integers $\neq 1$.

The set $\mathfrak{F}(d)$ of proper classes of primitive integral binary quadratic forms of determinant $-d$ (with d odd and squarefree, for simplicity) also forms a group under under composition. The class group of the number field $\mathbb{Q}[\sqrt{d}]$ is isomorphic to a factor group $\mathfrak{F}(d)/I$, where $|I| \leq 2$ (see [Coh93, sec. 5.2]).

By Gauß' Principal Genus Theorem [Gau89, art. 247, 261, 286f.], the 2-rank of $\mathfrak{F}(d)$ equals $2^{\omega(d)}$, where $\omega(d)$ is the number of distint prime divisors of d; hence $|\mathfrak{F}(d)|$ is divisible by $2^{\omega(d)}$. This power of two constitutes the 'deterministic part' of the class number: It is determined by the prime factorization of d.

Beyond that part determined by genus theory, class numbers seem to behave 'randomly'. The CLH accounts for the empirical findings with the conjecture that

$$\lim_{x\to\infty} \frac{\displaystyle\sum_{\substack{D\leq x\\ h_{\not 2}(D)=1}} 1}{\displaystyle\sum_{D\leq x} 1} = c_0, \tag{9.26}$$

see [CM87]. Here D ranges over all positive integers satisfying either $D \equiv 1 \bmod 4$ with D squarefree, or $D \equiv 8$ or $12 \bmod 16$ with $\frac{D}{4}$ squarefree. By $h(D)$ we denote the class number of $\mathbb{Q}[\sqrt{D}]$, and $h_{\not 2}(D)$ is its odd part, i. e. $h_{\not 2}(D)$ is odd and $h(D) = h_{\not 2}(D) \cdot 2^t$ for some t (note that $\mathbb{Q}[\sqrt{d}] = \mathbb{Q}[\sqrt{4d}]$). Finally

$$c_0 = \frac{1}{2\left(\prod_{i=2}^{\infty}\zeta(i)\right)\prod_{i=1}^{\infty}(1-2^{-i})} \approx 0.75446,$$

where $\zeta(\cdot)$ is the Riemann ζ-function.

The total exclusion of the prime 2 in this conjecture seems to be overcautious: If $\mathfrak{Cl}(d)$ denotes the class group of $\mathbb{Q}[\sqrt{d}]$, then only the index $|\mathfrak{Cl}(d) : \mathfrak{Cl}^2(d)|$ is determined by Gauß theorem, but not the even part of $|\mathfrak{Cl}^2(d)|$. It was conjectured in [Ger84], [Ger87] and, in contrast to the large remaining part of the heuristics, it was proven in [FK06, thm. 3] that the 2-part of $|\mathfrak{Cl}^2(d)|$ behaves as random as conjectured for the odd part of the class number; in particular it was shown that

$$\lim_{x\to\infty} \frac{\displaystyle\sum_{\substack{D\leq x\\ 2\nmid|\mathfrak{Cl}(D)^2|}} 1}{\displaystyle\sum_{D\leq x} 1} = c_1 \quad \text{with} \quad c_1 = \prod_{j=2}^{\infty}(1-2^{-j}) \approx 0.57758. \tag{9.27}$$

The special Cohen-Lenstra Heuristic. Now our variant of the CLH assumes that the probability of $h(D) = 1$ converges not too slowly to positive values, as in equations (9.26) and (9.27). Moreover, we implicitly assume that the convergencies (9.26) and (9.27) are compatible in the sense that the odd and the "random" even parts of the class number are simultaneously trivial with positive probability. Finally, we restrict the determinants to primes in certain arithmetic progressions.

At the same time, we relax our assumption by not insisting that the rate of trivial class numbers among the $h(\mathbb{Q}[\sqrt{p}])$ be exactly c_0c_1 (with c_i from (9.26), (9.27)); instead, we settle for any positive constant. Precisely, we state:

Special Cohen Lenstra Heuristics 9.3.1 There are $c, e > 0$ and a polynomial F such that the following holds:

Let $B > 0$ and primes $p_1, \ldots, p_k$ be given, where $k \leq e \log B$. Then

$$\#\{q \leq F(B) \,|\, q \text{ prime}, \ \left(\frac{q}{p_i}\right) = -1 \,\forall\, i; \quad |\mathfrak{Cl}(D(q))^2| = 1\} \ \geq c\, \frac{F(B)}{B \log F(B)}.$$

Here $D(q) = q$ if $q \equiv 1 \bmod 4$ and $D(q) = 4q$ otherwise.

It should be noted that our restriction to primes, and further to primes in specific residue classes, which is well supported by tables as [TW86], already pops up in the original publication (see [CL84, §9, II. C12]) and is explicitly encouraged in [CM94, sec. 3].

9.4 The Image of Binary Forms

We will now characterize the image of certain binary forms over $\mathbb{Z}_p$ and $\mathbb{Z}$. Over fields, an elementary results states that every isotropic form is *universal*, i. e. it represents all field elements. In this section we try to imitate this result for more general rings.

These results will be used in Sect. 9.7 to guarantee that a small list of quadratic forms represents an unknown fixed integer with high probability.

Lemma 9.4.1
Let p be a prime satisfying

$$\text{cls}\,\langle 1, -p\rangle = \text{gen}\,\langle 1, -p\rangle.$$

Let $m \in \mathbb{Z}$ be odd and satisfy

$$\left(\frac{m}{p}\right) = 1 \qquad \textit{and} \qquad \left(\frac{p}{q}\right) = -1 \quad \forall\, q \textit{ prime}, q|m.$$

(In particular, $p \not| m$.) Then:

(a) Whether the form $\langle 1, -p\rangle$ represents m primitively depends only on the residue classes of p and $m \bmod 8$.

(b) If $p \equiv 1 \bmod 4$, then $\langle 1, -p\rangle$ represents m primitively.

<u>*Proof :*</u> Let $f := x^2 - py^2$. We proceed by verifying the statement locally for all $\mathbb{Z}_r$ and then deducing it over $\mathbb{Z}$.

First consider f over $\mathbb{Z}/8\mathbb{Z}$ evaluated at primitive vectors, which here means tuples $(x, y)^t$ with at least one of x, y odd.

Obviously, the values it takes on primitive vectors depends only on $p \bmod 8$; moreover, if $p \equiv 1 \bmod 4$, then all odd classes modulo 8 are hit, as can be computed directly.

Now by the Local Square Theorem [O'M63, thm. 63.1] (or a strong version of Hensel's lemma as in [Eis95, p. 183ff.]), $a \in \mathbb{Z}_2 \backslash 8\mathbb{Z}_2$ has a primitive representation over $\mathbb{Z}_2$ if and only if it has one modulo 8.

Now consider the prime p. Obviously, f represents 1 primitively mod p; thus, by Hensel's lemma, every square mod p is primitively represented, and conversely, if $p \not| m$ is represented, then $\left(\frac{m}{p}\right) = 1$.

Next consider a prime $q|m$. By Lemma 3.1.1,

$$\begin{aligned} \langle 1, -p\rangle \xrightarrow{*}_{\mathbb{Z}_q} m \;\Leftrightarrow\; & \left(\frac{-1}{q}\right) = 1 = -\left(\frac{-p}{q}\right) \text{ or} \\ & \left(\frac{-1}{q}\right) = -1 = -\left(\frac{-p}{q}\right) \\ \Leftrightarrow\; & \left(\frac{p}{q}\right) = -1. \end{aligned}$$

We have used that $\left(\frac{-1}{r}\right) = 1 \;\Leftrightarrow\; p \equiv 1 \bmod 4.$

Furthermore, for an arbitrary prime $r \not| 2pm$, we always have $\langle 1, -p\rangle \xrightarrow{*}_{\mathbb{Z}_r} m$ by Lemma 3.1.1.

Finally, as f is indefinite every integer is represented by f over $\mathbb{R}$. Combining this information by means of the Chinese Remainder Theorem, we obtain that

$$\langle 1, -p\rangle \xrightarrow{*}_{\mathbb{Z}_r} m \text{ for all symbols } r \;\Leftrightarrow\; \langle 1, -p\rangle \xrightarrow{*}_{\mathbb{Z}_2} m,\; \left(\frac{m}{p}\right) = 1, \text{ and}$$
$$\Big(\, q|m \text{ prime } \rightarrow \left(\frac{p}{q}\right) = -1\Big). \tag{9.28}$$

By [Cas78, thm. 9.1.3], the left hand side of (9.28) holds if and only if m is represented by a form in the genus of f. But by hypothesis, there is only one equivalence classes of binary quadratic forms over $\mathbb{Z}$ in the genus of $\langle 1, -p\rangle$. Therefore, it represents all integers obeying the criterion of (9.28), which was to be proven. □

9.5 The One-Class Condition

The aim of this section is to assert that the forms f constructed in the reductions of Sect. 9.7 in fact satisfy $\operatorname{gen} f = \operatorname{cls}^+ f$. To show this, recall the one-class criteria of Sect. 8.2. Again we denote the group of R-automorphisms of f by $\mathcal{O}_R(f)$. We again employ the spinor norm

$$\theta : \mathcal{O}_K(f) \longrightarrow K^*/K^{*2}$$

for $K = \mathbb{Q}_p$.

Proposition 9.5.1 *Let b be odd, p an odd prime, and $p \not| b$. Then the forms*

$$f := 2x^2 + by^2 - pbz^2 \qquad \text{and} \qquad g := x^2 - byz$$

satisfy

$$\operatorname{gen} f = \operatorname{cls}^+ f, \qquad \operatorname{gen} g = \operatorname{cls}^+ g.$$

<u>*Proof :*</u> For the primes p and 2 the criteria of 8.2.2 obviously apply. Now for $q|b$, we have to verify that

$$\theta(\mathcal{O}^+_{\mathbb{Z}_q}(f)) \supseteq (\mathbb{Z}_q)^*.$$

But since $\langle b, -pb\rangle$ is an orthogonal component of f, we have

$$\begin{aligned} \theta(\mathcal{O}^+_{\mathbb{Z}_q}(f)) &\supseteq \theta(\mathcal{O}^+_{\mathbb{Z}_q}\langle b, -pb\rangle) \\ &= \theta(\mathcal{O}^+_{\mathbb{Z}_q}\langle 1, -p\rangle) && \text{by 8.2.3,} \\ &\supseteq (\mathbb{Z}_q)^* && \text{by 8.2.2.} \end{aligned}$$

Hence we have verified the statement for the form f. For g, the proof is completely analogous. □

9.6 Orbits of Representations

Let $S \in \mathrm{SL}_3\mathbb{Z}$ be an *(proper) automorphism* of the quadratic form f, i. e. $f\,S = f$ and $\det S = 1$, and let $f(v) = m$. Then obviously $f(Sv) = m$. This means that the group of proper automorphisms of f acts on the set $X(m)$ of representation of a number m by f. Hence $X(m)$ decomposes into orbits

$$X(m) = \biguplus_{i \in I} X_i$$

with the properties

$$v, w \in X_i \;\Rightarrow\; \exists\, S \in \mathrm{SL}_3\mathbb{Z} \text{ such that } f\,S = f,\ Sv = w$$

and

$$i \neq j,\ v \in X_i,\ w \in X_j \;\Rightarrow\; \nexists\, S \in \mathrm{SL}_3\mathbb{Z} \text{ such that } f\,S = f,\ Sv = w.$$

To reduce the interval representation problem to the interval transformation problem in the proofs of Theorems 9.1.8 and 9.1.9 in Sect. 9.7, we have to classify these orbits of representations of ternary forms. This is accomplished in Proposition 9.6.2.

The first step consists in determining the possible genera of an orthogonal complement to e_1 in f.

Lemma 9.6.1 *Let $m \in \mathbb{N}$ be odd and squarefree, and let $b \in \mathbb{N}$ be coprime to $2m$. Let f be a ternary quadratic form over $\mathbb{Z}$ and*

$$T = \begin{pmatrix} 1 & \alpha & \beta \\ & 1 & \\ & & 1 \end{pmatrix} \in SL_3\mathbb{Q}. \tag{9.29}$$

(a) Let

$$(mf)\,T = \langle m^2 \rangle \perp g. \tag{9.30}$$

Then the form g is integral. Moreover, if

$$f \sim \langle 1, b, -qb \rangle$$

with $q \not| 2bm$ a prime, then the genus of g is uniquely determined by b, q, and m.

(b) Let

$$(2mf)\,T = \langle 4m^2 \rangle \perp g. \tag{9.31}$$

Then the form g is integral. Moreover, If f is equivalent to the form with associated matrix

$$\begin{pmatrix} 2 & & \\ & & b \\ & b & \end{pmatrix}$$

then g falls into one of at most 4 genera. These are determined by m and b.

In both cases, the list of genera for g can be efficiently generated as soon as the factorization of mb is given.

Proof :

(a) Let

$$\begin{pmatrix} m' & c_1 & c_2 \\ c_1 & & \\ & Q & \\ c_2 & & \end{pmatrix} \tag{9.32}$$

be the associated matrix of f. Then (9.29) and (9.30) imply that $m = m'$, $\alpha = -\frac{c_1}{m}$, and $\beta = -\frac{c_2}{m}$. Consequently, $f\,T$ has associated matrix

$$\begin{pmatrix} m & 0 & 0 \\ 0 & & \\ & Q - \frac{1}{m}cc^t & \\ 0 & & \end{pmatrix}$$

where $c = (c_1, c_2)^t$. Hence g is integral. Moreover, as $\det T = 1$, it holds that $\det g = \frac{m^3 \det f}{m^2} = -mqb^2$.

We now determine the class of g over all rings $\mathbb{Z}_p$. Over $\mathbb{Z}_\infty = \mathbb{R}$, we have

$$\langle m \rangle \perp \frac{1}{m} g \sim_{\mathbb{R}} f \sim_{\mathbb{R}} \langle 1, 1, -1 \rangle$$

since $m, q > 0$. Therefore $g \sim_{\mathbb{R}} \langle 1, -1\rangle$ by Witt's lemma 1.2.5.

Next let p be an odd prime, $p \not| m$. Then $T \in \mathrm{SL}_3\mathbb{Z}_p$ is p-adically integral, and we obtain

$$\langle 1, b, -qb\rangle \ \sim_{\mathbb{Z}_p} \ \langle m\rangle \perp \frac{1}{m} g. \tag{9.33}$$

We distinguish three subcases: If $p \not| qb$, then $p \not| \det g$ and we necessarily have $g \sim_{\mathbb{Z}_p} \langle 1, \det g\rangle$.

Second, let $p|b$, say $\nu_p(b) = k \geq 1$. We know that m is a square modulo b and hence p since $\langle 1, b, -qb\rangle$ primitively represents m. For such p, (9.33) implies

$$\langle 1, p^k, -qp^k\rangle \ \sim_{\mathbb{Z}_p} \ \langle 1\rangle \perp \frac{1}{m} g,$$

(where we have applied Lemma 1.2.14 to simplify f). Then by Witt's Lemma for $\mathbb{Z}_p$ 1.2.5, it follows that $g \sim_{\mathbb{Z}_p} \langle p^k, -qp^k\rangle$.

Thirdly, if $p = q$, we claim that

$$g \sim_{\mathbb{Z}_q} m \cdot \langle mb, -bq\rangle.$$

If $\left(\dfrac{m}{q}\right) = 1$ or if $\left(\dfrac{m}{q}\right) = -1 = \left(\dfrac{b}{q}\right)$, this again follows by applying Witt's lemma to (9.33). In case $\left(\dfrac{m}{q}\right) = -1$, $\left(\dfrac{b}{q}\right) = 1$, observe that

$$\langle 1, b\rangle \sim_{\mathbb{Z}_q} \langle m, m\rangle$$

by Lemma 1.2.14, and then argue with Witt's Lemma.

We continue with the case $p|m$. We know that $\det g = -mqb^2$; hence g is $\mathbb{Z}_p$-equivalent to one of

$$\langle 1, -qm\rangle \qquad \text{or} \qquad \langle \rho, -\rho qm\rangle \tag{9.34}$$

by Theorem 1.2.13 (where $\rho \in \mathbb{Z}$, $\left(\frac{\rho}{p}\right) = -1$). Equation (9.30) implies that

$$m\, f \sim_{\mathbb{Q}_p} \langle m^2\rangle \perp g,$$

hence by Theorem 1.2.8,

$$c_p(mf) = c_p(\,\langle m^2\rangle \perp g\,). \tag{9.35}$$

To evaluate the right hand side, compute

$$c_p\langle 1, b, -qb\rangle = \left(\frac{b,\,-qb}{p}\right) = 1.$$

Moreover, it follow directly from the definition of c_p and the properties of $\left(\frac{\cdot,\cdot}{p}\right)$ (Sect. 1.2.5) that generally

$$c_p(mf) = \left(\frac{m, m}{p}\right)^{\dim f} \cdot c_p(f) \cdot \left(\frac{m, \det f}{p}\right)^{\dim f - 1}, \tag{9.36}$$

hence in our case we obtain

$$c_p(mf) = \left(\frac{m, m}{p}\right) = \left(\frac{-1}{p}\right).$$

Compute the left hand side of (9.35) by substituting g by each of the possibilities (9.34). If $g \sim_{\mathbb{Z}_p} \langle 1, -qm\rangle$, then

$$c_p(\,\langle m^2\rangle \perp g\,) = \left(\frac{m^2, -mq}{p}\right) = 1.$$

Otherwise if $g \sim_{\mathbb{Z}_q} \langle \rho, -\rho qm\rangle$, then

$$c_p(\,\langle m^2\rangle \perp g\,) = \left(\frac{\rho, m}{p}\right) = -1.$$

Hence we have verified that depending on $\left(\frac{-1}{p}\right)$, the form g is $\mathbb{Z}_p$-equivalent to exactly one of the forms (9.34).

Finally, consider the prime $p = 2$. Note that the determinant $\det g = -qmb^2$ is odd. Hence by Theorem 1.2.15, g is $\mathbb{Z}_2$-equivalent to one of

$$\langle 1, -qm\rangle, \qquad \langle 3, 3\rangle, \qquad \text{or} \qquad \langle 3, 7\rangle, \tag{9.37}$$

where the second and third options are only applicable if $qm \equiv -1 \bmod 8$, or $qm \equiv 3 \bmod 8$, respectively.

Furthermore, from (9.30) we conclude that

$$m\,f \sim_{\mathbb{Q}_2} \langle m^2\rangle \perp g \sim_{\mathbb{Z}_2} \langle 1\rangle \perp g. \tag{9.38}$$

Hence

$$c_2(mf) = \left(\frac{m, m}{2}\right) \cdot c_2(f) = (-1)^{(m-1)/2} \cdot \left(\frac{b, -qb}{2}\right).$$

The latter can be easily computed by use of the arithmetic rules of $\left(\frac{\cdot,\cdot}{p}\right)$. Now the forms (9.37) differ pairwise in either determinant or Hasse-Minkowski invariant c_2. Therefore, the $\mathbb{Z}_2$-class of g is uniquely determined by $\det g$ and (9.38).

To sum up, we have established that for every symbol p, the $\mathbb{Z}_p$-class of g is uniquely determined. This is equivalent to saying that the genus of g is unique, which had to be shown.

(b) Proceed analogously to the proof of a). In this case we can compute that $m' = 2m$, $\alpha = \frac{c_1}{2m}$, $\beta = \frac{c_2}{2m}$ if (9.32) is the associated matrix of f. Integrity of g follows exactly as above. Moreover, it holds that $\det g = -4mb^2$.

To determine the genus of g, it is useful to note that

$$f \sim_R \langle 2, 2b, -\frac{b}{2}\rangle$$

for $R = \mathbb{Q}$ and $R = \mathbb{Z}_p$, $p \neq 2$, by application of Lemma 1.2.3. Now we can apply similar arguments as above. We obtain:

For $p = \infty$, we have $g \sim_{\mathbb{R}} \langle 1, -1 \rangle$.

If $p \not| mb$ is odd, then $g \sim_{\mathbb{Z}_p} \langle 1, -m \rangle$.

If $p|b$, say $\nu_p(b) = k \geq 1$, then $g \sim_{\mathbb{Z}_p} \langle p^k, -p^k \rangle$.

If $p|m$, then we find two candidates for the $\mathbb{Z}_p$-class of g:

$$g_1 := \langle 1, -2m \rangle \qquad \text{and} \qquad g_2 := \langle \rho, -2\rho m \rangle$$

with $\rho \in \mathbb{Z}$, $\left(\frac{\rho}{p}\right) = -1$. Then $c_p(\,\langle 4m^2 \rangle \perp g_1\,) = 1$ and $c_p(\,\langle 4m^2 \rangle \perp g_2\,) = -1$. On the other hand, by (9.36) it holds that

$$c_p(m\,f) = \left(\frac{m, m}{p}\right) c_p(f) = \left(\frac{-1}{p}\right).$$

Hence g is p-adically integrally equivalent to exactly one of g_1, g_2, depending on $\left(\frac{-1}{p}\right)$.

Finally, we have to take into account the possible $\mathbb{Z}_2$-classes of g. Note that the Hasse-Minkowski invariant $c_2(g)$ is uniquely determined by m and f since

$$c_2(2mf) = c_2(\langle 4m^2 \rangle \perp g) = c_2(g) \cdot \underbrace{\left(\frac{4m^2, \det g}{2}\right)}_{=1} = c_2(g).$$

Moreover, we have $\det g = -4mb^2 \equiv 4 \bmod 8$. By [Jon44, lm. 1], either g decomposes as

$$g \sim_{\mathbb{Z}_2} \langle r, 4t \rangle \tag{9.39}$$

with r, t odd, or $g = 2\,g_0$ for a binary form g_0 of odd determinant.

In the former case, [Jon44, lm. 6] allows us to chose $r \in \{1, 3\}$ in (9.39). Then t is uniquely determined by r as

$$rt = \frac{1}{4} \det g \equiv -m \bmod 8.$$

Therefore, g falls into one of at most two $\mathbb{Z}_2$-classes if (9.39) holds.

So turn to the latter case, i. e. $g = 2\,g_0$ with $\det g_0 = -m$. If g_0 is properly primitive, then Theorem 1.2.15 gives candidates for its class. Note that

$$c_2(g) = c_2(2g_0) = \underbrace{\left(\frac{2, 2}{2}\right)}_{=1} \left(\frac{2, \det g_0}{2}\right) c_2(g_0) =$$

$$= \begin{cases} c_2(g_0) & \text{if } m \equiv 1, 7 \bmod 8, \\ -c_2(g_0) & \text{else.} \end{cases}$$

As the binary forms listed in Theorem 1.2.15 differ pairwise by determinant or c_2, the $\mathbb{Z}_2$-equivalence class of g_0 is uniquely determined by $\det g$ and $c_2(g)$.

Otherwise, i. e. if g_0 is improperly primitive, then it is $\mathbb{Z}_2$-equivalent to form with associated matrix either

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

again by [Jon44, lm. 1]. As these have inequivalent determinants modulo 8, we have to consider at most one additional class for g per cogruence class m mod 8 and value of $c_2(g)$.

Summing up, this makes a total of 4 $\mathbb{Z}_2$-classes g may be contained in.

Note that each of the possible genera of g, we have (implicitly) specified a form $\mathbb{Z}_p$-equivalent to g for every $p|2(\det g)\infty$. Therefore, a form from such a genus can be efficiently constructed using Theorem 4.2.7. □

As usual, denote by $\omega(m)$ the number of distinct prime factors of m.

Proposition 9.6.2 *Let $m \in \mathbb{N}$ be odd and squarefree, and let $b \in \mathbb{N}$ be coprime to $2m$.*

(a) If

$$f \sim \langle 1, b, -qb \rangle$$

with $q \not| 2bm$ a prime, then the set of representations of m by f decomposes into at most $2^{\omega(m)}$ orbits under the action of the proper automorphisms of f.

(b) If f is equivalent to a form with associated matrix

$$\begin{pmatrix} 2 & & \\ & & b \\ & b & \end{pmatrix},$$

then the set of representations of $2m$ by f decomposes into at most $4 \cdot 2^{\omega(m)}$ orbits under the action of the proper automorphisms of f.

In both cases, we can produce a list of forms f_j, $j = 1, \ldots, N$ with length $N \leq 2^{\omega(m)}$ (or $N \leq 4 \cdot 2^{\omega(m)}$ for part b)) such that for every primitive representation v of m (or $2m$, respectively) by f, there exists $1 \leq j \leq N$ and $B \in SL_3\mathbb{Z}$ with first column v such that

$$f\,B = f_j.$$

If the factorization of mb is given, this list can be computed by a probabilistic algorithm in time $2^{\omega(m)}$ times a polynomial in the input length.

Proof :

(a) If $v \in \mathbb{Z}^3$ is a primitive representation of m by f, then f is equivalent to a form f' satisfying $f(e_1) = m$. By Lemma 9.6.1, we can associate to f' a binary form g whose genus is uniquely determined by v. Let G be the associated matrix of an arbitrary form in that genus. Define $\mathcal{C}$ as the quotient set of

$$\{c \in \mathbb{Z}^2 \,|\, cc^t \equiv -G \bmod m\}$$

modulo the equivalence relation

$$c \approx c' \;\Leftrightarrow\; c' = m\,u + Sc \tag{9.40}$$

for some $u \in \mathbb{Z}^2$ and some automorphism S of g. Then by [Zhu97, lm. 2.1], there is a bijection between the orbits of primitive representations v of m by f and $\mathcal{C}$.

Let us compute the cardinality of $\mathcal{C}$. By the Chinese Remainder Theorem and the squarefreeness of m,

$$|\mathcal{C}| = \prod_{p|m} |\mathcal{C} \bmod p|.$$

So let $p|m$ be a prime and consider the set $\mathcal{C} \bmod p$. In the proof of Lemma 9.6.1 we have shown that

$$g \sim_{\mathbb{Z}_p} \langle r, -rqm\rangle,$$

where $r \in \mathbb{Z}$ satisfies

$$\left(\frac{r}{p}\right) = \left(\frac{-1}{p}\right). \tag{9.41}$$

We may thus choose

$$G \equiv \begin{pmatrix} r & 0 \\ 0 & 0 \end{pmatrix} \bmod p.$$

Then the elements of $\mathcal{C} \bmod p$ are represented by vectors $c \in \mathbb{Z}^2$ solving the congruences

$$\begin{aligned} c_1^2 \equiv &\;\; -r \bmod p, \\ c_1c_2 \equiv &\;\;\;\; 0 \bmod p, \\ c_2^2 \equiv &\;\;\;\; 0 \bmod p. \end{aligned} \tag{9.42}$$

Note that by (9.41), the first congruence is always solvable, and hence the whole system is so. If c is a solution, then necessarily $c_1 \in (\mathbb{Z}/p\mathbb{Z})^*$. Therefore $c_2 \equiv 0 \bmod p$. We have shown that solutions of the congruences (9.42) correspond to the square roots of $-r$ modulo p. Therefore

$$|\mathcal{C} \bmod p| \leq 2.$$

Consequently,

$$|\mathcal{C}| \leq 2^{\omega(m)}.$$

(b) As in part a), we consider the forms g which arise from primitive representations of $2m$ by f in the assignment of Lemma 9.6.1. For such a form g with associated matrix G, let $L_{f,m,g}$ be the set of representations of $2m$ by f which lead to a form in the genus of g under the asignment of Lemma 9.6.1. Then again by [Zhu97, lm. 2.1], the set of orbits of $\mathcal{C}_g$ under the automorphisms of f is in bijection with

$$\{c \in \mathbb{Z}^2 \mid cc^t \equiv -G \bmod m\}$$

modulo the equivalence relation (9.40).

We can estimate the cardinality of $\mathcal{C}_g$ by computing $|\mathcal{C} \bmod p|$ for primes $p|2m$. If p is odd, we have

$$|\mathcal{C}_g \bmod p| \leq 2$$

analogously to part a). Modulo 2, a system of congruences

$$\begin{aligned} c_1^2 &\equiv G_{11} \quad \bmod 2, \\ c_1 c_2 &\equiv G_{12} \quad \bmod 2, \\ c_2^2 &\equiv G_{22} \quad \bmod 2. \end{aligned}$$

has at most one solution modulo 2, namely $c_1 \equiv G_{11}$, $c_2 \equiv G_{22} \bmod 2$. Hence $|\mathcal{C} \bmod 2| \leq 1$.

Altogehter, we have at most

$$\sum_g |\mathcal{C}_g| \leq \sum_g \prod_{p|2m} |\mathcal{C}_g \bmod p| \leq 4 \cdot 2^{\omega(m)},$$

where g runs through a set of representatives of the genera in question.

It remains to describe the algorithm to enumerate forms corresponding to the orbits of representations. Let $m' = m$ for part a) and $m' = 2m$ for part b). Then proceed as follows:

1.) Enumerate representatives for all possible genera for g in Lemma 9.6.1.
2.) For each such g, compute the general solution $c \in (\mathbb{Z}/m'\mathbb{Z})^2$ of the congruence system $cc^t \equiv -G \bmod m'$.
3.) For each g and each vector c corresponding to g, output the form with associated matrix

$$\begin{pmatrix} m' & c^t \\ c & \frac{1}{m'}(G + cc^t) \end{pmatrix}.$$

Note that the output matrices are integral by construction of c. The output forms cover all orbits of representations by [Zhu97]. Moreover, it is obvious that the algorithm needs only polynomial-time per output form (step 1 takes random polynomial-time by Lemma 9.6.1, step 2 uses the factorization of m'). The number of output forms is bounded by $2^{\omega(m)}$, respectively $4 \cdot 2^{\omega(m)}$, as shown above. □

9.7 Conclusion of the Proofs

Proof of Theorem 9.1.2: Let $M \in \mathbb{N}$ and Φ an instance of 3SAT. By Proposition 9.2.1, this is probabilistically transformed into an instance (a, b, c) of $\mathbf{MS}'(M)$. In particular, $(b, M) = 1$, and it is NP-complete to decide whether there are $x, y \in \mathbb{Z}$, $0 \leq x \leq c$ with

$$x^2 + by = a. \tag{9.43}$$

Clearly the binary form xy represents all integers primitively. Hence (9.43) is solvable for $x, y \in \mathbb{Z}$, $0 \leq x \leq c$, if and only if the form

$$x^2 + byz \tag{9.44}$$

represents a primitively. Obviously, (9.44) is indefinite. Moreover, by Proposition 9.5.1, the form (9.44) belongs to a one-proper-class genus. □

Proof of Theorem 9.1.3: Let Φ' be an instance of 3SAT, i. e. a boolean formula in 3-CNF. Denote by $\varphi := |\Phi'|$ the binary length of Φ.

Then by Proposition 9.2.1, Φ is randomly reduced to an instance of $\mathbf{MS}'(M)$. For the resulting problem instance proceed as follows:

```
input:   $\mathbf{MS}'(M)$-instance $(a, b, c)$.
reply := false;
repeat polynomially many times
    select random $k \in [0, b]$;
    $a' := a + kb$;
    repeat polynomially many times
        select random prime $p \equiv 1 \bmod 4$ such that $p > \max\left(\left\lceil \frac{c+|2a'+b|}{|b|} \right\rceil, |b|\right)$,
            and $\left(\frac{-2b}{p}\right) = -1$;

        ask oracle if there is $(x, y, z)^t \in \mathbb{Z}^3$ satisfying
            $2x^2 + by^2 - bpz^2 = 2a' + b$ and $-c \leq x \leq c$

        reply := reply $\vee$ (oracle reply)
 output reply.
```

Here k is chosen according to the uniform distribution in $[0, b]$, and p is chosen by means of Proposition 4.1.1 b) with starting point N uniformly distributed in some suitable interval of the integers.

This establishes a polynomial-time oracle algorithm (for prime selection, see Proposition 4.1.3). Let us examine its correctness for solving $\mathbf{MS}'(M)$. At first, note that if it returns `true` then there are $|x| \leq c$, $z_1, z_2 \in \mathbb{Z}$ such that

$$2\,x^2 + bz_1^2 - bpz_2^2 = 2a' + b,$$

hence, putting $y := z_1^2 - pz_2^2$, we have, in particular, that there are x, y such that $2x^2 + by = 2a' + b$ and thus

$$x^2 \equiv a' \equiv a \bmod b$$

since 2 is invertible modulo the odd integer b. Thus, the $\mathbf{MS}'(M)$ instance has a solution (x, y) and so is a 'yes'-instance.

Conversely, if the algorithm returns `false`, but nevertheless (a, b, c) is a 'yes'-instance, then there is $|x| \leq c$ such that

$$x^2 \equiv a \bmod b;$$

and thus there is $y \in \mathbb{Z}$, necessarily odd, such that $2x^2 + by = 2a' + b$, but y is not represented by any of the binary quadratic forms $\langle 1, -p\rangle$. For each of these forms, one of two things may have happened: Either y is represented by the genus of $\langle 1, -p\rangle$, but this genus consists of several classes; or y is not even represented by the genus of $\langle 1, -p\rangle$.

First, the sCLH 9.3.1 gives us an upper bound on the probability that the first case occurs if the second does not. The second case, however, implies that

$$\forall\, (x, y) \in \mathbb{Z}^2,\ |x| \leq c,\ x^2 + by = a, \quad \exists\, q|y \text{ prime: } \left(\frac{q}{p}\right) \neq -1$$

by Lemma 9.4.1 b). As the q are odd, the symbol $\left(\frac{q}{p}\right)$ takes the values $1, -1$ according to the uniform distribution and independently for different q as p is randomly chosen; hence if (x, y) is any solution of the **MS**' instance, the probability that the second case applies is bounded by

$$1 - 2^{-\omega(y)}$$

(where $\omega(y)$ counts the number of distinct prime factors of y). We now have to show that if we start with a 'yes'-instance of $\mathbf{MS}'(M)$, then with high probability, in some iteration we obtain an instance of (a', b, c) which has solution (x, y) with y decomposing into only logarithmically many prime factors in the input length.

Observe that for all solutions (x, y), $|y|$ is bounded from above by

$$\left\lceil \frac{2|x^2 - a'| + b}{b} \right\rceil \leq \frac{2\max(c^2, a + k\, b)}{b} + 2 < 2(b + 1). \tag{9.45}$$

Assume that (a, b, c) is a 'yes'-instance with some solution (x_0, y_0). Then, for $k = 0, \ldots, b$, the problem instance $(a' = a + kb, b, c)$ necessarily has a solution, namely $(x_0, y_0 + k)$. The range over which y varies thus is an interval

$$[y_0, y_0 + b] \cap \mathbb{Z}, \quad \text{where} \quad y_0 < 2b. \tag{9.46}$$

As follows directly from a result of Erdős and Nicolas [EN81, §5, prop. 3][†] it holds that for $B > 0$,

$$\#\{Y \leq B \,|\, \omega(Y) > 2\ln\ln B\} < \frac{6}{\pi^{5/2}} \frac{B}{(\ln B)^{2\ln(2)-1}\sqrt{\ln\ln B}} \left(1 + \mathcal{O}\left(\frac{1}{\ln\ln B}\right)\right). \quad (9.47)$$

Combining (9.46) with (9.47), we conclude that the innermost *repeat* loop produces at most

$$\mathcal{O}\left(\frac{b}{(\ln b)^{2\ln(2)-1}\sqrt{\ln\ln b}}\right)$$

different a' for which there exists *no* solution (x, y) with y having less than $2\ln\ln y$ prime factors. This implies that after $\log b$ iterations, we have seen at least one instance with a solution of few prime divisors with exponentially large probability.

Now that we have established the occurrence of at least one solution in which y_0 has few prime divisors with high probability, we may conclude that for every choice of p, the probability of failure according to case two is in each iteration independently bounded from above by

$$1 - 2^{-\omega(y_0)} \leq 1 - 2^{-2\ln\ln y_0} \leq 1 - \frac{1}{\ln^2\left\lceil\frac{c^2+|a|}{|b|}\right\rceil},$$

which after special treatment of finitely many instances is bounded away from 1. Together with the sCLH in the first case, we have an bounded the error probability away from 1, and hence this is a one-sided error probabilistic reduction.

It remains to be shown that the forms constructed here satisfy all the properties entailed on them.

Obviously, all forms constructed here are indefinite, of dimension 3, and of determinant prime to M.

Next consider anisotropy: By [Cas78, sec. 4.2] and the Hasse principle, a ternary quadratic form f over $\mathbb{Z}$ is isotropic if and only if $c_q f = 1$ for all symbols q. But we have chosen $\left(\frac{-2b}{p}\right) = -1$, hence

$$c_p\langle 2, b, -pb\rangle = \left(\frac{2, -p}{p}\right)\left(\frac{b, -pb}{p}\right) = \left(\frac{2}{p}\right)\underbrace{\left(\frac{b, b}{p}\right)}_{=\left(\frac{-1}{p}\right)}\left(\frac{b, p}{p}\right) = \left(\frac{-2b}{p}\right) = -1,$$

hence our forms are anisotropic.

Finally, we have to establish the one-proper-class property for all forms constructed above. But this follows directly from Proposition 9.5.1. □

Proof of Theorem 9.1.6: Let Φ be an instance of 3SAT, i. e. a boolean formula in 3-CNF. Follow the reduction from the proof of Proposition 9.2.1 until

[†] Note that the reference [MSC96, §V.21 b)] to this result contains a small but severe misprint.

equation (9.23). By the Chinese Remainder Theorem, this equation system is solvable for $x \in \mathbb{Z}$, $|x| \leq c$, if and only if the equation

$$(K+1)^3 p^{m+2}(c^2 - x^2) - K(\tau^2 - x^2) - Kp^{m+2}y = 0 \tag{9.48}$$

is solvable for $x, y \in \mathbb{Z}$, $|x| \leq c$. We claim that for every pair $x, y \in \mathbb{Z}$ satisfying (9.48), the condition $|x| \leq c$ is equivalent to $y \geq 0$.

From (9.9) we obtain

$$|\tau| = \left| 1 - \sum_{k=1}^{m} (5 - \sum_{x_i \in \sigma_k} 1)p^k - \sum_{\sigma_k \in \Phi} p^k \right| \leq 6 \sum_{k=1}^{m} p^k - 1 < p^{m+1}. \tag{9.49}$$

Moreover, we have $\theta_j \geq \prod_{i \neq j} p_i$ by (9.12); hence

$$c = \sum_{j=0}^{n} \theta_j \geq \sum_{j=0}^{n} \prod_{i \neq j} p_i = \sum_{j=0}^{n} \frac{K}{p_j} \geq \frac{(n+1)K}{p_n} > p^{m+1}, \tag{9.50}$$

where the last estimate is due to the choice of p_n.

Now (9.49) and (9.50) together yield $|\tau| < c$. But then, again by the choice of p_0, it holds that

$$\frac{(K+1)^3}{K} p^{m+2} > 2.$$

These two equations impy that

$$\frac{c^2 - \tau^2}{\frac{(K+1)^3}{K} p^{m+2} - 1} \in]0, 1[.$$

But this establishes the claim by direct manipulations of (9.48).

By rearrangement, (9.48) can be written as

$$ax^2 + by = c \tag{9.51}$$

with $a, b, c > 0$. By the argument of the last paragraph, we have obtained that the decision problem whether (9.51) is solvable for $x \in \mathbb{Z}, y \in \mathbb{N}_0$ is NP-complete. But by Lagrange's Four Square Theorem, this is equivalent to whether a is represented by the quadratic form

$$\langle a, b, \ldots, b \rangle$$

of dimension $n \geq 5$. As $a, b > 0$, this form is positive. □

Proof of Theorems 9.1.9 and 9.1.8: We prove the reduction

$$\mathbf{MS}'(M) \preccurlyeq_{r,na} \mathbf{DITrafo}(\mathcal{P}) \tag{9.52}$$

for

$$\mathcal{P} = \mathcal{P}_M,\ \mathcal{P}'_M \tag{9.53}$$

from Theorems 9.1.9, 9.1.8, respectively. Let (a, b, c) be an instance of $\mathbf{MS}'(M)$. Then a, b are coprime. Hence by Proposition 4.1.3, we can find in random polynomial time a prime number $q \equiv a \bmod b$. Obviously (a, b, c) is a 'yes'-instance of $\mathbf{MS}'(M)$ if and only if (q, b, c) is one.

Now follow the proofs of Theorems 9.1.2, 9.1.3 with $\mathbf{MS}'(M)$-instance (q, b, c) instead of (a, b, c) (using sCLH for Theorem 9.1.3). We obtain a $\mathbf{DIRepr}(\mathcal{P})$-instance (f, q, c), for the respective $\mathcal{P}$ in (9.53), with q prime. Such an instance (f, q, c) asks whether

$$\exists\, v \in \mathbb{Z}^n \text{ primitive}, \ |v_1| \leq c, \quad f(v) = q$$

where f is a quadratic form satisfying $\operatorname{gen} f = \operatorname{cls}^+ f$.

Use Proposition 9.6.2 to produce a list of forms f_j, $j = 1 \ldots N$, such that every primitive representation v of q by f can be extended to a transformation $(v|B) \in \mathrm{SL}_3\mathbb{Z}$ from f to some f_j. By that proposition, we can choose $N = 8$ for 9.1.8 and $N = 2$ for Theorem 9.1.9, and these lists can be provided in random polynomial time. Then hand this list over to the **DITrafo**-oracle.

If there is a primitive representation $f(v) = q$ such that $|v_1| \leq c$, then at least for one j there exists a transformation $T \in \mathrm{SL}_3\mathbb{Z}$ such that $f\,T = f_j$ and $|T_{11}| \leq c$, hence at least one of the (f, f_j, c) is a 'yes'-instance of **DITrafo**. Conversely, if there is no primitive representation $f(v) = q$ such that $|v_1| \leq c$, then each (f, f_j, c) is a 'no'-instance of **DITrafo**.

Obviously, this establishes a randomized non-adaptive reduction. As in the **DITrafo**-instance we have used the form f which already occurred in the **DIRepr**-instances, (9.52) holds for both properties sets $\mathcal{P}$ from the statements of Theorems 9.1.9 and 9.1.8, respectively. □

Chapter 10

Relationship between Transformation and Representation Problems

10.1 Result and Proof Outline

In this chapter we prove a result that closely links the complexity of **FTrafo** and ***FRepr**. As defined in Chapter 8, ***FRepr** asks forrepresentations of integers coprime to the determinant of the respective form. For both problems, the factorization of determinants is included in the input.

Recall that for the security the identification scheme of Sect. 2.3, we require that both transformation and representation problems be intractable. It would be desirable to have a proof of computational equivalence of these two problems because this would reduce the number and clarify the content of our hardness assumptions. Theorem 10.1.1 comes close to this desire: In particular, for ternary forms we prove reductions in both directions if we allow the determinants of the forms to be squared.

Recall our definition of the problem union $\mathbf{A} \sqcup \mathbf{B}$ from Sect. 1.1.2.

Theorem 10.1.1 *Let $n \geq 3$ and let $d \in \mathbb{Z}$ be odd and squarefree.*

(a) $^*\mathbf{FRepr}_n(d) \preceq_{r,1} \mathbf{FTrafo}_n(d^{n-1})$.

(b) $\mathbf{FTrafo}_n(d) \preceq_r {}^*\mathbf{FRepr}_n(d^{n-1}) \sqcup \mathbf{Trafo}_{n-1}$.

For dimension three the statement takes a particularly simple shape.

Corollary 10.1.2 *Let d be odd and squarefree. Then*

$$\mathbf{FTrafo}_3(d) \preceq_{r,1} {}^*\mathbf{FRepr}_3(d^2) \qquad \textit{and} \qquad {}^*\mathbf{FRepr}_3(d) \preceq_{r,na} \mathbf{Trafo}_3(d^2).$$

Algorithm 7:

```
input:   instance $(f, m)$ of ${}^*\mathbf{FRepr}_n(d)$.
output:  $x \in \mathbb{Z}^n$ with $f(x) = m$
1. find binary quadratic form $g$ with $\det g = md^{n-2}$ such that $f^\# \xrightarrow{*} g$
2. construct ternary form $\bar{g}$ with $\varphi_{\bar{g}} = g$ and $f^\# \sim \bar{g}$
3. ask oracle for $S \in \mathrm{GL}_n(\mathbb{Z})$ such that $f^\# S = \bar{g}$
4. output $x := (S_{*,1..(n-1)})^\#$.
```

Proof of part a), overview: First we introduce some definitions. Let φ be a quadratic form of dimension $n-1$ with associated matrix A_φ. We say f *represents* φ if there is $\Sigma \in R^{n\times(n-1)}$ such that $\Sigma^t A \Sigma = A_\varphi$. We also denote this by $f\,\Sigma = \varphi$. The representation is called *primitive* if Σ is a primitive matrix, i. e. if the greatest common divisor of its maximal minors (the determinants of its $(n-1)\times(n-1)$ submatrices) equals one. Denote this by $f \xrightarrow{*} \varphi$.

Recall that if S is a square matrix then its adjoint is denoted by $S^\#$, and analogously for forms. If S is a $n \times (n-1)$-matrix then let $S^\# \in \mathbb{Z}^n$ be the vector defined by

$$(S^\#)_{i+1} = (-1)^i \det \tilde{S}_i,$$

where $\tilde{S}_i$ is obtained from S by deleting its i-th row. In [Min11], Minkowski revealed the following duality.

Proposition 10.1.3 (Minkowski) *Let f be a quadratic form of dimension d over $\mathbb{Z}$ and $m \in \mathbb{Z}\backslash\{0\}$.*

(a) If $S \in \mathbb{Z}^{n\times(n-1)}$ is primitive and $\det(f^\# S) = md^{n-2}$, *then $f(S^\#) = m$.*

(b) If $x \in \mathbb{Z}^n$ is primitive and $f(x) = m$, then there is a primitive $S \in \mathbb{Z}^{n\times(n-1)}$ such that $S^\# = x$ and $\det(f^\# S) = md^{n-2}$.

We fix the following notation: For every quadratic form h, we write A_h for its associated matrix; moreover, if h is a n-ary, we subdivide A_h according to

$$A_h = \begin{pmatrix} A_{\varphi_h} & b_h \\ b_h^t & c_h \end{pmatrix}; \tag{10.1}$$

here φ_h is a $(n-1)$-ary form (with associated matrix A_{φ_h}).

To prove part (a) of the theorem, proceed according to Algorithm 7. In the last step, $S_{*,1..(n-1)}$ is obtained from S by deleting the last column.

By Proposition 10.1.3 quoted above it is obvious that this algorithm is correct and polynomial-time if and only if steps 1 and 2 succeed in polynomial time. We will deal with these steps in Sections 10.2 and 10.3, respectively.

10.2 Construction of a Represented Form

To perform step 1, we first characterize which $(n-1)$-ary forms g of determinant md^{n-2} are represented by f. To this end, we determine the class of $f^{\#}$ and determine which genera g belongs to. Subsequently, we apply the algorithms of Sect. 4.2.4 to construct an integral form g which is primitively represented by $f^{\#}$.

Throughout this section, we keep the notation from the proof overview in Sect. 10.1: Namely, f is a n-dimensional quadratic form with $d = \det f$ odd and squarefree, and $m \in \mathbb{Z}$ is primitively represented by f.

Lemma 10.2.1 *Let $p|d$ be an odd prime. Then*

$$f^{\#} \sim_{\mathbb{Z}_p} \langle r_p, p, , \dots, p, s_p p \rangle \tag{10.2}$$

where $r_p, s_p \in \mathbb{Z}$ satisfy $p \not| r_p, s_p$ and

$$\left(\frac{r_p s_p (d/p)^{n-1}}{p} \right) = 1. \tag{10.3}$$

<u>*Proof :*</u> By Theorem 1.2.13, the form $f^{\#}$ is $\mathbb{Z}_p$-equivalent to a diagonal form $\langle a_1, \dots, a_n \rangle$. Let $a_i = p^{e_i} r_i$ with $r_i \in \mathbb{Z}_p^*$. Then since

$$f^{\#\#} = d^{n-2} f, \tag{10.4}$$

it holds that for each $1 \leq i \leq n$,

$$p^{n-2} | \prod_{\substack{j=1 \\ j \neq i}}^{n} a_j.$$

It follows that $e_i \leq 1$ for $i = 1, \dots, n$, and hence that $e_i = 1$ for all but one i. Then (10.2) follows directly from Theorem 1.2.13. Now Equation (10.4) implies (10.3). □

Note that r_p can easily be computed from the conditions $f \xrightarrow{*} r_p$ and $p \not| r_p$. Furthermore, s_p can be determined from the equation $\left(\frac{r_p s_p (d/p)^{n-1}}{p}\right) = 1$. Thus we can identify the class of $f^{\#}$ according to Lemma 1.2.17 by finding r_p and s_p for each $p|d$, and finally inspecting $\operatorname{sign} f^{\#}$. Next we look for a suitable g such that

$$f^{\#} \xrightarrow{*}_{\mathbb{Z}_p} g \quad \text{and} \quad \det g = md^{n-2}. \tag{10.5}$$

Here we have extended previous notation: The formula $F \xrightarrow{*}_R G$ should mean that F represents G primitively over the ring R.

Lemma 10.2.2 *Let $p|d$ be an odd prime. Then an $(n-1)$-ary form g_p satisfying (10.5) can be constructed in polynomial time.*

Proof : Let r_p, s_p as in Lemma 10.2.1. By the preceeding remark they can be determined efficiently. Let $d = d_0 p$. Without loss of generality we may assume that $r_p, s_p \in \mathbb{Z}$ and that $r_p s_p \equiv d_0^{n-1} \bmod p$. Now if $\left(\frac{m s_p d_0}{p}\right) = 1$, then it is obivious from Lemma 10.2.1 that $f^\#$ primitively represents

$$g_p := \langle m r_p s_p d_0, p, \ldots, p \rangle.$$

If else $\left(\frac{m s_p d_0}{p}\right) = -1 = \left(\frac{s}{p}\right)$ then $f^\#$ primitively represents

$$g_p := \langle r_p, p, \ldots, p, m s_p d_0 p \rangle$$

over $\mathbb{Z}_p$. Finally if $\left(\frac{m d_0}{p}\right) = -1$ and $\left(\frac{s}{p}\right) = 1$, then there are $\alpha, \beta \in \mathbb{Z}$ with $\alpha^2 + \beta^2 s \equiv m d_0 \bmod p$. Sending e_{n-2} to $\alpha e_{n-2} + \beta e_{n-1}$ we obtain that $f^\#$ primitively represents

$$g_p := \langle r_p, p, \ldots, p, m d_0 p, s_p p \rangle.$$

□

Let us determine g_p for the remaining symbols p. For $p = \infty$ we can clearly choose

$$g_\infty := \langle -1, \ldots, -1, 1, \ldots, 1 \rangle \tag{10.6}$$

of signature $\operatorname{sign} f^\#$ or $\operatorname{sign} f^\# - 1$ depending on the condition

$$(-1)^s = \operatorname{sign}(m d^{n-2}).$$

Moreover, for all $p \not| d$ odd we can identically choose

$$g_p := \langle 1, \ldots, 1, d \rangle. \tag{10.7}$$

For $p = 2$ some additional care is needed. Compute

$$c := \prod_{p | d\infty} c_p(g_p)$$

Then g_2 can be selected such that

$$c_2(g_2) = c \qquad \text{and} \qquad \det g_2 \equiv d \bmod 8.$$

We may choose

$$g_2 = \begin{cases} \langle \underbrace{1, \ldots, 1}_{n-1}, d \rangle & \text{if } c = 1, \quad \text{and} \\ \langle \underbrace{1, \ldots, 1}_{n-3} \perp g' & \text{if } c = -1, \end{cases} \tag{10.8}$$

where g' is taken from the table

$$g' = \begin{cases} \langle 1, 3, 3 \rangle & \text{if } d \equiv 1 \bmod 8, \\ \langle 3, 3, 3 \rangle & \text{if } d \equiv 3 \bmod 8, \\ \langle 1, 3, 7 \rangle & \text{if } d \equiv 5 \bmod 8, \quad \text{and} \\ \langle 3, 3, 7 \rangle & \text{if } d \equiv 7 \bmod 8, \end{cases}$$

cf. Theorem 1.2.15.

Note that the computations performed to determine the g_p concern only $\omega(d)+\mathcal{O}(1) = \mathcal{O}(\log d)$ different p. Thus all the local forms g_p can be constructed together in polynomial time.

Lemma 10.2.3 *There exists an $(n-1)$-ary integral form g of determinant md^{n-2} such that*

$$g \sim_{\mathbb{Z}_p} g_p$$

for all symbols p, where the g_p are chosen according to Lemma 10.2.2 and equations (10.6), (10.7), and (10.8).

Such a form g can be constructed in probabilistic polynomial time.

Proof : By Theorem 4.2.1, it suffices that $\det g_p \in md^{n-2}\mathbb{Z}_p^{*2}$, that $c_p(g_p) = 1$ for all $p \not| 2md^{n-2}\infty$, and that

$$\prod_p c_p(g_p) = 1.$$

The product ranges through all symbols p. The first of the two conditions is obviously satisfied. The last one holds by construction of g_2. By Theorem 4.2.7, if follows that g can be computed efficiently. □

10.3 Construction of an Equivalent Form

We now turn to the question of how to construct the form $\bar{g}$ in step 2 of Algorithm 7. We are given the left upper $(n-1) \times (n-1)$ corner of its associtated matrix as well as its equivalence class, which is that of $f^{\#}$.

Lemma 10.3.1 *Let F, g be integral quadratic forms with $n = \dim F = (\dim g) + 1 \geq 3$ and $e = \det F$ odd and n-power free such that $F \xrightarrow{*}_{\mathbb{Z}_p} g$ for all $p|e\infty$. Let $\bar{g}$ be an integral n-ary quadratic form of determinant e with $\bar{g} \xrightarrow{*} g$. Then $F \xrightarrow{*} g$.*

Proof : Let $p|e\infty$. Then $p \neq 2$, and it follows from $F \xrightarrow{*}_{\mathbb{Z}_p} g$ and [Cas78, sec. 9.3] that

$$F \sim_{\mathbb{Z}_p} g \perp \left\langle \frac{e}{\det g} \right\rangle;$$

in particular, $\frac{e}{\det g} \in \mathbb{Z}_p$. Since $\det \bar{g} = e$, we also have $\bar{g} \sim_{\mathbb{Z}_p} g \perp \left\langle \frac{e}{\det g} \right\rangle$ for all $p|e\infty$. Therefore $F \sim_{\mathbb{Z}_p} \bar{g}$ for these p. By Lemma 1.2.17, then $F \sim_g \bar{g}$. By Theorem 1.2.18 and because e is odd and n-power free, and $n \geq 3$, it holds that

$F \sim \bar{g}$. By hypothesis, $\bar{g} \overset{*}{\longrightarrow} g$, and therefore also $F \overset{*}{\longrightarrow} g$. □

It is clear that $f^{\#}$ satisfies the hypotheses of this lemma on F. Moreover, if step 2 is sucessful (as we will demonstrate now), we will have an integral form $\bar{g}$ of determinant e satisfying $\bar{g} \overset{*}{\longrightarrow} g$ in an obvious way, and this proves that $f^{\#} \overset{*}{\longrightarrow} g$ over $\mathbb{Z}$.

Now turn to step 2 of the algorithm. We have to find a form $\bar{g}$ of determinant d^{n-1} and of the shape

$$A_{\bar{g}} = \begin{pmatrix} A_g & b \\ b^t & c \end{pmatrix}. \tag{10.9}$$

Equivalently, we can state that we have to find the corresponding $b \in \mathbb{Z}^{n-1}$, $c \in \mathbb{Z}$.

Lemma 10.3.2 *If $\bar{g}$ is an n-ary form of determinant d^{n-1} satisfying (10.9), then c is uniquely determined by g, b, and d, and is efficiently computable from these.*

Given g, b, and d, an integer c exists such that (10.9) defines a quadratic form of determinant d^{n-1} if and only if

$$g^{\#}(b_1, \ldots, b_{n-1}) = d^{n-1} \bmod (md^{n-2}). \tag{10.10}$$

<u>*Proof :*</u> Expand the determinant as

$$d^{n-1} = \det(\bar{g}) = c(\det g) + g^{\#}(b_1, \ldots, b_{n-1}).$$

□

We can now finalize the proof of Theorem 10.1.1, part a).

<u>*Proof of Theorem 10.1.1 a):*</u> Follow Algorithm 7. Step 1 is accomplished in probabilistic polynomial-time by Lemma 10.2.3. To execute step 2, it suffices by Lemma 10.3.2 to find values $b_1, \ldots, b_{n-1}$ which satisfy (10.10).

Solubility of (10.10) follows from [Cas78, thm. 9.1.3] and Proposition 10.1.3 or direct calculation; note that $\det g^{\#}$ is a power of the modulus the factorization of which is known to us. As to algorithmic resolution, we can solve the modular quadratic equation (10.10) by use of [AEM87].

By the discussion after Algorithm 7, this completes the proof of part a). □

10.4 Reduction to Representations

Now let us turn to proving b) of Theorem 10.1.1. This means that we have to show how to solve a given instance of $\mathbf{FTrafo}(d)$ using an oracle answering ${}^*\mathbf{FRepr}_n(d^{n-1})$ and $\mathbf{Trafo}_{n-1}$ queries.

For any form h define

$$\Omega(h) = \{u \in \mathbb{Z} \mid h \xrightarrow{*} u,\ (u, \det h) = 1\}.$$

Let an instance (f, g) of $\mathbf{FTrafo}_n(d)$ be given, with n, d as in Theorem 10.1.1. By [Cas78, sec. 8.3, 9.1] for h satisfying the hypotheses of Theorem 1.2.18, $\Omega(h)$ is the union of several arithmetic progressions mod $8\prod_{p|d} p$; and this obviously applies to $g^{\#}$. Let $L_1, L_2 : \mathbb{N} \to \mathbb{N}$ be arbitrary polynomially bounded functions with $L_i(x) \geq 2$ for all x which grow at least linearly. Recall the notational convention on φ_h, b_h, c_h from (10.1).

Perform Algorithm 8. Its rough outline is as follows: In step 1, we find a representation of a random value m of the form $g^{\#}$. By Minkowski Duality 10.1.3, this corresponds to the representation of an $(n-1)$-dimensional form $\varphi = \varphi_g$ by g. The same computation is performed for $f^{\#}$. If we are lucky, then φ_f and φ_g will be equivalent, and a representation of φ_g by g can be extended to a transformation from f to g. For this step, we require a transformation from φ_g to φ_f which we obtain from the oracle.

Note that each single step of Algorithm 8 is in fact efficiently executable. In particular, we have $\det \varphi = md^{n-2}$ in step 2 by Proposition 10.1.3. The representation asked for after label 2 exists because $f \sim g$ and thus $f^{\#} \sim g^{\#}$. The boolean value in the **if**-query just before step 3 is evaluated using Theorem 3.1.3. The asterisk in the matrix equation above indicates that the equation for the (n, n) entry is not considered, ensuring that the system is in fact linear.

Lemma 10.4.1 *The* **goto** *statement is executed at most once at each iteration of the* **repeat** *loop, and if so then during the first pass through the* **for** *loop.*

Proof : From Lemma 10.2.1 and $f^{\#} \xrightarrow{*} \varphi$ it follows that

$$\varphi \sim_{\mathbb{Z}_p} \langle r_p, p, \ldots, p, v_p p \rangle.$$

Here r_p is uniquely determined by f up to a factor from $\mathbb{Z}_p^{*2}$. Once r_p has been chosen, we may take $v_p = \frac{\det \varphi}{r_p p^{n-2}}$. By Proposition 1.2.17 and Theorem 1.2.18, exactly two classes occur for φ, and they have different signatures. By $\operatorname{sign} m \neq \sigma$ it is enforced that the signature is changed if $f\, B \not\to \varphi$. □

Algorithm 8:

```
input:   Instance $(f, g)$ of $\mathbf{FTrafo}_n(d)$.
output:  transformation $T$ with $f\,T = g$.
$\sigma := 1$;
repeat
    1.select random $m \in \Omega(g^{\#}) \cap \sigma\,[-L_1(|d|), 0]$ (w.r.t. uniform distribution);
      ask oracle for a representation $w$ of $m$ by $g^{\#}$;
      compute primitive $C \in \mathbb{Z}^{n \times (n-1)}$ such that $C^{\#} = w$;
      find $r \in \mathbb{Z}^n$ such that $R := (C \mid r) \in \mathrm{GL}_n\mathbb{Z}$; $g := gR$, $\varphi := \varphi_g$, $\sigma := \mathrm{sign}\,(md^{n-2})$;
    2.for $i = 1, \ldots, L_2(\log|d|)$ do
          ask oracle for a representation $u$ of $m$ by $f^{\#}$
          compute primitive $B \in \mathbb{Z}^{n \times (n-1)}$ such that $B^{\#} = u$
          if $f\,B \not\sim \varphi$ then goto 1. fi
          3. ask oracle for $S \in \mathrm{GL}_{n-1}\mathbb{Z}$ such that $f\,B\,S = \varphi$;
             $B := B\,S$;
             try to solve the system of linear equations
                       $\det(B \mid a) = \pm 1,$
             $\begin{pmatrix} B^t \\ a^t \end{pmatrix} A_f(B \mid a) = \begin{pmatrix} A_\varphi & b_g \\ b_g^t & * \end{pmatrix}$
             ... for $a \in \mathbb{Z}^n$;
             if a solution $a$ has been found then output $\begin{pmatrix} B \mid a \end{pmatrix} R^{-1}$ fi
      od
end repeat
```

Lemma 10.4.2 *Let φ be as at label 3 in Algorithm 8. Then there are at most $2^{\omega(m)}$ orbits of representations of φ by f under the automorphism group $\{Q \in GL_n\mathbb{Z} \mid f\,Q = f\}$.*

Proof : By [AS87], these orbits are at most as numerous as square roots of $-d^{n-1}\varphi^{\#}$ in $(\mathbb{Z}/m\mathbb{Z})[x_1, \ldots, x_{n-1}]$, of which there are $2^{\omega(m)}$. □

Since every representation from at least one of these orbits can be extended to form a transformation, each iteration of the **for** loop has a probability of solving the problem of at least $2^{-\omega(m)}$. So if $\omega(m) \leq \frac{3}{2}(\log\log m)$ say, then the algorithm terminates successfully during the current iteration of the **repeat** loop with significant, i. e. reciprocally polynomial probability. Finally, $(1 - o(1))x$ of all integers $1 \leq m \leq x$ satisfy this bound [HW60, thm.431]. Now in step 1, the algorithm scans several arithmetic progressions, hence a positive fraction of the integers up (or down) to $\pm L_1(|d|)$. Therefore it hits such m with probability at least ℓ^{-1} where $\ell = P(L_1(|d|))$ for some positive polynomial P. Hence each single iteration of the **repeat** loop brings the algorithm to a successful termination with probability $\pi \geq \frac{1}{\ell}$. Since random bits used in step 1 are selected independently we obtain a geometric distribution:

$$\begin{aligned}\textbf{Prob}[\text{success after } \ell \text{ iterations}] &= \pi \sum_{i=0}^{\ell} (1-\pi)^i \\ &= 1 - (1-\pi)^{\ell+1} \geq 1 - (1-\ell^{-1})^{\ell+1} \geq 1 - \mathrm{e}^{-1}. \end{aligned} \tag{10.11}$$

where e is Euler's constant. Hence running the **repeat** loop $\lceil \ell \rceil$ times and returning **fail** in case of failure establishes the desired reduction.

This concludes the proof of Theorem 10.1.1. □

We can now as well prove the corollary for dimension three.

Proof of Corollary 10.1.2: The second statement of Corollary 10.1.2 is a direct application of Theorem 10.1.1 for $n = 3$, so let us turn to the first one.

Let (f, g) be an instance of $\textbf{FTrafo}_3(d)$. Then Theorem 10.1.1 transforms it into an instance (f', m) of $^*\textbf{FRepr}_3(d^2)$ and an instance (g, h) of $\textbf{FTrafo}_2$. By [Die03], there is a solution $T \in \mathrm{GL}_3\mathbb{Z}$ to the original transformation problem of polynomial size. By the proof of Theorem 10.1.1 b), every such transformation arises from a construction as in Algorithm 8; in particular, with the notation from there, BS consist of the first two columns of RT, where S is the binary equivalence transform. This implies that S depends linearly on B, R, and T, and thus is of polynomial size. Hence the oracle query for S can be replaced by an application of Theorem 7.2.1, which finds S in polynomial time. Since we cannot be sure that such an S can be found for the particular m chosen in the first step of Algorithm 8, we have to modify Algorithm 8 as follows: Interrupt the attempt to compute S after a polynomial time bound, reselect m, repeat the calculations up to the computation of S, increase the time bound polynomially, and retry. This proves the reduction without the $\mathbf{Trafo}_2$-oracle. □

Bibliography

[AD97] Ajtai, Miklós and Dwork, Cynthia, *A public-key cryptosystem with worst-case/average-case equivalence*, Proceedings of the 29th annual ACM symposium on theory of computing, El Paso, TX, USA, May 4-6, 1997, Association for Computing Machinery, 1997, pp. 284–293.

[AEM87] Adleman, Leonard M., Estes, Dennis R., and McCurley, Kevin S., *Solving bivariate quadratic congruences in random polynomial time*, Mathematics of Computation **48** (1987), no. 177, 17–28.

[AKS04] Agrawal, Manindra, Kayal, Neeraj, and Saxena, Nitin, *PRIMES is in P*, Annals of Mathematics **160** (2004), no. 2, 781–793.

[AS87] Arenas-Sola, Angela, *On integral representations by quadratic forms*, Linear and Multilinear Algebra **22** (1987), 149–160.

[Bae78] Baeza, Ricardo, *Quadratic forms over semilocal rings*, Lecture Notes in Mathematics, no. 655, Springer, 1978.

[BB97] Buchmann, Johannes and Biehl, Ingrid, *An analysis of the reduction algorithms for binary quadratic forms*, Tech. Report No. TI-26/97, TU Darmstadt, 1997.

[BBBV97] Bennett, Charles H., Bernstein, Ethan, Brassard, Gilles, and Vazirani, Umesh, *Strengths and weaknesses of quantum computing*, SIAM Journal of Computing **26** (1997), no. 5, 1510–1523.

[BBS86] Blum, Lenore, Blum, Manuel, and Shub, Michael, *A simple unpredictable pseudo-random number generator*, SIAM Journal of Computing **15** (1986), 364–383.

[BBT94] Biehl, Ingrid, Buchmann, Johannes, and Thiel, Christoph, *Cryptographic protocols based on discrete logarithms in real-quadratic orders*, Advances in cryptology - CRYPTO '94. 14th annual international cryptology conference, Santa Barbara, CA, USA, August 21-25, 1994. Proceedings (Desmedt, Yvo G., ed.), Lecture Notes in Computer Science, no. 839, Springer, 1994, pp. 56–60.

[BDG88] Balcázar, José Luis, Díaz, Josep, and Gabarró, Joaquim, *Structural complexity. I*, EATCS Monographs on Theoretical Computer Science, no. 11, Springer, 1988.

[BMM00] Buchmann, Johannes, Maurer, Markus, and Moran, Shlomo, *Cryptography based on number fields with large regulator*, Journal de Théorie des Nombres de Bordeaux **12** (2000), no. 2, 293–307.

[BR93] Bellare, Mihir and Rogaway, Phillip, *Random oracles are practical: a paradigm for designing efficient protocols*, Proceedings of the First ACM Conference on Computer and Communications Security, Association for Computing Machinery, 1993, pp. 62–73.

[Bro06] Bromwich, Thomas John I'Anson, *Quadratic forms and their classification by means of invariant-factors*, Cambridge Tracts in Mathematics and Mathematical Physics, no. 3, Cambridge University Press, 1906.

[BTW95] Buchmann, Johannes, Thiel, Christoph, and Williams, Hugh, *Short representation of quadratic integers*, Computational algebra and number theory. Based on a meeting on computational algebra and number theory, held at Sydney University, Sydney, Australia, November of 1992. (Bosma, Wieb (ed.) et al., ed.), Math. Appl., no. 325, Kluwer Academic Publishers, 1995, pp. 159–185.

[BV07] Buchmann, Johannes and Vollmer, Ulrich, *Binary quadratic forms. An algorithmic approach*, Algorithms and Computation in Mathematics, no. 20, Springer, 2007.

[BW90] Buchmann, Johannes and Williams, Hugh, *A key exchange system based on real quadratic fields: Extended abstract*, Advances in Cryptology - CRYPTO '89, Proc. Conf., Santa Barbara/CA (USA), Lecture Notes in Computer Science, no. 435, Springer, 1990, pp. 335–343.

[Cas78] Cassels, John W. S., *Rational quadratic forms*, London Mathematical Society Monographs, no. 13, Academic Press, 1978.

[CGH98] Canetti, Ran, Goldreich, Oded, and Halevi, Shai, *The random oracle model methodology revisited*, STOC '98. Proceedings of the 30th annual ACM symposium on theory of computing, Dallas, TX, USA, May 23-26, 1998. New York, NY, Association for Computing Machinery, 1998, pp. 209–218.

[CL84] Cohen, Henri and Lenstra jun., Hendrik W., *Heuristics on class groups of number fields*, Number Theory, Proc. Journ. arith., Noordwijkerhout 1983, Lecture Notes in Computer Science, no. 1068, Springer, 1984, pp. 33–62.

[CM87] Cohen, Henri and Martinet, Jacques, *Class groups of number fields: Numerical heuristics*, Mathematics of Computation **48** (1987), no. 177, 123–137.

[CM94] ______, *Heuristics on class groups: Some good primes are no too good*, Mathematics of Computation **63** (1994), no. 207, 329–334.

[CM98] Cochrane, Todd and Mitchell, Patrick, *Small solutions of the Legendre equation*, Journal of Number Theory **70** (1998), no. 1, 62–66.

[Coh93] Cohen, Henri, *A course in computational algebraic number theory*, Graduate Texts in Mathematics, vol. 138, Springer, 1993.

[Coh00] ______, *Advanced topics in computational number theory*, Graduate Texts in Mathematics, vol. 193, Springer, 2000.

[CP01] Crandall, Richard and Pomerance, Carl, *Prime numbers: A computational perspective*, Springer, 2001.

[CR03] Cremona, J. E. and Rusin, D., *Efficient solutions of rational conics*, Mathematics of Computation **72** (2003), no. 243, 1417–1441.

[CS93] Conway, J. H. and Sloane, N. J. A. (eds.), *Sphere packings, lattices and groups*, 2nd ed., Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen, vol. 290, Springer, 1993.

[Dic34a] Dickson, Leonard Eugene, *History of the theory of numbers: I. divisibility and primality*, Chelsea, New York, 1934, (reprint of the 1919 edition, Carnegie Inst., Washington DC).

[Dic34b] ______, *History of the theory of numbers: II. diophantine analysis*, Chelsea, New York, 1934, (reprint of the 1919 edition, Carnegie Inst., Washington DC).

[Dic34c] ______, *History of the theory of numbers: III. quadratic and higher forms*, Chelsea, New York, 1934, (reprint of the 1919 edition, Carnegie Inst., Washington DC. Reprinted again in 1966).

[Die03] Dietmann, Rainer, *Small solutions of quadratic Diophantine equations*, Proceedings of the London Mathematical Society, III. Ser. **86** (2003), no. 3, 545–582.

[Dix81] Dixon, John D., *Aymptotically fast factorization of integers*, Mathematics of Computation **36** (1981), 255–260.

[Eic52] Eichler, Martin, *Quadratische Formen und Orthogonale Gruppen*, Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen, no. 63, Springer, 1952.

[Eis95] Eisenbud, David, *Commutative algebra: With a view toward algebraic geometry*, Graduate Texts in Mathematics, no. 150, Springer, 1995.

[EN81] Erdős, Pál and Nicolas, Jean-Louis, *Sur la fonction: Nombre de facteurs premiers de n*, L'Enseignement Mathématique, II. Sér. **27** (1981), 3–27.

[Erd49] Erdős, Pál, *On some applications of Brun's method*, Acta Mathematica **13** (1949), 57–63.

[FK06] Fouvry, Étienne and Klüners, Jürgen, *On the 4-rank of class groups of quadratic number fields*, preprint, 2006.

[FS87] Fiat, Amos and Shamir, Adi, *How to prove yourself: Practical solutions of identification and signature problems*, Advances in Cryptology – Proceedings of CRYPTO '86 (Odlyzko, Andrew M., ed.), Lecture Notes in Computer Science, vol. 263, Association for Computing Machinery, 1987, pp. 186–194.

[Gau89] Gauß, Carl Friedrich, *Untersuchungen über höhere Arithmetik (Disquisitiones Arithmeticae)*, Springer, 1889, German translation by H. Maser.

[Ger84] Gerth III, F., *The 4-class ranks of quadratic fields*, Inventiones Mathematicae **77** (1984), no. 3, 489–515.

[Ger87] ______, *Extension of conjectures of Cohen and Lenstra*, Expositiones Mathematicae **5** (1987), no. 2, 181–184.

[GGH97] Goldreich, Oded, Goldwasser, Shafi, and Halevi, Shai, *Public-key cryptosystems from lattice reduction problems*, Advances in Cryptology - CRYPTO '97, 17th annual international cryptology conference. Santa Barbara, CA, USA. (jun. Kaliski, Burton S., ed.), Lecture Notes in Computer Science, vol. 1294, Springer, 1997, pp. 112–131.

[GJ79] Garey, Michael R. and Johnson, David S., *Computers and intractibility – a guide to the theory of np-completeness*, Freeman, San Francisco, 1979.

[Gol01] Goldreich, Oded, *Foundations of cryptography. I: Basic tools*, Cambridge University Press, 2001.

[Gre69] Greenberg, Marvin J., *Lectures on forms in many variables*, Benjamin, New York, 1969.

[GS04] Grunewald, Fritz J. and Segal, Daniel, *On the integer solutions of quadratic equations*, Journal für die reine und angewandte Mathematik **569** (2004), 13–45.

[Hah94] Hahn, Alexander J., *Quadratic algebras, Clifford algebras, and arithmetic Witt groups*, Universitext, Springer, 1994.

[Har07] Hartung, Rupert Josef, *Cryptography based on quadratic forms: Complexity considerations*, Proceedings of WEWoRC 07, Bochum, to appear in: Lecture Notes in Computer Science, Springer, 2007.

[Has24] Hasse, Helmut, *äquivalenz quadratischer Formen in einem beliebigen algebraischen Zahlkörper*, Journal für Mathematik **153** (1924), 158–162.

[HHGP$^+$03] Hoffstein, Jeffrey, Howgrave-Graham, Nick, Pipher, Jill, Silverman, Joseph H., and Whyte, William, *NTRUSign: Digital signatures using the NTRU lattice*, Topics in cryptology – CT-RSA 2003. The cryptographers' track at the RSA conference 2003, San Francisco, CA, USA, April 13–17, 2003 (Joye, Marc, ed.), Lecture Notes in Computer Science, vol. 2612, Springer, 2003, pp. 122–140.

[HPS98] Hoffstein, Jeffrey, Pipher, Jill, and Silverman, Joseph H., *NTRU: A ring-based public key cryptosystem*, Algorithmic number theory. 3rd international symposium, ANTS-III, Portland, OR, USA, June 21–25, 1998 (Buhler, J. P., ed.), Lecture Notes in Computer Science, vol. 1423, Springer, 1998, pp. 267–288.

[HPS01] ______, *NSS: an NTRU lattice-based signature scheme*, Advances in cryptology - EUROCRYPT 2001. 20th international conference on theory and application of cryptographic techniques, Innsbruck, Austria, May 6-10, 2001 (Pfitzmann, Birgit, ed.), Lecture Notes in Computer Science, vol. 2045, Springer, 2001, pp. 211–228.

[HS07a] Hartung, Rupert Josef and Schnorr, Claus-Peter, *Identification and signatures based on NP-hard problems of indefinite quadratic forms*, preprint, 2007.

[HS07b] ______, *Public key identification based on the equivalence of quadratic forms*, Mathematical Foundations of Computer Science 2007, 32nd International Symposium, MFCS 2007, Český Krumlov, Czech Republic, August 26–31, 2007, Proceedings (Kučera, Luděk and Kučera, Antonín, eds.), Lecture Notes in Computer Science, Springer, 2007, pp. 333–345.

[HW60] Hardy, G. H. and Wright, E. M., *An introduction to the theory of numbers*, fourth ed., Oxford: At the Clarendon Press, 1960.

[IAS96] Ivanyos, Gábor and Ágnes Szánto, *Lattice basis reduction for indefinite forms and an application*, Journal on Discrete Mathematics **153** (1996), no. 1–3, 177–188.

[IM88] Imai, Hideki and Matsumoto, Tsutomu, *Public quadratic polynomial-tuples for efficient signature-verification and message-encryption*, Advances in cryptology. Theory and application of cryptographic techniques, Proc. Workshop, EUROCRYPT '88, Lecture Notes in Computer Science, vol. 330, 1988, pp. 419–453.

[IW97] Impagliazzo, R. and Wigderson, A., *P= BPP, unless E has subexponential circuits: Derandomizing the XOR-lemma*, Proceedings of the 29th Annual ACM Symposium on Theory of Computing, 1997, pp. 220–229.

[Jon44] Jones, Burton Wadsworth, *A canonical quadratic form for the ring of 2-adic integers*, Duke Mathematical Journal **11** (1944), 715–727.

[Jon50] ______, *The arithmetic theory of quadratic forms*, Carus Mathematical Monographs, no. 10, John Wiley & Sons, 1950.

[Kan87] Kannan, Ravi, *Minkowski's convex body theorem and integer programming*, Mathematics of Operations Research **12** (1987), no. 3, 415–440.

[Kho05] Khot, Subash, *Hardness of approximating the shortest vector problem in lattices*, Journal of the ACM **52** (2005), no. 5, 789–808.

[Kit93] Kitaoka, Yoshiyuki, *Arithmetic of quadratic forms*, Cambridge Tracts in Mathematics and Mathematical Physics, no. 106, Cambridge University Press, 1993.

[KM07] Koblitz, Neal and Menezes, Alfred J., *Another look at "provable security"*, Journal of Cryptology **20** (2007), no. 1, 3–37.

[Kne02] Kneser, Martin, *Quadratische Formen*, Springer, 2002.

[Lag80] Lagarias, J. C., *Worst-case complexity bounds for algorithms in the theory of integral quadratic forms*, Journal of Algorithms **1** (1980), 142–186.

[Lam05] Lam, Tsit-Yuen, *Introdction to quadratic forms over fields*, Graduate Studies in Mathematics, no. 67, American Mathematical Society, 2005.

[Leg85] Legendre, Adrien-Marie, *Recherche d'analyse indéterminée*, Histoire de l'Académie Royale des Sciences **1788** (1785), 465–559.

[Leg98] ———, *Essai sur la théorie des nombres*, Duprat, Paris, 1798.

[Lem95] Lemmermeyer, Franz, *The Euclidean algorithm in algebraic number fields*, Expositiones Mathematicae **13** (1995), no. 5, 385–416.

[Len82] Lenstra jun., Hendrik W., *On the calculation of regulators and class numbers of quadratic fields*, Journees arithmetiques, Exeter 1980 (Armitage, J. V., ed.), Lecture Note Series, vol. 56, London Mathematical Society, 1982, pp. 123–150.

[Len87] ———, *Factoring integers with elliptic curves*, Mathematische Annalen (2) **126** (1987), 649–673.

[LL83] Lenstra, Arjen K. and Lenstra jun., Hendrik W. (eds.), *The development of the number field sieve*, Lecture Notes in Computer Science, no. 1554, Springer, 1983.

[LLL82] Lenstra jun., Hendrik W., Lenstra, Arjen K., and Lovász, László, *Factoring polynomials with rational coefficients*, Mathematische Annalen **261** (1982), 515–534.

[LLMP90] Lenstra, Arjen K., Lenstra jun., Hendrik W., Manasse, Mark S., and Pollard, John M., *The number field sieve*, Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC), Baltimore, May 14–16, 1990, 1990, pp. 564–572.

[LO85] Lagarias, J. C. and Odlyzko, Andrew M., *Solving low-density subset sum problems*, Journal of the ACM **32** (1985), 229–246.

[LP92] Lenstra jun., Hendrik W. and Pomerance, Carl, *A rigorous time bound for factoring integers*, Journal of the American Mathematical Society **5** (1992), 483–516.

[MA78] Manders, Kenneth L. and Adleman, Leonard M., *NP-complete decision problems for binary quadratics*, Journal of Computer and System Sciences **16** (1978), 168–184.

[Man74] Manin, Yuri Ivanovitch, *Cubic forms: Algebra, geometry, arithmetic*, North-Holland Mathematical Library, vol. 4, North-Holland, 1974, Translated by M. Hazewinkel.

[Mat93] Matiyasevich, Yuri V., *Hilbert's tenth problem: With a foreword by Martin Davis*, MIT Press, Cambridge MA, 1993.

[Mey91] Meyer, Arnold, *Zur Theorie der indefiniten ternären quadratischen Formen*, Journal für Mathematik **CVIII** (1891), 125–139.

[MG02] Micciancio, Daniele and Goldwasser, Shafi, *Complexity of lattice problems: a cryptographic perspective*, The Kluwer International Series in Engineering and Computer Science, vol. 671, Kluwer Academic Publishers, March 2002.

[Mih94] Mihailescu, P., *Fast generation of provable primes using search in arithmetic progressions*, Advances in cryptology - CRYPTO '94. 14th annual international cryptology conference, Santa Barbara, CA, USA, August 21-25, 1994. Proceedings (Desmedt, Yvo G., ed.), Lecture Notes in Computer Science, no. 839, Springer, 1994, pp. 282–293.

[Min11] Minkowski, Hermann, *Grundlagen einer Theorie der ganzzahligen quadratischen Formen*, H. Minkowski: Gesammelte Werke (Hilbert, David, ed.), Teubner, Wiesbaden, 1911, pp. 3–144.

[MSC96] Mitrinović, D. S., Sándor, J., and Crstici, B. (eds.), *Handbook of number theory*, Mathematics and Its Applications, vol. 351, Kluwer Academic Publishers, 1996.

[Neu92] Neukirch, Jürgen, *Algebraische Zahlentheorie*, Springer, 1992.

[O'M63] O'Meara, O. T., *Introduction to quadratic forms*, Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen, vol. 117, Springer, 1963, (reprinted in 2000).

[Pap94] Papadimitriou, Christos H., *Computational complexity*, Addison-Wesley, 1994.

[Per54] Perron, Oskar, *Die Lehre von den Kettenbrüchen*, 3rd ed., vol. I, Teubner, Stuttgart, 1954.

[Pom80] Pomerance, Carl, *A note on the least prime in an arithmetic progression*, Journal of Number Theory **12** (1980), 218–223.

[Pom82] ———, *Analysis and comparison of some integer factoring algorithms*, Computational Methods in Number Theory, Part I (Lenstra jun., Hendrik W. and Tijdeman, R., eds.), Math. Centre Tracts, vol. 154, Math. Centrum, 1982, pp. 89–139.

[Pom87] ———, *Fast rigorous factorization and discrete logarithm algorithms*, Discrete algorithms and complexity, Proc. Jap.-US Joint Semin., Kyoto/Jap. 1986, Perspect. Comput., no. 15, 1987, pp. 119–143.

[PP85] Plesken, Wilhelm and Pohst, Michael E., *Constructing integral lattices with prescribed minimum. I*, Mathematics of Computation **45** (1985), 209–221.

[Pra61] Prachar, Karl, *Über die kleinste Primzahl einer arithmetischen Reihe*, Journal für die reine und angewandte Mathematik **206** (1961), 3–4.

[PS85] Plesken, Wilhelm and Souvignier, B., *Computing isometries of lattices*, Mathematics of Computation **45** (1985), 209–221.

[PS87] Pollard, John M. and Schnorr, Claus-Peter, *An efficient solution of the congruence $x^2 + ky^2 = m \ (mod\ n)$*, IEEE Transactions on Information Theory **33** (1987), 702–709.

[PZ89] Pohst, Michael E. and Zassenhaus, H., *Algorithmic algebraic number theory*, Encyclopedia of Mathematics and its Applications, no. 30, Cambridge University Press, 1989.

[Sch04a] Schnorr, Claus-Peter, *Indefinite ternary quadratic form signatures*, preprint, 2004.

[Sch04b] ______, *Reduction of quadratic forms reconsidered*, preprint, 2004.

[Sch07] ______, *Progress on LLL and lattice reduction*, Proceedings LLL+25, Caen, France, June 29–July 1, 2007, 2007, to appear.

[Ser73] Serre, Jean-Pierre, *A course in arithmetic*, Graduate Texts in Mathematics, no. 7, Springer, 1973.

[SH07] Stehlé, Damien and Hanrot, Guillaume, *Improved analysis of Kannan's shortest lattice vector algorithm*, Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings (Menezes, Alfred J., ed.), Lecture Notes in Computer Science, vol. 4622, 2007, pp. 170–186.

[Sha72] Shanks, Daniel, *The infrastructure of a real quadratic field and its applications*, Proc. 1972 Number Theory Conf., Univ. Colorado, Boulder, 1972, pp. 217–224.

[Sha74] Shafarevich, I. R., *Basic algebraic geometry*, Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen, no. 213, Springer, 1974.

[Sie35] Siegel, Carl Ludwig, *Über die analytische Theorie der quadratischen Formen*, Annals of Mathematics (2) **36** (1935), 527–606.

[Sie36] ______, *Über die analytische Theorie der quadratischen Formen. II*, Annals of Mathematics (2) **37** (1936), 230–263.

[Sie37] ______, *Über die analytische Theorie der quadratischen Formen. III*, Annals of Mathematics (2) **38** (1937), 212–291.

[Sie63] ______, *Lectures on quadratic forms*, Tata Institute of Fundamental Research, Bombay, 1957 (reissued 1963), notes by K. G. Ramanathan.

[Sie72] ______, *Zur Theorie der quadratischen Formen*, Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse **1972** (1972), 21–46.

[Sim05a] Simon, Denis, *Quadratic equations in dimensions 4, 5 and more*, preprint, 2005.

[Sim05b] ______, *Solving quadratic equations using reduced unimodular quadratic forms*, Mathematics of Computation **74** (2005), no. 251, 1531–1543.

[Sma98] Smart, Nigel P., *The algorithmic resolution of diophantine equations*, Student Texts, vol. 41, London Mathematical Society, 1998.

[SO80] Scharlau, Winfried and Opolka, Hans, *Von Fermat bis Minkowski: Eine Vorlesung über Zahlentheorie und ihre Entwicklung*, Springer, 1980.

[TW86] Tennenhouse, M. and Williams, H.C., *A note on the class-number one in certain real quadratic and pure cubic fields*, Mathematics of Computation **46** (1986), no. 173, 333–336.

[Val91] Vallée, Brigitte, *Generation of elements with small modular squares and provably fast integer factoring algorithms*, Mathematics of Computation **56** (1991), no. 194, 823–849.

[VV07] Vallée, Brigitte and Vera, Antonio, *Lattice reduction in two dimensions: Analyses under realistic probabilistic models*, preprint, 2007.

[Wag79] Wagstaff, Jr., Samuel S., *Greastest of the least primes in arithmetic progressions having a given modulus*, Mathematics of Computation **33** (1979), no. 147, 1073–1080.

[Wat76] Watson, G.L., *The 2-adic density of a quadratic form*, Mathematika **23** (1976), 94–106.

[Wei84] Weil, André, *Number theory: An approach through history; From Hammurapi to Legendre*, Birkhäuser, 1984.

[Zhu97] Zhuravlev, V. G., *Representation of a form by the genus of quadratic forms*, St. Petersburg Mathematical Journal **8** (1997), no. 1, 15–84.

Index